REMIGIJUS LAPINSKAS

PRACTICAL ECONOMETRICS. II. TIME SERIES ANALYSIS

COMPUTER LABS

***

PRAKTINĖ EKONOMETRIJA. II. LAIKINĖS SEKOS

PRATYBOS KOMPUTERIŲ KLASĖJE

remigijus.lapinskas@mif.vu.lt

Vilnius 2013

Contents – Computer Labs

****

Contents – Lecture Notes

Introduction

These computer labs are to accompany the Lecture Notes „Practical econometrics.II. Time series analysis“. We shall use two software programs, GRETL and R, interchangeable.

1. Time Series: Examples

In Fig.1.1, one can see a graph of a typical time series y_t [1] together with three of its components. Time series can usually be split into three components: clearly seen regular part, a so-called *trend* Tr_t, *seasonal part* [2] S_t (there are less passengers in the winter time) and, hopefully, not very big random *errors* ε_t; in general, we shall write $Y_t = Tr_t + S_t + \varepsilon_t$. Note that in Fig. 1.1 not the trend Tr_t is drawn but only its estimate $\widehat{Tr}_t$ (the same applies to other components, $\hat{S}_t$ and $e_t = \hat{\varepsilon}_t$, but we shall not be too pedantic). One of the objectives of the time series analysis is to extract from Y_t these three components and use them for analysis, comparison or forecast.

```
help.search("AirPassengers");library(datasets)
data(AirPassengers);dec=decompose(log(AirPassengers)) # extracts three comp.
plot(dec)
```

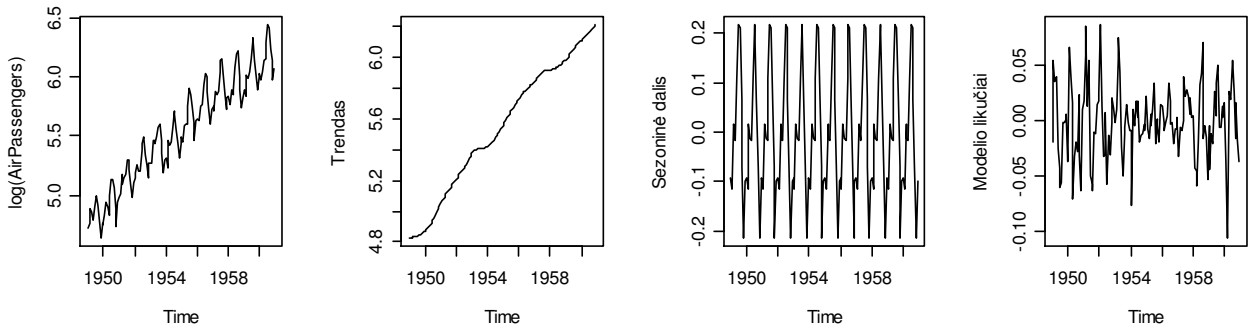


Fig. 1.1. The graphs of the time series `log(AirPassengers)` and its three components (trend, seasonal part and residuals)

If residuals make a stationary process, we assume that we have created a proper model (i.e., the components were extracted correctly). Recall that the process is stationary if its trajectories randomly and more or less regularly fluctuate around a constant (its mean). The stationary process also has a mean reversion property, that is, whatever is its value, it tends to get back to the mean. The right-most graph is similar to that of stationary process (though one can see something like periodicity in it; on the other hand, the amplitude of oscillations in the middle is less than elsewhere – anyway, we assume that our procedure (described in what follows) has correctly decomposed the time series).

- **White Noise**

This is a fundamental example of stationary process – the *white noise* is a sequence of uncorrelated random variables with zero mean and constant variance. Note that we do not mention the distribution of the random variables but often it is assumed that they are normal or

[1] This is the classical Box and Jenkins's *airline* data set (it contains monthly, 1949-1960, totals of international airline passengers; the set can be found in the `datasets` package, data set `AirPassengers`). Notice that in Fig. 1.1 the logarithms are drawn. This data is also analysed in 1.12 exercise.
[2] Both trend and seasonal components are treated as nonrandom curves.

Gaussian. The white noise is a model of absolutely chaotic process of uncorrelated observations, a process which immediately forgets its past.

Let us plot three trajectories of (normal) white noise, $y_t^{(1)}, y_t^{(2)}, y_t^{(3)}$ (it is impossible to plot all the trajectories of random process (because there are infinitely many of them), therefore we shall artificially create some of them).

We start with a few remarks on the `set.seed` function. R „knows“ how to generate a (very very long) sequence of uniformly in the interval [0,1] distributed random numbers (r.n.). A part of the sequence we can produce with the `runif` (=`r+unif`) function:

```
> runif(10)
 [1] 0.81878843 0.13345228 0.07379024 0.01138173 0.13101318
 [6] 0.20613918 0.16512315 0.51821997 0.47321645 0.52996317
```

If we apply the function once again, it will produce the next ten numbers of that sequence:

```
 > runif(10)
 [1] 0.01618378 0.15645068 0.10320850 0.46572912 0.53319667
 [6] 0.34754886 0.57468911 0.17348361 0.57527263 0.82605006
```

To generate always the same sequence of r.n., R must know from where to begin the sub-sequence. This can be achieved by indicating the starting point with the `set.seed` function before starting `runif` anew. All the other r.n. (normal, Poisson etc) can be obtained from the uniform, therefore, in order to get the same sequence, one has to point the starting point with `set.seed(int)`, where `int` is any integer.

```
set.seed(10)                                              # fix the beginning
opar=par(mfrow=c(1,3))
y=ts(rnorm(50));plot(y,ylim=c(-2.5,2.5));abline(0,0)   # 50 elements long seq.
y=ts(rnorm(50));plot(y,ylim=c(-2.5,2.5));abline(0,0)   # Another sequence
y=ts(rnorm(50));plot(y,ylim=c(-2.5,2.5));abline(0,0)   # Continuation
par(opar)
```

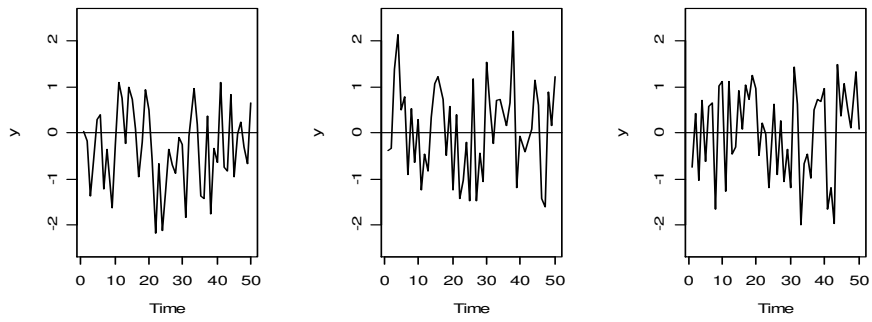


Fig. 1.2. Three trajectories (paths) $y_t^{(1)}, y_t^{(2)}, y_t^{(3)}$ of normal white noise (note that some trajectories do not seem „quite regular“)

We shall use the symbol W_t in the future to denote white noise; its trajectories (realizations) will be denoted by w_t.

- **Random Walk**

The random walk process is in a way opposite to stationary (it does not oscillate around a constant, it is characterized by long excursions up and down, it does not have the mean reversion property).

1.1 example. The gambler begins with an initial capital of $y_0 = \$10$ and during each play he wins or loses \$1 with probability 0.5. It is easy to verify that his capital at moment t is expressed by the formula $Y_t = y_0 + \sum_{i=1}^{t} \varepsilon_t$ where ε_t are his (uncorrelated) gains at moment t. The process Y_t is called the *random walk*. It does not „more or less regularly fluctuate around a constant" because its variance increases: $\operatorname{var} Y_t = ct \neq const$ (can you prove that?).

```
set.seed(10)
opar=par(mfrow=c(1,3))
wn1=sample(c(-1,1), 100, replace = TRUE)    # Random sampling from a set of
                                            # -1 and 1.
rw1=cumsum(wn1)+10                          # cumsum:(x1,x2) → (x1,x1+x2)

plot(rw1,type="l")
wn2=sample(c(-1,1), 100, replace = TRUE)    # generates gains
rw2=cumsum(wn2)+10                          # generates cumulative capital
plot(rw2,type="l")
wn3=sample(c(-1,1), 100, replace = TRUE)
rw3=cumsum(wn3)+10
plot(rw3,type="l")
par(opar)
```

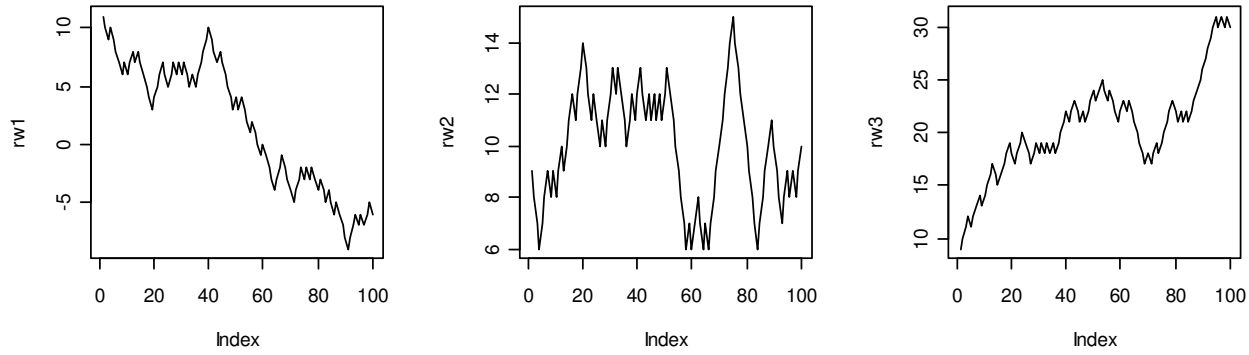


Fig. 1.3. Three trajectories of Bernoulli random walk; note long excursions up and down.

▶▶ Note that the construction of random walk `rw` consists of two steps:

1) Generate any white noise `wn` (for example, `wn1` – does it satisfy the definition of white noise?); it will represent random gains.
2) Use `cumsum(wn)` and generate your accumulated capital `rw`.

1.1 exercise. Generalize this (Bernoulli) random walk by assuming that the gain is a normal r.v. $N(0,1)$. *Hint*. To generate this `wn` use the function `rnorm(100)` (=`rnorm(100,mean=0)`).◀

This process can still be generalized by assuming that during each play the gambler gets not only his gainings but also a premium of a=\$0.20: $Y_t = Y_{t-1} + (\varepsilon_t + a) = (Y_{t-2} + \varepsilon_{t-1} + a) + \varepsilon_t + a = \ldots$

$= y_0 + \sum_{s=1}^{t} \varepsilon_s + ta$ (this is a *a random walk with a drift* a).

1.2 exercise. Draw two paths of a random walk with drift with $y_0 = 10$ and $a = 0.2$. *Hint.* Replace `rnorm(100)` by `rnorm(100,mean=0.2)`.

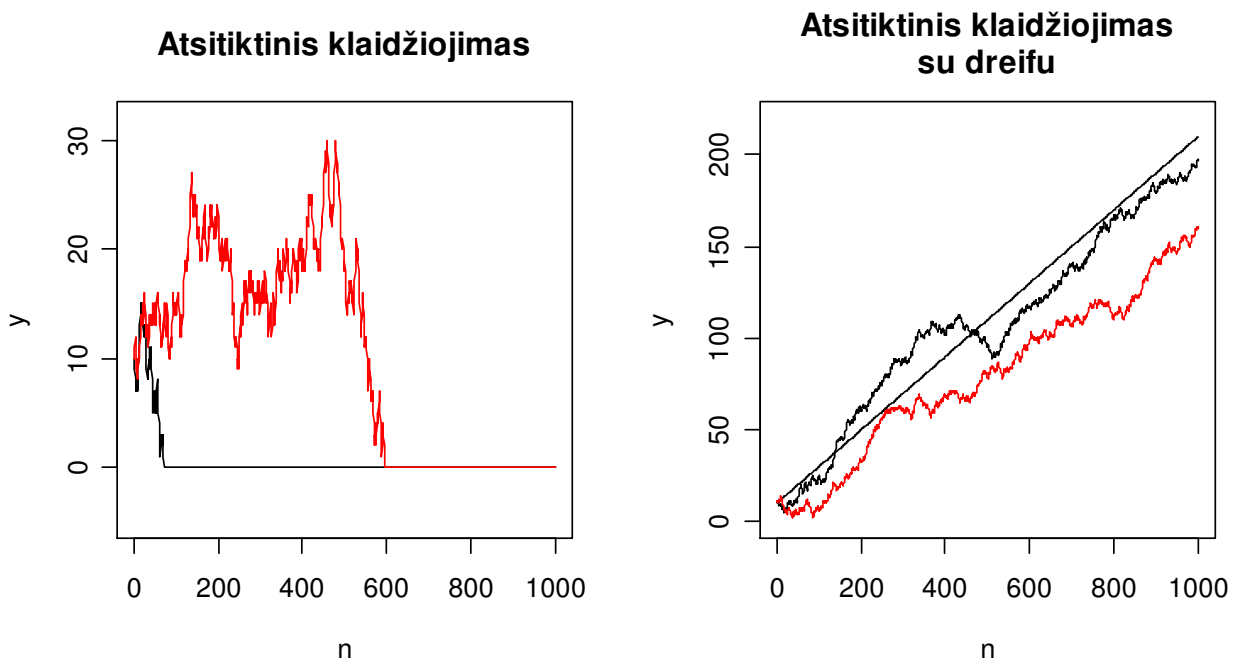


Fig. 1.4. Two graphs of a random walk till bankruptcy (left) and two paths of a random walk with drift (right; the black line denotes the mean value of the process: $y = 10 + 0.2n$).

- **Time Series and R**

In R, a time series is a vector with some additional attributes. Select the Courier New text below and using Copy+Paste move it to the console of R:

```
shipm1 = c(42523, 46029, 47485, 46692, 46479, 48513, 42316, 45717, 48208,
47761, 47807, 47772, 46020, 49516, 50905, 50226, 50678, 53124, 47252, 47522,
52612, 53800, 52019, 49705, 48864, 53281, 54668, 53740, 53346, 56421, 49603,
52326, 56724, 57257, 54335, 52095, 49714, 53919, 54750, 53190, 53791, 56790,
49703, 51976, 55427, 53458, 50711, 50874, 49931, 55236, 57168, 56257, 56568,
60148, 51856, 54585, 58468, 58182, 57365, 55241, 54963, 59775, 62049, 61767,
61772, 64867, 56032, 61044, 66672, 66557, 65831, 62869, 63112, 69557, 72101,
71172, 71644, 75431, 66602, 70112, 74499, 76404, 75505, 70639, 71248, 78072,
81391, 80823, 82391, 86527, 77487, 83347, 88949, 89892, 85144, 75406)

> class(shipm1)
[1] "numeric"
```

The vector `shipm1` is value of shipments, in millions of dollars, monthly from January, 1967 to December, 1974. This represents manufacturers' receipts, billings, or the value of products shipped, less discounts, and allowances, and excluding freight charges and excise taxes. Shipments by foreign subsidiaries are excluded, but shipments to a foreign subsidiary by a domestic firm are included. Source: U. S. Bureau of the Census, Manufacturer's Shipments, Inventories and Orders.

To transform the vector to time series object, one should indicate the initial date and monthly frequency[3]:

```
shipm = ts(shipm1, start=1967, freq=12)
```

```
> shipm
       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
1967 42523 46029 47485 46692 46479 48513 42316 45717 48208 47761 47807 47772
1968 46020 49516 50905 50226 50678 53124 47252 47522 52612 53800 52019 49705
.....................................................................................
1973 63112 69557 72101 71172 71644 75431 66602 70112 74499 76404 75505 70639
1974 71248 78072 81391 80823 82391 86527 77487 83347 88949 89892 85144 75406
```

Now `shipm` has a time series structure:

```
> class(shipm)
[1] "ts"
```

Inside R, `shipm` is now expressed as follows:

```
> dput(shipm) # The function dput describes the internal structure of shipm
structure(c(42523, 46029, 47485, 46692, 46479, 48513, 42316, 45717, 48208, 47761,
47807, 47772, 46020, 49516, 50905, 50226, 50678, 53124, 47252, 47522, 52612, 53800,
52019, 49705, 48864, 53281, 54668, 53740, 53346, 56421, 49603, 52326, 56724, 57257,
54335, 52095, 49714, 53919, 54750, 53190, 53791, 56790, 49703, 51976, 55427, 53458,
50711, 50874, 49931, 55236, 57168, 56257, 56568, 60148, 51856, 54585, 58468, 58182,
57365, 55241, 54963, 59775, 62049, 61767, 61772, 64867, 56032, 61044, 66672, 66557,
65831, 62869, 63112, 69557, 72101, 71172, 71644, 75431, 66602, 70112, 74499, 76404,
75505, 70639, 71248, 78072, 81391, 80823, 82391, 86527, 77487, 83347, 88949, 89892,
85144, 75406), .Tsp = c(1967, 1974.91666666667, 12), class = "ts")
```

```
> tsp(shipm)
[1] 1967.000 1974.917   12.000
> start(shipm)
[1] 1967    1
> end(shipm)
[1] 1974   12
> frequency(shipm)
[1] 12
> attributes(shipm)
$tsp
[1] 1967.000 1974.917   12.000
$class
[1] "ts"
```

The function `plot` draws the graphs of `shipm1` and `shipm`; note that the shape of the graph depends on the class of the object.

```
opar=par(mfrow=c(1,2))
plot(shipm1) # The graphs are different for different classes
```

[3] If `freq=12`, R will treat the data as monthly; if `freq=4` as quarterly.

```
plot(shipm)
par(opar)
```

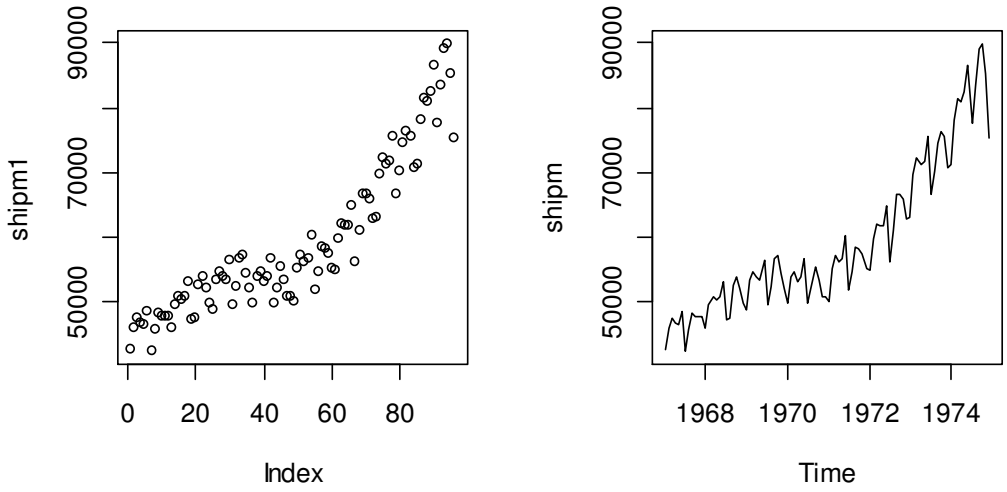


Fig. 1.5. The graphs of a vector `shipm1` and the time series `shipm`

1.3 exercise. Assign a time series structure to the random walk from 1.2 exercise and draw its graph. Experiment with number of observations, `start` and `freq` attributes.

1.4 exercise.

```
help.search("housing")  # find the data set housing
library(fma)          # attach the package fma
library(help=fma)     # description of the package
data(housing)         # unzip the file housing
housing               # the data set
?housing              # description of the data
class(housing)
dim(housing)
```

When begins and when ends this three dimensional time series? What is its frequancy? Plot its graph. What is the meaning of each series?

1.5 exercise. Find the data set `AirPassengers` and explore it (class, start, frequency etc).

2. Stationary Time Series

Once we have „correctly“ decomposed a time series into three components: $Y_t = T_t + S_t + \varepsilon_t$, the errors ε_t [1] must constitute a stationary process. If this stationary process is WN, its near and distant forecasts are trivial – it is just its mean, i.e., 0. In other cases, the forecast depends on the structure of a stationary process, more specifically, on its past values. This chapter is to describe some classes of stationary processes.

2.1. White Noise

The simplest stationary process is a white noise (this is a sequence of uncorrelated r.v. with a constant, often zero, mean and constant variance). In R, the functions `rnorm(n)`, `runif(n,-a,a)` and like generate namely a white noise. To learn that your sequence `Y(=lh)` is white noise, perform **two** steps:

```
library(datasets)
?lh
data(lh); plot(lh)
```

1) Visual inspection.

```
library(forecast)
tsdisplay(lh)       # lh is time series
```

Follow the rule: if all the first 5-10 bars in both ACF and PACF are inside the blue lines, the time series is most probably a white noise. In our case, `lh` is most probably not a white noise (the first ACF and PACF bars are outside the blue band). ◀◀

Before taking Step 2, a few words about the stationary ARIMA model whose general form is `ARIMA(p,0,q)(P,0,Q)[freq]`. The time series Y_t is a white noise if it is described by `ARIMA(0,0,0)(0,0,0)[freq]` or, in a shorter form, `ARIMA(0,0,0)` (in this case $Y_t =$ const + ε_t). Now, to take the final decision, go to Step 2:

2) perform the **Ljung-Box test** with H_0 : *Y is a white noise*. Follow the rule: if at least one of the first 5-10 bubles is below the 5% critical blue line, reject H_0 (in other words, *Y* is then not a white noise).

```
mod.000 <- arima(lh, c(0,0,0))    # to express lh as const + res.
mod.000
Coefficients:
      intercept
         2.4000  # Note: 2.400±2*0.0788 does not include 0⇒intrc. is signif.
s.e.     0.0788

sigma^2 estimated as 0.2979:  log likelihood = -39.05,  aic = 82.09
```

[1] In reality, we observe *residuals* $\hat{\varepsilon}_t = e_t$, which should not differ much from the errors ε_t.

```
tsdiag(mod.000)     # mod.000 is a model
```

Note that `tsdisplay` is applicable to time series `Y` whereas `tsdiag` to `arima` model (`tsdiag` expresses `lh` as `2.4000 + mod.000$res` and tests whether residuals make a white noise). Our conclusion: according to its graph, `lh` is a stationary time series but according to ACF and PACF as well as Ljung-Box test, not a white noise. ◀

2.1 example. The file cret.ret.txt contains montly returns on citigroup stock starting from 1990:01.

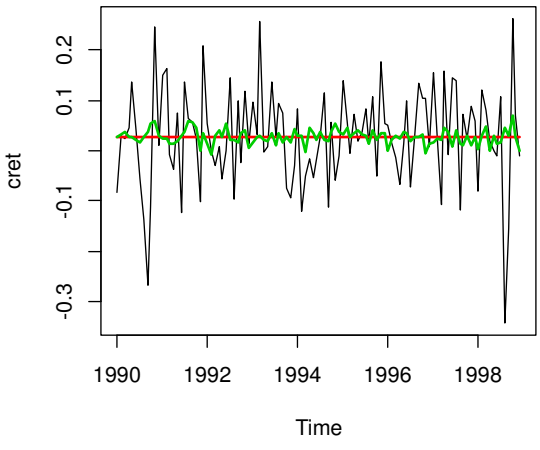
```
cret=ts(scan(file.choose(),skip=1),start=c(1990,1),freq=12)
cret
plot(cret)
tsdisplay(cret)
cret.000=arima(cret,c(0,0,0))
tsdiag(cret.000)
```

Both steps confirm that our data is WN (why?). In the future, we shall try to answer the question "is `cret` a WN or not?" using the function `auto.arima` from the `forecast` package:

```
(cret.auto=auto.arima(cret))
Series: cret
ARIMA(2,0,0)(1,0,0)[12] with non-zero mean

Coefficients:
          ar1      ar2     sar1  intercept
      -0.0393  -0.0889  -0.1289     0.0277
s.e.   0.0960   0.0959   0.1173     0.0075

sigma^2 estimated as 0.009589:  log likelihood=97.59
AIC=-185.18   AICc=-184.6   BIC=-171.77
```

The `auto.arima` function returns the best ARIMA model according to the smallest AIC value and claims that our model is not a white noise (because it is not an ARIMA(0,0,0)). Note that the function does not care whether coefficients are significant (the seasonal AR(1) term `sar1` is not significant because the confidence interval -0.1289 ± 2*0.1173 includes 0; we shall explain what seasonality means later) and whether residuals make a WN. To see whether these two models differ significantly, run

```
plot(cret)
lines(fitted(cret.000),col=2,lwd=2)
lines(fitted(cret.auto),col=3,lwd=2)
```



The `cret.auto` model is more precise (it follows `cret` more closely) but we would like to apply the general rule – all the coefficients must be significant. We improve the `cret.auto` model by forbidding insignificant seasonal term:

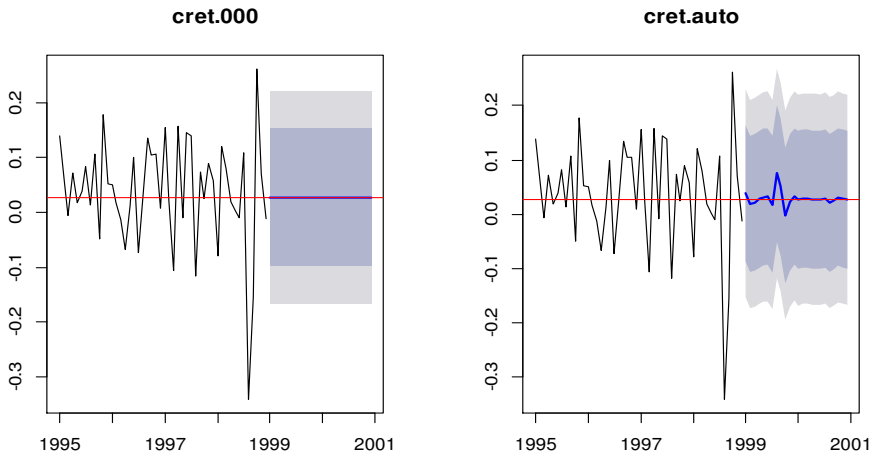
```
(cret.auto.noS=auto.arima(cret,seasonal=FALSE))
Series: cret
ARIMA(0,0,0) with non-zero mean
```

```
Coefficients:
      intercept
         0.0269
s.e.     0.0095

AIC=-189.19   AICc=-189.08   BIC=-183.83
```

Surprisingly, this single improvement brings us back to a more natural model ARIMA(0,0,0).

To forecast `cret` 24 months ahead, run

```
par(mfrow=c(1,2))
plot(forecast(cret.000),include=48,main="cret.000")    # 2*12 months forecast
abline(mean(cret),0,col=2)
plot(forecast(cret.auto),include=48,main="cret.auto")  # leave only 48 histor.
                                                       # observations in plot
abline(mean(cret),0,col=2)
```



Thus, if the model is WN, the forecast (left) is just the mean of the time series. If the model is not a WN (right), the forecast tends to the mean. ◀◀

2.2. Stationary Processes

Recall that Y_t is *stationary* if its mean and variance do not change in time (all trajectories of stationary process fluctuate around its mean with more or less constant spread). The process Y_t is *stable* if its distribution tends to stationary (if the process starts at a point far from its mean, its trajectory will tend towards the mean; we say that a stable process has the mean reverting property). The concept of stable process is close to stationarity: stationary process is stable, stable process tends to stationary.

The known Wold's decomposition theorem claims that practically any stationary[2] process Y_t, $t \in \mathbb{Z}$, may be expressed as a linear combination of the values w_{t-j} of some WN process:

[2] More specifically, stationary in *wide* or *covariance* sense.

$$Y_t = \mu + \sum_{j=0}^{\infty} k_j w_{t-j},\ \sum k_j^2 < \infty;$$

to simplify notation, we shall assume that the number μ (it is the (constant) mean of Y_t) equals zero. The mean of a stationary process μ and its variance $\sigma^2 (= \sigma_Y^2) = \sigma_w^2 \sum_{i=0}^{\infty} k_i^2$ are constant, therefore all the information about the process is contained in its (auto)covariance function (ACF) $\gamma(s)$ or (auto)correlation function $\rho(s)$:

$$\gamma(s) = \mathrm{cov}(Y_t, Y_{t+s}) = \sigma_w^2 \left(\sum_{i=0}^{\infty} k_i k_{i+s} \right),$$
$$\rho(s) = \mathrm{corr}(Y_t, Y_{t+s}) = (\text{what is its definition ?}),\ s = 1, 2, \ldots$$

One should distinguish a theoretical covariance function $\gamma(s)$ and its estimate (*sample* covariance function)

$$c(s) = \frac{1}{n} \sum_{i=\max(1,-s)}^{\min(n-s,n)} (Y_{i+s} - \bar{Y})(Y_i - \bar{Y}),$$

which is only for „large" n close (because of the law of large numbers) to $\gamma(s)$. The same proposition holds for the correlation function $\rho(s)$ and its estimate $r(s) = c(s)/c(0)$ (the ACF graph plots namely this function).

We usually have only a finite number of time series observations, therefore it is impossible to „restore" (that is, estimate) infinitely many coefficients k_j. The famous Box - Jenkins's ARMA model was introduced to approximate a stationary process by a process described by a finite number of parameters.

- A process Y_t is called the AR(p) process (AutoRegressive process of order p), if it satisfies the equation $Y_t = a_0 + \sum_{i=1}^{p} a_i Y_{t-i} + w_t$ (the simplest variant is AR(1) described by $Y_t = a_0 + a_1 Y_{t-i} + w_t$; it is stationary if $|a_1| < 1$ and its constant mean then equals $a_0/(1-a_1)$).

- A process Y_t is called the MA(q) process (Moving Average process of order q)), if[3] $Y_t = \mu + \sum_{i=0}^{q} b_j w_{t-j}$, $b_0 = 1$, (the simplest variant is MA(1) described by $Y_t = \mu + w_t + b_1 w_{t-1}$, it is always stationary and its mean equals μ).
- A process Y_t is called the ARMA(p,q) process[4], if $Y_t = \mu + \sum_{i=1}^{p} a_i Y_{t-i} + \sum_{j=0}^{q} b_j w_{t-j}$ (the simplest variant is ARMA(1,1) process $Y_t = \mu + a_1 Y_{t-1} + w_t + b_1 w_{t-1}$ and its mean equals $\mu/(1-a_1)$).

Each type can be recognized by its ACF and PACF plots but we shall use `auto.arima` function to do the job (if necessary, slightly correct the output and test whether residuals $\hat{w}_t$ make a WN)

[3] When defining the MA process, we do not need to know both the variance σ^2 of the process w_t and the coefficient b_0; we choose $b_0 = 1$.

[4] Some authors define ARMA in a different way.

Most of the functions for the analysis of the ARMA processes are in the `MASS`, `tseries` and `forecast` packages.

- **AR**

2.2 example. Generate and plot 100 values of the AR(1) process with $a_0 = 1$ and 1) $a_1 = 0.7$, 2) $a_1 = 1$, and 3) $a_1 = 1.2$. Forecast the 1*st* variant (why is it stationary?) 20 time periods ahead.

```
N=100
a0=1
a1=c(0.7,1,1.2)
y=vector("list",3)    # save space for 3 time series
set.seed(12)
for(j in 1:3)
{
y[[j]][1]=5           # all series begin in Y1=5
   for(i in 2:N)      # this loop generates 100 observations
   {
   y[[j]][i]=a0+a1[j]*y[[j]][i-1]+rnorm(1)
   }
}
yts07=ts(y[[1]],freq=4)   # assign ts attributes
yts07
yts10=ts(y[[2]],freq=4)
yts10
yts12=ts(y[[3]],freq=4)
yts12

par(mfrow=c(1,3))
plot(yts07)           # see Fig. 2.1
plot(yts10)
plot(yts12)
```

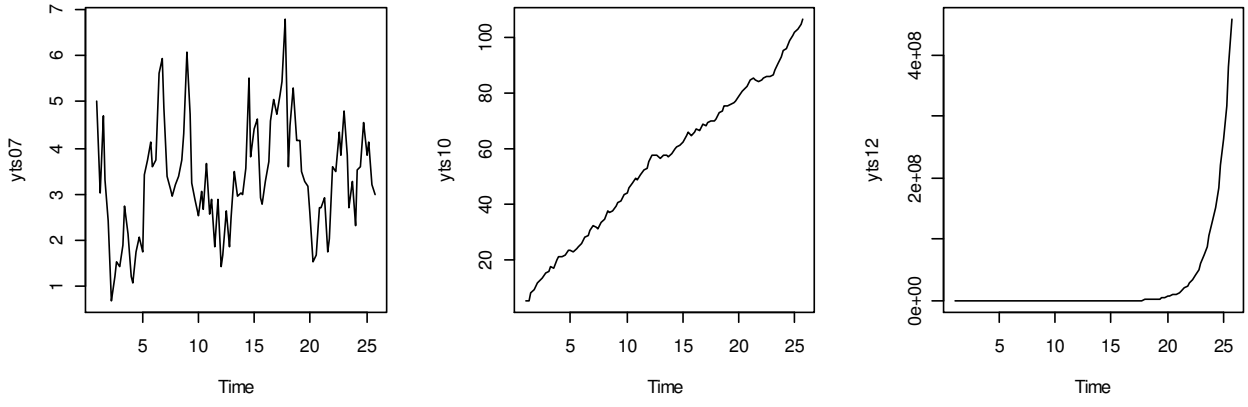


Fig. 2.1. Stationary with $a_1 = 0.7$ (left), random walk with $a_1 = 1.0$ (center), and explosion (for example, hyperinflation) with $a_1 = 1.2$ (right); only the left variant is stationary

```
(y.mod07=auto.arima(yts07))

ARIMA(1,0,0) with non-zero mean
Coefficients:
         ar1  intercept
      0.6976     3.3560
s.e.  0.0713     0.2794

par(mfrow=c(1,3))
plot(forecast(y.mod07,20),include=40)    # see. Fig. 2.2, left
abline(10/3,0,col=2)
```

The AR process (here it is AR(2)) may also be generated with the `arima.sim` function:

```
set.seed(1)
Y3 <- arima.sim(200, model=list(ar=c(0.5,0.1)))
plot(forecast(auto.arima(Y3),h=10),include=30)
abline(0,0,col=2)       # see Fig. 2.2, center

# the second path:
Y4 <- arima.sim(200, model=list(ar=c(0.5,0.1)))
plot(forecast(auto.arima(Y4),h=10),include=30)
abline(0,0,col=2)       # see Fig. 2.2, right
```

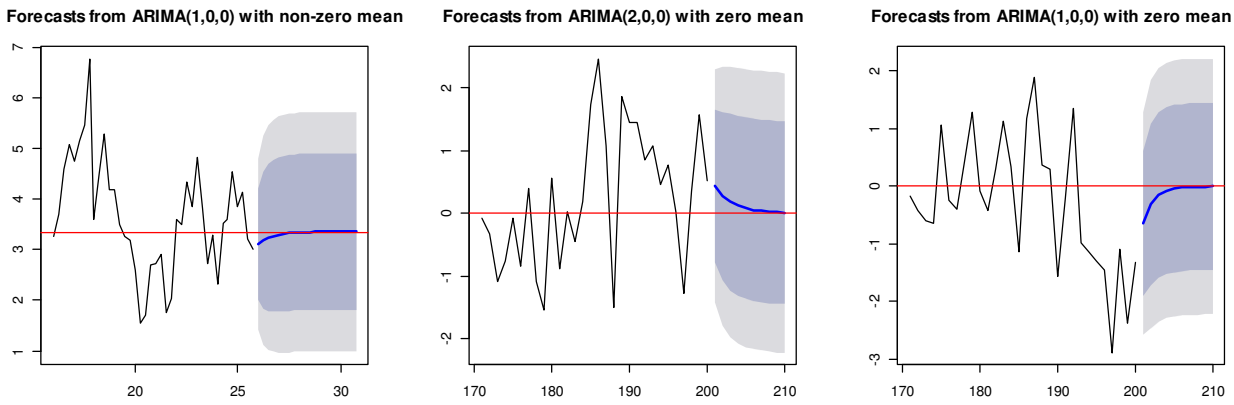


Fig. 2.2. The forecasts tend to the mean of the process.

2.1 exercise. Assume that we have forgoten the origin of the time series `Y3`. Create its model with `auto.arima`, test residuals for WN and forecast the model 24 time periods ahead.

2.2 exercise. The file unemp.txt in the folder PEdata contains quarterly data on the U.S. unemployment, 1948:01 iki 1978:01. Analyze the data and determine its model with `auto.arima`. Do the residuals constitute WN? (apply `tsdisplay` and `tsdiag`[5] functions from the `forecast` package). Forecast unemployment two years ahead, draw respective graph.

2.3 exercise. The R's data set `presidents` contains the (approximately) quarterly approval rating for the President of the United States from the first quarter of 1945 to the last quarter of 1974. Create an appropriate AR model.

[5] The argument to `tsdisplay` is time series and to `tsdiag` its `arima` model.

2.4 exercise. On p. 2 - 1 we saw that `lh` is not a WN. Can you find a better model for this time series?

• MA and ARMA

We have already mentioned that AR(p) in „good" cases is a stationary process (but not a WN; when AR(1) is stationary?) Here is one more example of a stationary process: the process $Y_t = \mu + b_0 w_t + b_1 w_{t-1} + ... + b_q w_{t-q}$, $b_0 = 1$, is called a Moving Average process of the q-th order and denoted MA(q) (the simplest MA process is MA(1): $Y_t = \mu + w_t + b_1 w_{t-1}$).

2.3 example. The internet site http://www.stat.pitt.edu/stoffer/tsa2/ contains the file varve.dat where the yearly time series `varve` is presented. Here is the description of the file:

Melting glaciers deposit yearly layers of sand and silt during the spring melting seasons, which can be reconstructed yearly over a period ranging from the time deglaciation began in New England (about 12,600 years ago) to the time it ended (about 6,000 years ago). Such sedimentary deposits, called varves, can be used as proxies for paleoclimatic parameters, such as temperature, because, in a warm year, more sand and silt are deposited from the receding glacier. Fig. 2.3 shows the thicknesses of the yearly varves collected from one location in Massachusetts for 634 years, beginning 11,834 years ago. Because the variation in thicknesses increases in proportion to the amount deposited, a logarithmic transformation could remove the nonstationarity observable in the variance as a function of time. Fig. 2.3 shows the original and transformed varves, and it is clear that this improvement has occurred. The ordinary first differences $\log(Y_t) - \log(Y_{t-1})$ will also improve stationarity.

One can import the data as follows:

```
varv = ts(scan("http://www.stat.pitt.edu/stoffer/tsa2/data/varve.dat"))
```

As proposed, we shall analyze logarithmic differences.

```
lvarv=log(varv)
dlvarv=diff(log(varve))    # create log-differences
par(mfrow=c(1,3))          # see Fig. 2.3
plot(varv)          # nonstationary with varying mean and amplitude
plot(lvarv))        # nonstationary with varying mean
plot(dlvarv)        # stationary, see Fig. 2.3, right
par(mfrow=c(1,1))
```

Recall that the meaning of the log-differences is the yearly percentage change (of `varv` in our case), this series is often stationary.

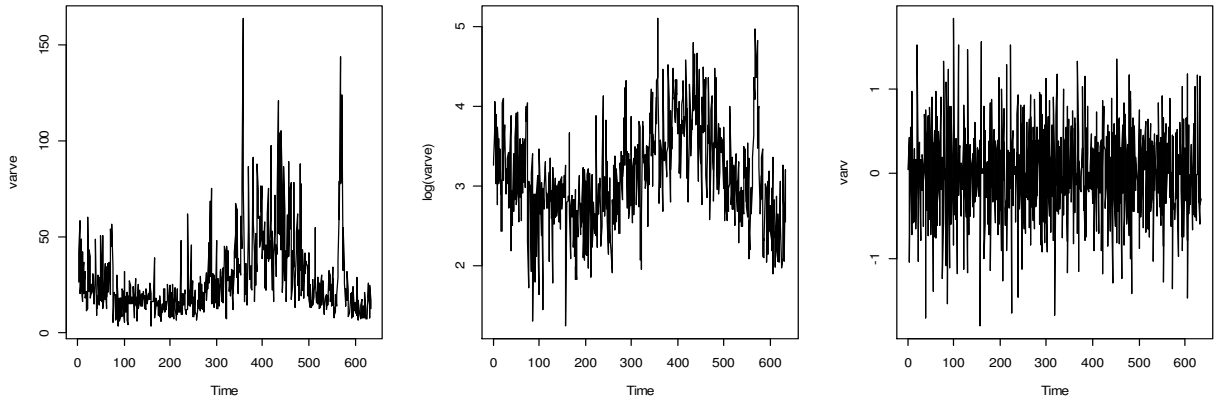


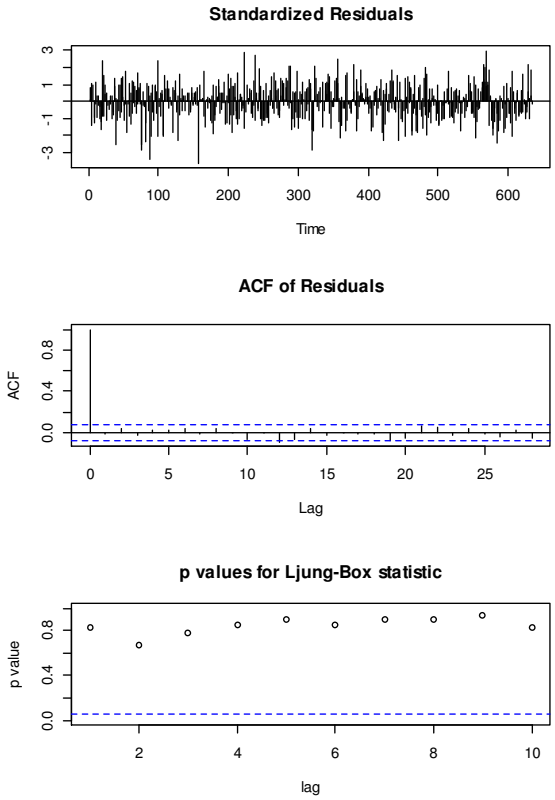
Fig. 2.3. The right graph of `varv` is stationary

To create the model of `dlvarv`, run

```
(varv.mod1=auto.arima(dlvarv))
Series: dlvarv
ARIMA(1,0,1) with zero mean

Coefficients:
         ar1      ma1
      0.2330  -0.8858
s.e.  0.0518   0.0292

AIC=868.88   AICc=868.91   BIC=882.23
```

(thus `dlvarv` is described by ARMA(1,1)) and it is a good model because its both terms are significant (why?) and residuals constitute a WN:

```
tsdiag(varv.mod1)     # see to the right
```



To forecast `varv` 10 years ahead, run

```
plot(forecast(varv.mod1),include=50)
```

Note that if Y_t is described by the model ARIMA(p,1,q), it means that the first difference $\Delta Y_t = Y_t - Y_{t-1}$ is ARIMA(p,0,q) or, in short, ARMA(p,q). In other words, if `lvarv = log(varv)` is described by ARIMA(1,1,1) then `dlvarv` by ARIMA(1,0,1) (this explains the meaning of the letter I). Indeed,

```
(varv.mod2=auto.arima(lvarv))
Series: lvarv
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.2330  -0.8858
s.e.  0.0518   0.0292
AIC=868.88   AICc=868.91   BIC=882.23
```

2.5 exercise. Use `arima.sim` to generate any MA(3) process (choose the parameters yourself). Use `auto.arima` to create a respective model. ◀◀

- **ARMA**

We already know that stationary processes can be written as infinite or finite sums: $Y_t = \mu + \sum_{j=0}^{\infty} k_j w_{t-j},\ \sum k_j^2 < \infty$. We cannot estimate infinitely many parameters therefore we try to simplify the model and replace the infinite sequence of coefficients 1, k_2, k_3,... with an appropriate finite set[6].

Recall that

- if the process $Y_t = \sum_{j=0}^{\infty} k_j L^j w_t$ (where $LY_t = Y_{t-1}$) is satisfactory described by the process $\left(\sum_{j=0}^{q} b_j L^j\right) w_t,\ b_0 = 1$ (i.e., $Y_t = w_t + b_1 w_{t-1} + ... + b_q w_{t-q}$), it is called an MA(q) process,
- if by the process $\frac{1}{1 - a_1 L - a_2 L^2 - ... - a_p L^p} w_t$ (i.e., $Y_t = a_1 Y_{t-1} + ... + a_p Y_{t-p} + w_t$), an AR(p) process,
- and if by the process $\frac{1 + b_1 L + ... + b^q L^q}{1 - a_1 L - a_2 L^2 - ... - a_p L^p} w_t$ (i.e., $Y_t = a_1 Y_{t-1} + ... + a_p Y_{t-p} + w_t + b_1 w_{t-1} + ... + b_q w_{t-q}$), an ARMA(p,q) process[7]. The ARMA(p,q) (or ARIMA(p,0,q)) process could also be expressed as $Y_t = \sum_{i=1}^{p} a_i Y_{t-i} + \sum_{j=0}^{q} b_j w_{t-j}$ or $(1 - a_1 L - ... - a_p L^p) Y_t = (1 + b_1 L + ... + b_q L^q) w_t$ where, if necessary, a constant on the right hand side is added.

2.6 exercise. Generate[8]

- 200-observation-long white noise
- 150-observation-long MA(2) process (for example, $y_t = 0.3 + (1 - 1.3L + 0.4L^2) w_t$)
- AR(1) process with 100 observations (for example, $(1 - 0.7L) y_t = 1.8 + w_t$)
- ARMA(1,1) process with 300 observations (for example, $(1 - 0.9L) y_t = 0.7 + (1 - 0.5L) w_t$)

Plot the processes and their sample ACF and PACF. Use the following rule to determine the type and order of respective process:

AR(p) – ACF declines, PACF = 0 if k > p
MA(q) – ACF = 0 if k > q, PACF declines
ARMA(p,q) – both ACF and PACF decline

This rule is easy to apply theoretically but not so easy for sample ACF and PACF, therefore parallel your analysis with `auto.arima`. ◀◀

[6] That is, to approximate original stationary time series with a simpler stationary process.
[7] It is not so easy to divide a polynomial by a polynomial, therefore the estimation procedure is rather complicated.
[8] This can be done with the `arima.sim` function (it generates a process with a zero mean, thus if necesary you have to calculate the mean of the process and add it to the time series generated) or directly following the definition and using the necessary loop in R.

2.7 exercise. Below you can see the seasonally adjusted quarterly Canadian employment data, 1961:01-1994:04.

```
caemp = structure(c(83.09, 82.8, 84.634, 85.377, 86.198, 86.579, 88.05, 87.925, 88.465, 88.398, 89.449, 90.556,
92.272, 92.15, 93.956, 94.811, 96.583, 96.965, 98.995, 101.138, 102.882, 103.095, 104.006, 104.777, 104.702,
102.564, 103.558, 102.986, 102.098, 101.472, 102.551, 104.022, 105.094, 105.195, 104.594, 105.813, 105.15,
102.899, 102.355, 102.034, 102.014, 101.836, 102.019, 102.734, 103.134, 103.263, 103.866, 105.393, 107.081,
108.414, 109.297, 111.496, 112.68, 113.061, 112.377, 111.244, 107.305, 106.679, 104.678, 105.729, 107.837,
108.022, 107.282, 107.017, 106.045, 106.371, 106.05, 105.841, 106.045, 106.651, 107.394, 108.669, 109.629,
110.262, 110.921, 110.74, 110.049, 108.19, 107.058, 108.025, 109.713, 111.41, 108.765, 106.289, 103.918, 100.8,
97.4, 93.244, 94.123, 96.197, 97.275, 96.456, 92.674, 92.854, 93.43, 93.206, 93.956, 94.73, 95.567, 95.546,
97.095, 97.757, 96.161, 96.586, 103.875, 105.094, 106.804, 107.787, 106.596, 107.31, 106.897, 107.211, 107.135,
108.83, 107.926, 106.299, 103.366, 102.03, 99.3, 95.305, 90.501, 88.098, 86.515, 85.114, 89.034, 88.823, 88.267,
87.726, 88.103, 87.655, 88.4, 88.362, 89.031, 91.02, 91.673, 92.015), .Tsp = c(1961, 1994.75, 4), class = "ts")
```

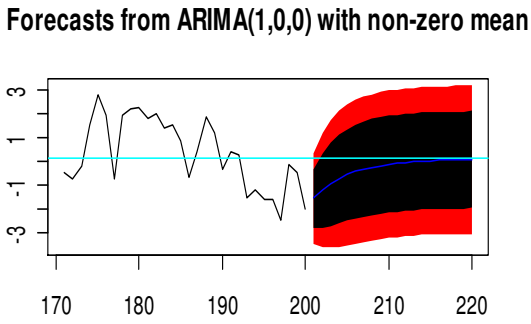
Analyze the series. ◀◀

• Forecasting Stationary Processes

The white noise process is a process with „no memory“[9], therefore its forecast is of no interest – it is always the mean. On the other hand, AR, MA, or ARMA processes correlate with their past, thus we may use this information. If the process is modeled with the `arima` function, forecast it with the `forecast` function from the forecast package.

AR(1)

```
library(forecast)
set.seed(1)
par(mfrow=c(1,1))
ar1=arima.sim(n=200,list(ar=0.8))
ar1.est=arima(ar1,order = c(1, 0, 0))
# or use auto.arima
plot(forecast(ar1.est,20),include=30)
abline(mean(ar1),0,col=5)
```



The forecast exponentially fast tends to the (sample) mean of the time series.

AR(2)

```
set.seed(2)
par(mfrow=c(1,2))
ar2=arima.sim(n=200,list(ar=c(-0.8,-0.6)))
ar2.est=arima(ar2,order = c(2, 0, 0))
# or use auto.arima
plot(forecast(ar2.est,10),include=30)
abline(mean(ar2),0,col=5)

ar2=arima.sim(n=200,list(ar=c(0.15,0.4)))
ar2.est=arima(ar2,order = c(2, 0, 0))
# or use auto.arima
plot(forecast(ar2.est,10),include=30)
abline(mean(ar2),0,col=5)
```

[9] Because it is a sequence of uncorrelated random variables.

In the first case, the roots of the inverse characteristic equation $1-a_1 z-...-a_p z^p=0$ are bigger in modulus than 1 (it means that the process is stationary) and complex , therefore the forecast tends to the mean oscillating; in the second case, the roots are real and therefore the convergence is monotone.

2.8 exercise. Analyze the behavior of the forecast in MA and ARMA cases.

2.9 exercise. Forecast the `caemp` time series two years ahead.

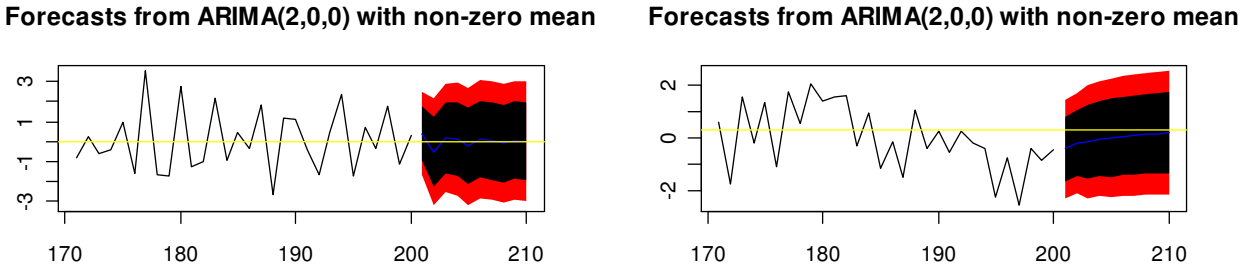


Fig. 2.4. If the inverse equation has a complex root, the forecast oscillates (left); if all the roots are real, the convergence to the mean is monotone (right).

2.10 exercise. In the data set GermanMoneySystem.txt, the variable `R` is the nominal long-term interest rate (quarterly data, 1972:1-1998:4). Plot necessary graphs. Remove the quadratic trend. Do residuals make WN? Choose the right model for residuals. Do the differences $D_t=\Delta R_t=R_t-R_{t-1}$ make WN? Choose the best model for differences. ◀◀

2.3. ARIMA model and `arima` and `auto.arima` functions

Here are some examples which should help you to systematize you knowledge about the stationary ARIMA(p,d,q) model (the necessary, but not sufficient, condition for stationarity is d=0; for example, the process ARIMA(1,0,0) is stationary if $|a_1|<1$).

- If X_t is ARIMA(p,0,q) or ARMA(p,q), i.e., $X_t-a_0-a_1X_{t-1}-...-a_pX_{t-p}=\varepsilon_t+b_1\varepsilon_{t-1}+...+b_q\varepsilon_{t-q}$, $\varepsilon_t\sim WN(0,\sigma_\varepsilon^2)$, its parameters a_i and b_j are estimated with `arima(x, c(p,0,q))`, for example,

```
library(lmtest)
library(forecast)
?unemployment
unp=unemployment[,1]
tsdisplay(unp)
arima(unp,c(2,0,3))      # ARIMA(2,0,3)
```

```
ARIMA(2,0,3) with non-zero mean
```

```
Coefficients:
         ar1      ar2      ma1      ma2     ma3  intercept
      1.7443  -0.7839  -0.6930  -0.4890  0.1821     6.4157
s.e.  0.1022   0.1010   0.1583   0.1168  0.1416     0.2279

AIC=423.26   AICc=424.63   BIC=440.76
```

Is the ARMA(2,3) model good for `unp`? - no, because `ma3` term is insignificant. Remove it and continue until you will get all significant terms (you will get ARMA(1,1)).

- If X_t is ARIMA(p,0,0) or ARMA(p,0) or AR(p), i.e., $X_t - a_0 - a_1 X_{t-1} - ... - a_p X_{t-p} = \varepsilon_t$, $\varepsilon_t \sim WN(0, \sigma_\varepsilon^2)$, its parameters a_i are estimated with `arima(x,c(p,0,0))` or `ar(x)` (the latter function chooses the order p by the minimum of AIC), for example,

```
ar(unp)

Coefficients:
      1         2         3         4
 1.0876  -0.4760    0.2978  -0.1920

Order selected 4  sigma^2 estimated as  6.165
```

or, once you know the order,

```
arima(unp,c(4,0,0))

ARIMA(4,0,0) with non-zero mean

Coefficients:
         ar1      ar2     ar3      ar4  intercept
      1.1171  -0.5594  0.4292  -0.2806     6.3341
s.e.  0.1020   0.1536  0.1598   0.1083     0.8399

AIC=422.83   AICc=423.85   BIC=437.83
```

(coefficients of the two models slightly disagree because the estimation formulas differ). The latter model is acceptable in the sense that all its terms are significant (for example, the interval 1.1171 ± 2*0.1020 does not include 0), but do its residuals make a WN?

```
library(forecast)
tsdiag(arima(unp,c(4,0,0)))    # residuals make WN
```

Yes, they do but is it the simplest (with minimum number of coefficients) correct model?

```
auto.arima(unp)

ARIMA(1,0,1) with non-zero mean

Coefficients:
         ar1     ma1  intercept
      0.6322  0.5105     6.3720
s.e.  0.0947  0.1143     1.0182

AIC=422.68   AICc=423.15   BIC=432.68

tsdiag(auto.arima(unp))   # residuals make WN
```

Thus, the simplest acceptable model of `unp` is ARMA(1,1) (it has only 3 coefficients, all the coefficients are significant and residuals are described by WN). This is quite a common situation – high order AR(p) can often be replaced by a simpler ARMA(1,1).

- The parameters of supposedly ARIMA(1,0,0) or ARMA(1,0) or AR(1) process $X_t - a_0 - a_1 X_{t-1} = \varepsilon_t$ are estimated with `arima(x,c(1,0,0))`. X_t is AR(1) if the term X_{t-1} is significant and residuals of the model make a WN.

2.11 exercise. Is the time series `unp` an AR(1) process?

- The parameter of supposedly WN or ARIMA(0,0,0) process $X_t - a_0 = \varepsilon_t$ is estimated with `arima(x,c(0,0,0))`. X_t is WN if residuals of the model make a WN.

2.12 exercise. Is the time series `unp` a WN?

Each type can be recognized by its ACF and PACF plots but we shall use `auto.arima` function to do the job (if necessary, slightly correct the output and leave only significant terms; then test whether residuals $\hat{w}_t$ make a WN)

The functions `arima` or `auto.arima` can be generalized to include the I part of the model (X_t is ARIMA(p,1,q) if its first differences $\Delta X_t = X_t - X_{t-1}$ form an ARIMA(p,0,q) time series). The ARIMA model can also include a seasonal part – the quarterly process $X_t = a_0 + a_{s1} X_{t-4} + \varepsilon_t$ can be treated as ARIMA(0,0,0)(1,0,0)[4] process and estimated with `arima(x,order=c(0,0,0),seasonal=list(order=c(1,0,0)))`.

The model

$$\begin{cases} X_t = \beta_0 + \beta_1 t + u_t \\ u_t = a_1 u_{t-1} + \varepsilon_t,\ \varepsilon_t \sim WN \end{cases}$$

or, what is the same,

$$X_t = a_1 X_{t-1} + \left(\beta_0 - a_1(\beta_0 - \beta_1) + (1 - a_1)\beta_1 t\right) + \varepsilon_t = a_1 X_{t-1} + B_0 + B_1 t + \varepsilon_t$$

describes a process such that its deviations from a straight line make an AR(1) process (it is estimated with `arima(x, order = c(1,0,0),xreg = 1:length(x))`.

2.4 example. As an artificial example, consider

```
uuu=unp+0.5*1:90
plot(uuu)
uuu.mod=arima(uuu, order = c(1,0,0),xreg = 1:90)
lines(fitted(uuu.mod),col=2)
tsdiag(uuu.mod)
```

Is it a good model? What if you replace 1 by 2?

2.4. ARCH and GARCH processes

Typically, for financial series, the return does not have a constant conditional variance (i.e., $\sigma_t^2 = \text{var}(Y_t \mid \Omega_{t-1}) \neq const$), and highly volatile periods tend to be clustered together. In other words, there is a strong dependence of sudden bursts of variability in a return on the series own past. Such processes are described by the endless family of ARCH models (ARCH, GARCH, EGARCH, TGARCH, GJRGARCH, AVGARCH, NGARCH, NAGARCH, APARCH, IGARCH etc).

The relevant R functions are contained in the fGarch, tseries, gogarch, and rugarch packages.

2.5 example. To start with, we shall repeat 2.8 example from the Lecture Notes. To import stock.xls from the dataPE folder, open the file, select and then copy both columns of it. Next, type

```
stock=ts(read.delim2("clipboard",header=T))
lStock=stock[,2]        # logged stock prices
d_lstock=diff(lStock)   # create log-differences, i.e., returns
mod1=lm(d_lstock~1)
uhat1=mod1$res; sq_uhat1 = uhat1^2 # centered and squared returns
par(mfrow=c(1,2)); plot(d_lstock); plot(sq_uhat1,col="blue",type="l")

library(dynlm)
mod2=dynlm(sq_uhat1~L(sq_uhat1)) # create the "naive" model
summary(mod2)
```

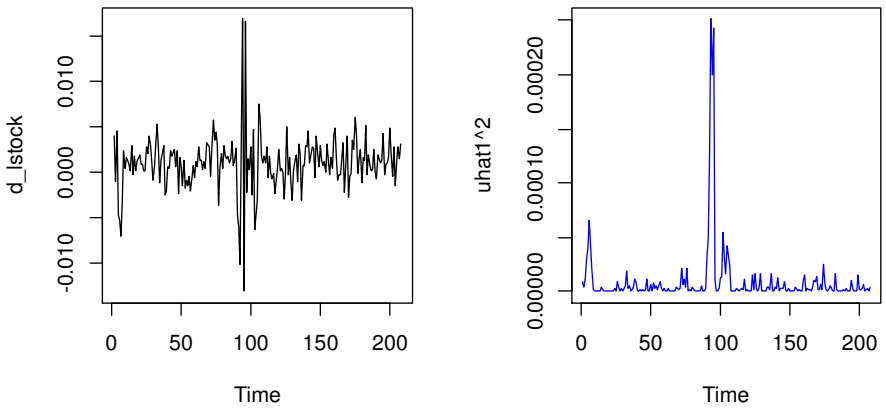


Fig. 2.5. Logged stock prices (left) and squared residuals (right)

```
Start = 2, End = 207

Coefficients:
             Estimate  Std. Error  t value Pr(>|t|)
(Intercept) 2.408e-06   1.484e-06    1.622    0.106
L(sq_uhat1) 7.370e-01   4.733e-02   15.572   <2e-16
```

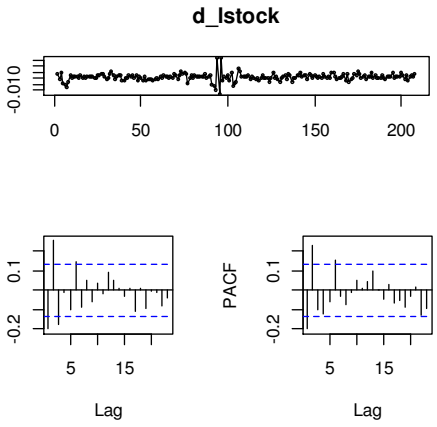
The significant coefficient in this "naive" model of volatility $u_t^2 = 0.0000024 + 0.737\, u_{t-1}^2 + w_t$ indicates that u_t is probably an ARCH(1) (=`garch(1,0)`) process. To create a respective model, type

```
library(fGarch)
```

```
mod3=garchFit(~garch(1,0), d_lstock, trace=F)
mod3
Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     1.048e-03   1.132e-04    9.259  < 2e-16 ***
omega  2.400e-06   3.904e-07    6.148 7.85e-10 ***
alpha1 6.599e-01   1.571e-01    4.199 2.68e-05 ***
```

which means that the respective model coincides with what was produced by GRETL:

$$\begin{cases} \widehat{d\_lstock} = \hat{\phi} = \mu = 0.001048 \\ \widehat{\sigma_t^2} = \omega + \alpha u_{t-1}^2 = 2.400 * 10^{-6} + 0.660 u_{t-1}^2 \end{cases}.$$



Now it is the right time to recall that `d_lstock` is not a WN:

```
library(forecast)
tsdisplay(d_lstock)
```

(the correlogram on the right indicates that `d_stock` is possibly an AR(6) process).

To find the proper conditional mean model `mod3` for `d_lstock` we use the `auto.arima` function. Then we create an AR(7) - ARCH(1) model `mod4` of the `d_lstock`:

```
mod4=auto.arima(d_lstock,max.p=10,max.q=0)
mod4   # auto.arima chooses an AR(7) process
mod5=garchFit(~arma(7,0)+garch(1,0), d_lstock, trace=F)
mod5
Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu      1.194e-03   1.730e-04    6.899 5.22e-12 ***
ar1    -1.237e-01   7.070e-02   -1.749 0.080257 .
ar2     8.081e-02   4.428e-02    1.825 0.067996 .
ar3    -3.826e-02   4.559e-02   -0.839 0.401336
ar4    -1.069e-01   3.933e-02   -2.719 0.006544 **
ar5     7.209e-03   3.970e-02    0.182 0.855914
ar6     1.636e-01   3.580e-02    4.568 4.92e-06 ***
ar7    -1.125e-01   3.389e-02   -3.318 0.000905 ***
omega   2.046e-06   3.567e-07    5.735 9.75e-09 ***
alpha1  6.503e-01   1.722e-01    3.777 0.000159 ***
```

To explore the GARCH predictions of volatility, we calculate and plot 51 observations from the middle of the data along with the one-step-ahead predictions of the corresponding volatility[10], $\widehat{\sigma_t^2}$:

```
sigma=mod5@sigma.t
plot(window(d_lstock, start=75, end=125),ylim=c(-0.020,0.035), ylab="l_stock")
lines(window(d_lstock-2*sigma, start=75, end=125), lty=2, col=4)
lines(window(d_lstock+2*sigma, start=75, end=125), lty=2, col=4)
abline(mean(d_lstock),0)
```

[10] The value of the `garchFit` function is an not a list (S3 object), but an S4 object; its components are accessed not with the $ symbol but with @.

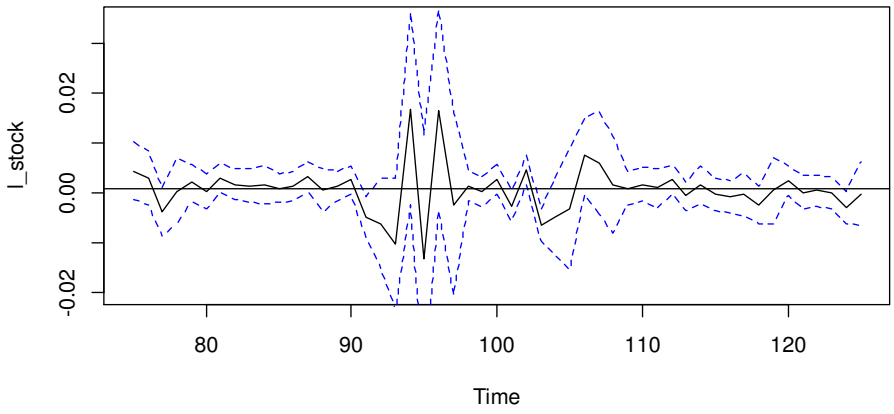


Fig. 2.6. `d_lstock` and its $\pm 2\sigma_t$ confidence region

To predict our process (see Fig. 2.7), use

```
predict(mod5, n.ahead = 2, mse="cond", plot=T)     # ◀◀
```

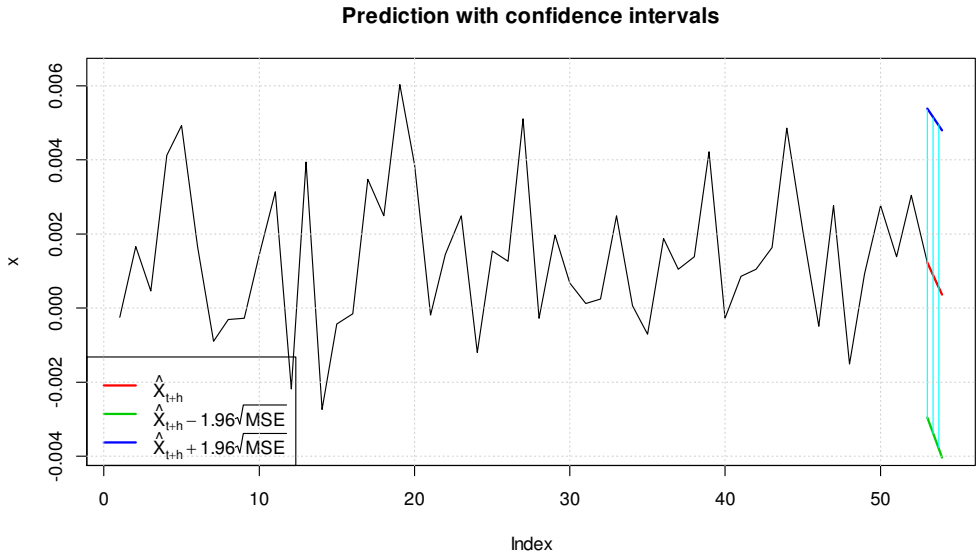


Fig. 2.7. Last 50 observations and 2-step-ahead prediction

2.6 example. (GRETL) The file S&P.txt contains monthly data on the S&P Composite index returns `ret` over the period 1954:1-2001:9, the consumer price index (CPI) inflation rate `infl`, and the change in the three-month Treasury bill (T-bill) rate `dTbill`[11]. We begin by modelling the returns series $\texttt{ret}_t$ as a function of a constant, one lag of returns $\texttt{ret}_{t-1}$, one lag of the inflation

[11] The returns series includes dividends and is calculated using the formula $r_t = (P_t + D_t - P_{t-1}) / P_{t-1}$. The inflation series is calculated as the first-difference of the natural logarithm of the CPI, and the T-bill series is the three-month T-bill rate, $dTbill_t = Tbill_t - Tbill_{t-1}$. All data are converted into percentages.

rate $\texttt{infl}_{t-1}$ and one lag of the first-difference of the three-month T-bill rate $\texttt{dTbill}_{t-1}$. We expect our model to be of the form $Y_t = \beta_0 + \beta_1 X_{1,t} + ... + \sigma(X_{1,t}, ...)w_t$ where w_t are i.i.d.r.v.'s with conditional mean 0 and conditional variance 1. We begin by adding the first lags of the necessary variables and create a usual OLS model; then we test the hypothesis H_0 : *residuals of Model 1 are normal*. In conventional regression models, rejection of the null would lead to the change of specification, but non-normality is an inherent feature of the errors from regression models for financial data, and here, as in all cases henceforth, robust standard errors must be calculated (this means that you have to check the Robust standard errors box). Note that all coefficients of corrected Model 2 are significant and have expected signs:

```
Model 2: OLS, using observations 1954:02-2001:08 (T = 571)
Dependent variable: ret
HAC standard errors, bandwidth 6 (Bartlett kernel)

             coefficient   std. error   t-ratio   p-value
  ---------------------------------------------------------
  const        1.17715     0.239720      4.911    1.19e-06 ***
  infl_1      -1.17945     0.533158     -2.212    0.0274   **
  dTbill_1    -1.25007     0.343281     -3.642    0.0003   ***
  ret_1        0.207991    0.0445415     4.670    3.77e-06 ***

R-squared             0.101059   Adjusted R-squared   0.096303
Log-likelihood       -1482.848   Akaike criterion     2973.697
Schwarz criterion     2991.086   Hannan-Quinn         2980.481
rho                   0.015237   Durbin-Watson        1.945707
```

Residuals of Model 2 constitute WN (test) but their squares do not, residuals are still non-normal (test), thus we shall test the residuals for ARCH(6) – in the Model 2 window, go to Tests| ARCH:

```
Test for ARCH of order 6

             coefficient    std. error    t-ratio    p-value
  ---------------------------------------------------------
  alpha(0)     7.40357       1.26960       5.831      9.31e-09 ***
  alpha(1)     0.105877      0.0440643     2.403      0.0166   **
  alpha(2)    -0.00339383    0.0442404    -0.07671    0.9389
  alpha(3)     0.0552574     0.0442307     1.249      0.2121
  alpha(4)    -0.000128459   0.0442403    -0.002904   0.9977
  alpha(5)     0.0521755     0.0442754     1.178      0.2391
  alpha(6)     0.102231      0.0441054     2.318      0.0208   **

  Null hypothesis: no ARCH effect is present
  Test statistic: LM = 16.1293
  with p-value = P(Chi-square(6) > 16.1293) = 0.0130763
```

The `alpha(6)` coefficient is significant and the hypothesis $H_0 : \alpha_1 = ... = \alpha_6 = 0$ in

$$\begin{cases} \varepsilon_t = \sigma_t w_t \\ \sigma_t^2 = E(\varepsilon_t^2 \mid \Omega_{t-1}) = \omega + \alpha_1 \varepsilon_{t-1}^2 + ... + \alpha_q \varepsilon_{t-q}^2 \end{cases}$$

is rejected, thus the residuals of Model 2 make ARCH(6). As a next step, we combine the Model 1 for conditional mean and Model 2 for conditional variance: in the main GRETL window, go to Mo-

del| Time series| GARCH... | check the „Robust standard errors“ box, and choose GARCH p: 0, ARCH q: 4 (alas, GRETL does not allow for higher order of ARCH):

```
Model 3: GARCH
Dependent variable: ret

             coefficient   std. error      z      p-value
  ---------------------------------------------------------
  const       1.32794      0.240500       5.522    3.36e-08 ***
  infl_1     -1.33020      0.567450      -2.344    0.0191   **
  dTbill_1   -1.10767      0.409915      -2.702    0.0069   ***
  ret_1       0.203983     0.0526545      3.874    0.0001   ***

  alpha(0)    6.78984      1.18074        5.750    8.90e-09 ***
  alpha(1)    0.155438     0.100847       1.541    0.1232
  alpha(2)    0.0338954    0.0441612      0.7675   0.4428
  alpha(3)    0.190236     0.123270       1.543    0.1228
  alpha(4)    0.0189580    0.0389964      0.4861   0.6269

Log-likelihood      -1473.048   Akaike criterion      2966.096
Schwarz criterion    3009.570   Hannan-Quinn          2983.057
```

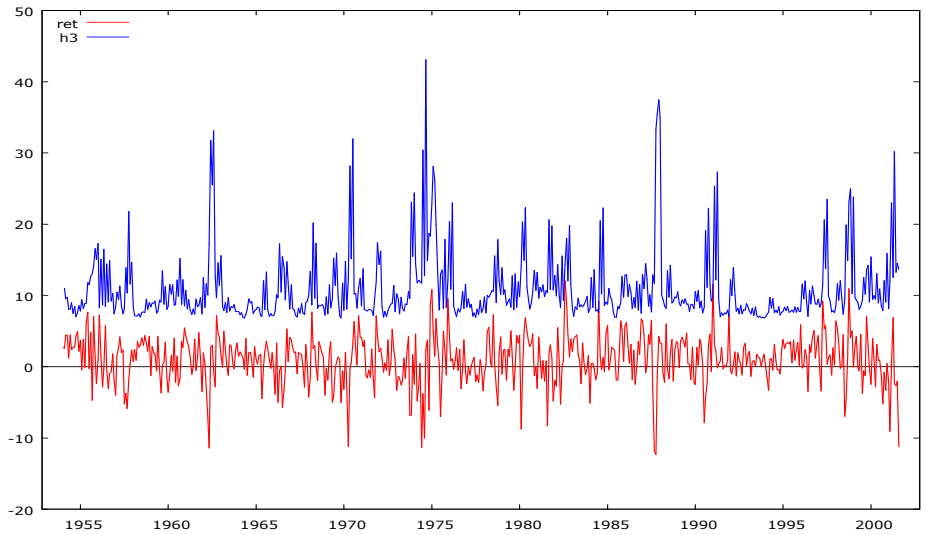
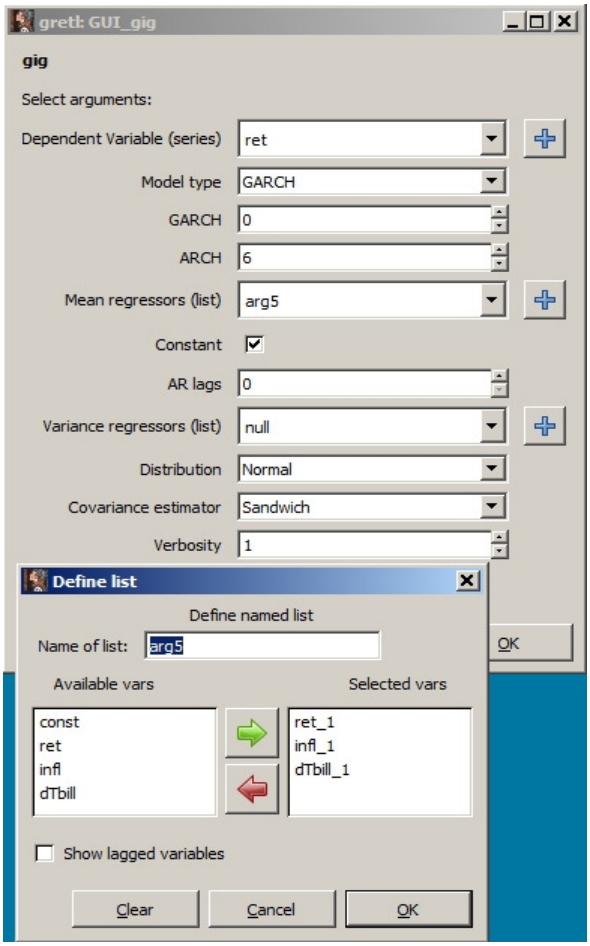


Fig. 2.8. The conditional heteroscedasticity of the S&P returns is clearly visible in both graphs (to obtain conditional error variance `h3`, go to Save| Predicted error variance); the spike in the conditional variance around observation 1988 corresponds to the crash of October 1987

GRETL, like R, allows to extend it by down-loading packages from the main server: in the GRETL window, go to File| Function files| On server...| gig (the `gig` function allows to take order higher than 4 and also to choose out of many more ARCH variants). If you do not find gig (= GARCH in GRETL) package on server, download it from http://www.econ.univpm.it/lucchetti/gretl/gig.zip and copy the unzipped folder to Program Files\ gretl\functions). Now go to File| Function files| On local machine…, click on `gig` and fill in the boxes as shown on the right.




```
Model: GARCH(0,6) [Bollerslev] (Normal)*
Dependent variable: ret
Sample: 1954:02-2001:08 (T = 571), VCV me-
thod: Robust

    Conditional mean equation

             coefficient   std. error      z       p-value
  ---------------------------------------------------------
  const         1.21848     0.270645       4.502   6.73e-06 ***
  ret_1         0.218684    0.0542242      4.033   5.51e-05 ***
  infl_1       -0.918096    0.796982      -1.152   0.2493
  dTbill_1     -0.966632    0.427802     ^-2.260   0.0239   **

    Conditional variance equation

             coefficient   std. error       z      p-value
  ----------------------------------------------------------
  omega       5.33612       1.20031       4.446     8.76e-06 ***
  alpha_1     0.123039      0.0855170     1.439     0.1502
  alpha_2     0.0146264     0.0508976     0.2874    0.7738
  alpha_3     0.133004      0.104548      1.272     0.2033
  alpha_4     0.00135661    0.0325198     0.04172   0.9667
  alpha_5     0.124912      0.129797      0.9624    0.3359
  alpha_6     0.147888      0.0853039     1.734     0.0830   *

  Llik:  -1467.41584  AIC:   2956.83168
  BIC:    3004.65296  HQC:   2975.48864
```

The Model is ARCH(6), i.e., a model of high order with many parameters, and a common advice is to replace the model by GARCH(1,1).

```
Model 4: GARCH, using observations 1954:02-2001:08 (T = 571)
Dependent variable: ret
QML standard errors

             coefficient   std. error      z       p-value
  ---------------------------------------------------------
  const         1.31222     0.224066       5.856   4.73e-09  ***
  infl_1       -1.31716     0.577148      -2.282   0.0225    **
```

```
  dTbill_1    -1.19681     0.370105     -3.234   0.0012    ***
  ret_1        0.185278    0.0449396     4.123   3.74e-05  ***

  alpha(0)     1.19981     0.556395      2.156   0.0311    **
  alpha(1)     0.119459    0.0511436     2.336   0.0195    **
  beta(1)      0.775665    0.0691383    11.22    3.29e-029 ***

Mean dependent var   0.994713   S.D. dependent var   3.428556
Log-likelihood      -1472.312   Akaike criterion     2960.624
Schwarz criterion    2995.403   Hannan-Quinn         2974.193
```

The parameters on `infl_1` and `dTbill_1` maintain their negative signs, and the estimated parameters in the equation for the conditional variance are highly significant. The information criteria computed (Akaike, Schwarz and Hannan-Quinn) allow the fit of competing models to be compared while penalizing for additional variables (the aim is to minimize the criteria). On the basis of the Schwarz criteria, which is often the preferred method of comparing fitted models in applied time series analysis, the GARCH(1,1) specification would be preferred to the ARCH(4) specification.

(R) Similar analysis may be performed in R with the help of the following script.

```
SP = ts(read.table(file.choose(),header=TRUE),
freq=12,start=1954) # navigate to S&P.txt
head(SP)
library(dynlm)
mod1 = dynlm(ret~L(ret)+L(infl)+L(dTbill),data=SP)
summary(mod1)
shapiro.test(mod1$res) # normality is rejected
library(sandwich)
library(lmtest)
# variance-covariance matrix Heteroskedasticity Corrected
# „HC“ means White‘s estimator
# z test (using a normal approximation) is performed
coeftest(mod1,df=Inf,vcov=vcovHC(mod1,"HC")) # all coefficients are still signi-
                                             # ficant
z test of coefficients:

             Estimate Std. Error z value  Pr(>|z|)
(Intercept)  1.177149   0.228283  5.1565 2.516e-07 ***
L(ret)       0.207991   0.048629  4.2771 1.893e-05 ***
L(infl)     -1.179451   0.521536 -2.2615 0.0237285 *
L(dTbill)   -1.250071   0.328941 -3.8003 0.0001445 ***

library(forecast)
mod1res = mod1$res
# mod1$res is probably an ARCH process:
tsdisplay(mod1res)      # WN
windows()               # open a new graphical window
tsdisplay(mod1res^2)    # not WN

library(fGarch)
fit60 = garchFit(formula = ~ garch(6,0), trace = FALSE, data=mod1res)
summary(fit60)

fit11 = garchFit(formula = ~ garch(1,1), trace = FALSE, data=mod1res)
summary(fit11)

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
```

```
mu      -1.510e-16   1.292e-01     0.000  1.00000
omega    1.151e+00   5.183e-01     2.221  0.02636 *
alpha1   1.139e-01   4.002e-02     2.846  0.00442 **
beta1    7.851e-01   6.584e-02    11.925  < 2e-16 ***

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  36.679    1.084565e-08 # Residuals are
 Shapiro-Wilk Test  R    W      0.9851698 1.482773e-05 # non-normal.
 Ljung-Box Test     R    Q(10)  10.01125  0.4395069    # Residuals
 Ljung-Box Test     R    Q(15)  22.05404  0.1063948    # form
 Ljung-Box Test     R    Q(20)  27.39449  0.1245256    # a WN.
 Ljung-Box Test     R^2  Q(10)  7.909245  0.6377014    # Squared residuals
 Ljung-Box Test     R^2  Q(15)  13.07198  0.5967378    # form
 Ljung-Box Test     R^2  Q(20)  17.00733  0.6524979    # a WN.
 LM Arch Test       R    TR^2   8.202932  0.7690776    # No more ARCH in res.

Information Criterion Statistics:
     AIC      BIC      SIC     HQIC
5.172060 5.202514 5.171962 5.183941
```

Here we did the ARCH analysis in two steps and the green output is quite close to GRETL's Model 4. The R's package `rugarch` allows to include the regression part into the model and combine these two steps in one:

```
library(rugarch)

ret=SP[,1];infl=SP[,2];dTbill=SP[,3]
SPEC06=ugarchspec(variance.model=list(garchOrder=c(0,6)),
mean.model=list(armaOrder=c(0,0),include.mean=TRUE,
external.regressors=cbind(ret[1:571],infl[1:571],dTbill[1:571])))
FIT06 = ugarchfit(data = ret[2:572], spec = SPEC06)
FIT06

SPEC11=ugarchspec(variance.model=list(garchOrder=c(1,1)),mean.model=
list(armaOrder=c(0,0),include.mean=TRUE,
external.regressors=cbind(ret[1:571],infl[1:571],dTbill[1:571])))
FIT11 = ugarchfit(data = ret[2:572], spec = SPEC11)
FIT11

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu       1.31189    0.230362   5.6949 0.000000
mxreg1   0.18500    0.043915   4.2126 0.000025
mxreg2  -1.31970    0.547745  -2.4093 0.015982
mxreg3  -1.19513    0.361096  -3.3097 0.000934
omega    1.19863    0.597928   2.0046 0.045001
alpha1   0.11992    0.047741   2.5118 0.012011
beta1    0.77568    0.069656  11.1358 0.000000
```

(the `FIT11` model is very close to Model 4!) Explain the meaning of numbers below:

```
Q-Statistics on Standardized Residuals
------------------------------------
             statistic p-value
Lag[1]         0.09045  0.7636
Lag[p+q+1][1]  0.09045  0.7636
Lag[p+q+5][5]  4.87537  0.4313
d.o.f=0
H0 : No serial correlation
```

```
Q-Statistics on Standardized Squared Residuals
------------------------------------
                statistic p-value
Lag[1]          0.002856  0.9574
Lag[p+q+1][3]   1.136267  0.2864
Lag[p+q+5][7]   7.423113  0.1910
d.o.f=2

ARCH LM Tests
------------------------------------
             Statistic DoF P-Value
ARCH Lag[2]      1.179   2  0.5547
ARCH Lag[5]      2.436   5  0.7862
ARCH Lag[10]     7.898  10  0.6388

forc11 = ugarchforecast(FIT11, n.ahead=20)
forc11
plot(forc11,which="all")
```

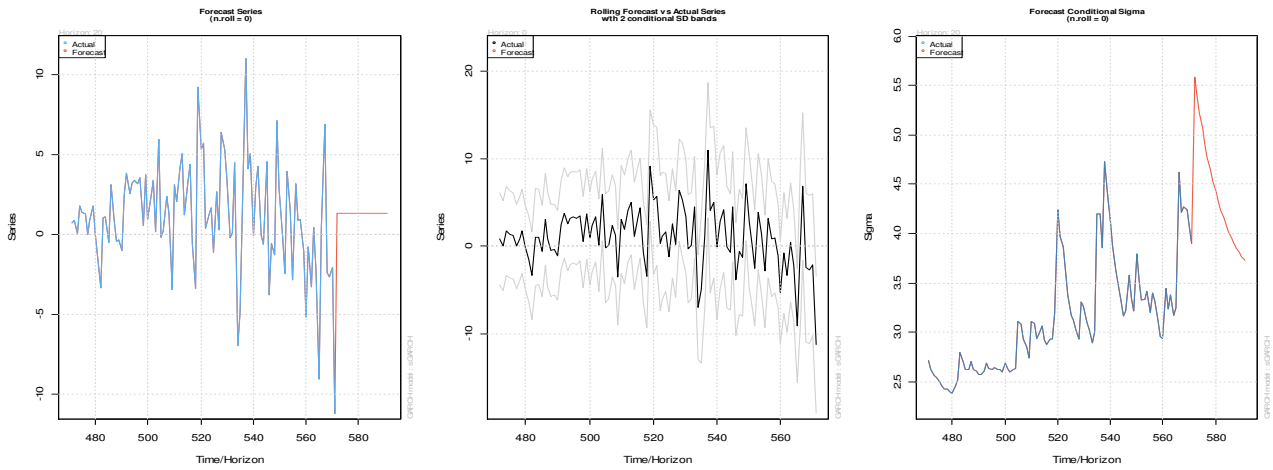


Fig. 2.9. Historical data of the `ret` series ± 2 conditional standards (center) and the 20-steps-ahead forecast of conditional standard (right)

2.13 exercise. The ARCH effects are better seen in high frequency (for example, daily) data. The data set `dmbp` in `rugarch` contains two columns:

1. The daily percentage nominal returns `ret` computed as $100\,(\log P_t - \log P_{t-1})$ where P_t is the bilateral Deutschemark/British pound rate constructed from the corresponding U.S. dollar rates.
2. A dummy variable `mo` that takes the value of 1 on Mondays and other days following no trading in the Deutschemark or British pound/ U.S. dollar market during regular European trading hours and 0 otherwise.

a) Some people say that means and/or variances of returns on `mo==1` differ from those on the rest days. Test.

b) Use GRETL or R and test `ret` for ARCH effects. Add `mo` to both conditional mean and conditional variance parts of the model. Are these terms necessary? Create a proper GARCH(1,1) model.
c) Forecast conditional variance or standard 10 days ahead.
d) Plot several relevant graphs.

```
ret=dmbp[,1]
mo=dmbp[,2]
plot(ret,type="l")
abline(mean(ret),0)
mean(ret[mo==0])
mean(ret[mo==1])
mod=lm(ret~mo)
summary(mod)
var(ret[mo==0])
var(ret[mo==1])
var.test(ret[mo==0],ret[mo==1])
mod=lm(ret~mo)
eps=mod$res
tsdisplay(eps)
tsdisplay(eps^2)
# with mo
sp11=ugarchspec(variance.model=list(garchOrder=c(1,1),external.regressors=matrix
(mo,ncol=1)),
mean.model=list(armaOrder=c(0,0),include.mean=TRUE,external.regressors=matrix(mo
,ncol=1)))
fi11 = ugarchfit(data = ret, spec = sp11)
fi11

# without mo
s11=ugarchspec(variance.model=list(garchOrder=c(1,1),external.regressors=matrix(
mo,ncol=1)),
mean.model=list(armaOrder=c(0,0),include.mean=TRUE))
f11 = ugarchfit(data = ret, spec = s11)
f11

for11 = ugarchforecast(f11, n.ahead=10)
for11
plot(for11,which="all")
```

3. Time Series: Decomposition

- **Once Again on White Noise**

Often it is important to establish whether a stationary process under consideration is WN. To achieve this, we shall use the graphs of the ACF and PACF functions produced by the `tsdisplay` function of the forecast package:

```
library(forecast)
set.seed(50)
rn=rnorm(100)     # what is the difference between
rn1=ts(rn)        # rn and rn1?
tsdisplay(rn1)
rn2=ts(rnorm(100))
tsdisplay(rn2)
```

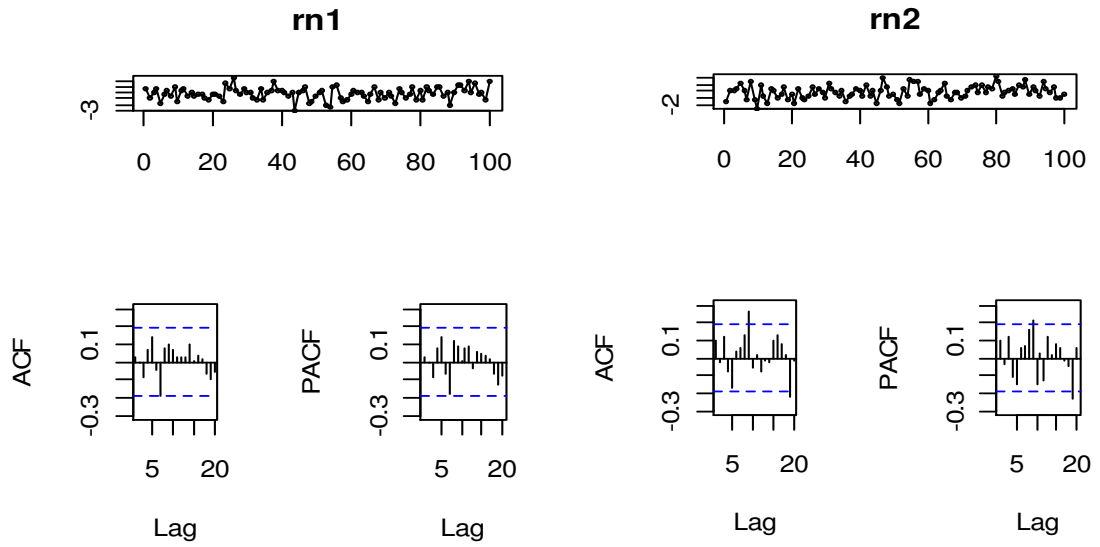
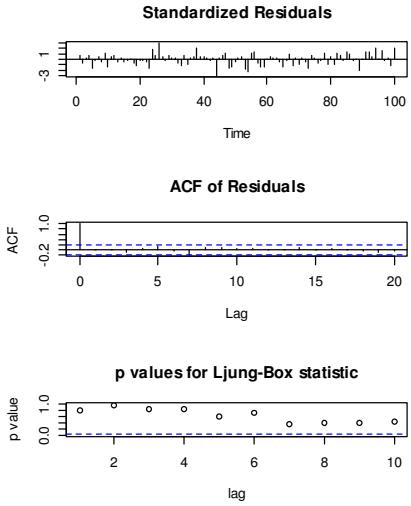


Fig. 3.1. **Rule**: if the first five to ten bars of both ACF and PACF plots are inside the blue strip, the process under consideration is WN

In "clear cases" it suffices to use the `tsdisplay` procedure. However, if the correlogram is ambiguous (bars in ACF and/or PACF are close to or slightly outside the blue band), the above could be followed by the `tsdiag` procedure which tests the residuals of the model for WN:

```
mod1=arima(rn1,c(0,0,0))      # rn1_t = β0 + ε_t
tsdiag(mod1)                  # mod1$res ~ WN?
```



(note: if ε_t is zero-mean WN, then $rn1_t$ is β_0 – mean WN). All the bubles (p values) in the bottom graph are above the 0.05 blue line, thus `rn1` is a white noise.

Maybe the simplest way, but not equivalent[1] to the previous ones, is to use

```
auto.arima(rn1)

ARIMA(0,0,0) with zero mean
sigma^2 estimated as 0.9883:  log likelihood=-141.31
AIC=284.61   AICc=284.65   BIC=287.22
```

Thus, the `auto.arima` also chooses the ARIMA(0,0,0) or WN model for `rn1` as the "best".

The following function conducts a tripple test on whiteness:

```
trippleWN <- function(x) {
library(forecast)
print(auto.arima(x))
tsdisplay(x)
oask <- devAskNewPage(TRUE)
tsdiag(arima(x,c(0,0,0)))
on.exit(devAskNewPage(oask))
}
```

As an illustration:

```
set.seed(123)
trippleWN(rnorm(100))

Series: x
ARIMA(0,0,0) with zero mean     # yes, WN

sigma^2 estimated as 0.8331:  log likelihood=-132.76
AIC=267.52   AICc=267.57   BIC=270.13
Waiting to confirm page change... # click on the graph
```

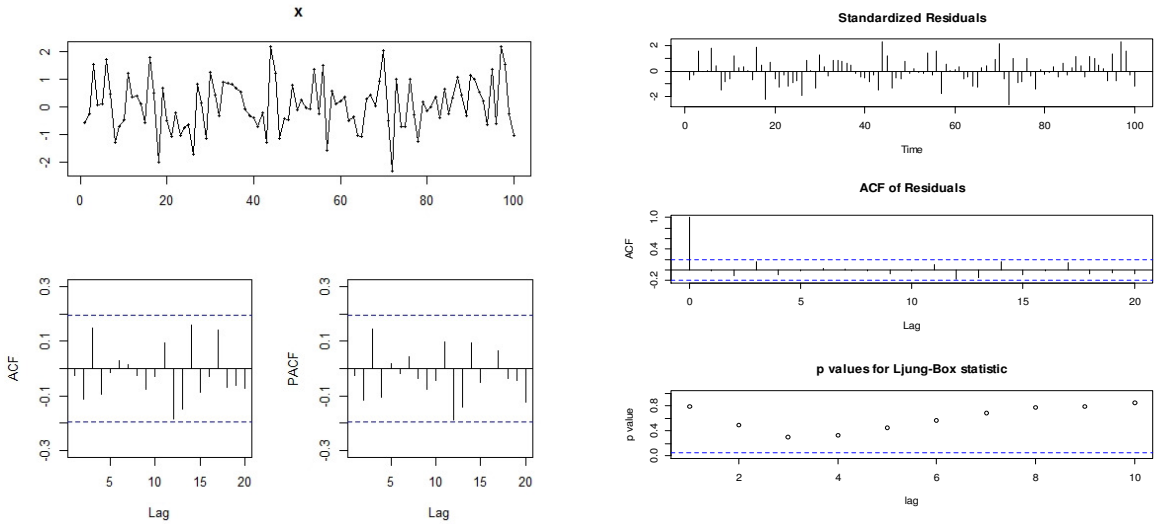


Fig. 3.2. ... and also both the correlogram (left) and the Ljung-Box test (right) confirm whiteness.

[1] Recall that `auto.arima` looks for an ARIMA model with minimum AIC (the function does not care whether the coefficients are significant nor whether residuals make white noise).

- **Decomposition**

The general mathematical representation of the decomposition approach is:

$$Y_t = f(Tr_t, S_t, \varepsilon_t), \qquad (*)$$

where

Y_t is the time series value (actual data) at period t

Tr_t is the trend-cycle component at period t

S_t is the seasonal component (or index) at period t

ε_t is the irregular (or remainder) component at period t

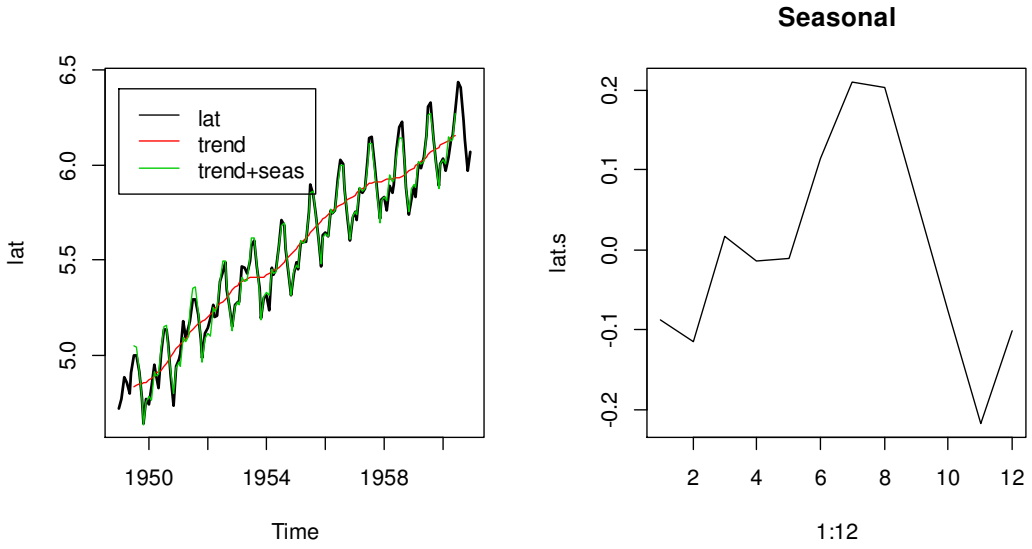
The exact functional form of (*) depends on the decomposition method actually used. A common approach is to assume that the (*) equation has the additive form

$$Y_t = Tr_t + S_t + \varepsilon_t .$$

That is, the trend-cycle, seasonal and irregular component are simply added together to give the observed series.

There are many methods to decompose a time series into its components.

3.1 example. To plot Fig. 3.2 in Lecture Notes, run the following script where the moving average method is used:




```
library(fma); data(airpass)
lat=log(airpass)
par(mfrow=c(1,2))
plot(lat,lwd=2)                    # plot original time series
lat.tr=filter(lat,filter=c(1/24,rep(1/12,11),1/24))
lines(lat.tr,col=2)                # add trend
lat.sez=lat-lat.tr
lat.se=matrix(lat.sez,ncol=12,byrow=TRUE)
```

```
lat.s=apply(lat.se,2,mean,na.rm=TRUE)      # estimate the seasonal component
lines(lat.tr+lat.s,col=3)                  # lat.s cyclically repeats
legend(1949,6.4,c("lat","trend","trend+seas"),lty=1,col=c(1,2,3))
plot(1:12,lat.s,type="l",main="Seasonal")
```

Note that this method does not allow to produce forecasts. However, this is possible when using the methods introduced below.

- **The Global Method of OLS**

Consider the `airpass` data set from the fma package.

```
library(fma); data(airpass);plot(airpass)
at=time(airpass); at                  # at is the sequence of dates
# only quadratic trend
air1.lm=lm(airpass~at+I(at^2))
summary(air1.lm)
time.f=seq(1949,1964-1/12,by=1/12)   # our time sequence plus 36 months for forecast
air1.f=predict(air1.lm,data.frame(at=time.f))
par(mfrow=c(1,3)); plot(time.f,air1.f,type="l",lty=2);lines(airpass,col=2)

# quadratic trend and monthly seasonality
month = seasonaldummy(airpass)
air2.lm=lm(airpass~at+I(at^2)+month);summary(air2.lm)
air2.f=predict(air2.lm,data.frame(at=time.f,month=I(rbind(month,seasonaldummyf(airpass
,36)))))
plot(time.f,air2.f,type="l",lty=2);lines(airpass,col=2)

# model for logs
air3.lm=lm(log(airpass)~at+I(at^2)+month);summary(air3.lm)
air3.f=predict(air3.lm,data.frame(at=time.f,month=
I(rbind(month,seasonaldummyf(airpass,36)))))
plot(time.f,air3.f,type="l",lty=2)
lines(log(airpass),col=2)
```

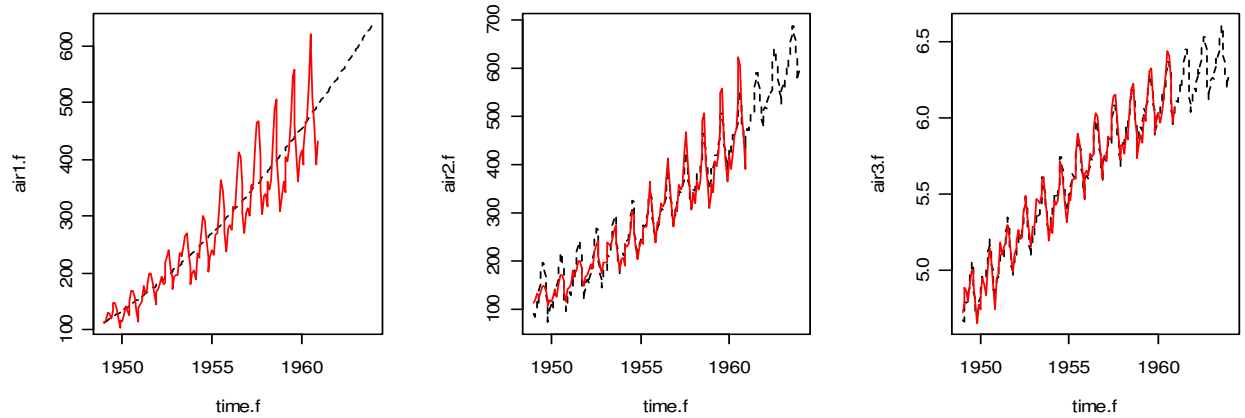
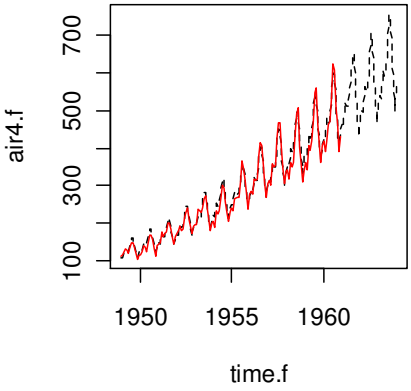


Fig. 3.3. Three quadratic trends and forecasts. Dependent variable is `airpass` - without seasonal part (left) and with seasonal component (center); dependent variable is `log(airpass)` – with seasonal component (right)

The central graph in Fig. 3.2 corresponds to the OLS model $airpass_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_{3,1} dm1_t + ... + \beta_{3,11} dm11_t + \varepsilon_t$, extended for 36 months into the future. Clearly, it is not a very good model since `airpass` fluctations are increasing in size together with the level of

`airpass` and additive model is unable to catch this effect. The common solution to this problem is to take logarithms of `airpass`. The model $\log(airpass)_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_{3,1} dm1_t + ... + \beta_{3,11} dm11_t + \varepsilon_t$ is free of the above mentioned weakness. The only problem appears when we move back to `airpass` – the "true" formula is $\widehat{airpass}_t = \exp(\widehat{\log(airpass)}_t + \hat{\sigma}^2/2)$ (note that $\exp(\hat{\sigma}^2/2) = 1.02$, thus this correction is almost invisible):




```
par(mfrow=c(1,1))
plot(time.f,exp(air3.f+
summary(air3.lm)$sigma/2),
type="l",lty=2,ylab="air4.f")
lines(airpass,col=2)
```

3.1 exercise. Analyze the residuals of all three models using the functions of the form `plot(air1.lm$res,type = "l")`. Are the residuals stationary? (decide by eye). Do the residuals make a WN? (use `tsdisplay`).

3.2 exercise. Extract linear, quadratic, and cubic trends from `shipm`. Do the residuals make WN? Should we include a seasonal component? Wouldn't it be better to use logs of `shipm`? Choose the best model (that is, the one whose 1. Residuals are WN and 2. AIC is minimal) and forecast `shipm` 36 months ahead.

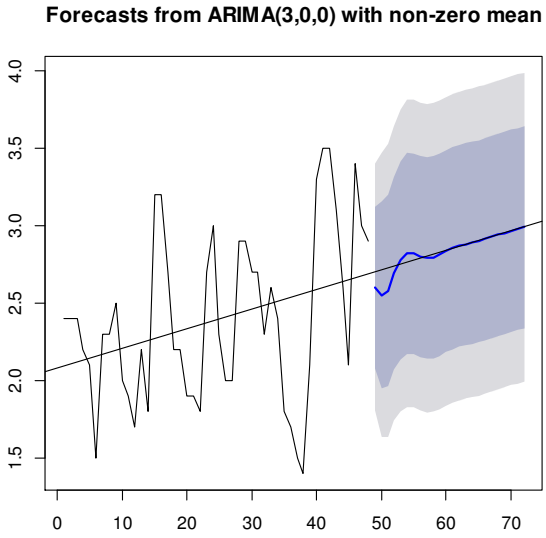
3.3 exercise. The monthly time series `co2`, 1959:01 – 1997:12, contains atmospheric concentrations of CO_2 expressed in parts per million (ppm). Describe the data through linear, quadratic and cubic trends with, probably, seasonal component. Which model is the best?

3.2 example. Earlier we assumed that the `lh` series is stationary and created its model (see 2.4 exercise). However, it seems that the series is trending (TS?), therefore we shall present another model of that series here.

```
library(datasets)
data(lh)
lh                      # length=48
plot(lh)                # it seems that the series contains incr. trend(?)

lh.lm=lm(lh~time(lh))   # what is time(lh)?
abline(lh.lm,col=2)

summary(lh.lm)          # significant trend; however, residuals
trippleWN(lh.lm$res)    # do not make WN, lm is not reliable

library(forecast)
lh1.arima=auto.arima(lh,xreg=time(lh))     # with linear trend
summary(lh1.arima)      # ar(3) term not significant, trend significant
lh2.arima=auto.arima(lh,max.p=1,xreg=time(lh))
summary(lh2.arima)        # ar(1)term is significant, trend is not
trippleWN(lh2.arima$res)  # residuals make WN
```

The results of our analysis are ambiguous, the last two lines of our code say that the trend is insignificant (if it were, then `lh` would be called a TS series). Just for illustration, taking into account that the trend in `lh1.arima` is significant, we can forecast the series 24 months ahead:

```
plot(forecast(lh1.arima,
xreg=48+(1:24)))
abline(lh1.arima$coef[4:5])
```



◀◀

- **The Global Method of Nonlinear Least Squares**

If economics increases every year the same percent, then it grows exponentially. This argument explains why the exponential model is so popular. However, modeling is rather complicated because to extract trend here we need nonlinear regression.

```
library(fma)
data(shampoo) # ?shampoo - sales of shampoo over a three year period; do
              # the sales are seasonal?
TIME=as.numeric(time(shampoo))
par(mfrow=c(1,2))
plot(shampoo) # Sales grow roughly exponential (see Fig. 3.3)
shampoo.nls=nls(shampoo~a+b*exp(c*TIME),start=list(a=100,b=10,c=1))
# I had to experiment when choosing the starting values
lines(TIME,predict(shampoo.nls))
plot(summary(shampoo.nls)$res) # The spread of residuals is more or less
                               # constant, however they are probably
                               # non-normal (their histogram is not symmetric
                               # - test)
abline(0,0)
summary(shampoo.nls)
```

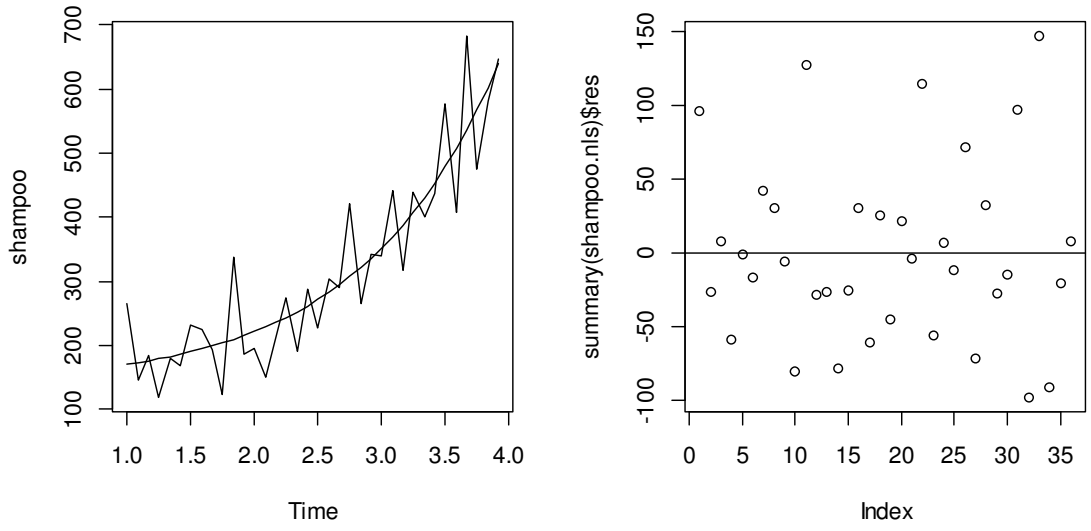


Fig. 3.4. `shampoo` time series and exponential trend (left); residuals (right)

One of the drawbacks of the OLS is its globality – if the time series departs for some reason from its trend at the final moment *T*, this will change all the coefficients of the model and therefore the predicted value at moment $t = 1$. The local methods are free of this defect – their forecast at moment *t* are (mostly) influenced only by the values at close time moments.

- **<u>The Local Method of Exponential Smoothing</u>**

We know three variants of exponential smoothing, they are for series without trend, with linear trend and with seasonality. In the first case (simple exponential smoothing), the procedure to produce smoothed series is described as follows:

1. Initialize at $t = 1$: $\hat{Y}_1 = Y_1$
2. Update: $\hat{Y}_t = \alpha Y_t + (1-\alpha)\hat{Y}_{t-1},\ t = 2,...,T.$
3. Forecast: $\hat{Y}_{T+h,T} = \hat{Y}_T,\ h = 1,2,....$

We call $\hat{Y}_t$ the estimate of the *level* at time *t*. The smoothing parameter α is in the unit interval, $\alpha \in [0,1]$. The smaller α is, the smoother the estimated level. As α approaches 0, the smoothed series approaches constancy, and as α approaches 1, the smoothed series approaches point-by-point interpolation.

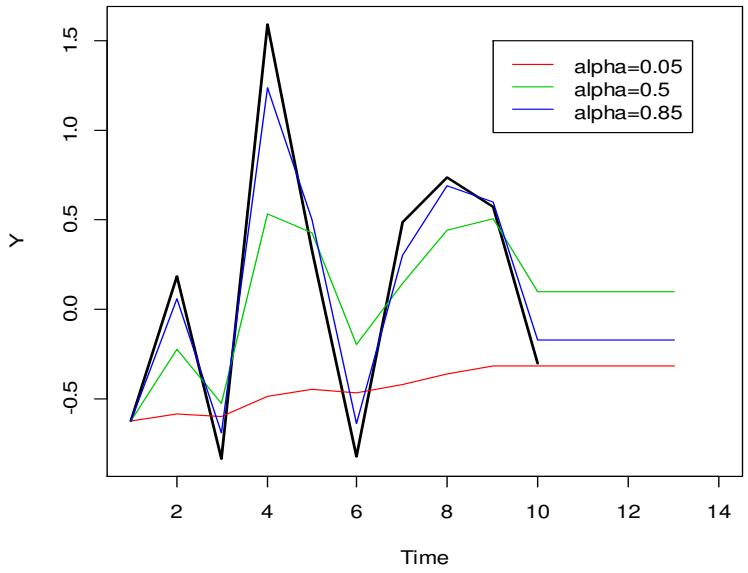


Fig. 3.5. Three variants of smoothing the original series (black)

R allows to use the automated procedure to choose the smoothing parameters: the function `ets` selects the best variant automatically (by the minimum of AIC) and estimates its parameters.

```
library(forecast)
library(datasets)
data(AirPassengers)
l.ap.ets <- ets(log(AirPassengers))
summary(l.ap.ets)
plot(forecast(l.ap.ets),include=36)
```

3.4 exercise.

```
library(fma)
data(housing)
?housing      # A three-dimensional time series
tsp(housing)
plot(housing) # Examine the 2nd component
hc=housing[,"construction"]
```

Use exponential smoothing to estimate trend and seasonal part; predict `hc` 12 months ahead.

```
plot(forecast(hc,h=12),include=36) # here, to write hc is the same as to
                                   # write ets(hc)
```

3.5 exercise. Predict `shampoo` 1 year ahead with the `nls` function.

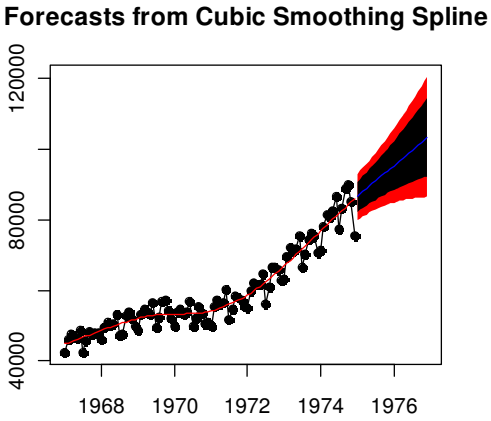
```
plot(shampoo,xlim=c(1,5),ylim=c(100,1400))
newTIME=seq(1,5-1/12,by=1/12)
new.pred=predict(shampoo.nls,newdata=data.frame(TIME=newTIME))
lines(newTIME,new.pred,col=2) # not very convincing
```

- **Local Linear Forecast Using Cubic Splines**

The cubic smoothing spline model is equivalent to an ARIMA(0,2,2) model (we shall present this model later) but with a restricted parameter space. The advantage of the spline model over the full ARIMA model is that it provides a smooth historical trend as well as a linear forecast function.

We shall use the function `splinef` from the forecast package to extract linear forecast from the `shipm` time series and forecast it for 24 months:

```
library(forecast)
?splinef
fcast <- splinef(shipm,h=24)
plot(fcast)
summary(fcast)
```



3.6 exercise. Find the `uspop` data set and their description in R. Estimate the trend and forecast this time series for 10 years ahead. Use the method of nonlinear least squares, exponential smoothing and smoothing splines. Which of the forecasts fits best todays data (find them from the internet).

3.3 example. Lithuanian quarterly GDP, 1993:01-2012:04, can be downloaded from http://db1.stat.gov.lt/statbank/default.asp?w=1280; it is also given in L.gdp.txt. We want to decom-pose the DGP into trend and seasonal components and also random (hopefully, stationary) component. Many R functions can do the job, but only some allow to forecast the series.

```
l.gdp=ts(scan(),start=1993,freq=4) # copy and paste data from L.gdp.txt
l.gdp
opar=par(mfrow=c(1,2))
plot(l.gdp, main="Lithuanian GDP")
ll.gdp=log(l.gdp)
ll=window(ll.gdp,start=2000)  # extract a subset of ll.gdp
plot(ll, main="logs of Lithuanian GDP")
par(opar)
```

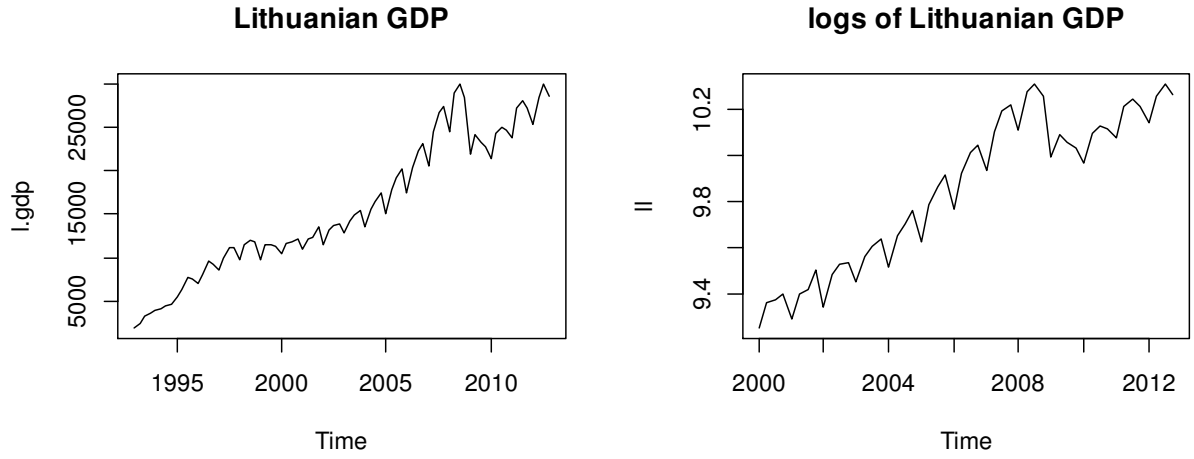


Fig. 3.6. Quarterly data of Lithuanian GDP, 1993:01-2012:04 (left) and logs of Lithuanian GDP, 2000:01-2012:04 (right)

The Lithuanian GDP is rather unregular till appr. 1996 due to the transition to the free market economy. The development of Lithuanian economy was also heavily disturbed by the 1998 Russian financial crisis, therefore we restrict ourselves to 2000:01-2012:04. We choose to analyze logarithms because its seasonal part appears to be of constant amplitude; thus, we want to find the additive decomposition $\log(\texttt{l.gdp}) = Tr_t + S_t + \varepsilon_t$. It is clear from the graph in Fig. 3.4, right, that it would be difficult to describe the trend of `ll` in analytic terms, therefore we shall apply the `decompose` function and use nonparametric decomposition.

```
dec.ll=decompose(ll)
dec.ll                  # the result of decomposition
plot(dec.ll)
library(forecast)
tsdisplay(dec.ll$rand) # draw a residuals correlogram
```

The seasonal effects can be extracted from `dec.ll` (the additive quarterly extras of `log(l.gdp)` are -0.099, 0.018, 0.041, and 0.040). The random component of `dec.ll` seems to be stationary (see Fig. 3.5), but not a white noise (see Fig. 3.5, right). However, the main problem with our procedure is that it does not allow to forecast `ll`.

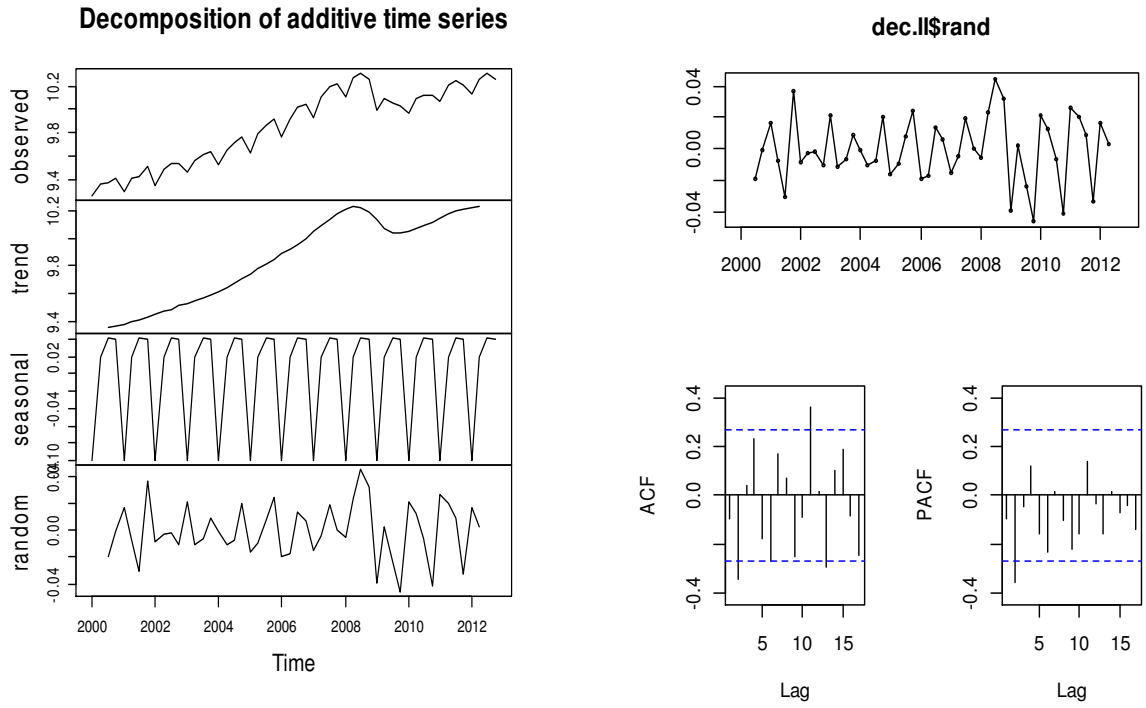


Fig. 3.7. Decomposition of `ll` (left) and the correlogram of `dec.ll$rand` (right)

One of the procedures which allows forecasting is exponential smoothing:

```
ets.ll=ets(ll)  # automatic exponential smoothing
ets.ll   # AIC -116.35
plot(forecast(ets.ll),include=16)
# plot(forecast(ll),include=16) will produce the same result
tsdisplay(ets.ll$resid)  # residuals form a WN
```

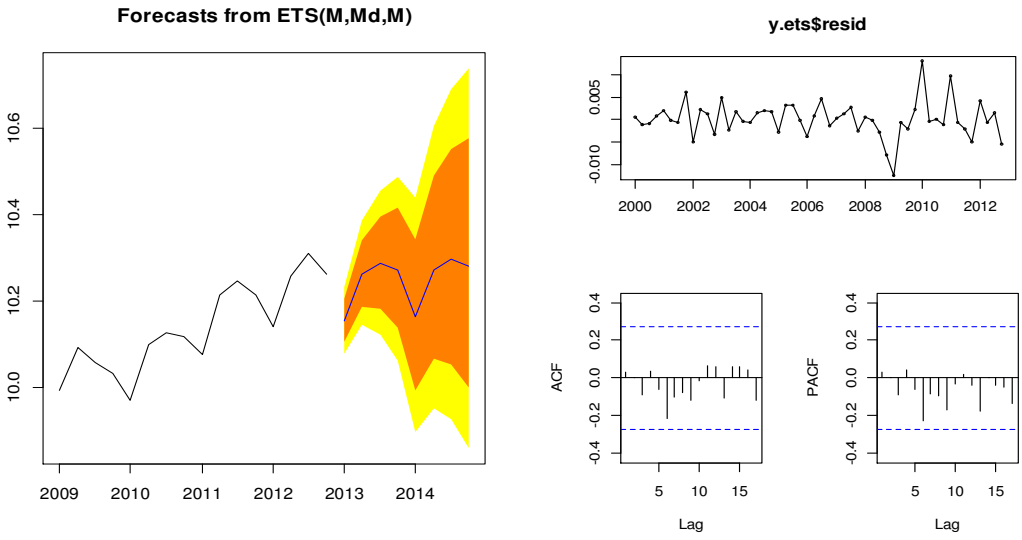


Fig. 3.8. Exponential forecasting (left) and the correlogram of residuals (right)

Later, in 4.10 exercise, we shall present more methods to analyze and forecast `ll`. ◀◀

4. Difference Stationary Time Series

4.1. Unit Roots.

In econometrics, nonstationary processes are more important than stationary. In this chapter, we shall discuss one class of such processe, the so-called processes with unit root.

Let the AR(1) time series Y_t is described by $Y_t = aY_{t-1} + w_t$, where w_t is a WN; if the root of an inverse characteristic equation $1 - az = 0$, namely, $z = 1/a$ equals 1 (to put it simply, $a = 1$), then we say that Y_t has a unit root. If Y_t is an AR(p) process, i.e., $Y_t = a_1 Y_{t-1} + a_2 Y_{t-2} + \ldots + a_p Y_{t-p} + w_t$, it is called a process with unit root if at least one root of the inverse characteristic equation $1 - a_1 z - \ldots - a_p z^p = 0$ equals +1. Warning – this is not the same as $a_1 = 1$!

Remark. If the time series Z_t is of the form $Z_t = T_t + Y_t$, where T_t is a deterministic trend (for example, $T_t = a + bt$), and (the deviation) process Y_t is AR(1) or AR(p) process with unit root, then sometimes Z_t is also called a process with unit root.

The series $Y_t = 1 \cdot Y_{t-1} + w_t$ is not a stationary process because $Y_t = (Y_{t-2} + w_{t-1}) + w_t = \ldots = Y_0 + w_1 + w_2 + \ldots + w_t$, thus $DY_t = DY_0 + t\sigma_w^2 \neq const$. In other words, the graphs of sample ACF and PACF are senseless, however they give some information about potential unit root (see Fig. 4.1).

The trajectories of the process with unit root make long "excursions" up and down. The (sample) ACF of the process decays very slowly and the first bar of the (sample) PACF is almost equal to 1 (other bars are almost zeros).

If Y_t is an AR(1) process with a unit root, then the process of differences $\Delta Y_t = Y_t - Y_{t-1} = w_t$ is a WN. If Y_t is an AR(p) process with one[1] unit root, then the process of differences is a stationary AR(p-1) process. Indeed, $w_t = (1 - a_1 L - \ldots - a_p L^p) Y_t = (1 - b_1 L - \ldots - b_{p-1} L^{p-1}) \cdot (1 - L) Y_t = (1 - b_1 L - \ldots - b_{p-1} L^{p-1}) \Delta Y_t$.

In Fig. 4.1 below, we depict three AR(1) processes $Y_t = aY_{t-1} + w_t$ (with, respectively, a=0.9, 1 and 1.1). Note that despite the fact that the a coefficients are almost equal, trajectories of the processes are notably different – in the first case the series is stationary, in the second case the series is a random walk, and in the third is a rarely met „explosion" (sometimes the process of hyperinflation can be described as explosion). In this chapter, we shall mostly deal with unit roots, namely: i) how to detect such series? and ii) how to extend such series into the future, i.e., to forecast them.

[1] Economical time series sometimes have two unit roots, but (almost) never more.

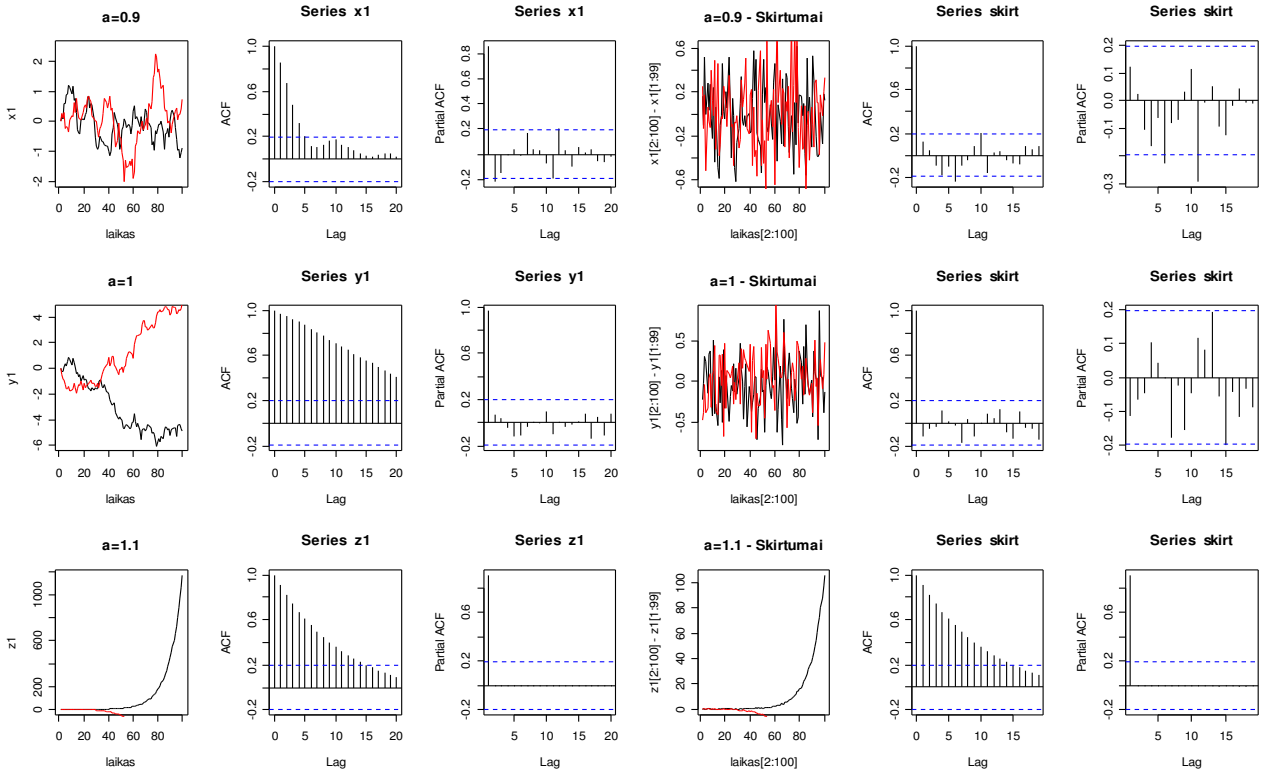


Fig. 4.1. The first row: a=0.9 (stationary process), second row: a=1 (this is a random walk; it is not a stationary process but its differences are), third row: a=1.1 (explosion – neither the process nor its differences are stationary)

Here are some definitions. If the sequence of the first differences $\Delta Y_t = Y_t - Y_{t-1}$ is stationary, the sequence Y_t is called a (one time) integrated series and is denoted by I(1); if only the series of the d*th* differences[2] is stationary, the series Y_t is the d*th* order integrated series and denoted by I(*d*) (the stationary Y_t is denoted by *I*(0)). We say that the ARMA(p+1,q) process y_t: $\Phi_{p+1}(L)y_t = \Theta_q(L)w_t$ has a unit (autoregressive) root if at least one of *p*+1 autoregressive roots equals 1, that is, if $\Phi_{p+1}(L) = \Phi'_p(L)(1-L)$. It is equivallent to that the differenced process ΔY_t is ARMA(p,q): $\Phi'_p(L)(1-L)y_t = \Phi'_p(L)\Delta y_t = \Theta_q(L)w_t$ (the original ARMA(p+1,q) process Y_t is now called ARIMA(p,1,q) process). That is to say that ARIMA(1,1,1) is a process whose first differences make an ARMA(1,1) process etc.

4.1 example. Assume that our process is described by a simple ARIMA(1,1,1) process $(1-0.4L)(1-L)Y_t = (1+0.3L)w_t$ which can be rewritten as $\frac{1-0.4L}{1+0.3L}\cdot(1-L)Y_t = (1-0.4L)\cdot(1-0.3L+(0.3L)^2-...)(1-L)Y_t = w_t$. The infinite order polynomial on the lhs can be approximated by a (sometimes quite of a large order) finite polynomial, i.e., AR(p) process. The popular Dickey-Fuller unit root test always interpret the process under consideration as an AR process.

[2] The operator of the d*th* differences is defined as $\Delta^d = \Delta(\Delta^{d-1})$. For example, $\Delta^2 Y_t = \Delta(\Delta Y_t) = \Delta(Y_t - Y_{t-1}) = Y_t - Y_{t-1} - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$.

If the d*th* differences $\Delta^d Y_t$ make a stationary ARMA(p,q) process, then the process Y_t is called an ARIMA(p,d,q) process.

4.2 example. The data set …\PE.II.data.2010\nyse.txt contains monthly index of the NewYork Stock Exchange from 1952:1 to 1996:1.

```
nyse=ts(read.table(file.choose(),header=TRUE),start=1952,freq=12)
par(mfrow=c(1,3))
plot(nyse);plot(log(nyse));plot(diff(log(nyse)))
```

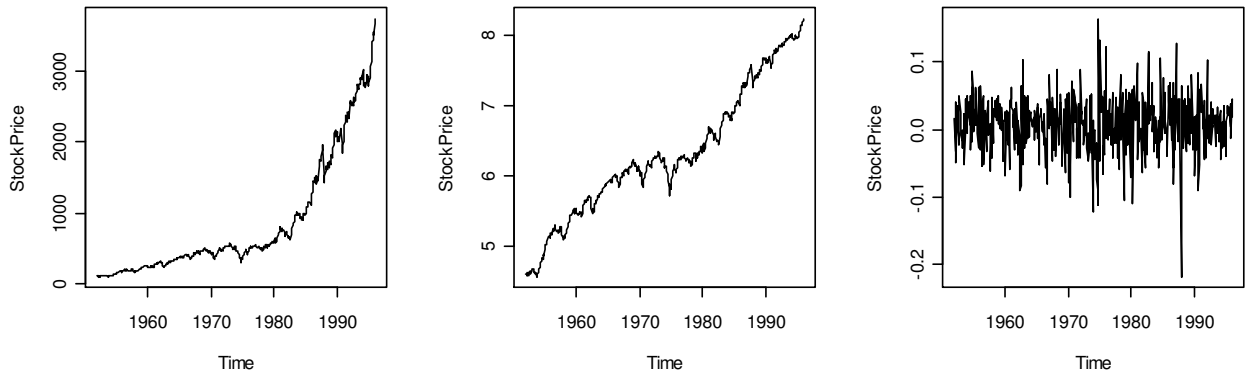


Fig. 4.2. `nyse` (left), `log(nyse)` (center), and `diff(log(nyse))` (right)

We shall model the variable `log(nyse)` (see Fig. 4.2, center) in two different but equivalent ways:

1) as a process with a linear trend whose deviations from the trend are described as the AR(1) process (possessing, probably, a unit root):

$$\begin{cases} Y_t = \alpha + \delta t + u_t \\ u_t = \varphi\, u_{t-1} + w_t \end{cases}$$

```
mod.a=arima(lnyse,c(1,0,0),xreg=time(lnyse))
summary(mod.a)
```

```
Coefficients:
         ar1  intercept  time(lnyse)
      0.9832  -144.2528       0.0763
s.e.  0.0076    12.4969       0.0063
```

2) as an autoregressive process with a linear trend:

$$Y_t = \alpha + \varphi Y_{t-1} + \tilde{\delta} t + w_t =$$
$$= (\alpha(1-\varphi) + \varphi\delta) + \varphi Y_{t-1} + \delta(1-\varphi)t + w_t$$

```
library(dynlm)
lnyse=log(nyse)
mod.1=dynlm(lnyse~time(lnyse)+L(lnyse))
```

```
summary(mod.1)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.169524   1.124569  -1.929   0.0542 .
time(lnyse)  0.001152   0.000595   1.936   0.0534 .
L(lnyse)     0.984518   0.008150 120.792   <2e-16 ***
```

The coefficients of the two models differ because of different parameterization, however both models practically coincide (their residuals in fact are zeros):

```
> summary(mod.1$res-mod.a$res)
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-1.021e-03 -6.636e-04  3.215e-05 -1.198e-04  3.638e-04  8.119e-04
```

The most important coefficient is that of the lagged term – in the first case it equals 0.984518, and in the second 0.9832 (do not forget that these are only the estimates of a_1 from $Y_t = a_1 Y_{t-1} + w_t$). The estimate is very close to 1, however we cannot test the unit root hypothesis $H_0 : a_1 = 1$ by using the presented $p-$value <2e-16 (it is to test the hypothesis $H_0 : a_1 = 0$). We also cannot test the hypothesis $H_0 : a_1 = 1$ in a standard way, i.e., through constructing the confidence interval 0.984518 $\pm 2 \cdot 0.008150$, due to the fact that $(\hat{a}_1 - 1)/s.e.$, provided H_0 is true, has not the Student but another, namely, **Dickey-Fuller** distribution.

A. We shall apply the OLS method and proceed as follows.

1. Since the order of AR is unknown, we begin with a "sufficiently high" order $p_{\max}$. It is easy to verify that the process $Y_t = \alpha + a_1 Y_{t-1} + a_2 Y_{t-2} + \ldots + a_p Y_{t-p} + \delta t + w_t$ can be rewritten as: $\Delta Y_t = \alpha + \rho Y_{t-1} + \gamma_1 \Delta Y_{t-1} + \ldots + \gamma_{p-1} \Delta Y_{t-p+1} + \delta t + w_t$ (for example, the process $Y_t = \alpha + a_1 Y_{t-1} + a_2 Y_{t-2} + \delta t + w_t$ can be expressed as $\Delta Y_t = \alpha + (a_1 + a_2 - 1) Y_{t-1} + a_2 \Delta Y_{t-1} + \delta t + w_t$). Now the original hypothesis $H_0 : \textit{the process has a unit root}$ will transform to $H_0 : \rho = 0$.

2. Let us find the "right" order of the model. To do this, use the model's output table to test the hypothesis $\gamma_{p_{\max}-1} = 0$ (if the $p-$value is >0.05, reduce the order and repeat the calculations etc). When the "right" order is found, we shall test the significance of δ.

3. The significance of ρ is tested in a **different** way:

i. If the final version contains the linear term δt, the 5% Dickey–Fuller critical value is appr. -3.45. Thus, if the $t-$ratio of the term Y_{t-1} is less than -3.45, the null $H_0 : \textit{the process has a unit root}$ is rejected and we decide that the AR process is stationary.

ii. If the final version of the model does not contain δt, the 5% Dickey–Fuller critical value is appr. -2.89.Thus, if the $t-$ratio of the term Y_{t-1} is less than -2.89, the null $H_0 : \textit{the process has a unit root}$ is rejected and we decide that the AR process is stationary.

****

We shall apply the procedure to `lnyse`. Let us begin with $p_{\max} - 1 = 4$.

```
mod.4=dynlm(d(lnyse)~L(lnyse)+L(d(lnyse),1:4)+time(lnyse))
summary(mod.4)
```

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.3687713  1.1485801  -2.062   0.0397 *
L(lnyse)          -0.0172567  0.0083590  -2.064   0.0395 *
L(d(lnyse), 1:4)1  0.0544182  0.0439376   1.239   0.2161
L(d(lnyse), 1:4)2 -0.0282789  0.0439795  -0.643   0.5205
L(d(lnyse), 1:4)3  0.0150925  0.0439341   0.344   0.7313
L(d(lnyse), 1:4)4  0.0364520  0.0439285   0.830   0.4070    nereikšmingas
time(lnyse)        0.0012584  0.0006077   2.071   0.0389 *
```

Now, instead of `L(d(lnyse),1:4)` we write `L(d(lnyse),1:3)` etc until we get such a model:

```
mod.0=dynlm(d(lnyse)~L(lnyse))
summary(mod.0)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0077315  0.0121721   0.635    0.526
L(lnyse)    -0.0001374  0.0019082  -0.072    0.943
```

We have no ground to reject $\rho = 0$ (since the $t-$value -0.072 is greater than -2.89). Thus `lnyse` has a unit root and it can be modelled as

```
mod.f=dynlm(d(lnyse)~1)
summary(mod.f)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.006865   0.001768   3.884 0.000116 ***
```

In other words, $lnyse_t = 0.0069 + lnyse_{t-1} + w_t$, i.e., this is a random walk with drift.

B. The 5% Dickey-Fuller approximate critical values were presented above. In our case, -0.072 was definitely greater than -2.89 but, generally, once the order and the necessity of δt are established, use the augmented Dickey–Fuller test implemented in `ur.df()`). Recall that this test is applied in the cases

$$Y_t = a_1 Y_{t-1} + w_t \quad (\text{„none“ option})$$
$$Y_t = a + a_1 Y_{t-1} + w_t \quad (\text{„drift“ option})$$
$$Y_t = a + a_1 Y_{t-1} + \delta t + w_t \quad (\text{„trend“ option})$$

In all three cases we test the null $H_0 : a_1 = 1$ (it is equivalent to $H_0 : \rho = 0$), its critical values depend on option and are equal to, respectively, `tau1`, `tau2` and `tau3`. Note that the critical values remain the same even if the process is not AR(1) but AR(p).

```
library(urca);?ur.df
lnyse.df=ur.df(lnyse,0,type = "drift")
```

```
summary(lnyse.df)

Call:
lm(formula = z.diff ~ z.lag.1 + 1)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0077315  0.0121721   0.635    0.526
z.lag.1     -0.0001374  0.0019082  -0.072    0.943

Value of test-statistic is: -0.072 7.5294

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.43 -2.86 -2.57
phi1  6.43  4.59  3.78
```

Since the $t-$value -0.072 is closer to zero than the 5% critical value: -2.86<-0.072<0, there is no ground to reject the null H_0 : *the process has a unit root*.

C. Testing for the unit root with the **A** and **B** procedures last long. The same `ur.df` function allows us to automate the procedure: once you choose the maximum number of lags (e.g., 4) and note to include linear trend, the `ur.df` function creates all possible models with `lags`=4, =3, =2, =1, =0 and chooses the one with least BIC. Note that this model will differ from the previous, however it is hard to say which one is more accurate.

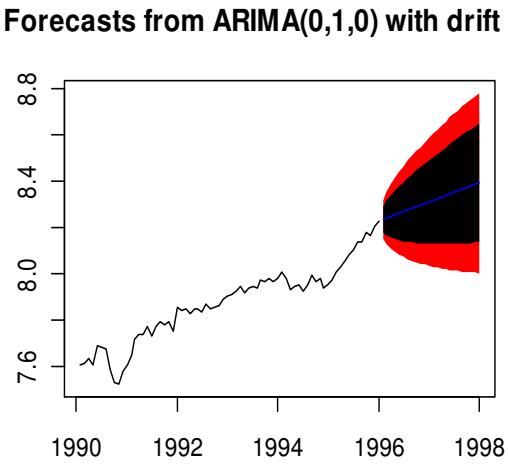
```
lnyse.df=ur.df(lnyse,type="trend",lags=4,selectlags="BIC")
summary(lnyse.df)

Call:
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.559e-02  3.915e-02   2.186   0.0292 *
z.lag.1     -1.680e-02  8.209e-03  -2.046   0.0412 *
tt           1.023e-04  4.983e-05   2.053   0.0406 *
z.diff.lag   5.261e-02  4.380e-02   1.201   0.2302

Value of test-statistic is: -2.0463 6.0166 2.1302

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -3.96 -3.41 -3.12
phi2  6.09  4.68  4.03
phi3  8.27  6.25  5.34
```

Thus, the bottom line is: since the test statistics is greater than the critical value, `lnyse` is a process with (a linear trend and a) unit root.

If instead of `type="trend"` we choose `type="drift"`, once again we get that the process has a unit root. Thus, we have three models and, to forecasting, choose (without any deeper argumentation) the A model.

D. `lnyse` is forecasted with the formula $lnyse_t = 0.0069 + lnyse_{t-1} + w_t$; to do this in R, use the `Arima` function:

```
library(forecast)
lnyse.ur=Arima(lnyse,c(0,1,0),
include.drift=TRUE)
lnyse.ur

ARIMA(0,1,0) with drift
Coefficients:
       drift
      0.0069
s.e.  0.0018

plot(forecast(lnyse.ur),include=72)
```



****
****

4.1 exercise. Repeat the forecast by using other two models. Compare the forecasts.

4.2 exercise. Use different methods and investigate whether the variable `lc` from the `Raotbl13` file in the urca package has a unit root. ◀◀

When testing for unit root, it is important to understand the meaning of the hypotheses – the function ur.df has three options: `type="none"`, `"drift"` and `"trend"`.

1. "`none`" – our time series resembles a stationary AR(1) process with zero mean (however, we suspect that it could be a random walk):

$$H_1: Y_t = \varphi Y_{t-1} + w_t \quad \sim \quad Y_t - Y_{t-1} = (\varphi - 1)Y_{t-1} + w_t, |\varphi| < 1$$
$$H_0: Y_t = Y_{t-1} + w_t \quad \sim \quad Y_t - Y_{t-1} = \mathbf{0} \cdot Y_{t-1} + w_t$$

In this case, the null $\varphi = 1$ means that our process is a random walk without drift. If the hypothesis is rejected, we conclude that the process is stationary AR(1) with zero mean.

2. "`drift`" – our time series resembles a stationary AR(1) process with nonzero mean (however, we suspect that it couls be a random walk with drift):

$$H_1: Y_t = \alpha + \varphi Y_{t-1} + w_t \sim Y_t - Y_{t-1} = \alpha + (\varphi - 1)Y_{t-1} + w_t, |\varphi| < 1$$
$$H_0: Y_t = \alpha + Y_{t-1} + w_t \sim Y_t - Y_{t-1} = \alpha + \mathbf{0} \cdot Y_{t-1} + w_t$$

In this case, the null $\varphi = 1$ means that our process is a random walk with drift. If the hypothesis is rejected, we conclude that the process is stationary AR(1) with nonzero mean.

3. "`trend`" - our time series resembles a stationary AR(1) process around a linear trend (however, we suspect that it couls be a random walk with drift):

$$H_1: Y_t - a - bt = \varphi(Y_{t-1} - a - bt) + w_t \sim Y_t = [a(1-\varphi) + b\varphi] + b(1-\varphi)\cdot t + \varphi Y_{t-1} + w_t, \ |\varphi| < 1$$
$$H_0: Y_t = b + Y_{t-1} + w_t \sim Y_t - Y_{t-1} = b + \mathbf{0}\cdot Y_{t-1} + w_t$$

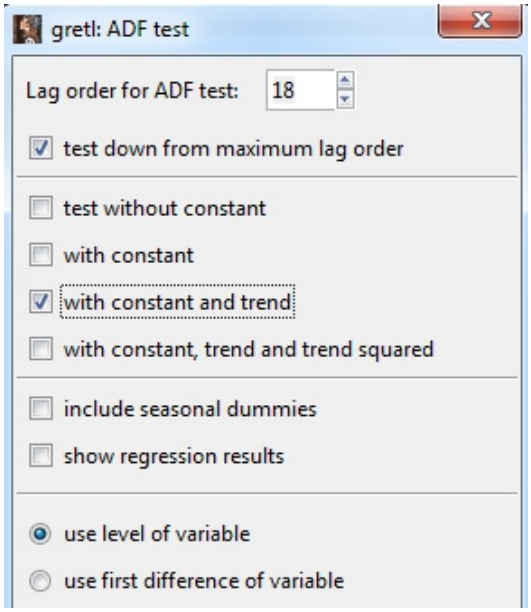
In this case, the null $\varphi = 1$ means that our process is a random walk with drift. If the hypothesis is rejected, we conclude that the process is stationary AR(1) around a line $a + bt$. ◀◀

We shall repeat the `nyse` analysis with gretl. The ADF test is, as implemented in gretl, the *t*-statistic on ρ in the following regression:

$$\Delta Y_t = \alpha + \rho Y_{t-1} + \gamma_1 \Delta Y_{t-1} + \ldots + \gamma_p \Delta Y_{t-p} + \delta t + w_t .$$

It is a one-sided test whose null hypothesis is $\rho = 0$ versus the alternative $\rho < 0$ (and hence large negative values of the test statistic lead to the rejection of the null). Under the null, Y_t must be differenced at least once to achieve stationarity; under the alternative, Y_t is already stationary and no differencing is required.

4.3 example. . Import the data file nyse.txt containing monthly data from 1952:1 through 1996:1 on a major stock price index provided by the New York Stock Exchange. Select `l_StockPrice`, go to Variable| Unit root tests| Augmented Dickey-Fuller test and fill in the window as shown on the right. If we choose the recommended order 18 as a starting point, gretl uses "the highest significant term" principle and finally takes 14 lags in Dickey-Fuller regression.




```
Augmented Dickey-Fuller test for l_StockPrice
including 14 lags of (1-L)l_StockPrice (max was 18)
sample size 514
unit-root null hypothesis: a = 1

   with constant and trend
   model: (1-L)y = b0 + b1*t + (a-1)*y(-1) + ... + e
   1st-order autocorrelation coeff. for e: -0.001
   lagged differences: F(14, 497) = 1.253 [0.2333]
   estimated value of (a - 1): -0.0151188
   test statistic: tau_ct(1) = -1.70429
   asymptotic p-value 0.7497
```

Note that if we replace 18 with 8, we get a recommendation to include only 5 lags, but the asymptotic p-value (=0.4378) is again more than 0.05, therefore in any case we do not reject the unit root hypothesis. To get the model, go to Model| Time series| ARIMA... and fill in the window as shown in 6.3 pav., left[3], below. The 12 months forecast of `l_StockPrice` is shown on the right.

[3] See Case **3**."`trend`" above: to apply ADF test, we selected "with constant and trend"; now, when we accepted $\rho = 0$ hypothesis, the model will contain only the drift (constant) α.

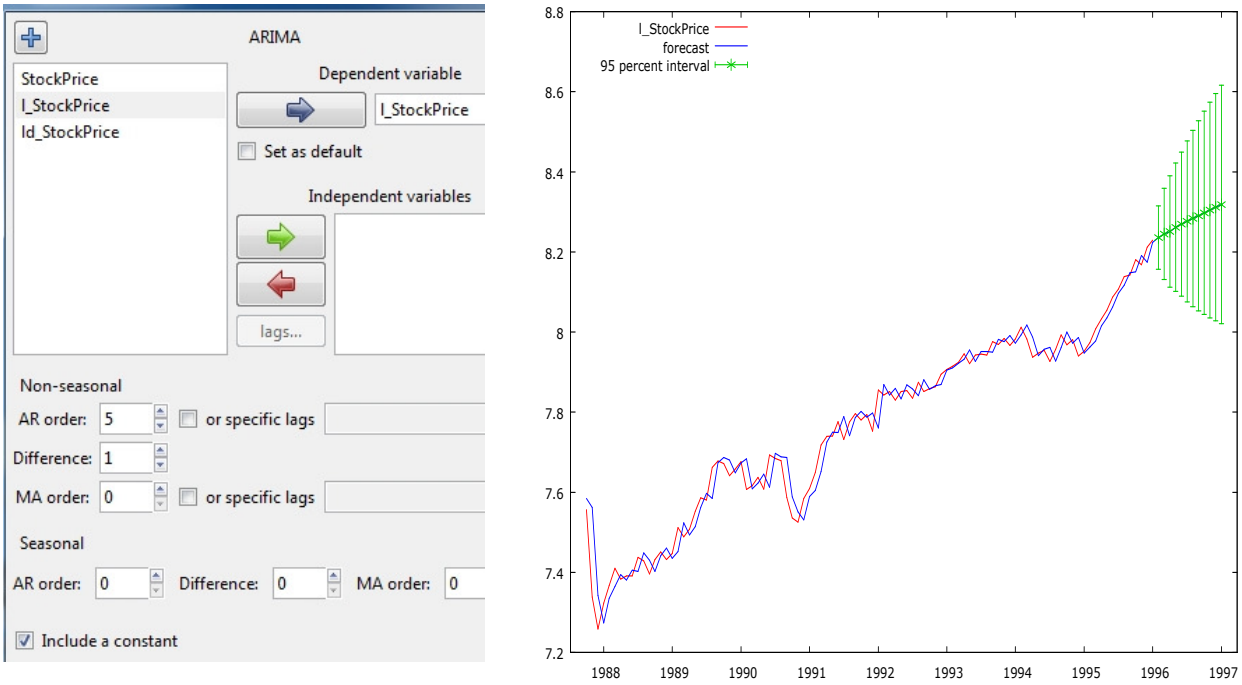


Fig. 4.3. ARIMA model (left) and 12 months forecast (right)

4.4 example. The data set …\dataPE\interestrates.xls contains quarterly long-run `Longrate` and short-run `Shortrate` interest rates, 1954:1-1994:4.

(a) Estimate descriptive statistics of both variables and their differences.
(b) Draw and comment the graphs of four respective variables.
(c) Draw and comment the `Longrate` – `Longrate(-1)` scatter diagram; do the same with `Shortrate`.
(d) Estimate `cor(`Y_t`,`Y_{t-1}`)` for both variables.
(e) Do (c) and (d) for the differences of our variables. Comment the differences.
(f) Plot ACF's of `Longrate` and `diff(Longrate)`.

```
rate= ts(read.delim2("clipboard",header=TRUE),start=1954,freq=4)
(a)
summary(rate)
summary(diff(rate))
boxplot(data.frame(rate))
(b)
plot(rate)
plot(diff(rate))
(c)
Longrate=rate[,1]
Shortrate=rate[,2]
lag.plot(Longrate)
lag.plot(diff(Longrate))
(d)
cor(Longrate[2:164],Longrate[1:163])              # [1]  0.9997162
cor(diff(Longrate[2:164]),diff(Longrate[1:163])) # [1] -0.002354711
```

(f)

```
acf(Longrate)
acf(diff(Longrate))
```

The results of this analysis allows us to guess that `Longrate`, most probably, has a unit root. We shall repeat the procedure of the previous example with $p_{\max}-1=4$ and finally end with the model $\Delta Y_t=\alpha+\rho Y_{t-1}+w_t$:

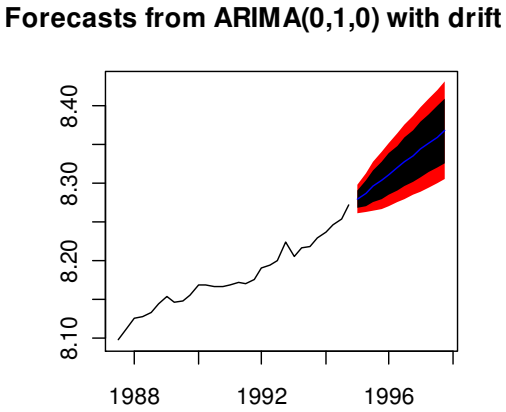
```
mod.0=dynlm(d(Longrate)~L(Longrate))
summary(mod.0)
Call:
dynlm(formula = d(Longrate) ~ L(Longrate))

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.038606   0.014392   2.682  0.00807 **
L(Longrate) -0.003983   0.001871  -2.130  0.03473 *
```

It would be wrong to say: since the $t-$value `|-2.13|>2`, we reject the null $\rho=0$. The right answer can be obtained with the Dickey-Fuller test:

```
Longrate.df=ur.df(Longrate,0,type="drift")
summary(Longrate.df)


Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.038606   0.014392   2.682  0.00807 **
z.lag.1     -0.003983   0.001871  -2.130  0.03473 *

Value of test-statistic is: -2.1295 63.8967

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.46 -2.88 -2.57
phi1  6.52  4.63  3.81
```

Since the $t-$value statistics `-2.13>-2.88`, we have no ground to reject the null, therefore `Longrate` is a random walk with drift, $Y_t-Y_{t-1}=0.008+w_t$:

```
Long.auto=auto.arima(Longrate,d=1)
plot(forecast(Long.auto,12),include=30)
```



Repeat the analysis with `Shortrate`. ◀◀

4.1 exercise. The time series

```
lGNP = structure(c(14.1923, 14.2862, 14.328, 14.3646, 14.3577, 14.4125, 14.4322, 14.4513,
14.4464, 14.4995, 14.5197, 14.5445, 14.6042, 14.6470, 14.7053, 14.7661, 14.8290, 14.8546,
14.8987, 14.9293, 14.9293, 14.9622, 15.0173, 15.0738, 15.0680, 15.0592, 15.1122, 15.1624,
15.2127, 15.2413, 15.2413, 15.2679, 15.2486, 15.2868, 15.3513, 15.3836, 15.4119, 15.4435,
15.4807, 15.5055, 15.5138, 15.5017, 15.5221, 15.5517), .Tsp = c(1950, 1993, 1), class =
"ts")
```

contains the logarithms of the U.S. gross national product, 1950-1993, in total 44 years).

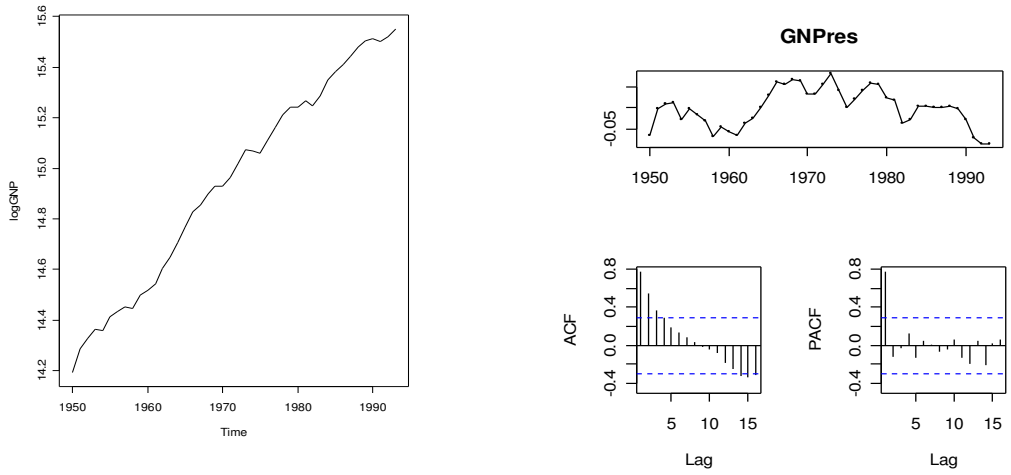


Fig. 4.4. The graph of `lGNP` (left) and the residuals of a linear model (right)

The time series has an evident, almost linear, trend, therefore we shall examine two variants: this is a process with **i**) a deterministic trend and **ii**) a stochastic trend (and, probably, a drift).

i)

(a) Use the `lm` function, remove linear trend, and analyse residuals.
(b) The residuals, it seems, are described by an AR(1) process, therefore once again remove the trend with

```
(GNPar100=arima(lGNP,c(1,0,0),xreg=time(lGNP)))
```

(the coefficient at `time(lGNP)*100` is an average percentage growth rate of GNP).

(c) Is it a proper model? (write the model). If your answer is positive, forecast `lGNP` seven years ahead:

```
plot(forecast(GNPar100,h=7,xreg=1994:2000),include=20)
```

(d) Since the residuals of the linear model have a unit root (can you substantiate your answer?), it seems reasonable to examine

ii)

(e) Is it true that `lGNP` itself has a unit root (and, probably, a positive drift)? Test the hypothesis with the `ur.df` function.

(f) Create a respective model (how does it differs from the previous one?)

```
(GNPar010=arima(lGNP,c(0,1,0),xreg=time(lGNP)))
```

The rate of growth, i.e., the drift, i.e., the coefficient at `time(lGNP)` equals ..., it is practically the same as earlier.

(g) Draw the forecast of `lGNP` seven years ahead. Both models give similar forecasts, the only difference is in the forecasting errors.

The width of the confidence interval of the forecast of the stationary AR process (the `GNPar100` model) tends to a constant.
The width of the confidence interval of the forecast of the process with unit root (the `GNPar010` model) widens as a square root.

(h) Which model is selected by the sequential procedure **A**? And by **C**? ◀◀

We conclude that all three models are of more or less the same accuracy. Below we shall present one more method to value the model: we remove the last few observations, create a new „shortened" model, and compare its forecast with the eliminated data. Then we choose the one with the smallest forecast error.

4.2 exercise. Generate a process with linear trend and correlated errors, i.e., $Y_t = -7 + 0.3t + 5e_t$ where $e_t = 0.8842e_{t-1} - 0.5316e_{t-2} + w_t$, $t = 1,...,150$. Use the sequential procedure to establish the type of the process. Do not forget to apply the Dicky-Fuller test. Find the 20-step-ahead forecast.

```
set.seed(1)
ee=arima.sim(n = 150, list(ar = c(0.8842, -0.5316)))
tt=time(ee)
yy=-7+0.3*tt+5*ee

mod.1=dynlm(d(yy)~L(yy)+L(d(yy),1)+tt)
summary(mod.1)
summary(ur.df(yy,lags=1,type="trend")) # vienet. šaknį atmetame
(yy2=arima(yy,c(2,0,0),xreg=tt))
plot(forecast(yy2,h=20,xreg=151:170),include=70)
```

4.3 exercise. Generate the random walk Y_t with 0.1 drift: $Y_t = 0.1 + Y_{t-1} + w_t$. Use sequential procedure to classify the process. Forecast the process for 20 time moments ahead.

4.4 exercise. Create two `lGNP` models using the data from 1950 till 1990 and compare the accuracy of forecasts of both models for 1991 till 1993.

```
lGNP.sh=window(lGNP, 1950, 1990) # shorten lGNP
```

(a) Create the models `GNPar100.sh` and `GNPar010.sh` and draw the graphs as in Fig. 4.5.

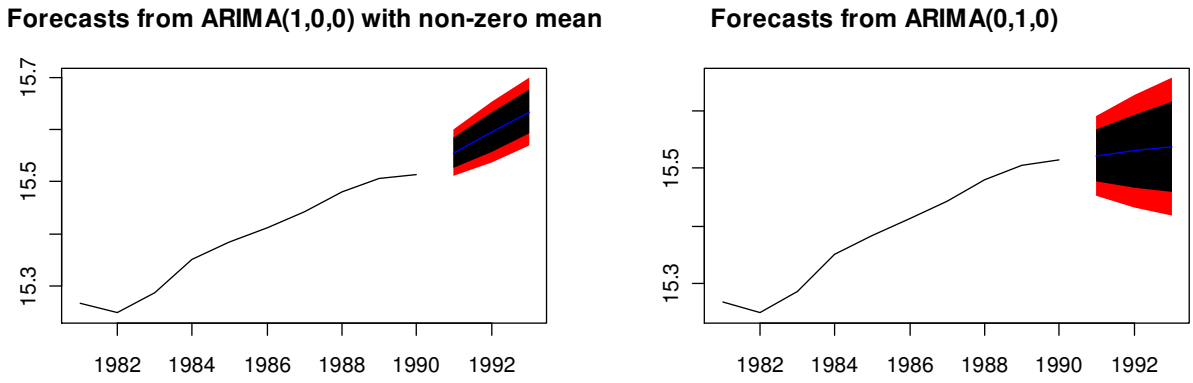


Fig. 4.5. The two models give different forecasts

The mean error of the forecast may be estimated in various ways:

Root-Mean-Square Error:	$\mathrm{RMSE} = \sqrt{(1/n)\sum_{i=1}^{n}\hat{u}_i^2}$
Mean Absolute Error:	$\mathrm{MAE} = (1/n)\sum_{i=1}^{n}\lvert\hat{u}_i\rvert$
Mean Squared Error	$\mathrm{MSE} = (1/n)\sum_{i=1}^{n}\hat{u}_i^2$
Mean Absolute Percentage Error	$\mathrm{MAPE} = (1/n)\sum_{i=1}^{n}\lvert\hat{u}_i / y_i\rvert\times 100$
Mean Error	$\mathrm{ME} = (1/n)\sum_{i=1}^{n}\hat{u}_i$

Compare the two models by RMSE:

```
sqrt(sum((lGNP[42:44]-forecast(GNPar010.sh,h=3,xreg=1991:1993)$mean)^2)/3)
sqrt(sum((lGNP[42:44]-forecast(GNPar100.sh,h=3,xreg=1991:1993)$mean)^2)/3)
```

According to this criterion, the unit root model is more accurate.

(b) Compare the models by MAPE. ◀◀

4.5 exercise. The data set .../PE.II.2010data/nporg.txt contains the fourteen U.S. economic time series (annual, 1860-1970) used by Nelson & Plosser in their seminal paper.

`year`	Time index from 1860 until 1970.
`gnp.r`	Real GNP, Billions of 1958 Dollars], [1909 – 1970]
`gnp.n`	Nominal GNP, [Millions of Current Dollars], [1909 – 1970]
`gnp.pc`	Real Per Capita GNP, [1958 Dollars], [1909 – 1970]
`ip`	Industrial Production Index, [1967 = 100], [1860 – 1970]
`emp`	Total Employment, [Thousands], [1890 – 1970]
`ur`	Total Unemployment Rate, [Percent], [1890 – 1970]
`gnp.p`	GNP Deflator, [1958 = 100], [1889 – 1970]
`cpi`	Consumer Price Index, [1967 = 100], [1860 – 1970]

`wg.n`	Nominal Wages (Average annual earnings per full-time employee in manufacturing), [current Dollars], [1900 – 1970]
`wg.r`	Real Wages, [Nominal wages/CPI], [1900 – 1970]
`M`	Money Stock (M2), [Billions of Dollars, annual averages], [1889 – 1970]
`vel`	Velocity of Money, [1869 – 1970]
`bnd`	Bond Yield (Basic Yields of 30-year corporate bonds), [Percent per annum], [1900 – 1970]
`sp`	Stock Prices, [Index; 1941 – 43 = 100], [1871 – 1970]

Until 1982, when these data and their analysis were published, most of the economists believed that all time series were TS (i.e., after removing the trend they become stationary). Nelson & Plosser proved that most of economic series are DS (i.e., their differences are stationary). Take `gnp.r` and `cpi` (or, probably, their logarithms) from the above file and examine whether they are TS or DS.

4.6 exercise. The urca package contains many macroeconomical time series. The data set `Raotbl1` has 5 columns which can be (though it is not necessary) converted to time series. What is the meaning of the variables in the 1st, 3rd, and 5th columns? Examine whether these series have a unit root.

4.2. SARIMA (=Seasonal ARIMA) Models

Many economic processes have a seasonal component. Usual procedure at first removes (the trend and then) the seasonal part and then analyzes the remainder, for example, identifies it as a stationary ARMA process. On the other side, better results are obtained if both procedures take place at the same time.

Here are two seasonal quarterly models: $y_t = a_4 y_{t-4} + w_t$, $|a_4|<1$ and $y_t = w_t + b_4 w_{t-4}$. It is easy to verify that in the first case $\rho_i = a_4^{i/4}$, if *i*/4 is an integer and =0 otherwise. In the second case, ACF has the only nonzero bar at 4. However, usually the processes have not only the seasonal component but also a trend, therefore the behavior of ACF is more complicated (also, do not forget that sample ACF may differ from the theoretical).

One more example: $y_t = a_1 y_{t-1} + w_t + b_1 w_{t-1} + b_4 w_{t-4}$. In principle, the seasonal effect may also be described as $y_t = a_1 y_{t-1} + a_4 y_{t-4} + w_t + b_1 w_{t-1}$. Both these models add a seasonal term (w_{t-4} or y_{t-4}, respectively), therefore they are termed *additive*. Here are two variants of *multiplicative* models[4]:

$$(1-a_1L)\,y_t = (1+b_1L)\times(1+b_4L^4)\,w_t \quad \text{(denote by SARIMA(1,0,1)(0,0,1)}_4\text{)}$$

(i.e.,, $y_t = a_1 y_{t-1} + w_t + b_1 w_{t-1} + b_4 w_{t-4} + b_1 b_4 w_{t-5}$) and

$$(1-a_1L)\times(1-a_4L^4)\,y_t = (1+b_1L)\,w_t \quad \text{(denoted by SARIMA(1,0,1)(1,0,0)}_4\text{).}$$

(i.e., ... – write it yourselves).

---

[4] The process $(1-L)^d(1-L^s)^D\phi(L)\Phi(L^s)Y_t = \theta(L)\Theta(L^s)w_t$ is denoted by $ARIMA(p,d,q)(P,D,Q)_s$ or $SARIMA(p,d,q)(P,D,Q)_s$.

Now we shall examine some models with seasonal component. We begin with GRETL.

4.5 example. The data set .../Thomas – ME1997/data4.txt contains quarterly, 1974:1-1984:4, data about consumption C and disposable income Y in UK. Test whether log(C) has a unit root, create a model of log(C), and forecast it 12 quarters ahead.

These series are not seasonally smoothed, therefore we say that log(C) has a unit root if the coefficient ρ in the equation

$$\Delta \log C_t = \alpha + \rho \log C_{t-1} + \gamma_1 \Delta Y_{t-1} + ... + \gamma_{p-1} \Delta Y_{t-(p-1)} + \delta t + \beta_2 D_{2,t} + \beta_3 D_{3,t} + \beta_4 D_{4,t} + \varepsilon_t \quad (6.1)$$

equals 0 (here D_i are quarterly dummy variables and the order p is chosen according to the minimum of AIC or by the significance of the highest term). Select `l_C` (=log(C)), go to Variable|Unit root tests| Augmented Dickey-Fuller test and fill in the test window as shown in 6.5 pav., right. The answer is a bit strange: it is well known that log(C) is $I(1)$, however the output suggests stationarity:

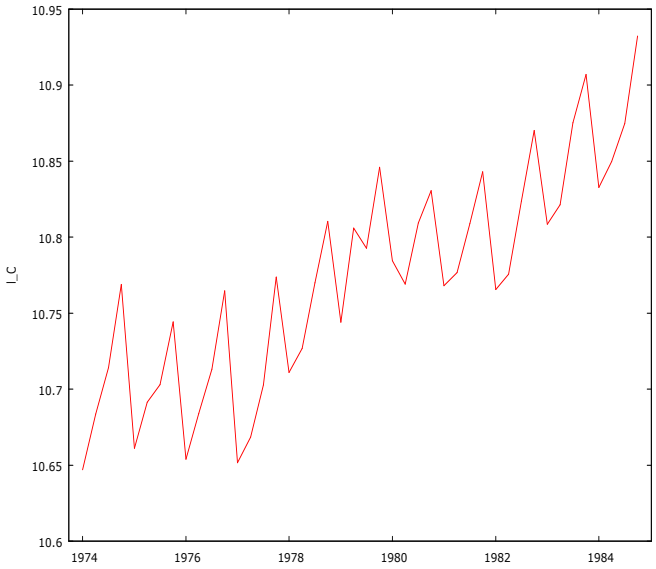
```
Augmented Dickey-Fuller test for l_C
including 10 lags of (1-L)l_C (max was 13)
sample size 33
unit-root null hypothesis: a = 1

   with constant and trend plus seasonal dummies
   model: (1-L)y = b0 + b1*t + (a-1)*y(-1) + ... + e
   1st-order autocorrelation coeff. for e: -0.138
   lagged differences: F(10, 17) = 2.020 [0.0970]
   estimated value of (a - 1): -1.72407
   test statistic: tau_ct(1) = -4.02327
   asymptotic p-value 0.008069
```

However, if one repeats the analysis with Lag order ... = 5, the ADF test will now claim that $p = 1$ in (6.1) and that unit root is present in the model:

```
Dickey-Fuller test for l_C
sample size 43
unit-root null hypothesis: a = 1

   with constant and trend plus seasonal dummies
   model: (1-L)y = b0 + b1*t + (a-1)*y(-1) + e
   1st-order autocorrelation coeff. for e: -0.178
   estimated value of (a - 1): -0.35111
   test statistic: tau_ct(1) = -2.78711
   p-value 0.2096
```

These different answers can be explained by the fact that a random walk could be quite close to stationary AR(p) with large p and the roots of the inverse characteristic equation close to 1. This assertion can be also supported by the forecasts of the two models. We start with the TS model and describe respective ARIMA[5] model as shown on the right (this model is similar to the model without dq2, dq3, and dq4 as independent variables, but with Seasonal AR order equal to 1; try to create the model yourself).

[5] Sometimes it is termed ARIMAX to indicate the presence of independent variable X.



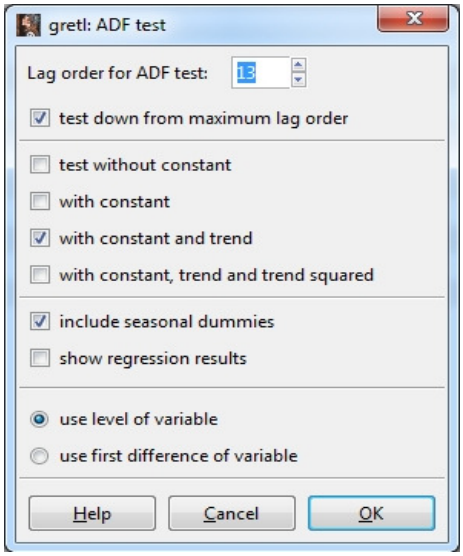
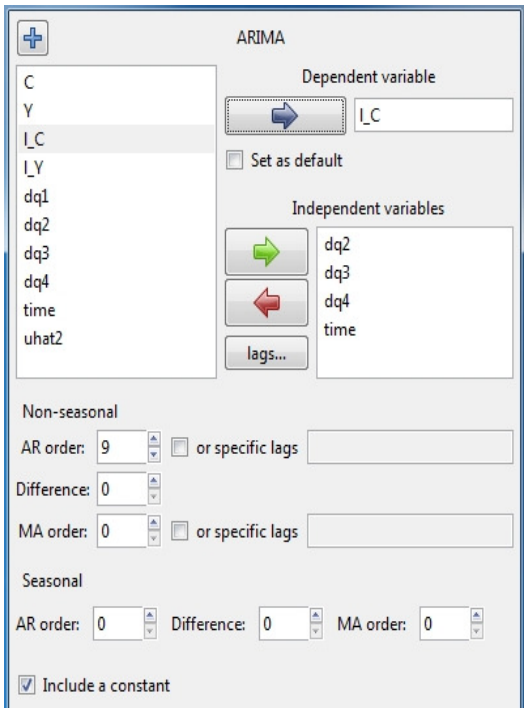


Fig. 4.6.

To create respective DS model, fill in the ARIMA window with AR order: 0, Difference:1 and remove the variable time from the list.
The forecasts are presented in 6.6 pav. where one can see that both models are similar. In fact, the only difference is in the width of the confidence intervals (in DS case they are broader but it is not quite clear which model is „more correct“).



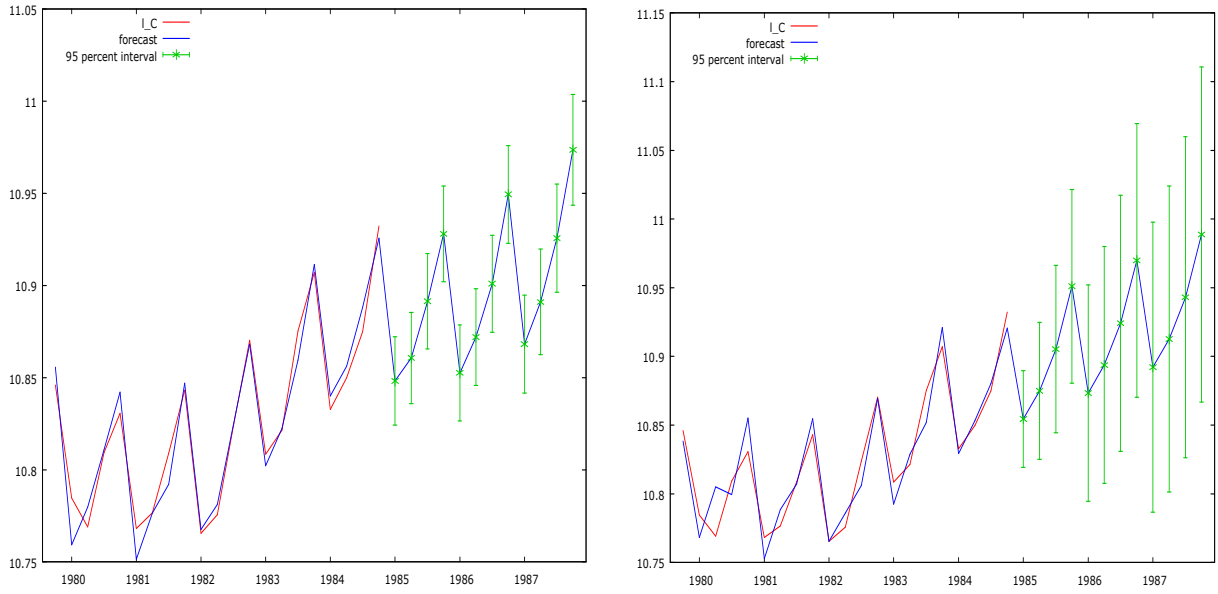


Fig. 4.7. The forecasts of the TS model (left) and DS model (right)

4.7 exercise. Parallel this analysis with `log(Y)`. ◀◀

****
****

Now we shall continue the analysis with R. Upon typing

```
spain=ts(scan(file.choose()),start=1970,freq=12) # Navigate to
                                                 # Data\Enders\Spain.txt
tsdisplay(spain)
```

you shall see the monthly numbers of tourists visiting Spain. The mean of `spain` is slowly increasing, but the data has evident seasonality.

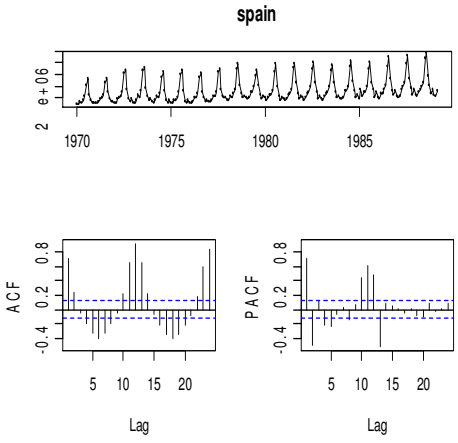


Fig. 4.8. The number of tourists visiting Spain and respective ACF and PACF graphs

Since the spread of data is increasing in time, we take logarithms first and then differentiate to make them stationary.

```
tsdisplay((y1 <- diff(log(spain))))
tsdisplay((y12 <- diff(log(spain),12)))
```

= $(1-L)y_t = y_t - y_{t-1}$

= $(1-L^{12})y_t = y_t - y_{t-12}$

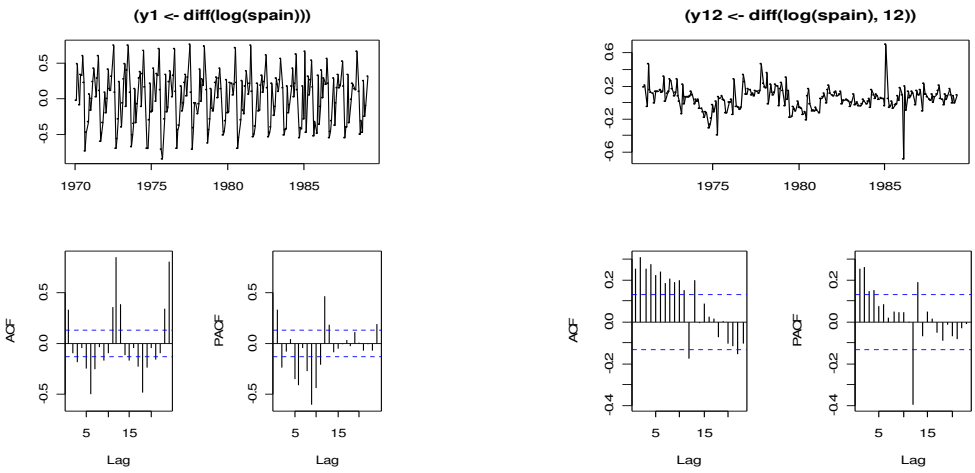


Fig. 4.9. The plots of simple and seasonal differences of `spain`

The seasonal differences are similar to stationary, therefore in the future we shall analyze `y12`. The remaining variations we eliminate with simple differentiating.

```
tsdisplay(diff(y12))
```

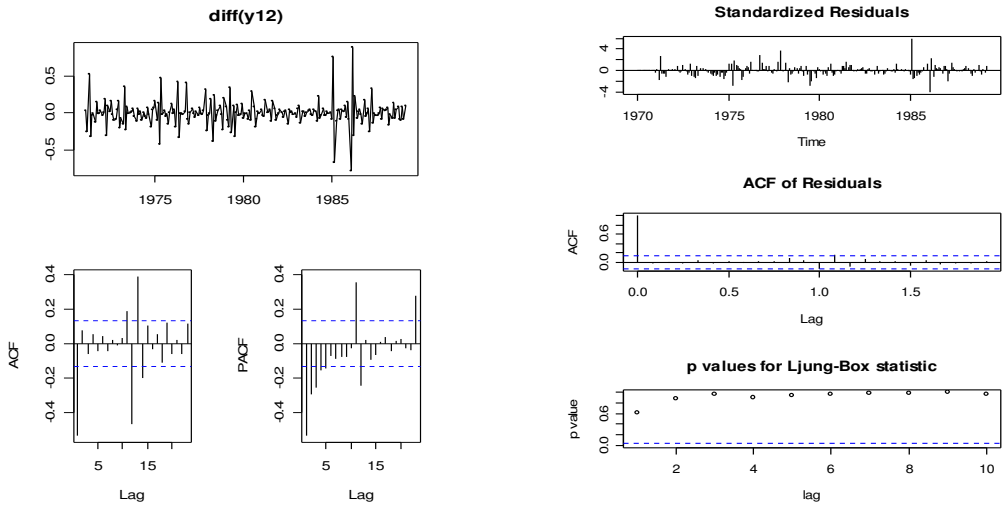


Fig. 4.10. The graphs of `diff(y12)` (left) and the diagnostics of the MAR (see below) model (right)

The only clear peak of the ACF of `diff(y12)` is at 1; coupling the fact with the vanishing PACF, we choose the MA(1) model (significant peaks around 12 may be explained by additive or multiplicative seasonal factors – we propose three variants to explain them (notation: `y=log(spain)`):

$(1-L^{12})(1-L)(1-a_{12}L^{12})y_t = (1+b_1L)w_t$ (multiplicative autoregressive model MAR)

$(1-L^{12})(1-L)y_t = (1+b_1L)(1+b_{12}L^{12})w_t$ (multiplicative moving average model MMA)

$(1-L^{12})(1-L)y_t = (1+b_1L+b_{12}L^{12})w_t$ (additive moving average model AMA)

```
y=log(spain)

MAR=arima(y, order = c(0,1,1), seasonal = list(order=c(1,1,0)))
MAR
Series: y
ARIMA(0,1,1)(1,1,0)[12] model

Coefficients:
          ma1     sar1
      -0.7452  -0.4085
s.e.   0.0425   0.0627

sigma^2 estimated as 0.01378:  log likelihood = 156.16,  aic = -306.32
tsdiag(MAR) # See Fig. 4.9, right


MMA= arima(y, order = c(0,1,1), seasonal = list(order=c(0,1,1)))
MMA
Series: y
ARIMA(0,1,1)(0,1,1)[12] model

Coefficients:
          ma1     sma1
      -0.7354  -0.7267
s.e.   0.0455   0.0515

sigma^2 estimated as 0.01118:  log likelihood = 175.52,  aic = -345.04
tsdiag(MMA)
AMA=arima(y,   order=c(0,1,12),   fixed=c(NA,0,0,0,0,0,0,0,0,0,0,NA),   seasonal=
list(order=c(0,1,0)))
AMA
Series: y
ARIMA(0,1,12)(0,1,0)[12] model

Coefficients:
          ma1  ma2  ma3  ma4  ma5  ma6  ma7  ma8  ma9  ma10  ma11     ma12
      -0.6930    0    0    0    0    0    0    0    0     0     0  -0.2777
s.e.   0.1068    0    0    0    0    0    0    0    0     0     0   0.1047

sigma^2 estimated as 0.01522:  log likelihood = 145.08,  aic = -284.17
tsdiag(AMA)
```

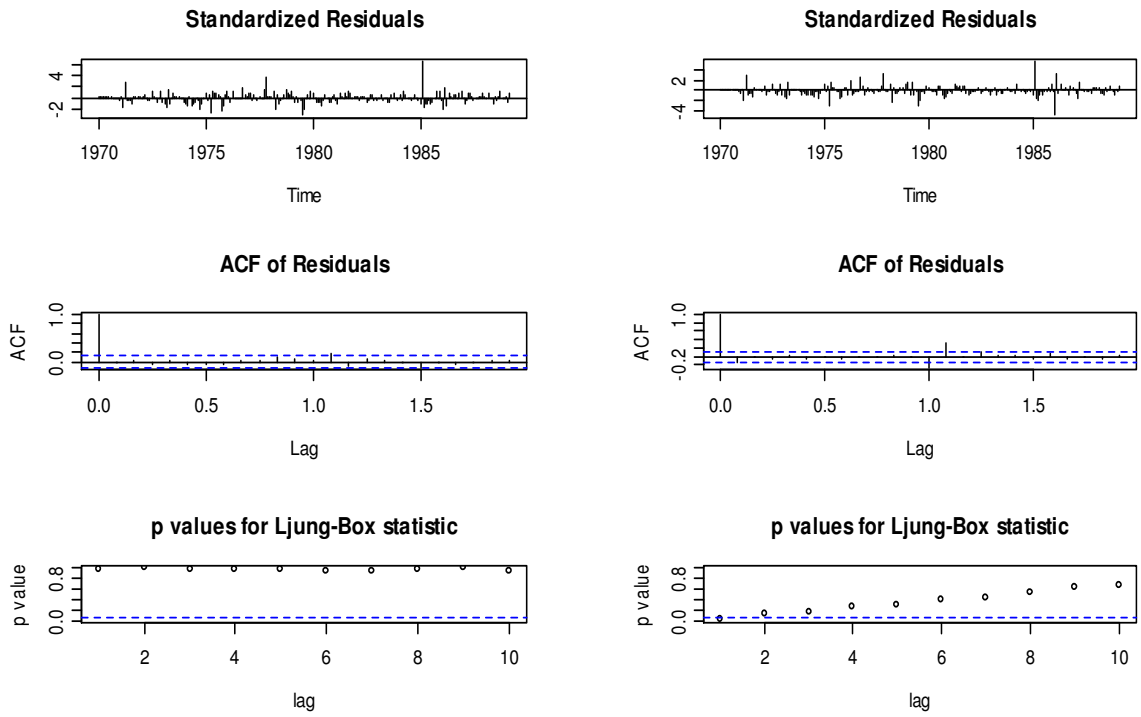


Fig. 4.11. The diagnostic graphs of MMA (left) and AMA (right; the model MMA is better)

According to AIC, the best among these models (and, possibly, among all the SARIMA models) is the MMA model which can be denoted as SARIMA(0,1,1)(0,1,1)$_{12}$. We can use this model (write it down explicitely) to forecast the logs of the Spain‘s tourists.

```
plot(forecast(MMA,12),include=36) # See Fig.4.12 below
```

It is interesting to note that a fully automated procedure of exponential smoothing gives almost the same result:

```
y.ets=ets(y)
plot(forecast(y.ets,h=12),include=36) # See Fig.4.12 below
```

4.8 exercise. 1. The `spain.p` model can also be characterized by its accuracy (i.e., calculate ME, MSE, MAE, MPE, and MAPE by its historical data, see `summary(spain.p)`). Estimate some of these characteristics for the MMA model. 2. Type the numeric values of the both forecasts.

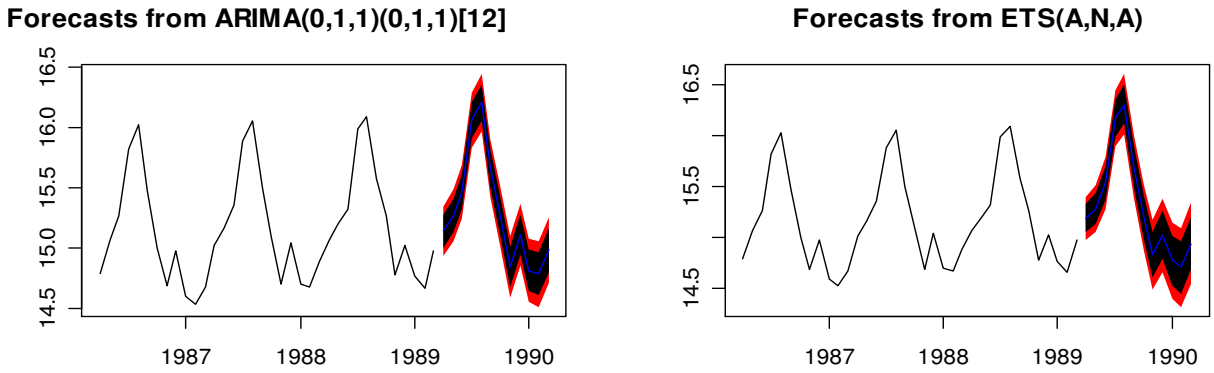


Fig. 4.12. Two forecasts using the MMA and exponential smoothing models

4.9 exercise. In the `fma` package, one can find the `airpass` data set; their logarithms (why logarithms but not `airpass` itself?) are usually described by the classical „airline“ model SARIMA$(0,1,1)(0,1,1)_{12}$. Create the model. Compare it with the exponential smoothing.

4.10 exercise. In 3.1 example, we have used exponential smoothing and created a model allowing to forecast the Lithuanian GDP.

```
l.gdp=ts(scan(),start=1993,freq=4) # copy and paste data from L.gdp.txt
l.gdp
ll.gdp=log(l.gdp)
ll=window(ll.gdp,start=2000)
plot(ll, main="logs of Lithuanian GDP")
```

Model `ll` (it is clearly not stationary and has a seasonal component) as i) `S.ll` = SARIMA$(0,1,1)(0,1,1)_4$ (analyze the model, estimate its AIC, and use it to forecast `ll` 8 months ahead), and as ii) the model with dummy seasonal variables:

```
library(forecast)
quarter=seasonaldummy(ll)
quarter
q.ll=Arima(ll,order=c(0,1,0),include.drift=TRUE,xreg=quarter)
q.ll
tsdiag(q.ll)
plot(forecast(q.ll,,h=8,xreg=seasonaldummyf(ll,8))) # AIC=?
```

Compare these two models with the model presented in 3.1 example. Which one do you prefer? The last one? Create all three models with GRETL. ◀◀

5. Regression with Time Lags: Autoregressive Distributed Lag Models

We shall redo 5.1 example from *Lecture Notes* with R.

5.1 example. The effect of bad news on market capitalization.

Select 2-62 rows in badnews.xls and copy them (to clipboard):

```
badnew=read.delim2("clipboard",header=TRUE)
badnews=ts(badnew,freq=12)
capit=badnews[,1]
price=badnews[,2]
library(dynlm)
mod4=dynlm(capit~L(price,0:4))
summary(mod4)
```

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          91173.32    1949.85  46.759  < 2e-16 ***
L(price, 0:4)Series 1  -131.99      47.44  -2.783 0.007589 **
L(price, 0:4)Series 2  -449.86      47.56  -9.459 1.01e-12 ***
L(price, 0:4)Series 3  -422.52      46.78  -9.032 4.40e-12 ***
L(price, 0:4)Series 4  -187.10      47.64  -3.927 0.000264 ***
L(price, 0:4)Series 5   -27.77      47.66  -0.583 0.562736
```

What can we conclude about the effect of news about the oil price on market capitalization? Increasing the oil price by one dollar per barrel in a given month is associated with:

1. An immediate reduction in market capitalization of $131,994, *ceteris paribus*.
2. A reduction in market capitalization of $449,860 one month later, *ceteris paribus*

and so on. To provide some intuition about what the *ceteris paribus* condition implies in this context note that, for example, we can also express the second of these statements as: "Increasing the oil price by one dollar in a given month will tend to reduce market capitalization in the following month by $449,860, **assuming that no other change in the oil price occurs**".

Since the *p*-value corresponding to the explanatory variable $\texttt{price}_{t-4}$ is greater than 0.05 we cannot reject the hypothesis that $\beta_4 = 0$ at 5% significance. Accordingly, we drop this variable from the model and re-estimate with lag length set equal to 3, yielding the results in the following table:

```
mod3=dynlm(capit~L(price,0:3))
summary(mod3)
```

```
Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          90402.22    1643.18  55.017  < 2e-16 ***
L(price, 0:3)Series 1  -125.90      46.24  -2.723 0.008792 **
L(price, 0:3)Series 2  -443.49      45.88  -9.666 3.32e-13 ***
L(price, 0:3)Series 3  -417.61      45.73  -9.131 2.18e-12 ***
L(price, 0:3)Series 4  -179.90      46.25  -3.890 0.000287 ***
```

The *p*-value for testing $\beta_3 = 0$ is 0.0003, which is much less than 0.05. We therefore conclude that the variable $\texttt{price}_{t-3}$ does indeed belong in the distributed lag model. Hence q = 3 is the lag

length we select for this model. In a formal report, we would present this table of results or the equation

$$capit_t = 90402.22 - 125.90\ price_t - 443.49\ price_{t-1} - 417.61\ price_{t-2} - 179.90\ price_{t-3}.$$

The results are similar to those discussed above, therefore we will not comment their interpretation.

5.1 exercise. [redacted] Consider different forms for a consumption function based on quarterly US macroeconomic data from 1950:1 through 2000:4 as provided in the data set USMacroG in the AER package. Consider two models,

$$consumption_t = \alpha_1 + \beta_0 dpi_t + \beta_1 dpi_{t-1} + ... + \beta_q dpi_{t-q} + \varepsilon_t$$
$$consumption_t = \alpha_1 + \varphi_1 consumption_{t-1} + \beta_0 dpi_t + \varepsilon_t$$

(here `consumption` is real consumption expenditures and `dpi` real disposable personal income). In the former model, a distributed lag model, consumption responds to changes in income only over q (where q is still to be found) periods, while in the latter specification, an autoregressive distributed lag model, the effects of income changes persist due to the autoregressive specification.

```
library(AER)
data(USMacroG)
USMacroG[1:6,]
```

```
        gdp consumption invest government    dpi  cpi      m1 tbill unemp
[1,] 1610.5      1058.9  198.1      361.0 1186.1 70.6 110.20  1.12   6.4
[2,] 1658.8      1075.9  220.4      366.4 1178.1 71.4 111.75  1.17   5.6
[3,] 1723.0      1131.0  239.7      359.6 1196.5 73.2 112.95  1.23   4.6
[4,] 1753.9      1097.6  271.8      382.5 1210.0 74.9 113.93  1.35   4.2
[5,] 1773.5      1122.8  242.9      421.9 1207.9 77.3 115.08  1.40   3.5
[6,] 1803.7      1091.4  249.2      480.1 1225.8 77.6 116.19  1.53   3.1
     population inflation interest
[1,]    149.461        NA       NA
[2,]    150.260    4.5071  -3.3404
[3,]    151.064    9.9590  -8.7290
[4,]    151.871    9.1834  -7.8301
[5,]    152.393   12.6160 -11.2160
[6,]    152.917    1.5494  -0.0161
```

```
plot(USMacroG[, c("dpi", "consumption")], lty = c(3, 1),plot.type = "single",
ylab = "")
legend("topleft", legend = c("income", "consumption"),lty = c(3, 1), bty = "n")
```

What about the models in logs and with a trend t? Fit all the models, visualize and compare them with `AIC`. Estimate the short- and long-run multipliers of the best model. ◀◀

6. Regression with Time Series Variables

Recall that regression models are aimed to explain the response variable Y in terms of the predictive variables X's.

6.1. Time Series Regression when Both Y and X are Stationary

Let us repeat our argument which was used in our discussion on the ADL(p,q) processes: the dynamic model

$$Y_t = \alpha + \delta t + \varphi_1 Y_{t-1} + ... + \varphi_p Y_{t-p} + \beta_0 X_t + ... + \beta_q X_{t-q} + \varepsilon_t$$

is equivalent to the model

$$\Delta Y_t = \alpha + \delta t + \rho Y_{t-1} + \gamma_1 \Delta Y_{t-1} + ... + \gamma_{p-1} \Delta Y_{t-p+1} + \theta X_t + \omega_1 \Delta X_t + ... + \omega_q \Delta X_{t-q+1} + \varepsilon_t .$$

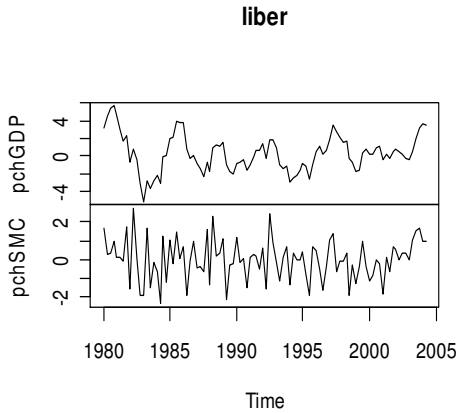
The second form is often more convenient than the first because 1) it allows us to easier test the unit root hypothesis (now it is just $H_0 : \rho = 0$) and 2) in the first model one is often challenged by the multicollinearity problem. The coefficients of both models have their usual interpretation, however in economics more popular is the concept of a multiplier. Assume that X and Y are in a long-running equilibrium, i.e., , $... = X_{t-2} = X_{t-1} = X_t \equiv X$, and $... = Y_{t-2} = Y_{t-1} = Y_t \equiv Y$; thus the equilibrium model is given by $Y = \frac{\alpha}{1-\varphi_1 - ... - \varphi_p} + \frac{\delta}{1-\varphi_1 - ... - \varphi_p} t + \frac{\beta_0 + \beta_1 + ... + \beta_q}{1-\varphi_1 - ... - \varphi_p} X$. Now, if the process transfers to the new constant level, i.e., $X_{t+1} = X_{t+2} = ... \equiv X + 1$, this will change the equilibrium value of Y. This change of Y is called the long run or total multiplier and it equals $(\beta_0 + \beta_1 + ... + \beta_q) / (1 - \varphi_1 - ... - \varphi_p)$ or $-\theta / \rho$.

6.1 example. The effect of financial liberalization on economic growth.

Researchers in the field of international finance and development are interested in whether financial factors can play an important role in encouraging growth in a developing country. The purpose of this example is to investigate this issue empirically using time series data from a single country. Data set LIBERAL.XLS contains data from Country A for 98 quarters starting at 1980:1 on GDP growth and a variable reflecting financial liberalization: the expansion of the stock market. In particular, the dependent and explanatory variables are:

`Y = pchGDP` the percentage change in GDP.
`X = pchSMC` the percentage change in total stock market capitalization.

```
liber = ts(read.delim2("clipboard"),
start=1980,freq=4)
```



```
mean(data.frame(liber))
     pchGDP      pchSMC
0.29677662 0.01256935
plot(liber)
```

The mean of these two variables is 0.30% and 0.01% per quarter, indicating that stock markets in Country A have not expanded by much **on average**. Note, however, that this average hides wide variation. In some quarters market capitalization increased considerably, while in other quarters it decreased. Assuming that both variables are stationary, we can estimate an ADL(2, 2) model using OLS. Remember that, if the variables in a model are stationary, then the standard regression quantities (e.g., OLS estimates, *p*-values, confidence intervals) can be calculated in an ordinary way.

We shall use the above applied sequential procedure and begin with the model

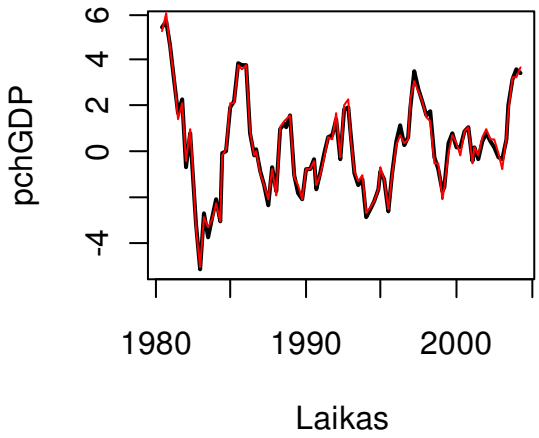
```
library(dynlm)
summary(dynlm(d(pchGDP)~time(liber)+L(pchGDP,1)+L(d(pchGDP),1)+pchSMC+
L(d(pchSMC),0:1),data=liber))
```

which, after removing insignificant terms, ends in

```
mod.f = dynlm(d(pchGDP)~L(pchGDP,1)+L(d(pchGDP),1)+pchSMC+d(pchSMC),data=liber)
summary(mod.f)
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.007127   0.020386    0.35  0.72747
L(pchGDP, 1)   -0.120092   0.011414  -10.52  < 2e-16 ***
L(d(pchGDP), 1) 0.799661   0.029680   26.94  < 2e-16 ***
pchSMC          0.124421   0.041605    2.99  0.00358 **
d(pchSMC)       0.839340   0.036458   23.02  < 2e-16 ***
Adjusted R-squared: 0.9736
```

How to interpret the long run multiplier $-\theta / \rho = -0.124421/(-0.120092) = 1.036$? We know that until now the market capitalization grew on average 0.01% per quarter and GDP 0.30% per quarter. If the stock capitalization began to grow 1.01% each quarter, then after some time the GDP will begin to grow 0.30+1.036=1.336% each quarter.



The inclusion of `pchSMC` into our model not only improved it (when compared with the autoregressive model of `d(pchGDP)`; compare their `Adjusted R-squared`), but also allowed to analyze the influence of `pchSMC` (it is described by the lung run multiplier).

Note that to forecast `pchGDP` we use the formula $\texttt{pchGDP}_t = \texttt{pchGDP}_{t-1} + \texttt{fitted(mod.f)}$:

```
ttt=time(liber)
length(ttt)
```

```
[1] 98
plot(ttt[3:98],pchGDP[3:98],type="l",
ylab="pchGDP",xlab="Laikas",lwd=2)
lines(ttt[3:98],pchGDP[2:97]+
fitted(mod.f),col=2)
```

6.2 example. xxxxxxxxxxxxxxxxxxxxxx

6.2. Time Series Regression when Both Y and X are Trend Stationary: Spurious Regression

Consider the regression

$$Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + \varepsilon_t \qquad (*)$$

and assume that some (or, maybe, even all) of the variables have a linear trend, e.g., $X_{1t} = \alpha_{10} + \alpha_{11} t + \varpi_{1t}$. In this case, the seemingly (or spuriously) significant coefficients of regression (the spurious regression) can be obtained not because the response „truly" depends on predictive variables, but because of the trends „hidden" in these variables (both Y and X increase because of "progress", but Y increases not because of X increases). To clear the „true" dependence of Y on X, we can act twofoldly: **1**) to augment the (*) equation with a linear or, when necessary, polynomial trend:

$$Y_t = \gamma_0 + \gamma_1 X_{1t} + \gamma_2 X_{2t} + \delta t + \varepsilon_t \qquad (**)$$

or **2**) to rewrite the (*) equation in the form of deviations from the trend:

$$\tilde{Y}_t = \beta_0 + \gamma_1 \tilde{X}_{1t} + \gamma_2 \tilde{X}_{2t} + \varepsilon_t ; \qquad (***)$$

here, for example, $\tilde{X}_{1t} = X_{1t} - \hat{\alpha}_{10} + \hat{\alpha}_{11} t$. Interestingly, the γ coefficients in (**) and (***) (note that namely these coefficients describe the "true" dependence of Y on X's) are the same.

6.3 example. The files ...\PE.II.data.2010\PRMINWGE.RAW and ...PRMINWGE.DES contain the yearly data on Puerto Rican employment rate, minimum wage and other variables (this data was used to study the effects of the U.S.A. minimum wage on employment in Puerto Rico).

```
PUERTO <- scan(file.choose(),what=list(rep(0,25)),multi.line=T)
puerto = matrix(unlist(PUERTO),byrow=T,ncol=25)
head(puerto)
colnames(puerto)=c("year","avgmin","avgwage","kaitz","avgcov","covt","mfgwage",
"prdef","prepop","prepopf","prgnp","prunemp","usgnp","tt","post74",
"lprunemp","lprgnp","lusgnp","lkaitz","lprun_1","lprepop","lprep_1",
"mincov","lmincov","lavgmin")
head(puerto)
uerto=ts(puerto,start=1950,freq=1)
```

Two simple models are

$$\log(prepop_t) = \beta_0 + \beta_1 \log(mincov_t) + \beta_2 \log(usgnp_t) + \varepsilon_t$$

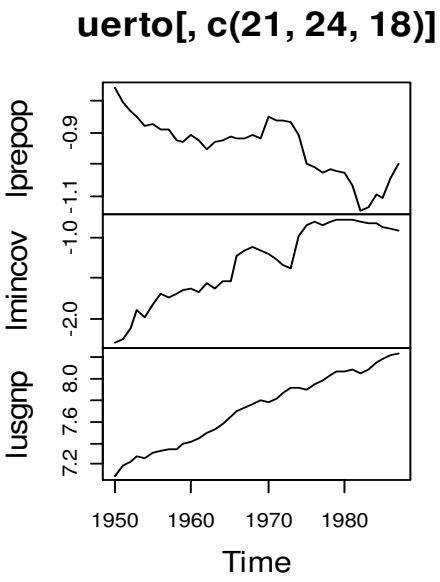
$$lprepop_t = \beta_0 + \beta_1 lmincov_t + \beta_2 lusgnp_t + \varepsilon_t$$

where

`lprepop`	log(PR employ/popul ratio)
`lmincov`	log((avgmin/avgwage)*avgcov)
`lusgnp`	log(US GNP)

avgmin is the average minimum wage, *avgwage* is the average overall wage, and *avgcov* is the average coverage rate (the proportion of workers actually covered by the minimum wage law).

```
plot(uerto[,c(21,24,18)])
```



All the variables have a trend close to linear (accurate analysis would require to test for unit root first!).

```
summary(lm(lprepop~lmincov+lusgnp,data=uerto))
Coefficients:
            Estimate    Std.Error t-value Pr(>|t|)
(Intercept) -1.05441    0.76541   -1.378   0.1771
lmincov     -0.15444    0.06490   -2.380   0.0229 *
lusgnp      -0.01219    0.08851   -0.138   0.8913
```

Thus, if the minimum wage increases then employment declines which matches classical economics. On the other hand, the GNP is not significant but we shall scrutinize the claim right now:

```
> summary(lm(lprepop~lmincov+lusgnp+tt,data=uerto))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.696292   1.295770  -6.711 1.04e-07 ***
lmincov     -0.168695   0.044246  -3.813 0.000552 ***
lusgnp       1.057350   0.176638   5.986 8.98e-07 ***
tt          -0.032354   0.005023  -6.442 2.31e-07 ***
```

Here, we interpret the coefficient of `lusgnp` as follows: if `usgnp` raises 1% more then it should accordind to its long run trend, `prepop` will raise extra 1.057%.

The above regression is equivallent to this:

```
summary(lm(lm(lprepop~tt)$res~lm(lmincov~tt)$res+lm(lusgnp~tt)$res,data=uerto))
Coefficients:
                      Estimate Std. Error  t value Pr(>|t|)
(Intercept)           2.351e-18  6.064e-03 3.88e-16 1.000000
lm(lmincov ~ tt)$res -1.687e-01  4.361e-02   -3.868 0.000456 ***
lm(lusgnp ~ tt)$res   1.057e+00  1.741e-01    6.073 6.18e-07 ***
```

6.1 exercise. (A spurious regression example). The file …\Shumway\globtemp2.dat has 142 observations starting from the year 1856 (yearly average global temperature deviations from the 1961-1990 average). Properly organize and import the file. Create a two-dimensional (from 1964 till

1987) time series containing the `Annual_Mean` column and also `lusgnp` column from the uerto data set. How do you think, does the GDP of the U.S.A. depends on the global temperature? Plot the necessary graphs. Create two regression models: the first, without the `year` variable on the rhs and another containing the variable. If your conclusions differ, explain. ◀◀

6.2 exercise. Generate two series $Y_t = \beta_0 + \beta_1 t + \varpi_{1t}$ and $X_t = \alpha_0 + \alpha_1 t + \varpi_{2t}$ where both ϖ_{1t} and ω_{2t} are WN. Create two regression models: $Y_t = \gamma_0 + \gamma_1 X_t + \varepsilon_{1t}$ and $Y_t = \chi_0 + \chi_1 X_t + \delta t + \varepsilon_{2t}$. Comment the results.

6.3. Time Series Regression when Y and X Have Unit Roots: Spurious Regression

Let us consider the regression model

$$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t \qquad (6.1)$$

and assume that both variables are integrated, i.e., have unit roots. If, in addition, the variables are cointegrated (that is, the errors ε_t of the OLS model constitute a stationary process) all the usual procedures of estimating the coefficients, making inferences etc are still valid (the case will be analyzed in the next section). On the other hand, if the errors ε_t are not stationary, but still have unit root, the relationship between Y_t and X_t will only be seeming or spurious (despite big R^2 and „significant“ coefficient), the regression, as a rule, will have no economic meaning. The reason of this phenomenon is the fact that the OLS estimators are now inconsistent and the $t-$statistics has another, not Student‘s, distribution.

Let us consider a revealing example: generate two <u>independent</u> random walks $Y_t = Y_{t-1} + w_{Yt}$ and $X_t = X_{t-1} + w_{Xt}$ where w_{Yt} and w_{Xt} are two independent white noises. It is clear that equation (6.1) is senseless (in other words, β_1 must be 0 since Y_t and X_t are not related in any sense), however, once 1000 Monte-Carlo experiments are performed, it is easy to verify that the hypothesis $H_0 : \beta_1 = 0$ will be rejected with, say, 5% significance, much more often than 5 times in 100.

```
ilgis=50        # The length of random walk
kartai=1000     # The number of Monte-Carlo experiments
set.seed(1)
p.value=numeric(kartai)    # A vector to place the p-values of H0
for(i in 1:kartai)         # Loop
{
y=ts(cumsum(rnorm(ilgis)))   # Generate random walk y
x=ts(cumsum(rnorm(ilgis)))   # Generate random walk x
p.value[i]=summary(lm(y~x))$coef[2,4]  # The p-value of H0
}
print(sum(ifelse(p.value<0.05,1,0))/kartai) # The frequency of regressions
                                            # with p<0.05
[1] 0.67   # 67 kartus iš 100 H0 atmesime
```

One outcome of this experiment is plotted in Fig. 6.1 where one can see two independent paths of random walks. Obviously, there should not be any ties between these variables, however, their scatter diagram is visibly organized and the coefficient of regression is „very significant“ (the *p*-value equals 0.000003). This quasi-significant regression is called spurious regression. Also draw your attention to the graph of residuals, it is more similar to the random walk rather than to stationary process.

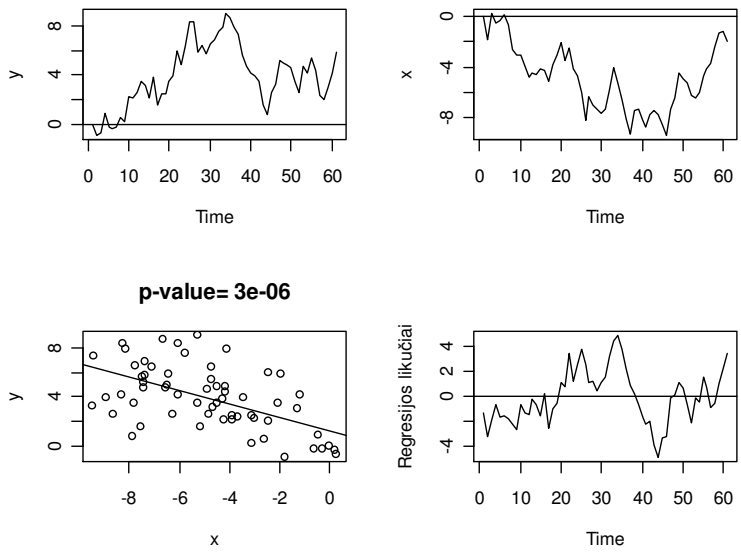


Fig. 6.1. Two graphs of random walks `y` and `x` (top row), the scatter diagram of `x` and `y` together with the regression line (bottom, left) and the graph of residuals (bottom, right)

6.4 example. Let us analyze daily IBM stock prices spanning May 17, 1961 to November 2, 1962 (369 days in all) and daily closing prices of German DAX index starting at the 130th day of 1991.

```
library(waveslim); ?ibm
library(datasets); ?EuStockMarkets
DAX=EuStockMarkets[1:369,1]
par(mfrow=c(1,3))
plot(ibm); plot(DAX,type="l")
iD=lm(ibm~DAX)
plot(DAX,ibm); abline(iD)
summary(iD)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 31.99302   74.03495   0.432    0.666
DAX          0.27347    0.04527   6.040 3.78e-09 ***
```

Though, because of their nature, `ibm` and `DAX` should not have any relationship, the coefficient of regression is very significant. This is an example of spurious regression which can be explained through the fact that the errors of the model have a unit root.

1) Establish that `ibm` has a unit root.
2) Establish that `DAX` has a unit root.
3) Now we shall verify that the errors of the `iD` model have a unit root:

```
iD.res=ts(iD$res)
summary(dynlm(d(iD.res)~L(iD.res)+time(iD.res)))
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.898716   1.088535   1.744   0.0820 .
L(iD.res)    -0.013392   0.007109  -1.884   0.0604 .
time(iD.res) -0.011303   0.005366  -2.107   0.0358 *
```

Recall that the unit root hypothesis H_0 claims that the coefficient at `L(iD.res)` equals zero. In this cointegration case, we have a further complication: here the $t-$statistics has not the Dickey – Fuller, but the Engle-Granger distribution whose critical values are given in this table:

Table 6.1 The asymptotical critical values of the Engle-Granger distribution (when testing the unit root hypothesis in the error process in the model with constant)

Number of variables (incl. Y_t)	Significance level		
	1%	5%	10%
2	−3.90	−3.34	−3.04
3	−4.29	−3.74	−3.45
4	−4.64	−4.10	−3.81
5	−4.96	−4.42	−4.13

In order to test the cointegration of these two variables note that the $t-$statistics equals -1.884 which is closer to 0 than -4.10, therefore there is no ground to reject H_0; thus the errors have a unit root or, in other words, `ibm` and `DAX` are not cointegrated and the strong regression bonds are only **spurious**. ◀◀

6.4. Time Series Regression when Y and X Have Unit Roots: Cointegration

It is well known that many economic time series are DS and therefore the regression for levels is often misleading (the standard Student or Wald tests provide wrong $p-$ values). On the other hand, if these DS series are cointegrated, the OLS method is OK. Recall that if Y and X have unit roots (i.e., are nonstationary), but some linear combination $Y_t-\alpha-\beta X_t$ is stationary, then we say that Y and X are cointegrated. In order to establish cointegration, we estimate unknown coefficients α and β by means of OLS and then test whether the errors of the model $Y_t=\alpha+\beta X_t+\varepsilon_t$ have a unit root (respective test is applied to the residuals $e_t=\hat{\varepsilon}_t=\ Y_t-\hat{\alpha}-\hat{\beta}X_t$).

6.5 example. The file ...\PE.II.data.2010\hamilton.txt contains quarterly data, 1947:01 – 1989:3, where:

`lc` logarithms of the real quarterly personal expenditure
`ly` logarithms of the real quarterly aggregated disposable income

```
ham = read.table(file.choose(),header=T)
ha = ts(ham,start=1947,freq=4)    # attach time series structure to ham
ha[1:6,]
lc=ha[,2]
ly=ha[,3]
par(mfrow=c(1,3))
plot(lc,ylab="lc&ly")
lines(ly,col=2)
plot(ly,lc)
plot(lm(lc~ly)$res,type="l")
library(dynlm)
mod2c = dynlm(d(lc)~L(lc)+L(d(lc),1:2)+time(lc))
mod0y = dynlm(..)
```

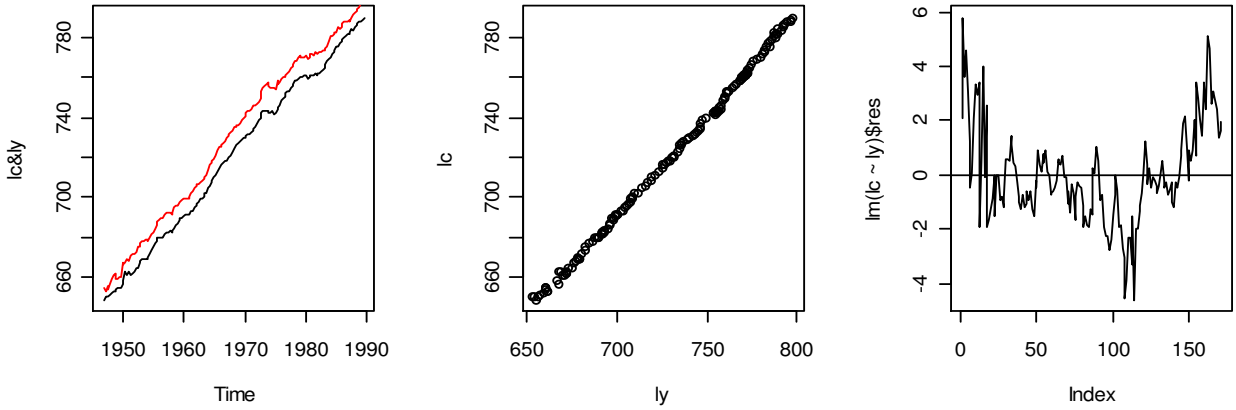


Fig. 6.2. The graphs of `lc` and `ly` (left), scatter diagram `lc` vs `ly` (center), and the residuals of the cointegration equation (right)

It is easy to verify that both series have unit roots and their final models are

```
summary(mod2c)
Call:
dynlm(formula = d(lc) ~ L(lc) + L(d(lc), 1:2) + time(lc))

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -307.49203  129.86620  -2.368   0.0191 *
L(lc)           -0.05204    0.02176  -2.392   0.0179 *    > -3.45
L(d(lc), 1:2)1   0.06550    0.07592   0.863   0.3895
L(d(lc), 1:2)2   0.23949    0.07586   3.157   0.0019 **
time(lc)         0.17553    0.07390   2.375   0.0187 *
summary(mod0y)
Call:
dynlm(formula = d(ly) ~ L(ly))

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.064366   1.478031   1.397    0.164
L(ly)       -0.001679   0.002024  -0.829    0.408          > -2.89
```

Now we shall test whether the errors of the model $lc_t = \alpha + \beta \cdot ly_t + \varepsilon_t$ make a stationary process, i.e., do not have unit root. The arguments of the function `dynlm` must be time series, therefore we shall endow this structure to the residuals of the cointegration model:

```
cy.res=ts(lm(lc~ly)$res,start=1947,freq=4)
```

We want to test whether $|\varphi|<1$ in $cy.res_t = \varphi\ cy.res_{t-1} + \varepsilon_t$. However, if the errors of this model do not constitute WN, the standard OLS procedure gives the erroneous estimate of φ. One approach is to replace the AR(1) model by an AR(p) model. Another approach is to use the AR(1) modeliu, but to take into account the fact that the errors are not a WN (this is done by the Phillips – Ouliaris test).

1st method

This method is also called the Engle – Granger method, it tests the unit root in errors hypothesis. However, the OLS procedure proposes the coefficients to the cointegration equation such that the variance of the residuals were minimal, thus residuals are „too stationary". In other words, the null hypothesis will be rejected too often. This is why we use other critical values to test the null hypothesis $H_0 : \rho = 0$ in[1] $\Delta e_t = \rho e_{t-1} + \gamma_1 \Delta e_{t-1} + ... + \gamma_{p-1} \Delta e_{t-p+1} + w_t$.

Table 6.1 The asymptotic critical values to test the unit root hypothesis in the errors of the cointegration equation containing an intercept

Number of variables (incl. Y_t)	Significance level			Intercept + trend
	1%	5%	10%	5%
2	−3.90	−3.34	−3.04	-3.78
3	−4.29	−3.74	−3.45	-4.12
4	−4.64	−4.10	−3.81	-4.43
5	−4.96	−4.42	−4.13	-4.72

It seems that R does not have such a function therefore we shall perform the test "manually". Here is the final model of residuals:

```
summary(dynlm(d(cy.res)~-1+L(cy.res)+L(d(cy.res),1:4)))

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
L(cy.res)           -0.14504    0.05271  -2.752 0.006603 **
L(d(cy.res), 1:4)1 -0.27251    0.07981  -3.414 0.000809 ***
L(d(cy.res), 1:4)2 -0.08063    0.08171  -0.987 0.325222
L(d(cy.res), 1:4)3 -0.03778    0.08076  -0.468 0.640568
L(d(cy.res), 1:4)4 -0.22957    0.07317  -3.138 0.002025 **
```

Since -2.752 is closer to zero than -3.34, we do not reject the unit root in errors hypothesis, thus the processes `lc` and `ly` are not cointegrated. Too, the test statistics is close to critical value and the residual graph is similar to that of RW (see Fig. 6.2, right).

2nd method

The Phillips – Ouliaris cointegration test can be performed with the `ca.po` funkction from the urca package:

[1] The average value of residuals is 0, therefore the intercept is not included.

```
library(urca)
cy.data=cbind(lc,ly)
cy.po <- ca.po(cy.data, type="Pz")
summary(cy.po)

Test of type Pz
detrending of series none

Call:
lm(formula = lc ~ zr - 1)

Coefficients:
     Estimate Std. Error t value Pr(>|t|)
zrlc  0.96528    0.03449   27.98   <2e-16 ***
zrly  0.03541    0.03406    1.04      0.3

Call:
lm(formula = ly ~ zr - 1)


Coefficients:
     Estimate Std. Error t value Pr(>|t|)
zrlc  0.18636    0.04627   4.028 8.52e-05 ***
zrly  0.81713    0.04569  17.886  < 2e-16 ***

Value of test-statistic is: 51.4647
Critical values of Pz are:
                 10pct    5pct    1pct
critical values 33.9267 40.8217 55.1911
```

The test statistics 51.4647 is further from 0 than 40.8217, therefore we again reject the null hypothesis: `lc` and `ly` are cointegrated. ◀◀

6.3 exercise. Use the data on Y = `Longrate` = long-term interest rates and X = `Shortrate` = short-term interest rates in ...\PE.II.data.2010\interestrates.xls.

(a) Use a sequential procedure and/or the Dickey–Fuller tests to verify that Y and X have unit roots.
(b) Run a regression of Y on X and save the errors.
(c) Carry out a unit root test on the residuals using an AR(1) model.
(d) Carry out a unit root test on the residuals using an AR(2) model.
(e) Carry out a unit root test on the residuals using an AR(3) model.
(f) What can you conclude about the presence of cointegration between Y and X?

(If you have done this question correctly, you will find that cointegration does seem to be present for some lag lengths, but not for others. This is a common occurrence in practical applications, so do not be dismayed by it. Financial theory and time series plots of the data definitely indicate that cointegration should occur between Y and X. But the Engle–Granger test does not consistently indicate cointegration. One possible explanation is that the Engle–Granger and Dickey–Fuller tests are known to have low power.)

(**g**) Use Phillips-Ouliaris test to verify that Y and X are cointegrated.

6.4 exercise. The data set .../DUOMENYS/Thomas – ME1997/data1.txt has yearly, 1963-1992, data on the food demand in the U.S.A.

NF	expenditure on food in current prices		
RF	expenditure on food in constant prices	= Q	demand for food
NTE	total expenditure in curent prices	= X	total expenditure
RTE	total expenditure in constant prices		

The variable P (price of food) is defined as the ratio NF/RF and G (general price index) is NTE/RTE. Test whether ln(Q) (=expenditure on food in constant prices) and ln(X/G) (=total expenditure in constant prices) are cointegrated.

6.5 exercise. The data set .../DUOMENYS/Thomas – ME1997/data4.txt contains quarterly, 1974:1 – 1984:4, data on consumption C and disposable income Y in the United Kingdom. Test these variables on cointegration. These series are not seasonally smoothed, therefore try such a cointegration equation:

$$\ln C_t = \alpha + \beta_2 D_2 + \beta_3 D_3 + \beta_4 D_4 + \beta \ln Y_t + \varepsilon_t .$$

The not very favorable outcome can be explained by fact that some important variables (property, interest rate etc) are missing in the model. ◀◀

6.5. Time Series Regression when Y and X are Cointegrated: the Error Correction Model

If X and Y are integrated but not cointegrated, their ties[2] can only be spurious. In order to get sensible results, the model may be augmented with new variables or another ADL model for (stationary differences) created:

$$\Delta Y_t = \varphi + \gamma_1 \Delta Y_{t-1} + ... + \gamma_p \Delta Y_{t-p} + \omega_0 \Delta X_t + ... + \omega_q \Delta X_{t-q} + \varepsilon_t .$$

Similar model can also be analyzed in the case of cointegration, but then one more term, the so-called error correction term e_{t-1} (here $Y_t = \hat{\alpha} + \hat{\beta} X_t + e_t$ is a long-run equilibrium or cointegration equation) should be included:

$$\Delta Y_t = \varphi + \delta t + \lambda e_{t-1} + \gamma_1 \Delta Y_{t-1} + ... + \gamma_p \Delta Y_{t-p} + \omega_0 \Delta X_t + ... + \omega_q \Delta X_{t-q} + \varepsilon_t . \quad (▲)$$

This is the error correction model (ECM) where the coefficient λ is usually negative[3] (e.g., if $\lambda = -0.25$, then unit deviation from equilibrium will be compensated in 1/0.25=4 time units).

The ECM is usually created in two steps.

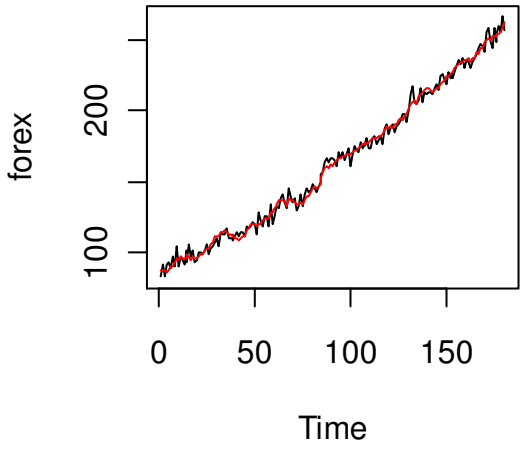
[2] We mean regression ties for levels.

[3] If e_{t-1} is positive, that is, Y_{t-1} is above the equilibrium $\hat{\alpha} + \hat{\beta} X_{t-1}$, then negative value of λe_{t-1} will shift Y_t towards equilibrium.

1. Estimate the long-run equilibrium model $Y_t = \hat{\alpha} + \hat{\beta} X_t + e_t$ and save its residuals.
2. Estimate the regression model (▲).

6.6 example. 181 monthly data on the spot and forward rate of a certain currency are located in ...\PE.II.data.2010\forexN.xls. Both rates are similar to a random walk with drift (see figure on the right, `fr` is red), however, it seems that they are wandering in a similar manner (in other words, they are cointegrated).

```
forex = ts(read.delim2("clipboard",
header=TRUE),start=1,freq=1)
colnames(forex)=c("sr","fr")
forex[1:6,]
```



Let us test their cointegration (...)

(...) – then the ECM

6.7 example. We have already analyzed (in 6.4 exercise) the data set .../DUOMENYS/Thomas – ME1997/data1.txt and found that the variables $\ln Q$ and $\ln(X/G)$ are cointegrated, that is, ... (end the definition).

```
d1=ts(read.table(file.choose(),header=TRUE),start=1963)
d1[1:6,]
X=d1[,"NTE"]
Q=d1[,"RF"]
P=d1[,"NF"]/d1[,"RF"]
G=d1[,"NTE"]/d1[,"RTE"]
c1=lm(log(Q)~log(X/G))
summary(c1)           # cointegration (equilibrium) equation
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.11378    0.25406   28.00   <2e-16 ***
log(X/G)     0.35537    0.01778   19.99   <2e-16 ***

Residual standard error: 0.02478 on 28 degrees of freedom
Multiple R-squared: 0.9345,     Adjusted R-squared: 0.9322
F-statistic: 399.5 on 1 and 28 DF,  p-value: < 2.2e-16
c1res=ts(c1$res,start=1963)
```

We use the residuals of the cointegration equation `c1` and create a regression model

$$\Delta \ln Q_t = \lambda \cdot \mathrm{c1res}_{t-1} + \gamma_1 \Delta \ln Q_{t-1} + \omega_0 \Delta \ln(X/G)_t + \omega_1 \Delta \ln(X/G)_{t-1} + \varepsilon_t .$$

```
library(dynlm)
```

```
mod1 = dynlm(d(log(Q))~ -1 + L(c1res) + L(d(log(Q))) + L(d(log(X/G)),0:1))
summary(mod1)
Time series regression with "ts" data:
Start = 1965, End = 1992

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
L(c1res)             -0.4233     0.1208  -3.504 0.001823 **
L(d(log(Q)))          0.4051     0.1592   2.545 0.017794 *
L(d(log(X/G)), 0:1)0  0.5368     0.1347   3.984 0.000548 ***
L(d(log(X/G)), 0:1)1 -0.2532     0.1387  -1.825 0.080523 .

Residual standard error: 0.01307 on 24 degrees of freedom
Multiple R-squared: 0.7338,     Adjusted R-squared: 0.6894
F-statistic: 16.54 on 4 and 24 DF,  p-value: 1.242e-06
```

The error correction term has the right sign and is significant, the model itself is quite acceptable. We can still improve the model by including other differences of $I(1)$ variables, e.g., log-differences of relative food prices P/G:

$$\Delta \ln Q_t = \lambda \cdot \mathrm{c1res}_{t-1} + \gamma_1 \Delta \ln Q_{t-1} + \omega_0 \Delta \ln(X/G)_t + \omega_1 \Delta \ln(X/G)_{t-1} + \varpi_0 \Delta \ln(P/G)_t + \varpi_1 \Delta \ln(P/G)_{t-1} + \varepsilon_t$$

After deleting insignificant terms, we get such an ECM:

```
mod2 = dynlm(d(log(Q))~-1+L(c1res)+L(d(log(Q)))+d(log(X/G))+d(log(P/G)))
summary(mod2)

Time series regression with "ts" data:
Start = 1965, End = 1992

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
L(c1res)    -0.40437    0.10795  -3.746 0.000999 ***
L(d(log(Q)))  0.24010    0.13413   1.790 0.086080 .
d(log(X/G))   0.36579    0.08985   4.071 0.000440 ***
d(log(P/G))  -0.32361    0.10135  -3.193 0.003906 **

Residual standard error: 0.01169 on 24 degrees of freedom
Multiple R-squared: 0.7872,     Adjusted R-squared: 0.7518
F-statistic:  22.2 on 4 and 24 DF,  p-value: 8.984e-08
```

Here the short-run demand elasticities with respect to total real expenditure and relative price are 0.36579 and 0.32361, respectively. In the long-run, however, demand is independent of relative price, while the expenditure elasticity falls sligthly to the 0.35537 in the equilibrium relationship.

7. Multivariate Models

7.1. Granger Causality

Assume that X and Y are stationary processes generated by the ADL(1,1) equation[1]

$$Y_t = \alpha + \varphi_1 Y_{t-1} + \beta_1 X_{t-1} + \varepsilon_t$$

where, if the coefficient β_1 is significant, we say that X Granger causes[2] Y. Recall that we usually use the Student test to verify the (in)significance hypothesis $H_0 : \beta_1 = 0$. On the other hand, we can use the Fisher test to this end: the previous model is called unrestricted and the model without X_{t-1} restricted; if[3] $F = \dfrac{(SSR_R - SSR_{UR})/q}{SSR_{UR}/(T-q-p)}$ is "small"[4], that is, if $SSR_{UR} \approx SSR_R$, that is, if the model augmented with X_{t-1} has only marginally better SSR, then H_0 is accepted, i.e., X does not Granger cause Y.

In general case, we consider the ADL(p, q) model

$$Y_t = \alpha + \delta t + \varphi_1 Y_{t-1} + ... + \varphi_p Y_{t-p} + \beta_1 X_{t-1} + ... + \beta_q X_{t-q} + \varepsilon_t ; \qquad (7.1)$$

if the hypothesis $H_0 : \beta_1 = 0, ..., \beta_q = 0$ is rejected (use Fisher's test), we say that X Granger causes Y.

7.1 example. The file .../dataPE/stockpab.xls contains 133 monthly data of the most important stock price indices in Countries A and B (the data are logged) and respective stock market returns:

.....
`pchA` stock returns in Country A
`pchB` stock returns in Country B

It is easy to verify that both indices have unit roots but are not cointegrated. On the other hand, the logarithmic differences, i.e., returns, are stationary. We are interested in whether the stock returns of country A Granger causes the returns in B.

```
stock = read.delim2("clipboard",header=TRUE)[-1,]
stoc = ts(stock[,4:5],freq=12,start=2)
plot(stoc)
```

[1] Note that only lagged variables are on the rhs.
[2] This does not mean that X causes Y, it only means that information on X helps to reduce the forecast error of Y.
[3] Here q is the number of restrictions (in our case $q = 1$), and p is the number of coefficients in the restricted model (in our case $p = 2$).
[4] That is, less than the 95% quantile of the Fisher distribution with $(q, T-q-p)$ degrees of freedom.

We have already learned of the sequential procedure to choose p and q in the (7.1) model. On the other hand, in the Granger causality case, this is not the question of the first importance (and, also, we usually assume $p = q$).

```
library(dynlm)
mod1NR=dynlm(pchA~time(stoc)+L(pchA,1:2)+L(pchB,1:2),data=stoc)  #  NR  mode-
lis
summary(mod1NR)
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.0850199  1.0204824  -1.063  0.28974
time(stoc)     0.4478695  0.1432894   3.126  0.00221 **
L(pchA, 1:2)1  0.0489767  0.1659134   0.295  0.76834
L(pchA, 1:2)2 -0.0007376  0.1656757  -0.004  0.99646
L(pchB, 1:2)1  0.8607950  0.1977275   4.353 2.77e-05 ***
L(pchB, 1:2)2 -0.2339125  0.2026224  -1.154  0.25055
mod1R=dynlm(pchA~time(stoc)+L(pchA,1:2),data=stoc) # R modelis
summary(mod1R)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.33585    1.08277  -1.234  0.21960
time(stoc)     0.42007    0.15196   2.764  0.00656 **
L(pchA, 1:2)1  0.65583    0.08872   7.392 1.76e-11 ***
L(pchA, 1:2)2 -0.07830    0.08876  -0.882  0.37936
```

To compare these two models by their *SSR*, we can use the F statistics and `anova` function:

```
anova(mod1NR,mod1R)
Model 1: pchA ~ time(stoc) + L(pchA, 1:2) + L(pchB, 1:2)
Model 2: pchA ~ time(stoc) + L(pchA, 1:2)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    124 2258.8
2    126 2604.9 -2   -346.13 9.5005 0.0001449 ***
```

The models definitely differ, thus the information on `pchB` notably improves the forecast of `pchA`. On the other hand, on reversing the procedure, we can see that `pchA` is not the Granger cause of `pchB`:

```
mod2NR=dynlm(pchB~time(stoc)+L(pchB,1:2)+L(pchA,1:2),data=stoc)
summary(mod2NR)
mod2R=dynlm(pchB~time(stoc)+L(pchB,1:2),data=stoc)
summary(mod2R)
anova(mod2NR,mod2R)
Model 1: pchB ~ time(stoc) + L(pchB, 1:2) + L(pchA, 1:2)
Model 2: pchB ~ time(stoc) + L(pchB, 1:2)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    124 1604.3
2    126 1610.5 -2   -6.2165 0.2402 0.7868
```

Note that we can shorten the whole procedure with

```
library(vars)
var.2c <- VAR(stoc,p=2,type="both") # "both" means that (7.1) contains
                                    # constant and trend
```

```
causality(var.2c)
$Granger
        Granger causality H0: pchA do not Granger-cause pchB


data:  VAR object var.2c
F-Test = 0.2402, df1 = 2, df2 = 248, p-value = 0.7866
```

7.1 exercise. Import the file ...\Data\GHJ_learning(1993)\TABLE21.6.txt. The quarterly U.S. data on consumption and disposable income, 1951:2 - 1969:4, were first seasonally adjusted and then, to make them stationary, differenced (the file contains these differences). Clearly, consumption z_{1t} in the given quarter depends on this quarter income z_{2t} but also, because of constancy of habits, on the previous quarter consumption $z_{1,t-1}$. On the other hand, this quarter income is similar to that in the previous quarter as well as the previuos quarter consumption (since the increase of consumption stimulates the growth and, therefore, income). Thus, the consumption-income system can be described by

$$\begin{cases} z_{1t} = a_{10} + a_{11} z_{1,t-1} + a_{12} z_{2,t-1} + e_{1t} \\ z_{2t} = a_{20} + a_{21} z_{1,t-1} + a_{22} z_{2,t-1} + e_{2t} \end{cases}$$

Investigate the Granger causality in both equations. ◀

7.2 exercise. Employ quarterly series for 1974:1 through 1984:4 in …/dataPE /DataCY.txt to examine the direction of causality between UK disposable income `Y` and UK consumer expenditure `C`. Test whether `Y` Granger-causes `C`.

The question of whether income "causes" consumption is not as straightforward as it sounds. Both the major theories of the consumption function – the life-cycle hypothesis and the permanent income hypothesis – suggest that current consumption depends not so much on current disposable income as on some measure of total lifetime resourses. Some researches suggest that changes in consumption are entirely random and not related to income changes at all.

Using a maximum lag of three quarters, apply the Granger test and estimate a version of (7.1) with seasonal dummies. Can you prove that `Y` Granger-cause `C`? ◀

7.2. Cointegration

Let all of the $Y_{1t}, Y_{2t}, ..., Y_{Mt}$ have unit roots, i.e., are nonstationary . If there exists a linear combination $Y_t = \alpha_1 Y_{1t} + ... + \alpha_M Y_{Mt}$ such that Y_t is stationary, the collection $Y_{1t}, Y_{2t}, ..., Y_{Mt}$ is called cointegrated. If such a combination exists, the coefficients can be found with the help of the least squares procedure.

7.2 example. data4.txt ...

7.3. VAR: Estimation and Forecasting

7.3 example. Economists often use such important macroeconomic variables as: `R` = the interest rate, `M` = the money supply, `P` = the price level and `Y` = real GDP. Due to the symbols used, models using these variables are sometimes informally referred to as RMPY models (pronounced "rumpy"). The file ...\PE.II.data.2010\rmpy.xls contains quarterly data on the variables for the US from 1947:1 through 1992:4. To be precise:

- `R` three-month Treasury bill rate
- `M` money supply (M2) measured in billions of dollars
- `P` price level measured by the GDP deflator (a price index with 1987 = 1.00)
- `Y` GDP measured in billions of 1987 dollars

Before carrying out an analysis using time series data, you must conduct unit root tests. Remember that, if unit roots are present but cointegration does not occur, then the spurious regression problem exists. In this case, you should work with differenced data. Alternatively, if unit roots exist and cointegration does occur, then you will have important economic information that the series are trending together and use ECM.

In the present case, tests indicate (verify) that we cannot reject the hypothesis that unit roots exist in all variables and that cointegration does not occur. In order to avoid the spurious regression problem, we work with differenced data. In particular, we take logs of each series, then take differences of these logged series, then multiply them by 100. This implies that we are working with percentage changes in each variable (e.g., a value of 1 implies a 1% change). Thus,

- `dR` percentage change in the interest rate.
- `dM` percentage change in the money supply.
- `dP` percentage change in the price level (i.e., inflation).
- `dY` percentage change in GDP (i.e., GDP growth).

We choose somewhat arbitrarily a VAR(1) model with a linear trend.

```
RMPY=read.delim2("clipboard",header=TRUE)
RMPY[1:6,]
  Year.Qtr      Y      P      R      M  X        dY       dP        dR         dM
1    47.00 1239.5 0.1829 0.3800 197.05 NA        NA       NA        NA         NA
2    47.25 1247.2 0.1849 0.3800 199.98 NA 0.6192966 1.087558  0.000000  1.4759858
3    47.50 1255.0 0.1872 0.7367 202.09 NA 0.6234534 1.236243 66.200950  1.0495781
4    47.75 1269.5 0.1930 0.9067 203.92 NA 1.1487550 3.051262 20.763088  0.9014617
5    48.00 1284.0 0.1956 0.9900 204.47 NA 1.1357083 1.338157  8.789331  0.2693505
6    48.25 1295.7 0.1994 1.0000 203.33 NA 0.9070884 1.924110  1.005034 -0.5590991
.....................................................................

rmpy=RMPY[-1,7:10]
attach(rmpy)
library(vars)
var.1t <- VAR(rmpy, p = 1, type = "both")
summary(var.1t)

 VAR Estimation Results:
=========================
Endogenous variables: dY, dP, dR, dM
```

```
Deterministic variables: both
Sample size: 182

Estimation results for equation dY:
===================================
dY = dY.l1 + dP.l1 + dR.l1 + dM.l1 + const + trend

        Estimate Std. Error t value Pr(>|t|)
dY.l1  0.3085537  0.0757405   4.074 6.99e-05 ***
dP.l1 -0.1168849  0.0995695  -1.174  0.24202
dR.l1  0.0003808  0.0050367   0.076  0.93982
dM.l1  0.2830971  0.0840506   3.368  0.00093 ***
const  0.4986157  0.1758513   2.835  0.00511 **
trend -0.0031337  0.0014759  -2.123  0.03513 *

Estimation results for equation dP:
===================================
dP = dY.l1 + dP.l1 + dR.l1 + dM.l1 + const + trend

        Estimate Std. Error t value Pr(>|t|)
dY.l1 -0.0387799  0.0466774  -0.831  0.40721
dP.l1  0.5190143  0.0613627   8.458 1.02e-14 ***
dR.l1  0.0099351  0.0031040   3.201  0.00163 **
dM.l1  0.1206121  0.0517987   2.328  0.02102 *
const  0.1588937  0.1083737   1.466  0.14439
trend  0.0018117  0.0009096   1.992  0.04793 *


Estimation results for equation dR:
===================================
dR = dY.l1 + dP.l1 + dR.l1 + dM.l1 + const + trend

      Estimate Std. Error t value Pr(>|t|)
dY.l1  3.22423    1.11748   2.885  0.00440 **
dP.l1  1.77875    1.46905   1.211  0.22759
dR.l1  0.22188    0.07431   2.986  0.00323 **
dM.l1  3.39062    1.24008   2.734  0.00689 **
const -3.57467    2.59451  -1.378  0.17002
trend -0.05618    0.02178  -2.580  0.01070 *


Estimation results for equation dM:
===================================
dM = dY.l1 + dP.l1 + dR.l1 + dM.l1 + const + trend

        Estimate Std. Error t value Pr(>|t|)
dY.l1 -0.0315759  0.0446037  -0.708  0.47993
dP.l1  0.0606118  0.0586367   1.034  0.30270
dR.l1 -0.0129930  0.0029661  -4.380 2.03e-05 ***
dM.l1  0.7494549  0.0494975  15.141  < 2e-16 ***
const  0.3351330  0.1035592   3.236  0.00145 **
trend  0.0003412  0.0008692   0.393  0.69515
```

To compare the original series with the fitted data, one can use `plot(var.1t)` (see Fig. 7.1); the 12 months forecast can be produced with (see Fig. 7.2)

```
var.1t.prd <- predict(var.1t, n.ahead = 12, ci = 0.95)
plot(var.1t.prd)
```

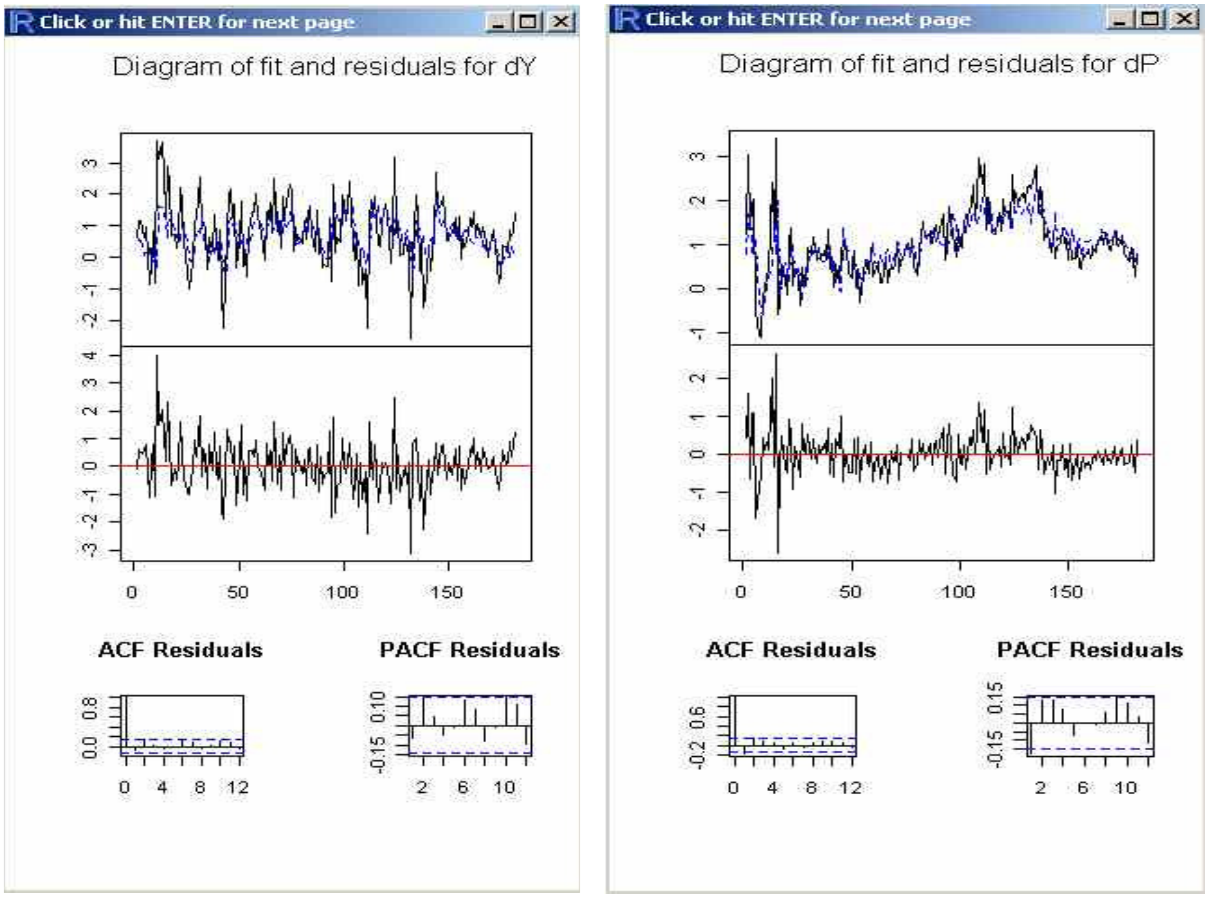


Fig. 7.1. Two (out of four) graphs of VAR

We have chosen a VAR(1) model with a constant and trend without any deeper consideration. On the other hand, one can automate the selection (use the `VARselect` function which allows to use different information criteria to choose the order):

```
> VARselect(rmpy, lag.max = 5, type="both")
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     3      2      1      3

$criteria
                1         2         3         4         5
AIC(n)  2.507590  2.335251  2.333008  2.382359  2.346143
HQ(n)   2.681563  2.625206  2.738945  2.904277  2.984044
SC(n)   2.936595  3.050259  3.334019  3.669372  3.919160
FPE(n) 12.276570 10.336949 10.322340 10.860509 10.498417
```

(thus the most conservative SC criterion suggests the 1st order).

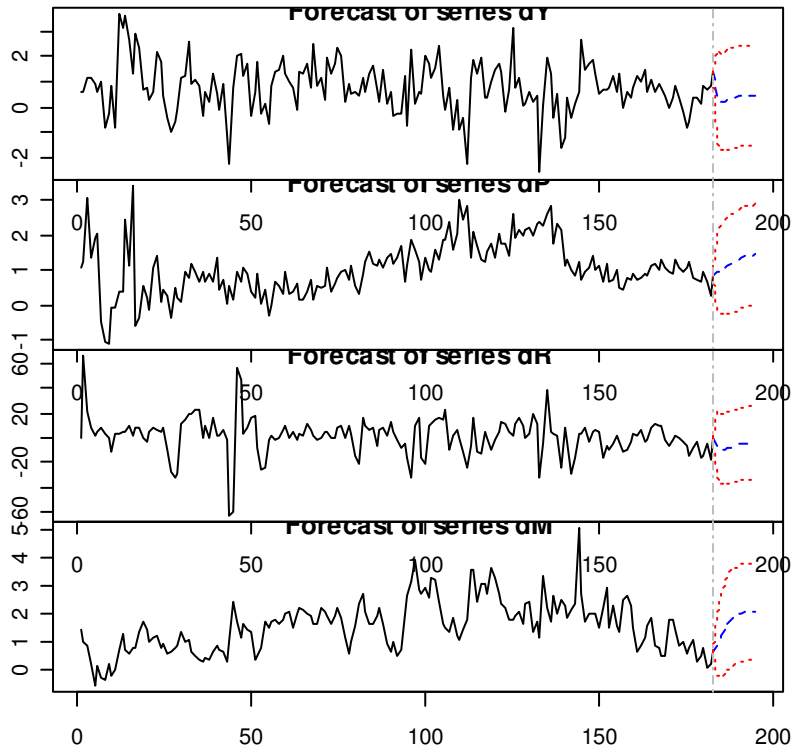


Fig. 7.2. The forecast of all four `rmpy` components.

To plot the impulse-response graphs we use the `plot(irf(...))` function. Recall that the response, generally, depends on the order of variables.

```
rmpy
          dY       dP        dR         dM
2 0.6192966 1.087558  0.000000  1.4759858
3 0.6234534 1.236243 66.200950  1.0495781
4 1.1487550 3.051262 20.763088  0.9014617
5 1.1357083 1.338157  8.789331  0.2693505
6 0.9070884 1.924110  1.005034 -0.5590991
7 0.6231988 2.035315  4.879016  0.1572559
```

```
plot(irf(VAR(rmpy, p = 1, type = "both"),impulse="dY",boot=FALSE))
```

In this case, we created the VAR(1) model using the original order of variables in the `rmpy` matrix and analyzed how all the variables reacted to the unit `dY` impulse (see Fig. 7.3, left). Now we will change the order of columns in the `rmpy` matrix:

```
plot(irf(VAR(rmpy[,4:1], p = 1, type = "both"),impulse="dY",boot=FALSE))
```

the VAR model is the same but reaction to the unit `dY` impulse is different (see Fig. 7.3, right; basically, the differences are not big).

Another variant

```
plot(irf(var.1t))
```

will plot all the responses to all the impulses together with the confidence intervals obtained via the bootstrapping procedure.

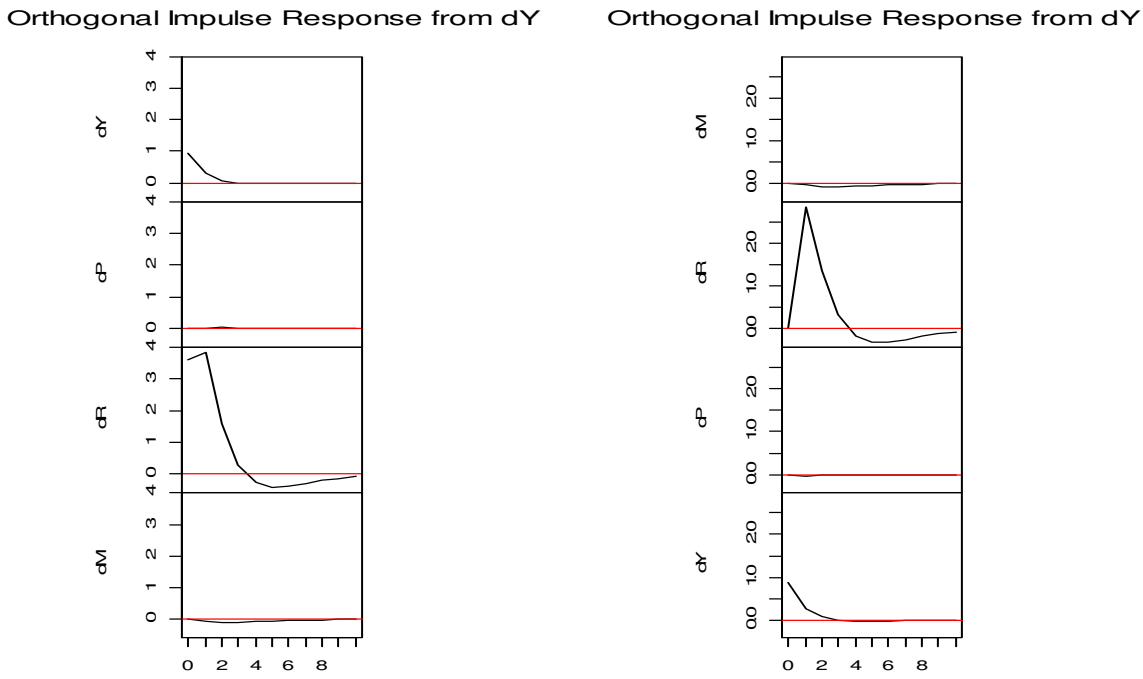


Fig. 7.3. Two variants of the reaction to the unit `dY` impulse

7.3 exercise. The package vars contains the data set `Canada`. Draw relevant graphs. Choose the right order of VAR (consider all four types: `type=c("const","trend","both","none")`). Choose a model with minimum value of SC. Estimate the model. Forecast all the variables 12 quarters ahead. Draw the impulse-response plots.

7.4. Vector Error Correction Model (VECM)

We shall model three VAR processes following the 7.8 example in the Lecture Notes. Recall that its equation

$$\begin{pmatrix} Y_{1t} \\ Y_{2t} \end{pmatrix} = \vec{\mu}_t + \Theta_1 \vec{Y}_{t-1} + \vec{\varepsilon}_t = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}$$

This two-variable VAR(1) process can be modelled in several ways. In the first case, we shall use the Dyn package.

```
library(tsDyn)
par(mfrow=c(1,3))
varcov <- matrix(c(36, 0, 0, 25),2)

B.ur <-  matrix(c(15.5, 0.5, 0, 0, 1, 1), 2) # unrestricted constant
B.ur
var.ur <- VAR.sim(B=B.ur,n=100,include="const")
ts.plot(var.ur, type="l", col=c(1,2), main="unrestricted")
# Both components are with a drift, (average) distance between them is
# constant and not equal to zero

B.rc <-  matrix(c(15.5, 0, 0, 0, 1, 1), 2) # restricted constant
B.rc
var.rc <- VAR.sim(B=B.rc,n=100,include="const")
ts.plot(var.rc, type="l", col=c(1,2), main="restricted")
# Both components are without drift, distance between them is constant and
# does not equal zero

B.nc <- matrix(c(0, 0, 1, 1), 2) # no constant
B.nc
var.nc <- VAR.sim(B=B.nc,n=100,include="none")
ts.plot(var.nc, type="l", col=c(1,2), main="no constant")
# Both components are without drift, distance between them is constant and
# equals zero
```

Explain the differences among the graphs.

7.4 exercise. The above process can be generated differently:

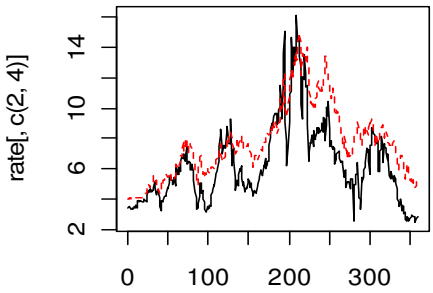
```
par(mfrow=c(1,3))

NN=100
y1=numeric(NN+3)
y2=numeric(NN+3)
mu1=15.5 # unrestriced
mu2=0.5  # this component does not equal zero
y1[1]=0
y2[1]=0
for(i in 2:(NN+3))
{
y1[i]=mu1+y2[i-1]+rnorm(1,sd=6)
y2[i]=mu2+y2[i-1]+rnorm(1,sd=5)
}
Y1=ts(y1[4:(NN+3)])   # Remove the „warming“ starter
Y2=ts(y2[4:(NN+3)])
plot(Y1, main = "unrestricted")
lines(Y2,col=2)
```

Generate three trajectories of this two-dimensional process. Do they vary? Why? Experiment with the drift `mu2`. Generate restricted and no constant processes. Test the components for the unit root. ◀◀

7.4 example. We shall create a VEC model of the vector (Y_1M, Y_5Y): the data file ustbill.txt contains monthly, 1964:01 through 1993:12, interest rates of US treasure bills for maturities of one month Y_1M and five years Y_5Y. Both series are integrated and even cointegrated (check), We have earlier created (not a very succesful) VAR(2) model for levels and differences. Since the sequences are cointegrated, the difference model lacks the error correction term; we include it using the Johansen method.

```
rate=ts(read.table(file.choose(),header=TRUE),start=1964,freq=12)
rate
     date  Y_1M  Y_1Y  Y_5Y
[1,]    1 3.419 3.789 3.984
[2,]    2 3.500 3.947 4.023
.........................
```

```
matplot(rate[,c(2,4)],type="l")
# it seems that both series do not have
# drift and the difference  in cointegra
# tion is constant
rrate=rate[,c(2,4)]
```




```
library(urca)
library(vars)
H1 <- ca.jo(rrate, ecdet="const",
K = 2) # Johansen's procedure
summary(H1)
```

Summary of the cointegratiom matrix $\Pi = \alpha\beta^T$:

```
######################
# Johansen-Procedure #
######################

Test type: maximal eigenvalue statistic (lambda max) ,
without linear trend and constant in cointegration

Eigenvalues (lambda):
[1] 7.619916e-02 1.306153e-02 6.612704e-20

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct Reject H0:r=0 because 28.37>15.67
r <= 1 |  4.71  7.52  9.24 12.97 Do not reject H0:r=1 because 4.71<9.24
r = 0  | 28.37 13.75 15.67 20.20

Eigenvectors, normalised to first column:
(These are the cointegration relations)

             Y_1M.l2   Y_5Y.l2   constant
Y_1M.l2    1.0000000    1.0000   1.000000
Y_5Y.l2   -0.9236070  113.3143  -1.183872
constant   0.9097674 -888.7860 -27.049571

Weights W:
(This is the loading matrix)
```

Here is the matrix β^T, equation $\hat{z}_t = 0.9098 + \text{Y\_1M}_t - 0.9236\,\text{Y\_5Y}_t$ is the cointegration or long-run equilibrium equation

```
           Y_1M.l2       Y_5Y.l2      constant
Y_1M.d -0.10977230 -0.0002484112  2.799703e-19
Y_5Y.d  0.04014519 -0.0001576087 -1.461470e-18
```

Here is the matrix α

The matrices β^T and α can be also expressed this way:

```
cajorls(H1)
$rlm

Call:
lm(formula = substitute(form1), data = data.mat)

Coefficients:
          Y_1M.d    Y_5Y.d
ect1      -0.10977   0.04015
Y_1M.dl1  -0.28757   0.01169
Y_5Y.dl1   0.71654   0.04408


$beta
                ect1
Y_1M.l2    1.0000000
Y_5Y.l2   -0.9236070
constant   0.9097674
```

Recall that the VECM equation is $\Delta\vec{Y}_t = \vec{\mu}_t + \Pi\vec{Y}_{t-1} + \Gamma_1\Delta\vec{Y}_{t-1} + \vec{\varepsilon}_t$. The function `cajools` parameterizes this equation differently (now the cointegration equation contains the second order lags):

```
summary(cajools(H1)) # VECM system

Response Y_1M.d :

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
Y_1M.dl1 -0.28970    0.05664  -5.114 5.18e-07 ***
Y_5Y.dl1  0.70251    0.10873   6.461 3.46e-10 ***
Y_1M.l2  -0.11002    0.03245  -3.390 0.000778 ***
Y_5Y.l2   0.07324    0.03434   2.133 0.033635 *
constant  0.12092    0.13944   0.867 0.386438

Response Y_5Y.d :

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
Y_1M.dl1  0.01034    0.03086   0.335  0.73786
Y_5Y.dl1  0.03517    0.05923   0.594  0.55301
Y_1M.l2   0.03999    0.01768   2.262  0.02432 *
Y_5Y.l2  -0.05494    0.01871  -2.937  0.00353 **
constant  0.17660    0.07596   2.325  0.02064 *
```

Recall that we can express VECM in the VAR form (and then to forecast the components of the model):

```
H1var=vec2var(H1, r=1) # transform VECM to VAR model
H1var                  # r is the rank of the matrix Π
```

Thus, the VEC form of our model is

$$\begin{cases} \Delta \mathtt{Y\_1M}_t = 0.121 - 0.290\,\Delta \mathtt{Y\_1M}_{t-1} + 0.703\,\Delta \mathtt{Y\_5Y}_{t-1} - 0.110\,\mathtt{Y\_1M}_{t-2} + 0.073\,\mathtt{Y\_5Y}_{t-2} \\ \Delta \mathtt{Y\_5Y}_t = 0.177 - 0.010\,\Delta \mathtt{Y\_1M}_{t-1} + 0.035\,\Delta \mathtt{Y\_5Y}_{t-1} + 0.040\,\mathtt{Y\_1M}_{t-2} - 0.055\,\mathtt{Y\_5Y}_{t-2} \end{cases}$$

and the VAR form

$$\begin{cases} \mathtt{Y\_1M}_t = -0.100 + 0.712\,\mathtt{Y\_1M}_{t-1} + 0.717\,\mathtt{Y\_5Y}_{t-1} + 0.178\,\mathtt{Y\_1M}_{t-2} - 0.615\,\mathtt{Y\_5Y}_{t-2} \\ \mathtt{Y\_5Y}_t = \;\; 0.037 + 0.012\,\mathtt{Y\_1M}_{t-1} + 1.044\,\mathtt{Y\_5Y}_{t-1} + 0.028\,\mathtt{Y\_1M}_{t-2} - 0.081\,\mathtt{Y\_5Y}_{t-2} \end{cases}$$

We use this model and forecast the components 60 months ahead:

```
H1pred=predict(H1var, n.ahead = 60)
plot(H1pred)


# We draw a more coherent graph (see Fig. 7.4)
Y_1M=rrate[,1]
Y_5Y=rrate[,2]
plot(seq(1964,1998+11/12,by=1/12),c(Y_1M,H1pred$fcst$Y_1M[,1]),type="l",
xlab="Time",ylab="Y")
lines(seq(1964,1998+11/12,by=1/12),c(Y_5Y,H1pred$fcst$Y_5Y[,1]),type="l",
xlab="Time",ylab="Y",col=2)
```

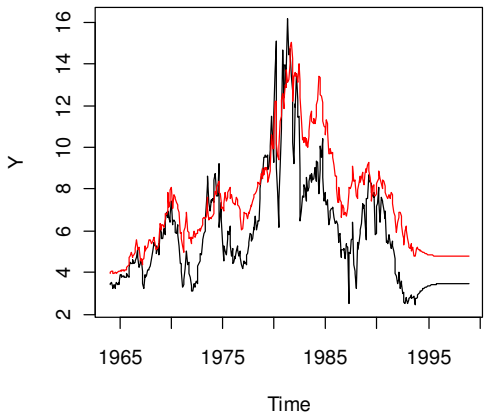


Fig. 7.4. 60 months (5 years) forecasts for `Y_1M` and `Y_5Y`

The forecast is quite reasonable and in line with the economic theory (the differences between cointegrated series must tend toward a constant). ◀◀

7.5 exercise. Repeat the previous exercise with the data set `RRate=rate[,2:4]`.

7.6 exercise. The package vars has a data set `Canada`. Describe the data. 1) Plot all the four series. 2) Use `ur.df` from the urca package and test whatever series on the unit root. Your results should be close to given in this table:

Variable	Deterministic terms	Lags	Test value	Critical values		
				1%	5%	10%
`prod`	constant, trend	2	−1.99	−4.04	−3.45	−3.15
`Δprod`	constant	1	−5.16	−3.51	−2.89	−2.58
`e`	constant, trend	2	−1.91	−4.04	−3.45	−3.15
`Δe`	constant	1	−4.51	−3.51	−2.89	−2.58
`U`	constant	1	−2.22	−3.51	−2.89	−2.58
`ΔU`		0	−4.75	−2.60	−1.95	−1.61
`rw`	constant, trend	4	−2.06	−4.04	−3.45	−3.15
`Δrw`	constant	3	−2.62	−3.51	−2.89	−2.58
`Δrw`	constant	0	−5.60	−3.51	−2.89	−2.58

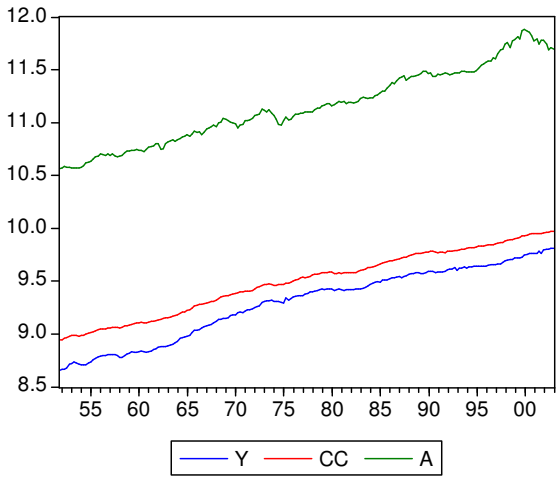
(thus, what about the unit root?) 3) Use `VARselect` to determine the VAR order (`lag.max` must be „sufficiently large“, the deterministic part of the trend of VAR, that is, choose `type="both"` (it follows from the graphs – why?). The conclusion – one can choose the 1st, 2nd or 3rd order (why?) 4) Use the `VAR` function to create the model `c.var1` for `Canada` (choose p = 1 and `type="both"`). 5) Use the command `plot(c.var1, names = "e")` to establish that the residuals do not make WN. Repeat modeling with p=2 and p=3. 6) Forecast `c.var3` 12 quarters ahead. 7) Use `ca.jo` to estimate the number of cointegration equations in the models `c.var2` (`..., ecdet=trend, K=2, ...`) and `c.var3` (`K=3`) (one equation?). 8) Transform the VEC model obtained into the VAR model (function `vec2var`) and forecast it 12 months ahead; plot relevant graphs. 9) Plot in one graph the historical values of `prod` and `rw` togerher with the VAR forecast. Do the same with the VEC forecast. ◀◀

[redacted]

7.7 exercise. We shall investigate the presence of cointegration using US *cay* quarterly data from 1951Q4 through 2003Q1 on c which is consumption (formally it is the log of real per capita expenditures on nondurables and services excluding shoes and clothing), a which is our measure of assets (formally it is the log of a measure of real per capita household net worth including all financial and household wealth as well as consumer durables), and y which is the log of after-tax labor income. This data is available in cay.txt.

⇒ In EViews

Import ccay.txt and plot the three series. They seem to be cointegrated but we have to establish the number of cointegration relations.



To test for unit root in `a`, click on `a` and choose View| Unit root test...| Include Trend and intercept| OK: we see (see below) that trend must be present in the model and respective p-value is greater than 0.05, thus we do not reject the unit root hypothesis. The same conclusion holds for `cc` and `y`.

Null Hypothesis: A has a unit root
Exogenous: Constant, Linear Trend
Lag Length: 0 (Automatic based on SIC, MAXLAG=14)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-2.525775	0.3154
Test critical values:	1% level	-4.003449	
	5% level	-3.431896	
	10% level	-3.139664	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(A)
Method: Least Squares
Date: 05/13/11 Time: 10:39
Sample(adjusted): 1952:1 2003:1
Included observations: 205 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
A(-1)	-0.064201	0.025418	-2.525775	0.0123
C	0.682687	0.267849	2.548772	0.0116
@TREND(1951:4)	0.000376	0.000153	2.454859	0.0149

To choose the lag length in the VAR model, in the Workfile window, choose Objects| New Object...| VAR| in Endogenous Variables window type `a cc y`| OK| the last lines in Var window show (that for a 2nd order model without trend)

Akaike Information Criteria	-19.88742
Schwarz Criteria	-19.54585

Note that this is the minimum value of AIC and SC among similar models. Recall that this implies that the VECM order will be 1 which is partially confirmed by the following table:

coint(s,4) a cc y

Date: 05/13/11 Time: 11:52
Sample: 1951:4 2003:1
Included observations: 201
Series: A CC Y
Lags interval: 1 to 4

Data Trend:	None	None	Linear	Linear	Quadratic
Rank or No. of CEs	No Intercept No Trend	Intercept No Trend	Intercept No Trend	Intercept Trend	Intercept Trend
Selected (5% level) Number of Cointegrating Relations by Model (columns)					
Trace	1	1	0	0	0
Max-Eig	1	1	0	0	0
Akaike Information Criteria by Rank (rows) and Model (columns)					
0	-19.80021	-19.80021	-19.89568	-19.89568	-19.87264
1	-19.86739	-19.86944	-19.91056*	-19.90438	-19.89101
2	-19.83528	-19.86886	-19.86172	-19.89247	-19.88726
3	-19.78439	-19.81006	-19.81006	-19.83303	-19.83303
Schwarz Criteria by Rank (rows) and Model (columns)					
0	-19.20857	-19.20857	-19.25474*	-19.25474*	-19.18240
1	-19.17714	-19.16276	-19.17101	-19.14840	-19.10216
2	-19.04643	-19.04715	-19.02357	-19.02145	-18.99980
3	-18.89693	-18.87331	-18.87331	-18.84697	-18.84697

The results are ambiguous – Akaike suggests that only one cointegrating relation exists while Schwarz claims that there is no such relations. We choose one.

⇒ In R

Create the VEC model. ◀◀

10. Panel Data Analysis

Panel data (also known as longitudinal or cross-sectional time-series data) is a dataset in which the behavior of entities are observed across time. These entities could be states, companies, individuals, countries, etc.

country	year	Y	X1	X2	X3
1	2000	6.0	7.8	5.8	1.3
1	2001	4.6	0.6	7.9	7.8
1	2002	9.4	2.1	5.4	1.1
2	2000	9.1	1.3	6.7	4.1
2	2001	8.3	0.9	6.6	5.0
2	2002	0.6	9.8	0.4	7.2
3	2000	9.1	0.2	2.6	6.4
3	2001	4.8	5.9	3.2	6.4
3	2002	9.1	5.2	6.9	2.1

Panel data allows you to control for variables you cannot observe or measure like cultural factors or difference in business practices across companies; or variables that change over time but not across entities (i.e. national policies, federal regulations, international agreements, etc.). This is, it accounts for individual heterogeneity[1].

In this chapter we focus on three techniques to analyze panel data:

- Pooled model
- Fixed effects
- Random effects

In the sequel, we shall use R and its `plm` package whose manual can be downloaded from http://cran.r-project.org/web/packages/plm/vignettes/plm.pdf .

****

The fundamental advantage of a panel data set over a cross section is that it will allow the researcher great flexibility in modeling differences in behavior across individuals. The basic framework for this discussion is a regression model of the form

$$Y_{it} = \beta_1 X_{it}^{(1)} + ... + \beta_K X_{it}^{(K)} + \alpha + \alpha_1 Z_i^{(1)} + ... + \alpha_H Z_i^{(H)} + \varepsilon_{it} =$$
$$\beta_1 X_{it}^{(1)} + ... + \beta_K X_{it}^{(K)} + c_i + \varepsilon_{it}, i = 1,...,I, t = 1,...,T$$

There are *K* regressors, not including a constant term. The heterogeneity, or individual effect, is $c_i = \alpha Z_i^{(0)} (=1) + \alpha_1 Z_i^{(1)} + ... + \alpha_H Z_i^{(H)}$ where $\vec{Z}_i$ contains a constant term and a set of individual or group specific variables, which may be observed, such as race, sex, location, and so on, or unobserved, such as family specific characteristics, individual heterogeneity in skill or preferences, and so on, all of which are taken to be constant over time *t*. As it stands, this model is a classical regression model. If $\vec{Z}_i$ is observed for all individuals, then the entire model can be treated as an ordinary linear model and fit by least squares. The complications

[1] The quality of being diverse and not comparable in kind.

arise when c_i is unobserved, which will be the case in most applications (for example, analyses of the effect of education and experience on earnings from which "ability" will always be a missing and unobservable variable).

The main objective of the analysis will be consistent and efficient estimation of the partial effects $\beta_1,...,\beta_K$. Whether this is possible depends on the assumptions about the unobserved effects.

1. Pooled Regression. If $\vec{Z}_i$ contains only a constant term: $Y_{it} = \beta_1 X_{it}^{(1)} + ... + \beta_K X_{it}^{(K)} + \alpha + \varepsilon_{it}$, then OLS provides consistent and efficient estimates of the common α and the slope vector $\vec{\beta}$.

2. Fixed Effects. If $\vec{Z}_i$ is unobserved, but correlated with $\vec{X}_{it}$, then the least squares estimator of $\vec{\beta}$ is biased and inconsistent as a consequence of an omitted variable. This fixed effects approach takes c_i to be a group-specific constant term in the regression model $Y_{it} = \beta_1 X_{it}^{(1)} + ... + \beta_K X_{it}^{(K)} + c_i + \varepsilon_{it}$. It should be noted that contrary to the frequent claim the term "fixed" as used here signifies the correlation of c_i and $\vec{X}_{it}$, not that c_i is nonstochastic.

3. Random Effects: If the unobserved individual heterogeneity, however formulated, can be assumed to be uncorrelated with the included variables, then the model may be formulated as

$$Y_{it} = \beta_1 X_{it}^{(1)} + ... + \beta_K X_{it}^{(K)} + \alpha + \nu_i + \varepsilon_{it}$$

that is, as a linear regression model with a compound disturbance that may be consistently, albeit inefficiently, estimated by least squares. This random effects approach specifies that ν_i is a group-specific random element, similar to ε_{it} except that for each group, there is but a single draw that enters the regression identically in each period. Again, the crucial distinction between fixed and random effects is whether the unobserved individual effect embodies elements that are correlated with the regressors in the model, not whether these effects are stochastic or not. We will examine this basic formulation, then consider an extension to a dynamic model.

If each individual in the data set is observed the same number of times, usually denoted *T*, the data set is a *balanced panel*. An *unbalanced panel* data set is one in which individuals may be observed different numbers of times. Some functions are operational only for balanced data.

10.1. Static Panel Data Models

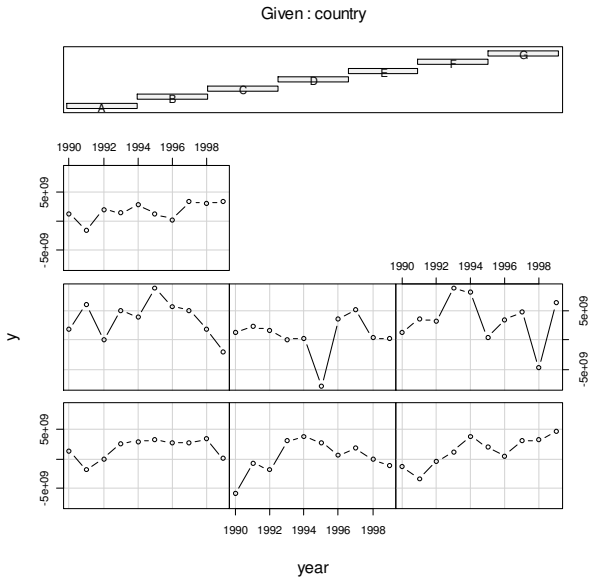
The file pp.txt contains panel data in a stacked cross-section form (7 countries: A, B etc, 10 years: 1990-1999, `y` is the output and `x1` predictive variable; this is a balanced panel):

```
pp=read.table(file.choose(),header=T)
pp
   country year           y          x1
1        F 1990  1342787840 -0.56757486
2        B 1990 -5934699520 -0.08184998
```

```
3         E 1990  1342787840  0.45286715
4         D 1990  1883025152 -0.31391269
5         G 1990  1342787840  0.94488174
6         C 1990 -1292379264  1.31256068
7         A 1990  1342787840  0.27790365
8         C 1991 -3415966464  1.17748356
9         G 1991 -1518985728  1.09872830
.....................................................
66        E 1999   243920688  0.60662067
67        D 1999 -2025476864 -0.07998896
68        B 1999 -1174480128  0.23696731
69        G 1999  3296283392  1.23420024
70        A 1999    39770336  0.39584252
```

To explore our data, type

```
attach(pp)
coplot(y~year|country,type="b")
```



Here `y` is plotted vs `year` (go left to right: the left-most graph in the bottom line is for country `A`, next to the right is for `B` etc)

Now, to explore the `country` and `year` effects on (the mean of) `y`, type

```
library(gplots)
plotmeans(y ~ country, main="Heterogeineity across countries")
plotmeans(y ~ year, main="Heterogeineity across years")
```

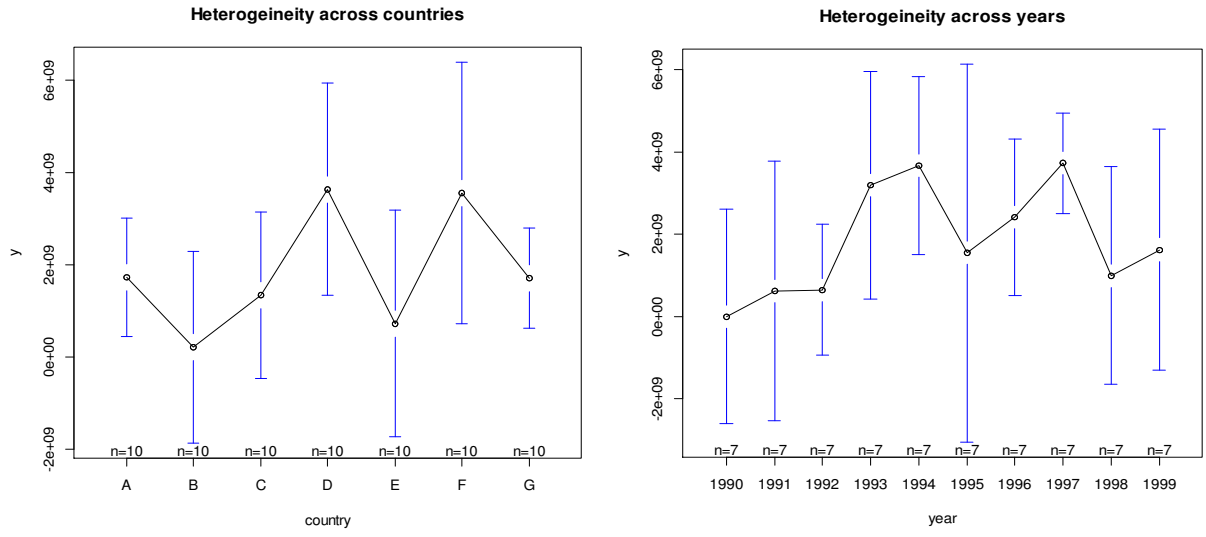


Fig. 10.1. The `country` effect on the mean of `y` (left) and the `year` (progress) effect on `y` (right) (95% confidence interval around the means is included)

The main purpose of the panel data analysis is to quantify the x1 effect on y. We start either with the pooled model

$$y_{it} = \alpha + \beta\, x1_{it} + \varepsilon_{it},\ i = 1,...,I,\ t = 1,...,T:$$

```
pooled = lm(y~x1)
summary(pooled)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.524e+09  6.211e+08   2.454   0.0167 *
x1          4.950e+08  7.789e+08   0.636   0.5272
```

or with OLS models restricted to individual countries, for example,

```
countryB=lm(y~x1,data=pp[country=="B",])
summary(countryB)
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.454e+09  1.059e+09  -3.261   0.0115 *
x1           7.139e+09  1.756e+09   4.067   0.0036 **
```

(note the difference in the green coefficients, see Fig. 10.2, center).

To compare the models, we use (see Fig. 10.2, left and center)

```
plot(x1,y,col=country,pch=15)
abline(pooled,lwd=5,lty=2)

library(car)
scatterplot(y ~ x1|country,legend.coords="topleft",smoother=FALSE,lwd=3)
```

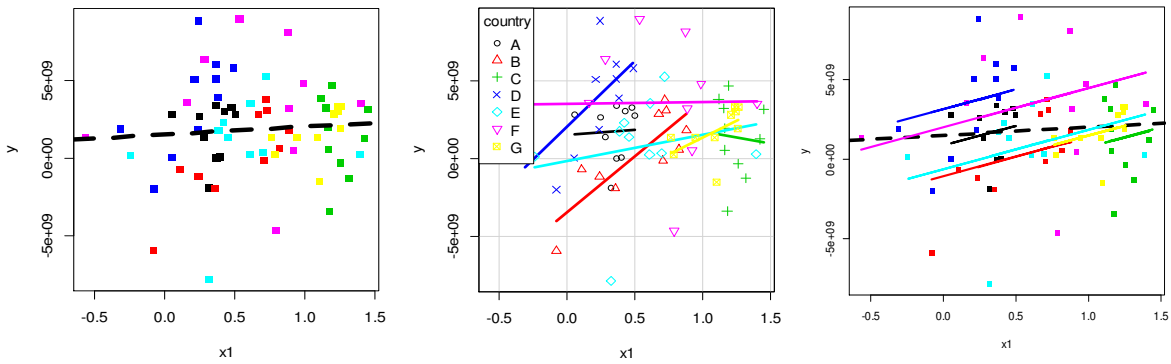


Fig. 10.2. `pooled` model (left), individual countries models (center), and `fixed` effects model together with the pooled model (right)

Both approaches have some drawbacks – the `pooled` model does not take into account heterogeneity across countries while individual model are based on small number of observations and do not consider common features of the countries (all they interact and experience the same influence of the progress). One possibility to take this common environment into account is to use the fixed effects (FE) model. To allow for the `country` effect, we introduce dummy variables (D_i =) `factor(country)`:

$$y_{it} = \alpha + \beta x1_{it} + \sum_{i=1}^{I} \nu_i D_i + \varepsilon_{it} = \begin{cases} \alpha + \beta x1_{1t} + \nu_1 + \varepsilon_{1t}, & i = 1 \\ \alpha + \beta x1_{2t} + \nu_2 + \varepsilon_{2t}, & i = 2 \\ \ldots \end{cases}$$

```
fixed = lm(y~x1+factor(country)-1)
summary(fixed)
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
x1                2.476e+09  1.107e+09   2.237  0.02889 *
factor(country)A  8.805e+08  9.618e+08   0.916  0.36347
factor(country)B -1.058e+09  1.051e+09  -1.006  0.31811
factor(country)C -1.723e+09  1.632e+09  -1.056  0.29508
factor(country)D  3.163e+09  9.095e+08   3.478  0.00093 ***
factor(country)E -6.026e+08  1.064e+09  -0.566  0.57329
factor(country)F  2.011e+09  1.123e+09   1.791  0.07821 .
factor(country)G -9.847e+08  1.493e+09  -0.660  0.51190
plot(x1,y,col=country,pch=15) # we plot regression lines for each country
abline(pooled,lwd=5,lty=2)
lines(x1[country=="A"],predict(fixed,newdata=pp[country=="A",]),col=1,lwd=3)
lines(x1[country=="B"],predict(fixed,newdata=pp[country=="B",]),col=2,lwd=3)
lines(x1[country=="C"],predict(fixed,newdata=pp[country=="C",]),col=3,lwd=3)
lines(x1[country=="D"],predict(fixed,newdata=pp[country=="D",]),col=4,lwd=3)
lines(x1[country=="E"],predict(fixed,newdata=pp[country=="E",]),col=5,lwd=3)
lines(x1[country=="F"],predict(fixed,newdata=pp[country=="F",]),col=6,lwd=3)
lines(x1[country=="G"],predict(fixed,newdata=pp[country=="G",]),col=7,lwd=3)
```

Note that now, when we take into account `country`, the coefficient at `x1` is significant and quite different from that of the `pooled` model (see Fig. 10.2, right).

To use a more systematic approach, we shall apply the `plm` (linear model for panel data) package.

```
library(plm)
pooled = plm(y ~ x1, data=pp, index=c("country", "year"), model="pooling")
summary(pooled)
Balanced Panel: n=7, T=10, N=70 # Balanced means every country was observed
Coefficients :                  # the same number of years (=10)
             Estimate Std. Error t-value Pr(>|t|)
(Intercept) 1524319070  621072624  2.4543  0.01668 *
x1           494988914  778861261  0.6355  0.52722


fixed = plm(y ~ x1, data=pp, index=c("country", "year"), model="within")
summary(fixed)
Coefficients :
     Estimate Std. Error t-value Pr(>|t|)
x1 2475617827 1106675594   2.237  0.02889 *

fixef(fixed) # display the fixed effects (constants for each country)
          A           B           C           D           E           F           G
 880542404 -1057858363 -1722810755  3162826897  -602622000  2010731793  -984717493

pFtest(fixed, pooled) # The F-test is used to test H0:all the constants
                      # equal 0
F test for individual effects
data:  y ~ x1
F = 2.9655, df1 = 6, df2 = 62, p-value = 0.01307 # if p-value is less than
alternative hypothesis: significant effects      # 0.05, then the fixed ef-
                                                 # fects model is better
```

Usually, there are too many parameters in the FE model and the loss of degrees of freedom can be avoided if ν_i in $y_{it} = \alpha + \beta x1_{it} + \nu_i + \varepsilon_{it}$ are assumed random (more specifically, $\nu_i \sim i.i.d.(0, \sigma_\nu^2)$, $\varepsilon_{it} \sim i.i.d.(0, \sigma_\varepsilon^2)$, $E(\nu_i + \varepsilon_{it} \mid \mathbf{X}) = \sigma_\nu^2 + \sigma_\varepsilon^2$, $E((\nu_i + \varepsilon_{it}) \cdot (\nu_j + \varepsilon_{jt}) \mid \mathbf{X}) = \sigma_\nu^2$.
The just presented conditions mean that the random effects (RE) model fits into the framework of a generalized LS model with autocorrelated within a group disturbances (see PE.I, Lecture Notes, 4.9.2). In particular, the parameters of the RE model can be estimated consistently, though not efficiently, by OLS. The RE model is an appropriate specification if we are drawing I individuals randomly from a large population (this is usually the case for household panel studies; in this case, I is usually large and a fixed effects model would lead to an enormous loss of degrees of freedom).

```
random = plm(y ~ x1, data=pp, index=c("country", "year"), model="random")
summary(random)
Effects:
                    var   std.dev share
idiosyncratic 7.815e+18 2.796e+09 0.873 = σ̂ε²
individual    1.133e+18 1.065e+09 0.127 = σ̂ν²
theta:  0.3611                          -> see below

Coefficients :
             Estimate Std. Error t-value Pr(>|t|)
(Intercept) 1037014284  790626206  1.3116   0.1941
x1          1247001782  902145601  1.3823   0.1714
Total Sum of Squares:    5.6595e+20
Residual Sum of Squares: 5.5048e+20
R-Squared      :  0.02733
     Adj. R-Squared :  0.026549
F-statistic: 1.91065 on 1 and 68 DF, p-value: 0.17141
```

Interpretation of the coefficient is tricky since it includes both the effects inside a country and between countries. In the case of time series-cross sectional data, it represents the average effect of *X* over *Y* when *X* changes across time and between countries by one unit.

Which of the three models to use? One hint is given by the quantity theta = $\theta = 1 - \frac{\sigma_\varepsilon}{\sqrt{\sigma_\varepsilon^2 + T\sigma_\nu^2}}$. We always have $0 \le \theta \le 1$; if $\theta = 0$, the model becomes a pooled model, and if $\theta = 1$ a FE model. As a rule, σ_ν^2 is much bigger than σ_ε^2, thus θ or, more exactly, its estimate $\hat{\theta} = 1 - \sqrt{\hat{\sigma}_\varepsilon^2 / (\hat{\sigma}_\varepsilon^2 + T\hat{\sigma}_\nu^2)}$ must be close enough to 1. The same applies when T is big (in both these cases FE and RE models are close).

To formally decide between fixed or random effects, you can run a Hausman test where the null hypothesis is that the preferred model is RE vs. the FE alternative. It basically tests whether the unique errors ν_i are correlated with the regressors (the null hypothesis is they are not) thus if the $p-$value is significant (for example <0.05) then use fixed effects, if not use random effects.

```
phtest(fixed,random)   #Hausman Test
chisq = 3.674, df = 1, p-value = 0.05527 # both models are close
alternative hypothesis: one model is inconsistent
```

To estimate a panel data model $Y_{it} = \beta_1 X_{it}^{(1)} + ... + \beta_K X_{it}^{(K)} + \alpha + \nu_i + \varepsilon_{it}$, we use the LS methods. In the pooled regression case, we assume $\nu_i \equiv 0$; in the fixed effect case, we treat ν_i as individual dummy variables; in random effect case, the errors $\nu_i + \varepsilon_{it}$ are autocorrelated, thus we apply a generalized LS method.

10.1 example. The NLS.txt data set contains data on $N = 716$ employed women interviewed in 1982, 1983, 1985, 1987, and 1988:

```
  id year    lwage age educ south black union    exper   exper2   tenure  tenure2
1  1   82 1.808289  30   12     0     1     1 7.666667 58.77777 7.666667 58.77777
2  1   83 1.863417  31   12     0     1     1 8.583333 73.67361 8.583333 73.67361
.....................................................................................
```

`id`	identifier for panel individual; 716 total
`year`	year interviewed (1982, 1983, 1985, 1987, 1988)
`lwage`	log(wage/GNP deflator)
`age`	age in current year
`educ`	current grade completed
`south`	1 if south
`black`	1 if black; 0 if white
`union`	1 if union member
`exper`	total work experience
`exper2`	exper^2
`tenure`	job tenure, in years
`tenure2`	tenure^2

We shall create three panel models, pooled, FE, and RE, for `lwage`:

$$lwage = \beta_0 + \beta_1 \text{educ} + \beta_2 \text{exper} + \beta_3 \text{exper2} + \beta_4 \text{tenure} + \beta_5 \text{tenure2} + \beta_6 \text{black} + \beta_7 \text{south} + \beta_8 \text{union} + \varepsilon$$

```
nls=read.table(file.choose(),header=T)
library(plm)
pooled = plm(lwage ~ educ+exper+exper2+tenure+tenure2+black+south+union,
data=nls, index=c("id", "year"), model="pooling")
summary(pooled)
```

```
Balanced Panel: n=716, T=5, N=3580
Coefficients :
               Estimate  Std. Error t-value  Pr(>|t|)
(Intercept)  0.47660003  0.05615585  8.4871 < 2.2e-16 ***
educ         0.07144879  0.00268939 26.5669 < 2.2e-16 ***
exper        0.05568506  0.00860716  6.4696 1.116e-10 ***
exper2      -0.00114754  0.00036129 -3.1763 0.0015046 **
tenure       0.01496001  0.00440728  3.3944 0.0006953 ***
tenure2     -0.00048604  0.00025770 -1.8860 0.0593699 .
black       -0.11671387  0.01571590 -7.4265 1.387e-13 ***
south       -0.10600257  0.01420083 -7.4645 1.045e-13 ***
union        0.13224320  0.01496161  8.8388 < 2.2e-16 ***
```

```
Total Sum of Squares:    772.56
Residual Sum of Squares: 521.03
R-Squared      :  0.32559
      Adj. R-Squared :  0.32477
F-statistic: 215.496 on 8 and 3571 DF, p-value: < 2.22e-16
```

```
fixed = plm(lwage ~ educ+exper+exper2+tenure+tenure2+black+south+union,
data=nls, index=c("id", "year"), model="within")
summary(fixed)
Coefficients :
           Estimate  Std. Error t-value  Pr(>|t|)
exper     0.04108317  0.00662001  6.2059 6.226e-10 ***
exper2   -0.00040905  0.00027333 -1.4965    0.1346
tenure    0.01390894  0.00327784  4.2433 2.272e-05 ***
tenure2  -0.00089623  0.00020586 -4.3536 1.386e-05 ***
south    -0.01632240  0.03614900 -0.4515    0.6516
union     0.06369723  0.01425380  4.4688 8.172e-06 ***
```

```
Total Sum of Squares:    126.95
Residual Sum of Squares: 108.8
R-Squared      :  0.14297
      Adj. R-Squared :  0.11414
F-statistic: 79.4647 on 6 and 2858 DF, p-value: < 2.22e-16
```

Note that the `fixed` model

$$lwage = \delta_1 D_1 + ... + \delta_{716} D_{716} + \beta_2 exper + \beta_3 exper2 + \beta_4 tenure + \beta_5 tenure2 + \\ + \beta_7 south + \beta_8 union + \varepsilon$$

contains all individual dummies D_i but does not contain `educ` and `black` variables (they do not change in time and their presence would imply multicollinearity; indeed, if, for example, every odd individual is black, then the `black` variable (column) (1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1,…)′ may be expressed as $1 \cdot D_1 + 1 \cdot D_3 + ...$ which means exact collinearity) (complete the example and choose the best model yourselves). ◀◀

10.1 exercise. The data set fuel.xls contains the 1993-2000 information on some parameters of the 2382 enterprises in Russian fuel and energy industry. Is it possible to model their output with the Cobb-Douglas production function $Q = AK^{\alpha} L^{\beta}$?

To import the data to R, select necessary data in fuel.xls and type

```
fuell=read.delim2("clipboard",header=TRUE) # to import
fuell[1:10,] # the panel is unbalanced (why? )
fuel = fuell[apply(fuell,1,function(x) all(x>0 & is.na(x)==FALSE,
na.rm=TRUE)),] # to leave only all-positive rows
fuel[1:10,]
     okpo year okonh  emp  wor      rout        rk
5   29570 1996 11170  201  138  147.0641   398.2113
6   29570 1993 11170  158  125  226.5521   290.7230
7   29570 1995 11170  196  133  142.0815   370.2107
8   29570 1994 11170  169  132  204.5996   364.4585
23 100013 1994 11120 3769 2544 3603.0230 17221.2800
24 100013 1993 11120 3676 2493 6474.3810 16572.7000
46 100138 1993 11120 2655 1744 9686.3380 20248.3100
47 100138 1994 11120 2711 1779 4143.9430 20317.9500
```

```
48 100138 1995 11120 2757 1800 3350.6670 21066.1300
55 100167 1994 11120 1900 1087 2077.2120 11980.7400
.....
```

Here

`okpo`	enterprise number according to OKPO classification (2382 enterprises)
`year`	year (not in order, some are missing)
`okonh`	code of the branch of industry (23 different codes)
`emp`	the number of workers
`wor`	the number of involved workers
`rout`	production output
`rk`	real capital

Start with the pooled model

```
fuel.pool=lm(log(rout)~log(rk)+log(emp),data=fuel)
```

(what is the meaning of the coefficients?) Then estimate all other relevent models and choose the best. Add some graphs to your analysis. ◀◀

10.2 exercise. The `Gasoline` file in `plm` is a panel of 18 observations (`country`) from 1960 to 1978 on the gasoline consumption. Analyze the data, create all the models you know and choose the right one.

```
data(Gasoline,package="plm")
?Gasoline
Gas <- pdata.frame(Gasoline,c("country","year"),drop=TRUE)
Gas
              lgaspcar  lincomep       lrpmg   lcarpcap
AUSTRIA-1960  4.173244 -6.474277 -0.33454761  -9.766840
.....................................................
AUSTRIA-1978  3.922750 -5.762023 -0.46960312  -8.211041
BELGIUM-1960  4.164016 -6.215091 -0.16570961  -9.405527
.....................................................
U.S.A.-1977   4.811032 -5.256606 -1.17590974  -7.553458
U.S.A.-1978   4.818454 -5.221232 -1.21206183  -7.536176

formule <- lgaspcar~lincomep+lrpmg+lcarpcap
gas.pool <- plm(formule,data=Gas,model="pooling")
summary(gas.pool)                    ◀◀
```

10.2. Dynamic Panel Data Models

xxxxxxxxxxxxxxxxxxxxx