Econometrics: Time Series

Diego López Tamayo * Based on MOOC by Erasmus University Rotterdam

Contents

*El Colegio de México, diego.lopez@colmex.mx

"There are two things you are better off not watching in the making: sausages and econometric estimates." -Edward Leamer

Time series

What is a time serie.

Look at Dates and Times in R Without Losing Your Sanity to understand how to use correctly date labes in R. Datasets to be used:

```
revenue <- read_csv(
  "https://raw.githubusercontent.com/diego-eco/diego-eco.github.io/master/downloads/dataset61.csv")
revenue$YEAR<- as.Date(paste0(revenue$YEAR, '-01-01'))

production <- read_csv(
  "https://raw.githubusercontent.com/diego-eco/diego-eco.github.io/master/downloads/dataset62.csv")
# We replace the weird "M" before months.
production <- rename(production, date=`YYYY-MM`)
production$date <- gsub("M","-",production$date)
production$date <- as.Date(as.yearmon(production$date))

dataset_training <- read_csv(
  "https://raw.githubusercontent.com/diego-eco/diego-eco.github.io/master/downloads/trainexer61.csv")
```

Time series data are a specific type of data that need a somewhat special treatment when using econometric methods. The specific aspect of time series variables is that they are sequentially observed. That is, one observation follows after another. The sequential nature of time series observations has important implications for modeling and especially for forecasting and this is different from the cross-sectional data that we have mostly looked at so far.

Think of the shoe size of your next-door neighbor. Now, it is quite unlikely that the very fact that someone lives next to you, implies that this person's shoe size has predictive value for yours. But with time series data, this is different.

Yesterday's sales level, likely has predictive value for today's sales level. Just like last month's inflation has for current inflation and your last year's disposable income for this year's.

A time series variable is observed at a **regular frequency.** This can be once per year, once per month, every day and sometimes, like in some areas of finance, even each millisecond. You can imagine that recent observations on a certain time series variable can have predictive value for future observations. If it is winter-like weather today, it will most likely be so tomorrow. When unemployment is high this month, it probably is still going to be high next month.

So in terms of regression models, you may want to include the past of a variable in order to predict its future. That is, to predict a new observation of y, you can use another variable X, but you can also think of using y one period lagged. $y(-1)$.

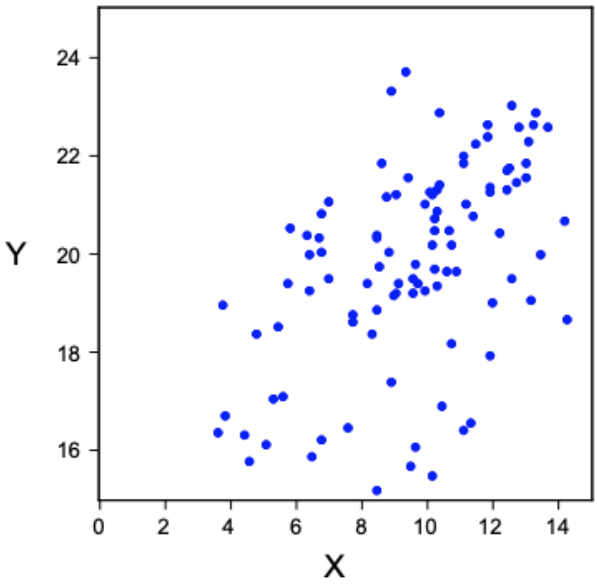
The inclusion of lagged values of the dependent variable in your regression model can also **prevent you from drawing spurious conclusions.** That is, you might think that another variable X helps to predict the variable of interest Y, while in reality, Y one period lagged predicts Y and X is irrelevant.

To illustrate this point consider two variables, X and Y, for which we know that the true data generating process is such that they depend with a factor 0.9 on their own previous value, whereas the variables X and Y are completely uncorrelated.

$$x_t = 1 + 0.9x_{t-1} + \epsilon_{x,t} \text{ and } y_t = 2 + 0.9y_{t-1} + \epsilon_{y,t}$$

$$\text{Two series completely uncorrelated } E(\epsilon_{y,t}, \epsilon_{x,s}) = 0 \forall t, s$$

A scatter of simulated Y and X variables with 100 observations may look like this.



Note that there seems to be some positive connection between the two, while we know that they are completely uncorrelated. You could be tempted to fit a simple regression model. Now suppose you would do so.

Dependent variable: Y (sample size $n = 100$)						
	Coef.	t-Stat.	p-value	Coef.	t-Stat.	p-value
Constant	15.99	23.45	0.000	2.91	2.87	0.005
X	0.40	5.78	0.000	0.07	1.53	0.129
Y(-1)	-	-	-	0.82	14.01	0.000
R-squared	0.254			0.753		

At the left-hand side of this table, you see that we estimate the slope parameter to be equal to 0.4 with a p-value of 0.000. So, this suggests that X has predictive value for Y. Now we know of course, this cannot be true given the way we created the data. The right-hand panel of the table shows what happens if we also include the Y variable one period lagged. The coefficient for this lagged variable is 0.82 and it is significant, whereas the coefficient of X is close to 0 and not statistically significant anymore.

You may now wonder whether we should have included not only X, but also X one period lagged. Consider the regression model where Y depends on Y one period lagged, X and also X one period lagged. Do X and its lag have any predictive power?

Dependent variable: Y (sample size $n = 100$)						
	Coef.	t-Stat.	p-value	Coef.	t-Stat.	p-value
Constant	2.88	2.83	0.006	2.69	2.66	0.009
Y(-1)	0.83	14.02	0.000	0.86	17.03	0.000
X	0.15	1.61	0.110	-	-	-
X(-1)	-0.09	-0.99	0.324	-	-	-
R-squared	0.756			0.747		

- Use F-test $F = \frac{(R_1^2 - R_0^2)/g}{(1-R_1^2)/(n-k)} \sim F_{(g,n-k)}$
- Number of restrictions: $g = 2$
- number of observations: $n = 100$
- number of parameters unrestricted model: $k = 4$
- values of R-squared: Unrestricted: $R_1^2 = 0.756$ and Restricted: $R_0^2 = 0.747$
- Substitute these values in formula for F-test: $F = 1.8 < 3.1$

- Joint effect of X and $X(-1)$ on Y is not significant

The larger model contains two extra variables, so the number of restrictions is two. We have 100 observations and the full model has 4 variables. The two R-squared values were reported in the table. Substituting these values in the familiar expression for the F-test gives a value of 1.8, which is smaller than the 5% critical value of 3.1. So even when we include X and one period lagged X, then these variables do not help to predict Y. Recall that the scatter of Y versus X was very suggestive, but proper analysis shows that pictures can sometimes fool us.

Example Airline revenue

Dataset: revenue

Simulated data set on yearly revenue passenger kilometers, 1975-2015 (estimation period 1976-2015, with pre-sample value for 1975).

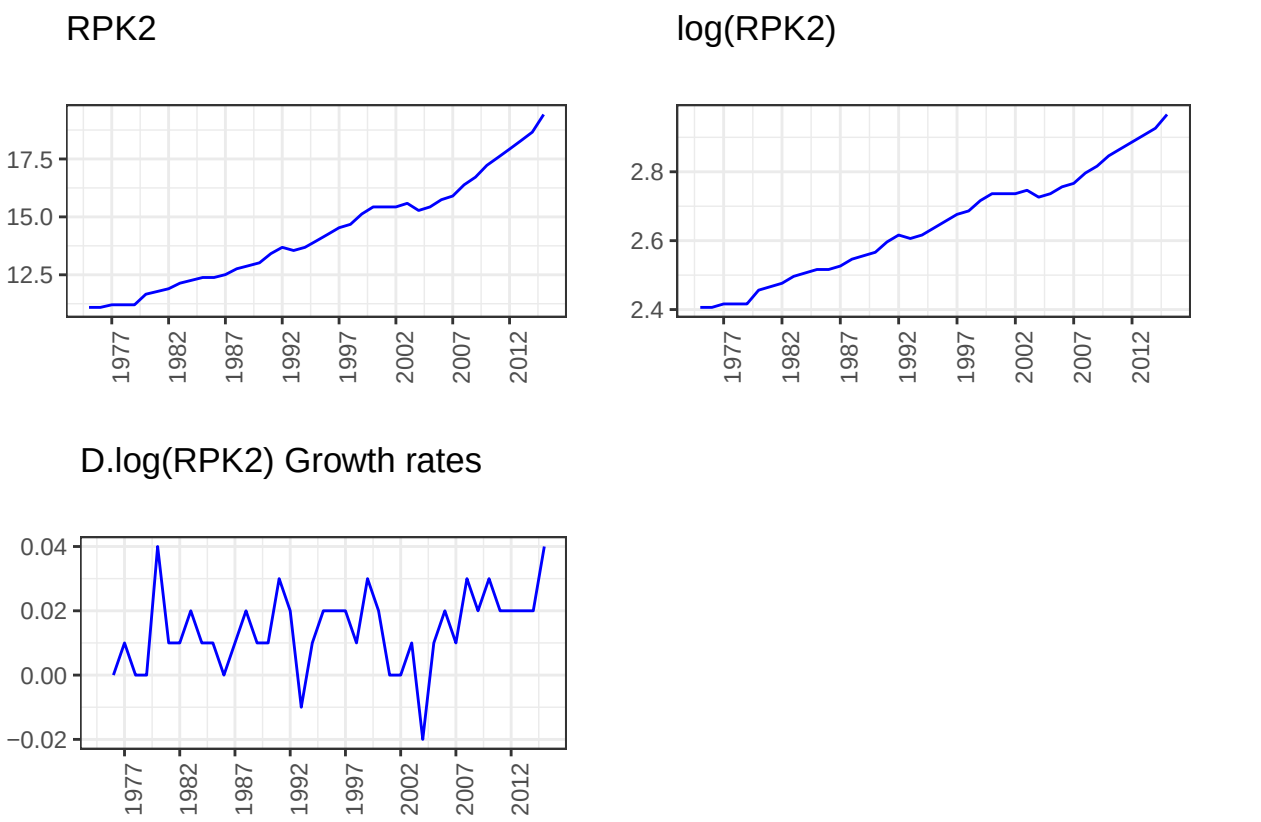
- RPK1: Revenue Passenger Kilometers of company 1 (1975-2015)
- RPK2: Revenue Passenger Kilometers of company 2 (1975-2015)
- X1: log(RPK1) (1975-2015)
- X2: log(RPK2) (1975-2015)
- DX1: first difference of X1, growth rate of RPK1 (1976-2015)
- DX2: first difference of X2, growth rate of RPK2 (1976-2015)
- Year: calendar year

Let us now look at how time series in economics and business can look like. Here is an example of passenger revenue data for an airline. The variable of interest is revenue passenger kilometers, which is the sum total over one year of the distance in kilometers traveled by each passenger on each flight of this airline company.

The left-hand graph gives the actual total number of kilometers traveled. The middle graph is obtained when taking natural logs and the right-hand graph shows the yearly growth rates.

```
# We create the log and the growth rate of both series
revenue <- revenue %>% mutate(X1=log(RPK1),DX1=c(NA,diff(log(RPK1))),X2=log(RPK2),DX2=c(NA,diff(log(RPK

plot_a <- ggplot(data=revenue, aes(x=YEAR)) +
  geom_line(aes(y=RPK2),col="blue") +
  labs(x = "", y = "", title = "RPK2",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .20),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

plot_b <- ggplot(data=revenue, aes(x=YEAR)) +
  geom_line(aes(y=X2),col="blue") +
  labs(x = "", y = "", title = "log(RPK2)",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .20),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

plot_c <- ggplot(data=revenue, aes(x=YEAR)) +
```

```
  geom_line(aes(y=DX2),col="blue") +
  labs(x = "", y = "", title = "D.log(RPK2) Growth rates",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .20),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

grid.arrange(plot_a, plot_b, plot_c, nrow = 2)
```



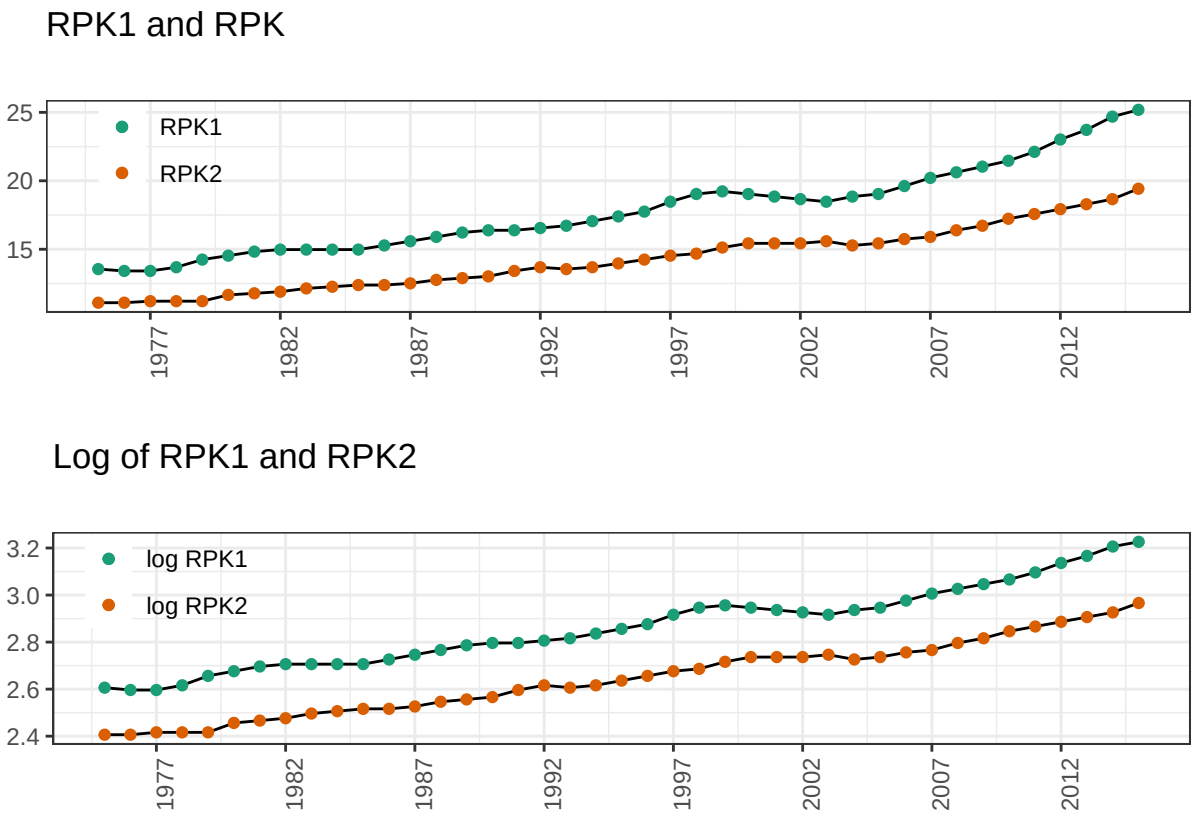
- RPK: Revenue Passenger Kilometers (in billions) yearly totals 1976-2015, trend somewhat exponential
- $log(RPK)$: more linear trend
- $D.log(RPK) = log(RPK_t) - log(RPK_{t-1}) \approx \frac{RPK_t - RPK_{t-1}}{RPK_{t-1}}$ yearly growth rate of RPK

The raw data on the left seems somewhat exponentially increasing, whereas the trend for the log of the time series seems more linear. The yearly growth rates fluctuate between minus 2% and plus 4%. The two leftmost graphs show that the data have a pronounced upward trend. When this occurs, it is not reasonable to assume that the mean of the data is constant over time. In fact, the mean increases with each new observation.

In the next section, we will deal with this important issue in more detail, as for proper statistical analysis, we need data with constant mean. **A constant mean is one aspect of what we call stationarity.** For a stationary time series like in the $D.log(RPK1)$ graph here, we have a straightforward modeling strategy. But for non-stationary time series, we will first need to get rid of this non-stationarity.

This issue of trends is even more important when two time series show similar trending behavior. Look at this graph that depicts the revenue passenger kilometers of two airlines.

```
plot_d <- ggplot(data=revenue, aes(x=YEAR)) +
  geom_line(aes(y=RPK2)) + geom_point(aes(y=RPK2,col="RPK2")) +
  geom_line(aes(y=RPK1)) + geom_point(aes(y=RPK1,col="RPK1")) +
  labs(x = "", y = "", title = "RPK1 and RPK",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.1, .8),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

plot_e <- ggplot(data=revenue, aes(x=YEAR)) +
  geom_line(aes(y=X2)) + geom_point(aes(y=X2,col="log RPK2")) +
  geom_line(aes(y=X1)) + geom_point(aes(y=X1,col="log RPK1")) +
  labs(x = "", y = "", title = "Log of RPK1 and RPK2",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.1, .8),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

grid.arrange(plot_d, plot_e, nrow = 2)
```



Clearly, they seem to have the same trend, especially when you take logs. This feature can be useful for forecasting in the following way. You may use both time series to estimate the common trend, then you can

forecast the trend. And finally, derive the individual forecast for each of the airlines. In case of a single or univariate time series, you can use its own past to make forecasts. When you have several or multivariate time series like in this example, you can try to use the other series to improve your forecasts.

Example Industrial Production

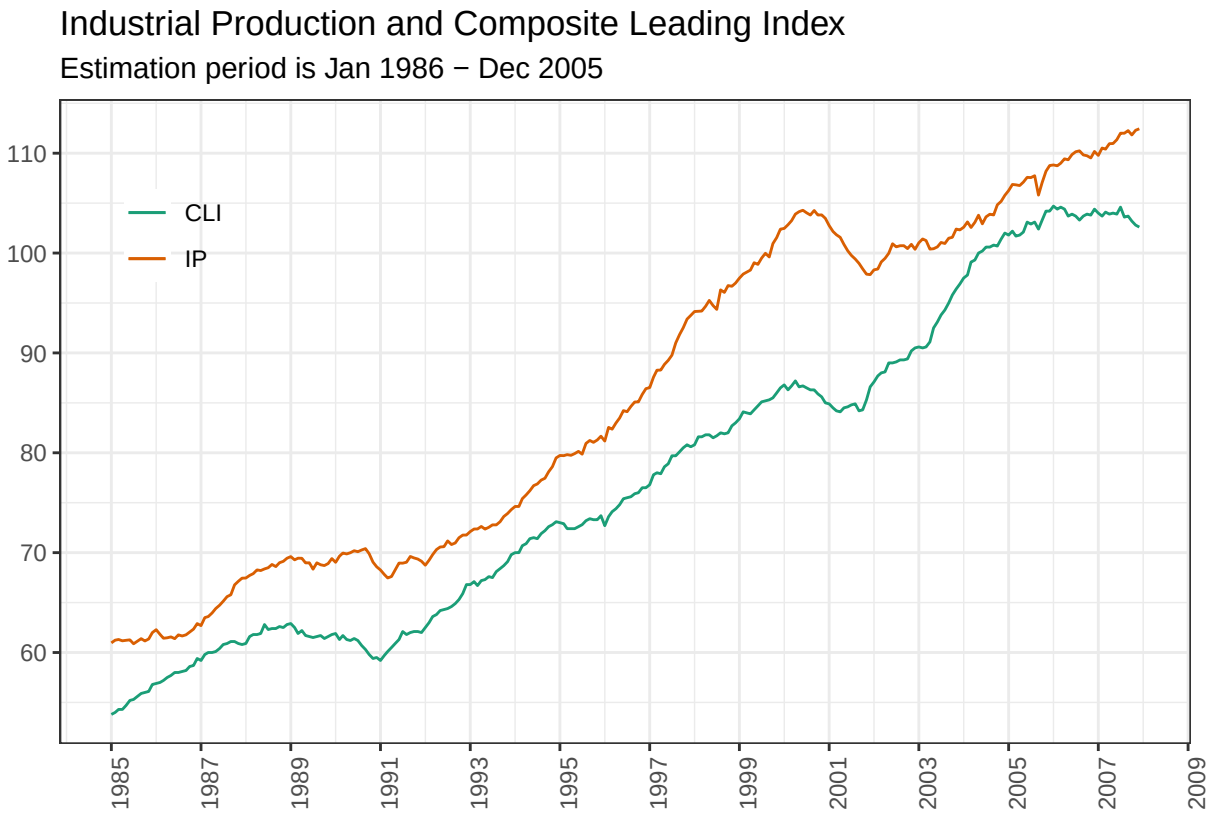
Dataset: production

Data set on Industrial Production and the Composite Leading Index for the USA, monthly data Jan 1985 - Dec 2007 (Source: Conference Board, USA). Estimation period is Jan 1986 - Dec 2005 (pre-sample values in 1985). Forecast evaluation period is Jan 2006 - Dec 2007.

- CLI: Composite Leading Index (based on 10 leading indicators)
- IP: Industrial Production (index, seasonally adjusted)
- LOGCLI: logarithm of CLI
- LOGIP: logarithm of IP
- GRCLI: monthly growthrate of CLI, first difference of LOGCLI
- GRIP: monthly growthrate of IP, first difference of LOGIP

Here is another pair of time series that are clearly related over time. These are the monthly industrial production index for the United States of America and the so-called **composite leading indicator or CLI.**

```
plot_f <- ggplot(data=production, aes(x=date)) +
  geom_line(aes(y=IP,col="IP")) +
  geom_line(aes(y=CLI,col="CLI")) +
  labs(x = "", y = "", title = "Industrial Production and Composite Leading Index ",
       subtitle = ("Estimation period is Jan 1986 - Dec 2005")) +
    scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.1, .8),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")
plot_f
```



The CLI is constructed by The Conference Board based on a set of ten variables like manufacturer's new orders, stock prices and consumer expectations. All these variables are forward looking. And therefore, they are believed to have predictive value for future macroeconomic developments. And for that reason, it may be useful to consider the CLI in case you want to forecast a variable like industrial production.

As with the airlines, the trends in industrial production and the Composite Leading Index seem to follow a similar pattern, which here associates with the business cycle. In our last section on time series, you will see if industrial production can indeed be predicted by means of this index.

Example spurious regression

Dataset: dataset_training

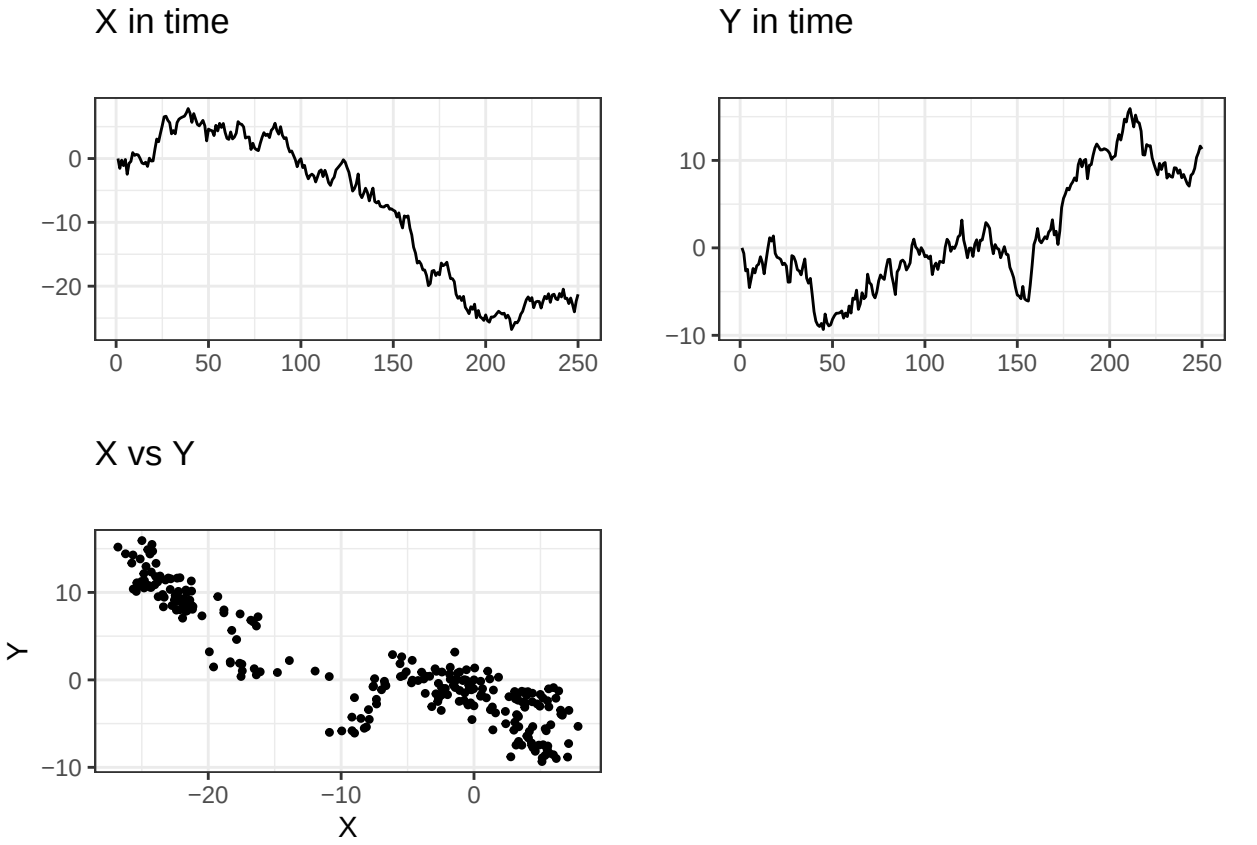
- epsx: sample of 250 values from normally and independently white noise with mean 0 and variance 1 (independent of ϵ_{yt})
- epsy: sample of 250 values from normally and independently distributed white noise with mean 0 and variance 1 (independent of ϵ_{xt})
- x: random walk generated from epsx: $x_1 = 0$, and $x_t = x_{t-1} + \epsilon_{xt}$
- y: random walk generated from epsy: $y_1 = 0$, and $y_t = y_{t-1} + \epsilon_{yt}$

The datafile contains values of four series of length 250. Two of these series are uncorrelated white noise series denoted by $\epsilon_{x,t}$ and $\epsilon_{y,t}$ where both variables are $NID(0,1)$ and $E(\epsilon_{y,t}, \epsilon_{x,s}) = 0 \forall t, s$. The other two series are so-called random walks constructed from these two white noise series by $x_t = x_{t-1} + \epsilon_{xt}$ and $y_t = y_{t-1} + \epsilon_{yt}$.

As ϵ_{xt} and ϵ_{yt} are independent for all values of t and s, the same holds true for all values of x_t and y_t The purpose of this exercise is to experience that, nonetheless, the regression of y on x indicates a highly significant relation between y and x if evaluated by standard regression tools. This kind of result is called **spurious regression** and is caused by the trending nature of the variables x and y.

- a) Graph the time series plot of x_t against time t, the time series plot of y_t against time t, and the scatter plot of y_t against x_t. What conclusion could you draw from these three graphs?

```
# We add an index column to the dataset for the time t
dataset_training <- dataset_training %>% mutate(time = row_number())
plot_1 <- ggplot(data=dataset_training, aes(x=time)) +
  geom_line(aes(y=X)) +
  labs(x = "", y = "", title = "X in time",
       subtitle = ("")) +
  theme_bw() +
  theme(legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")
plot_2 <- ggplot(data=dataset_training, aes(x=time)) +
  geom_line(aes(y=Y)) +
  labs(x = "", y = "", title = "Y in time",
       subtitle = ("")) +
  theme_bw() +
  theme(legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")
plot_3 <- ggplot(data=dataset_training, aes(x=X,y=Y)) +
  geom_point(shape=20) +
  labs(x = "X", y = "Y", title = "X vs Y",
       subtitle = ("")) +
  theme_bw() +
  theme(legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")
grid.arrange(plot_1, plot_2, plot_3, nrow = 2)
```



The two variables X,Y have completly random movements up and down. And the scatter plot seems to have a negative relation, so we could use X to forecast Y, but we know that this is not the case, the scatterplot is **misleading** in this sense.

- b) To check that the series ϵ_{xt} and ϵ_{yt} are uncorrelated, regress ϵ_{yt} on a constant and ϵ_{xt}. Report the t-value and p-value of the slope coefficient.

We use the summ() function to output our regression.

```
lm1 <- lm(EPSY ~ EPSX, data=dataset_training)
summ(lm1, digits = 3)
```

Observations	250
Dependent variable	EPSY
Type	OLS linear regression

F(1,248)	1.736
R^2	0.007
Adj. R^2	0.003

	Est.	S.E.	t val.	p
(Intercept)	0.031	0.064	0.484	0.629
EPSX	-0.088	0.067	-1.318	0.189

Standard errors: OLS

The t-value of the coefficient is around -1.32 and the p-value around 0.19, this shows that ϵ_{xt} and ϵ_{yt} have no significant relation.

- c) Extend the analysis of part (b) by regressing ϵ_{yt} on a constant, ϵ_{xt}, and three lagged values of ϵ_{yt} and of ϵ_{xt}. Perform the F-test for the joint insignificance of the seven parameters of ϵ_{xt} and the three lags of ϵ_{xt} and ϵ_{yt}. Report the degrees of freedom of the F-test and the numerical outcome of this test, and draw your conclusion. Note: The relevant 5% critical value is 2.0.

```
lm2 <- lm(EPSY ~ lag(EPSY,1) + lag(EPSY,2) + lag(EPSY,3) + EPSX + lag(EPSX,1) + lag(EPSX,2) + lag(EPSX,3
summ(lm2, digits = 3)
```

Observations	247 (3 missing obs. deleted)
Dependent variable	EPSY
Type	OLS linear regression

F(7,239)	0.546
R^2	0.016
Adj. R^2	-0.013

The $H_0 : \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = \gamma_6 = \gamma_7 = 0$ and the degrees of freedom of F test $df = (g, n-k)$ where g is the number of parameter restrictions on the null, n is the number of observations and k is the number of variables in the unrestricted model. In this case we have:

- $g = 7$ All 7 restrictions equal to 0.
- $n = 247$ Because 3 observarions are lost because the 3 lag values, so the first available observation is in $t = 4$.
- $k = 8$ Due to the 7 coefficient plus the constant term.

	Est.	S.E.	t val.	p
(Intercept)	0.046	0.066	0.698	0.486
lag(EPSY, 1)	0.025	0.064	0.387	0.699
lag(EPSY, 2)	-0.016	0.065	-0.244	0.807
lag(EPSY, 3)	-0.047	0.064	-0.734	0.464
EPSX	-0.097	0.069	-1.405	0.161
lag(EPSX, 1)	0.020	0.070	0.284	0.777
lag(EPSX, 2)	-0.060	0.070	-0.857	0.392
lag(EPSX, 3)	0.009	0.068	0.138	0.890

Standard errors: OLS

We can see the F-statistic at the top of the output or calculate it by hand $F = \frac{(R_1^2 - R_0^2)/g}{(1-R_1^2)/(n-k)} \sim F_{(g,n-k)}$ where $R_0^2 = 0$ because is a model with only a constant term. $F = 0.55$ and as is smaller of the critical value of 2. We do NOT reject the H_0. This is correct as the value of ϵ_{yt} is independent of all other observations.

- d) Regress y on a constant and x. Report the t-value and p-value of the slope coefficient. What conclusion would you be tempted to draw if you did not know how the data were generated?

```
lm3 <- lm(Y ~ X, data=dataset_training)
summ(lm3, digits = 3)
```

Observations	250
Dependent variable	Y
Type	OLS linear regression

F(1,248)	1090.611
R^2	0.815
Adj. R^2	0.814

	Est.	S.E.	t val.	p
(Intercept)	-2.487	0.214	-11.606	0.000
X	-0.515	0.016	-33.024	0.000

Standard errors: OLS

It seems by looking at the large t-value of X that X has a relevant explanatory power over Y. We know that this is not the case, so the regression is misleading, due to the trending nature of both variables. Look again at the scatterplot of a), it happens that X moves downward for long periods as Y moves upwards for long periods. This is why it seems to be a negative relation.

- e) Let e_t be the residuals of the regression of part (d). Regress e_t on a constant and the one-period lagged residual e_{t-1}. What standard assumption of regression is clearly violated for the regression in part (d)?

```
# We add the residuals of lm3 into the dataset
dataset_training <- dataset_training %>% mutate(lm3.res = resid(lm3))
lm4 <- lm(lm3.res ~ lag(lm3.res,1), data=dataset_training)
summ(lm4, digits = 3)
```

This coefficient is significant at 99%, this shows that the residuals are very strongly correlated. Therefore violates the standar regression assumption A7 that the error terms should be uncorrelated.

Observations	249 (1 missing obs. deleted)
Dependent variable	lm3.res
Type	OLS linear regression

F(1,247)	1457.056
R^2	0.855
Adj. R^2	0.854

	Est.	S.E.	t val.	p
(Intercept)	0.001	0.067	0.008	0.993
lag(lm3.res, 1)	0.925	0.024	38.171	0.000

Standard errors: OLS

Representing time series

Time series models typically are constructed with two main objectives. First, we want to describe the key properties of the time series data. In particular, the nature of the trend and the correlations with past values. And second, we may want to exploit these features to make forecasts of future observations.

The time series of interest is denoted as y, with the subscript t indicating the observation in period t. n refers to the number of available observations, or the length of the time series.

Time serie :

$$y_t \text{ where } t = 1, .., n \text{ is the time index.}$$

Stationarity

Let us begin with a very important issue, that of stationarity. A time series y is called **stationary, if its mean, variance, and covariances with past observations are constant over time.** Stationarity is an important condition that needs to be satisfied before we can even start thinking about designing a meaningful model for a given time series. Intuitively, if for each new observation of the time series properties like the mean and variance change, then we cannot reliably model such data, let alone provide reliable forecasts.

y_t is stationary if :

$$\begin{aligned} mean &= E(y_t) = \mu \text{ is fixed and same for all t} \\ autocovariance &= E[(y_t - \mu)(y_{t-k} - \mu)] = \gamma_k \text{ is same for all t} \end{aligned}$$

The autocovariances gamma k measure how strongly related observations at different points in time are. In terms of forecasting, they indicate whether past observations can be useful to make predictions of future observations.

Note that when all these autocovariances are zero, then the past carries no predictive value for the future. We call such a time series white noise.

Special case : $\gamma = k = 0 \forall k$

$$y_t \Rightarrow \text{ The time serie is White Noise}$$

Note that this relates to assumption A5 for regression models. A white noise time series cannot be predicted from its own past, and the only useful prediction is the mean of the variable itself.

A5. Uncorrelated error terms: $E(\epsilon_i \epsilon_j) = 0 \forall i \neq j$

Now this brings us to the **essence of time series modeling**. Our aim is to design a model that distills information from the past for forecasting. The model is deemed successful or adequate, if after all this distillation, there is nothing left that is informative for prediction. That is, the residuals are white noise.

Purpose of modeling time series: Create a time series model such that residuals are white noise.

In all what follows, we follow the usual convention to write a white noise variable as epsilon.

White noise

Uncorrelated series with mean zero : ϵ_t

Autoregressive model

Sometimes we call this ϵ_t the error but we also use the word **shock** to indicate that epsilon is something new to the variable y. A simple and popular time series model is the **autoregressive model**. An autoregression of order 1, or briefly **AR(1)**, is a model where the current observation of y in period t is explained by the previous observation of y in period t minus 1. This simple model provides a nice way to illustrate the relevance of stationarity.

$$\textbf{AR(1)} \; y_t = \alpha + \beta y_{t-1} + \epsilon_t$$

If the slope parameter β lies between -1 and +1, the effects of past shocks ϵ_t die out. So, the more distant in the past, the less impact those shocks have on current values of the variable y_t. This is a typical property of a stationary time series.

$$\textbf{Stationarity if } -1 < \beta < 1$$

$$y_t = \alpha + \beta y_{t-1} + \epsilon_t = \alpha + \beta(\alpha + \beta y_{t-2} + \epsilon_{t-1}) + \epsilon_t$$

$$y_t = \alpha(1+\beta) + \epsilon_t + \beta\epsilon_{t-1} + \beta^2 y_{t-2} = \alpha(1+\beta) + \epsilon_t + \beta\epsilon_{t-1} + \beta^2(\alpha + \beta y_{t-3} + \epsilon_{t-2}) = ...$$

$$y_t = \alpha \sum_{j=0}^{t-2} \beta^j + \sum_{j=0}^{t-2} \beta^j \epsilon_{t-j} + \beta^{t-1} y_1$$

$$\text{For } t \to \infty \text{ we get } \beta^{t-1} y_1 \to 0 \text{ and } y_t = \frac{\alpha}{(1-\beta)\sum_{j=0}^{\infty} \beta^j \epsilon_{t-j}}$$

Later, we will see that stationarity is lost if beta is equal to 1. The first order autoregression assumes that current y can be predicted by 1 period lagged y, but of course it might also be that 1 period lagged y and also 2 period lagged y are useful for predicting the current observation.

$$\textbf{AR(2)} \; y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \epsilon_t$$

In fact, the number of lags can run up to p, giving rise to the so-called AR(p) model.

$$\textbf{AR(p)} \; y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + + \beta_p y_{t-p} + \epsilon_t$$

Consider again the autoregression of order 1, where the current value of the white noise series epsilon is uncorrelated with the past of y.

$$\textbf{AR(1)}\ y_t = \alpha + \beta y_{t-1} + \epsilon_t$$
$$\epsilon_t \text{ uncorrelated with } y_{t-k} \forall k$$

If $\beta = 1$ then y_t can not be stationary because:

- If $\alpha \neq 0$ and $\beta = 1$, then y_t cannot have a fixed mean.

When the intercept alpha is not 0, then the mean will change by alpha for every new observation.

$$E(\epsilon_t) = 0 \text{ so } \mu = E(y_t) = \alpha + E(y_{t-1}) + 0 = \alpha + \mu \neq \mu$$

- If $\alpha = 0$ and $\beta = 1$, then y_t cannot have a fixed variance.

And when the intercept alpha is zero then the variance of the observations increases over time.

$$y_t = y_{t-1} + \epsilon_t \text{ so } (y_t - \mu) = (y_{t-1} - \mu) + \epsilon_t \text{ uncorrelated.}$$

$$E[(y_t - \mu)^2] = E[(y_{t-1} - \mu)^2] + E[\epsilon_t^2] > E[(y_{t-1} - \mu)^2]$$

Moving average model

Another useful time series model includes past shocks as explanatory variable. When you look at the epsilons as forecast errors, you can learn from these errors by taking them into account when making new forecasts. The so called **first order moving average model, or MA(1)**, includes epsilon 1 period lagged.

$$\textbf{MA(1)}\ y_t = \alpha + \epsilon_t + \gamma \epsilon_{t-1}$$

As ϵ_t is uncorrelated with it's own past and future, this model implies that y_t is correlated with y_{t-1} period lagged but not with more distant lags. y_{t-k} for $k = 2, 3...$

We can generalize this model to a moving average model of order q, which includes q lag forecast errors.

$$\textbf{MA(q)}\ y_t = \alpha + \epsilon_t + \gamma_1 \epsilon_{t-1} + ... + + \gamma_q \epsilon_{t-q}$$

ARMA model

It is also possible to combine the two models, which gives rise to an ARMA(1,1), if p and q are both equal to 1, or an ARMA(p,q), if these orders take different values.

$$\textbf{ARMA(p,q)}\ y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_p y_{t-p} + \epsilon_t + \gamma_1 \epsilon_{t-1} + ... + + \gamma_q \epsilon_{t-q}$$

Two autoregressive equations.

Moving average (MA) terms may arise when two autoregressive processes are related. Let $\epsilon_{x,t}$ and $\epsilon_{y,t}$ be be two mutually independent white noise processes, and let $y_t = \gamma x_t + \epsilon_{y,t}$ and $x_t = \delta x_{t-1} + \epsilon_{x,t}$. We can derive the orders p and q for the ARMA model for y_t (that does not include x_t).

- Consider that $y_t - \delta y_{t-1} = \gamma(x_t - \delta x_{t-1}) + \epsilon_{y,t} - \delta \epsilon_{y,t-1}$
- So we can express
$$y_t = \delta y_{t-1} + \gamma \epsilon_{x,t} + \epsilon_{y,t} - \delta \epsilon_{y,t-1}$$
Notice that this is an AR(1) order proces p=1, and error $w_t = \gamma \epsilon_{x,t} + \epsilon_{y,t} - \delta \epsilon_{y,t-1}$ is a MA(1) because $E(w_t w_{t-1}) = -\delta Var(\epsilon_{y,t-1})$ and $E(w_t w_{t-2}) = E(w_t w_{t-3}) = E(w_t w_{t-4}) = ... = 0$

We see that now y depends on y 1 period lagged, on the shock to x, the shock to y and, this is crucial, also the one period lagged shock to y. So the autoregressive order is one, whereas the moving average order is also one. Hence, **correlation, across joint autoregressive time series can lead to individual time series models of the ARMA type.**

(Partial) Autocorrelation Function

The time series models that we have discussed so far, autoregression and moving average and their combination, imply specific correlation properties of the time series. This relation can be reversed, that is, when you see certain properties of the data in the real world you can decide which model to use. And the **autocorrelation** is a very useful tool for this purpose.

k-th order sample autocorrelation coefficient :

$$ACF_k = cor(y_t, y_{t-k}) = \frac{\sum_{t=k+1}^{n}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=k+1}^{n}(y_t - \bar{y})^2}$$

For example, when the data are generated by a moving average model of order q, then the sample autocorrelations after lag q, will all be close to zero. For example, when the data are generated by a moving average model of order q, then the sample autocorrelations after lag q, will all be close to zero.

- If y_t is MA(q), then $ACF_k \approx 0$ for all $k > q$.

Next to autocorrelations, we also have the concept of **partial autocorrelations.** These account for the fact that the observations of y at time t y_t, and at time t minus two y_{t-2}, may seem to be correlated due to the fact that they both are related to the observation of y at t minus one y_{t-1}.

The sample partial autocorrelations follow from regressions of y on its own past values. If in this regression, the k-th lag coefficient is insignificant for values larger than p, then this suggests to use an AR model of this order p.

- k-th order sample partial autocorrelation coefficient: $PACF_k$ k is the OLS coefficient b_k in regression model:

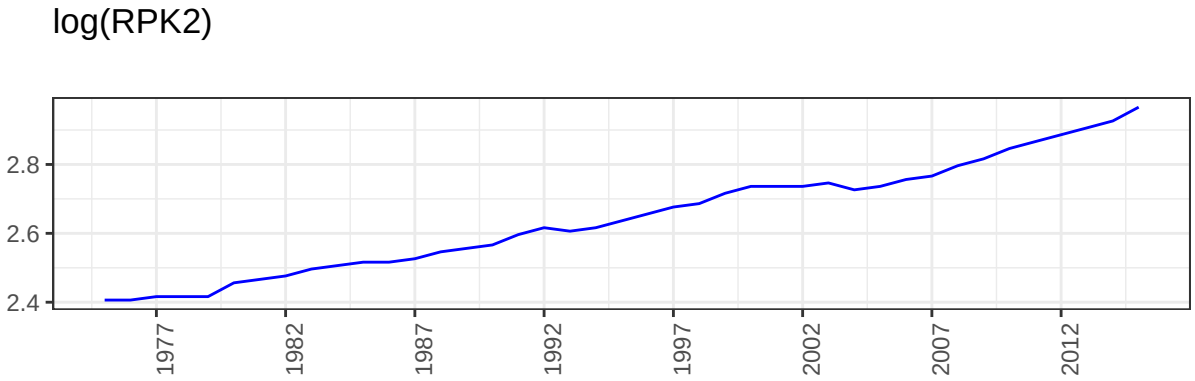
$$y_t = \alpha + \beta_1 y_{t-1} + ... + \beta_{k-1} y_{k-1} + \beta_k y_{t-k} + \epsilon_k$$

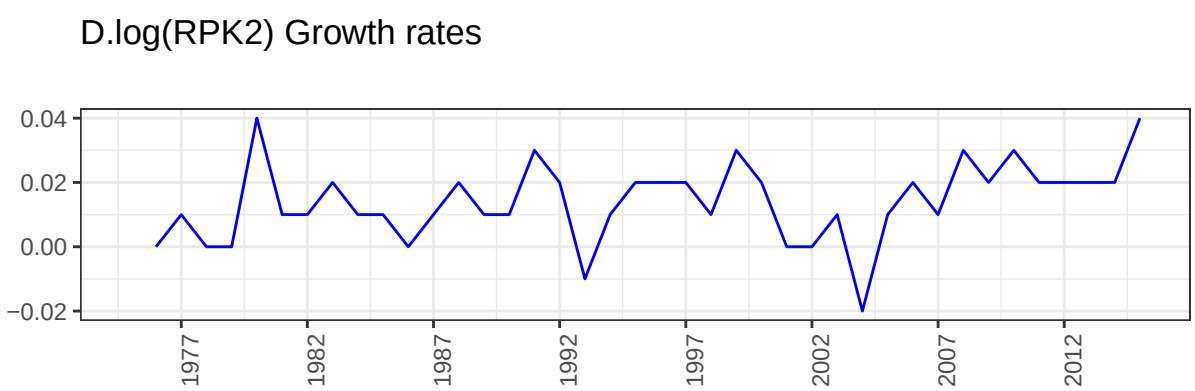
- If y_t is AR(p), then $PACF_k \approx 0$ for all $k > p$. The 95% confidence bounds around autocorrelations and partial autocorrelations are marked by plus and minus two divided by the square root of the sample size n. - 5% critical value: not significant if $-\frac{2}{\sqrt{n}} < (P)ACF < \frac{2}{\sqrt{n}}$.

Back to airlines example

Let us return now to the airline revenue passenger kilometers data. The data show a trend. And the growth rates do not.

```
grid.arrange(plot_b, plot_c, nrow = 2)
```

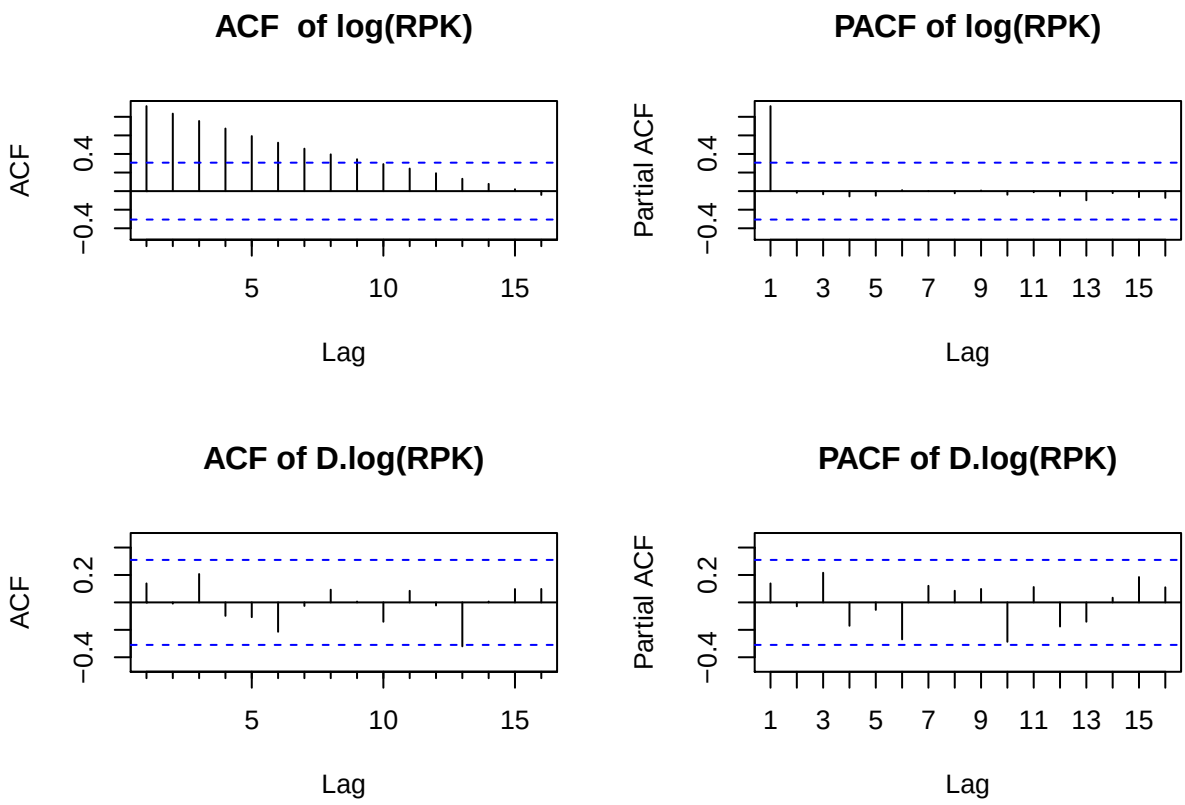




- log(RPK) is not stationary.
- first difference of log(RPK) (yearly growth rate) is stationary.

First we create the time series object in R And we use use the library forecast to plot the Acf and Pacf

```
# We create the time series object
ts1 <- ts(revenue[,-1], frequency = 1,start = 1975)
par(mfrow=c(2,2))
Acf(ts1[,4],main = "ACF  of log(RPK)")
Pacf(ts1[,4],main = "PACF of log(RPK)")
Acf(ts1[,6],main = "ACF of D.log(RPK)")
Pacf(ts1[,6],main = "PACF of D.log(RPK)")
```



The autocorrelations of the log series show a very slowly decaying pattern. And the partial autocorrelations are only large at lag 1. The values for the growth rates are not significant, as the number of observations is 39 and 2 divided by the square root of 39 is about 0.3.

Trends

Next, let us pay attention to the important issue of trends. Several trend models are available.

First, you have what is called a random walk. This is an autoregressive model, but with a slope parameter equal to 1.

$$y_t = y_{t-1} + \epsilon_t \text{ random walk, stochastic trend, no clear direction}$$

When the intercept alpha is unequal to 0, then we get trending data that looks like the airline's data, or the industrial production index considered before.

$$y_t = \alpha + y_{t-1} + \epsilon_t (\alpha \neq 0): \text{ stochastic trend}$$

When the model also contains a deterministic trend term, beta times t, then you get an explosive trend pattern.

$$y_t = \alpha + \beta t + y_{t-1} + \epsilon_t (\beta \neq 0): \text{ stochastic (explosive) trend}$$

If no lagged value of y is included, this gives a fully deterministic trend model.

$$y_t = \alpha + \beta t + \epsilon_t (\beta \neq 0): \text{ deterministic trend}$$

And if the lagged y term has a parameter smaller than 1, then this still results in a deterministic trend, but without random walk aspects.

$$y_t = \alpha + \beta t + \gamma y_{t-1} + \epsilon_t (\beta \neq 0, |\ \gamma\ | < 1): \text{ deterministic trend}$$

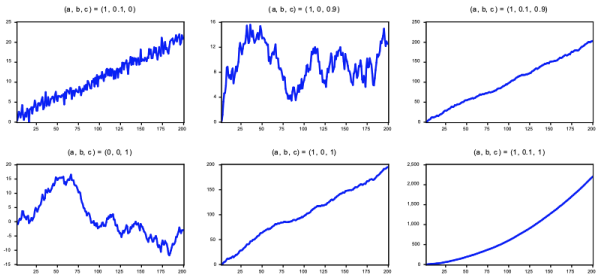
The main notion here, is that a **stochastic trend, that is where the autoregressive parameter is equal to 1**, can only be silenced by transforming the data, by taking their first difference. This is denoted by the symbol Delta.

$$y_t = \alpha + y_{t-1} + \epsilon_t \text{ then } \Delta y_t = y_t - y_{t-1} = \alpha + \epsilon_t$$

Example deterministic and stochastic trend

To give you some visual impression of how the parameters in a time series model determine how the data will look like, consider the following graphs of artificially generated data.

- DGP: $y_t = a + bt + cy_{t-1} + \varepsilon_t$
- Stochastic trend: $c = 1$ (bottom row)



Clearly the parameters matter a lot in how the data look like. This notion will be exploited in the actual analysis of real data. You look at the data and you see certain properties and from these properties, you can get ideas on the models that might be useful to fit and forecast this time series.

Cointegration

If two time series share the same stochastic trend, we say that they are cointegrated. In the next section, we will consider this in more detail.

- Sometimes: x_t and y_t each have stochastic trend, but $y_t - cx_t$ is stationary for some value of c.
- Cointegration (common stochastic trend)

Suppose that $z_t = z_{t-1} + \epsilon_{z,t}$ is unobserved, whereas $x_t = \alpha_1 + \gamma_1 z_t + \epsilon_{x,t}$ and $y_t = \alpha_2 + \gamma_2 z_t + \epsilon_{y,t}$ are observed, where $\epsilon_{z,t}, \epsilon_{x,t}, \epsilon_{y,t}$ are white noise processes. Show that x_t and y_t are cointegrated, and find the value of c for which $y_t - cx_t$ is stationary.

$$\gamma_1 y_t - \gamma_2 x_t = (\gamma_1 \alpha_2 - \gamma_2 \alpha_1) + (\gamma_1 \epsilon_{y,t} - \gamma_2 \epsilon_{x,t})$$

Where $\epsilon_t = \gamma_1 \epsilon_{y,t} - \gamma_2 \epsilon_{x,t}$ is white noise, therefore stationary.

$\gamma_1 y_t - \gamma_2 x_t = \gamma_1 (y_t - \frac{\gamma_2}{\gamma_1} x_t)$ So $c = \frac{\gamma_2}{\gamma_1}$.

We use the fact that y and x share the same stochastic trend z. A specific linear combination of y and x does not include that trend anymore.

Example of autocorrelation

If y_t is a stationary process with mean μ, then the k-th order autocovariance is defined as $\gamma_k = E[(y_t - \mu)(y_{t-k} - \mu)]$. In particular, the variance is $\gamma_0 = E(y_t - \mu)^2$. The k-th order autocorrelation is defined as $\rho_k = \frac{Cov(y_t, y_{t-k})}{Var(y_t)}$.

The AR(1) model is:

$$y_t = \alpha + \beta y_{t-1} + \epsilon_t$$

We know $E(\epsilon_t) = 0$ and ϵ_t is uncorrelated with all values of y_s for all $s < t$.

- a) Show that the mean of the AR(1) model is equal to $\mu = \frac{\alpha}{1-\beta}$

The expected value of y_t is equal to $E(y_t) = E(\alpha + \beta y_{t-1} + \epsilon_t) = \alpha + \beta E(y_{t-1}) + E(\epsilon_t) = \mu = \alpha + \beta\mu + 0$ And that means that $\beta \neq 1$

$$\mu = \frac{\alpha}{1-\beta}$$

- b) Define $z_t = y_t - \mu$. Show that $z_t = \beta z_{t-1} + \epsilon_t$ and that $Var(z_t) = \frac{\sigma^2}{(1-\beta^2)}$.

In part a) we see that $\alpha = \mu(1-\beta) = \mu - \beta\mu$. If we substitute this in the AR(1) equation we get:

$$y_t = \mu - \beta\mu + \beta y_{t-1} + \epsilon_t$$
$$y_t - \mu = \beta(y_{t-1} - \mu) + \epsilon_t$$

And as we can define $z_t = y_t - \mu$

$$z_t = \beta z_{t-1} + \epsilon_t$$

The expected value of $E(z_t) = E(y_t - \mu) = 0$ and the variance of $Var(z_t) = E(z_t - E(z_t))^2 = E(z_t^2)$ So if we use the definition of z_t we obtain:

$$Var(z_t) = E(z_t^2) = E((\beta z_{t-1} + \epsilon_t)^2) = \beta^2 E(z_{t-1}^2) + E(\epsilon_t^2) + 2\beta E(z_{t-1}\epsilon_t)$$

We use the fact that $E(z_{t-1}\epsilon_t) = 0$

$$Var(z_t) = \beta^2 Var(z_t^2) + \sigma^2 + 0$$

$$Var(z_t) = \frac{\sigma^2}{(1-\beta^2)}$$

That means that that $-1 < \beta < 1$

- c) Use the idea of part (b) to show that the autocorrelations of y_t are equal to $\rho_k = \beta^k$

Because y_t and z_t have the same autocovariances, we compute those of z_t, it is simplier because the mean of z_t is zero. The first order autocovariance γ_1

$$\gamma_1 = E(z_t z_{t-1}) = E(\beta z_{t-1} + \epsilon_t) z_{t-1} = \beta E(z_{t-1}^2) + E(\epsilon_t z_{t-1}) = \beta\gamma_0 + 0 = \beta\gamma_0$$

Hence the first order autocorrelation is:

$$\rho_1 = \frac{\gamma_1}{\gamma_0} = \beta$$

And the second order autovoariance γ_2 is equal:

$$\gamma_2 = E(z_t z_{t-2}) = E(\beta z_{t-1} + \epsilon_t) z_{t-2} = \beta E(z_{t-1} z_{t-2}) + E(\epsilon_t z_{t-2}) = \beta\gamma_1 + 0 = \beta(\beta\gamma_0) = \beta^2\gamma_0$$

Hence the second order autocorrelation is:

$$\rho_2 = \frac{\gamma_2}{\gamma_0} = \beta^2$$

And similarly, the k-th order autocovariance is equal to:

$$\gamma_k = E(z_t z_{t-k}) = E(\beta z_{t-1} + \epsilon_t) z_{t-k} = \beta E(z_{t-1} z_{t-k}) + E(\epsilon_t z_{t-k}) = \beta \gamma_{k-1} + 0 = \beta \gamma_{k-1}$$

Which is equal to:

$$\gamma_k = \beta^k \gamma_0$$

And the k-th order autocorrelation:

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \beta^k$$

- d) Argue that stationarity requires that $-1 < \beta < 1$

The correlations are always $-1 < \beta < 1$ so $\mid \beta \mid \leq 1$, furthermore $\beta = 1$ is exclueded in a) and $\beta = -1$ excluded in part b).

Specification and estimation

In this section, you will learn which steps to take to specify time series models, and to estimate parameters in such models. **Stationarity is crucial here**. And therefore, you should take care of any non-stationarity right at the start. Once a stationary series is obtained after proper transformation, you can use the autocorrelation (AC) and partial autocorrelation (PAC) functions to specify a first-guess model.

We start with the major motivation for using time series, that is, forecasting. Forecasts are based on a model that properly summarizes past information.

$$\text{Past values of time series model} \rightarrow \text{ Model } \rightarrow \text{ Forecast future values}$$

This past information can concern the own past of the dependent variable y, or possibly also the past of an explanatory factor, x. For notational convenience, we use PY and PX to denote this past information.

- In a univariate model, the forecast is based only on the past of the dependent variable y itself.
- And if we use also an explanatory factor x, then the forecast is a function of the past of both x and y.

Notation :

y_t : time series of interest $(t = 1, ..., n)$

x_t : time series possible explanatory factor (restrict to one)

Py_{t-1} : $[y_{t-1}, y_{t-2}, ..., y_1]$ past information on y at time t

Px_{t-1} : $[x_{t-1}, x_{t-2}, ..., x_1]$ past information on x at time t

Univariate time series forecast model : $\hat{y}_t = F(Py_{t-1})$

Forecast model with explanatory factor : $\hat{y}_t = F(Py_{t-1}, Px_{t-1})$

Of course, we want to use the past information in an optimal way, such that there is no predictive value anymore in the errors that we make. Indeed, we wish to arrive at a forecast error that is uncorrelated with the information in PY and PX.

- Aim: Optimal use of past information to get best forecasts.
- Wish: Forecast error $\epsilon_t = y_t - \hat{y}_t$ uncorrelated with past information.

Note that if the forecast error would be predictable, then some relevant information is still missing that could be used to improve the forecast.

Univariate time series model

Let us start with a univariate time series model. Here the forecast for y is a function of past observations of y only.

$$\text{Univariate time series forecast model: } : \hat{y}_t = F(Py_{t-1})$$

Now we first have to decide on the function F. And although many functions F can be chosen, a popular choice is the **linear function**. When the lagged information in PY is limited at lag p, the well-known autoregressive model of order p emerges. AR(p).

The true value of y is the forecast plus an error term that is equal to the forecast error. And together, this gives rise to the AR(p) model.

$$\hat{y}_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + +\beta_p y_{t-p}$$

$$y_t = \hat{y}_t + \epsilon_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + \beta_p y_{t-p} + \epsilon_t$$

Because ϵ_t is white noise that is, the future values of the epsilons cannot be predicted in a linear way. This can be seen in the following way:

Consider forecasts from an autoregression of order p, which says that there is useful information in the past until, and including, p observations ago $\hat{y}_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + ... + +\beta_p y_{t-p}$. And consider the forecast error epsilon, which should be uncorrelated with the past of y. $\epsilon_t = y_t - \hat{y}_t$ uncorrelated with uncorrelated with y_s for all s < t.

You can show that, in this situation, the epsilon process is white noise, that is, the future values of the epsilons cannot be predicted in a linear way.

- Without loss of generality, consider case s < t.
- $\epsilon_s = y_s - \alpha - \sum_{j=1}^{p} \beta_j y_{s-j}$ is a linear function of y_r, $r \leq s < t$
- e_t is uncorrelated with y_r for all $r < t$, so also uncorrealted with ϵ_s.

The crucial step here is that **epsilon is a linear function of current and past values of the dependent variable.**

Estimation of AR and ARMA

To estimate the parameters in an **AR model of order p**, we use the same ideas as in linear regression, where an optimal strategy is to **minimize the sum of squared errors**, which, of course, now are the forecast errors. So, you can use ordinary least squares.

$$\text{Forecast error: } : \epsilon_t = y_t - \hat{y}_t = y_t - \alpha - \sum_{j=1}^{p} \beta_j y_{t-j}$$

$$\textbf{To minimize via OLS: } : \sum_{t=p+1}^{n} \epsilon_t^2$$

As a moving average model **MA(q) includes also lagged forecast errors that are still unknown before estimation**, we have to resort to the method of **maximum likelihood in case of ARMA models**.

ADL(p,r) model

The usefulness of least squares extends to the case where the time series model also includes the past of an explanatory factor x. And again, here the popular choice is to use a linear forecast function.

$$\text{Forecast: } : \hat{y}_t = F(Py_{t-1}, Px_{t-1})$$

$$\text{find F such that } \epsilon_t = y_t - \hat{y}_t \text{ uncorrelated with } Py_{t-1}, Px_{t-1}$$

And this model that includes the lags of y and also the lags of x, when there are p lags of y, and r lags of x, we usually call this the autoregressive distributed lag model of order p and r, or shortly **ADL(p,r)**.

$$\text{ADL(p,r)} \ : \hat{y}_t = \alpha + \beta_1 y_{t-1} + ... + \beta_p y_{t-p} + \gamma_1 x_{t-1} + ... + \gamma_r x_{t-r}$$

Also for this ADL model, we can use the least squares method to estimate the parameters.

$$\textbf{To minimize via OLS: } : \sum_{t=m+1}^{n} \epsilon_t^2 \text{ where m=max(p,r)}$$

Granger causality

The ADL model is particularly useful to examine what is called Granger causality, named after the Nobel Laureate Sir Clive Granger. This idea of causality builds on the idea of forecastability. That is, when the past of one variable is helpful to predict the future of another, you might consider that as some form of causality.

What you do is to construct two ADL models. One for the variable y, and another for the variable x. Note that both models include the past of the dependent variable itself plus the past of the other variable.

$$y_t = \alpha + \sum_{j=1}^{p} \beta_j y_{t-j} + \sum_{j=1}^{r} \gamma_j x_{t-j} + \epsilon_t$$

$$x_t = \alpha^* + \sum_{j=1}^{p^*} \beta_j^* x_{t-j} + \sum_{j=1}^{r^*} \gamma_j^* y_{t-j} + \epsilon_t^*$$

-x_t helps to predict y_t if $\gamma_j \neq 0$ for some j -y_t helps to predict x_t if $\gamma_j^* \neq 0$ for some j

If some of the gamma parameters in the ADL model for y are different from 0, then the past of x helps to predict the future of y. And vice versa, in case the gamma star parameters are non-zero in the ADL model for x.

In case of such of non-zero parameters, you can say that **one variable is Granger causal to the other**. For example, we may find that x is Granger causal for y, but not the other way around. This indicates that the past of x can be helpful for predicting y. But for forecasting x, only its own past is relevant.

- x_t is Granger causal for y_t if it helps to predict y_t, whereas y_t does not help to predict x_t.

You can check the significance of, for example, the γ_j^* coefficients in the ADL model for x by means of the familiar F-test. And, conveniently, as these two models only include lag variables, you can still estimate the parameters using least squares for each of the two equations separately.

$$\text{Test } H_0 : \gamma_j^* = 0 \forall j = 1, ..., r^* \sim F - test$$

Consequences of non-stationarity

The first thing that needs to be done in modeling is to make sure that the time series you wish to analyze is stationary. The reason that we need stationarity is that this is required for proper statistical analysis. So, one should first somehow test if the variable of interest is stationary.

- Regression assumption A2 not satisfied: regressors y_{t-j} are random.
- Standard OLS t- and F-tests hold true in large enough samples provided all variables in equation are stationary.

So, one should first somehow test if the variable of interest is stationary. To do this, we can again make use of a time series model. For example, in the case of an autoregression of order 1, we know that when the parameter is equal to 1, then the time series is not stationary. And this suggests that we can test for stationarity by testing the value of this parameter.

$$\begin{aligned} &\text{Test for stationarity :} \\ \textbf{AR(1)}\ &y_t = \alpha + \beta y_{t-1} + \epsilon_t \text{ test } H_0 : \beta = 1 \text{ against} H_1 : -1 < \beta < 1 \end{aligned}$$

As we are familiar with statistical tests for parameters to be equal to 0, we usually rewrite the AR(1) model by subtracting the one period lagged y_{t-1} from both sides of the equation.

$$\begin{aligned} &\text{Rewrite AR(1) :} \\ &\Delta y_t = y_t - y_{t-1} = \alpha + (\beta - 1) y_{t-1} + \epsilon_t = \alpha + \rho y_{t-1} + \epsilon_t \\ &\text{Where } \rho = (\beta - 1) \end{aligned}$$

Unit root test

Now, the ρ parameter can be tested to be equal to 0 using a t-test. As we are interested in the parameter rho being 0, or smaller than 0, we reject non-stationarity when the t- ratio is more negative than minus 2.9. Note, that this is not the usual value of $t_{0.95} = -1.65$. And this is due to the fact that **under the null hypothesis, y is non-stationary, which gives rise to a different statistical theory.**

$$\begin{aligned} &\text{Test for stationarity :} \\ &\Delta y_t = \alpha + \rho y_{t-1} + \epsilon_t \text{ test } H_0 : \rho = 0 \text{ against } H_1 : \rho < 0 \\ &\text{Reject } H_0 : \text{non-stationarity if } t_{\hat{\rho}} < -2.9 \end{aligned}$$

For an AR(2) model we follow the same process, it is convenient to write the AR(2) model as a mixture of variables in first differences and in levels.

$$\begin{aligned} &\text{Rewrite AR(2) : } y_t = \alpha + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \epsilon_t \\ &\Delta y_t = y_t - y_{t-1} = \alpha + (\beta_1 - 1) y_{t-1} + \beta_2 y_{t-2} + \epsilon_t \\ &\text{Sum and substract } \beta_2 y_{t-1} \\ &\Delta y_t = \alpha + (\beta_1 + \beta_2 - 1) y_{t-1} - \beta_2 y_{t-1} + \beta_2 y_{t-2} + \epsilon_t \\ &\Delta y_t = \alpha + (\beta_1 + \beta_2 - 1) y_{t-1} - \beta_2 (y_{t-1} - y_{t-2}) + \epsilon_t \\ &\Delta y_t = \alpha + (\beta_1 + \beta_2 - 1) y_{t-1} - \beta_2 \Delta y_{t-1} + \epsilon_t \\ &\text{we get :} \\ &\Delta y_t = \delta + \rho y_{t-1} + \gamma \Delta y_{t-1} + \epsilon_t \\ &\text{Where } \delta = \alpha, \rho = (\beta_1 + \beta_2 - 1), \gamma = -\beta_2 \end{aligned}$$

Augmented Dickey Fuller

This rewriting of the AR(p) models provides the basis of the so-called Dickey-Fuller test. The test equation can either include a deterministic trend, or not, and the choice is usually based on the visual impression of the data.

The inclusion or exclusion of the deterministic trend term, βt, matters for the relevant 5% critical value. When the trend term is not included, the critical value for the t-test on rho is $t_{\hat{\rho}} < -2.9$, as we saw before. But when the trend is included, it becomes $t_{\hat{\rho}} < -3.5$.

If data NO clear trend direction :

$$\Delta y_t = \alpha + \rho y_{t-1} + \gamma_1 \Delta y_{t-1} + ... + \gamma_L \Delta y_{t-L} + \epsilon_t$$

Test without deterministic trend

Reject H_0 : non-stationarity if $t_{\hat{\rho}} < -2.9$

If data has clear trend direction :

$$\Delta y_t = \alpha + \beta t + \rho y_{t-1} + \gamma_1 \Delta y_{t-1} + ... + \gamma_L \Delta y_{t-L} + \epsilon_t$$

Test with deterministic trend

Reject H_0 : non-stationarity if $t_{\hat{\rho}} < -3.5$

In practice, we decide on the number of lags in the autoregression by testing for correlation in the residuals, or by using a model selection criteria.

- Choice lag L: serial correlation check, or AIC/BIC. See (Partial) Autocorrelation Function.

When the autoregression has more than one lag, the Dickey-Fuller test is usually called the Augmented Dickey-Fuller test, abbreviated as ADF.

Summary

So, now, how should you proceed to specify a time series model?

1. First you perform an ADF test. And when you can reject a unit root, or non-stationarity, this means that y_t is stationary, and you can model the series y_t without further transformation. But when it is not rejected, you should take the first difference, and continue with Δy_t.
2. Next, you can use OLS to estimate the parameters in an autoregression AR(p), or when you wish to consider an autoregressive distributed lag model ADL(p,r), you perform unit root tests for both series y and x, and proceed with levels or with first differences.

There is, however, one exceptional, and practically relevant case, namely, when y and x are not stationary, **but a linear combination of the two variables is.** When y_t and x_t are cointegrated.

Cointegration

Two variables are called cointegrated when a linear combination is stationary. This can only occur when y_t and x_t have the same stochastic trend.

y_t, x_t are are cointegrated if both series are non-stationary, but a linear combination is stationary : $y_t - cx_t$

You might, then, interpret this linear combination $y_t = cx_t$ as the **long-run equilibrium**, which is an attractive concept in economics.

Test for cointegration

A simple test for cointegration is the Engle-Granger test. This test amounts to regressing y_t on an intercept and x_t to estimate the long-run equilibrium relationship between these variables.

Engle-Granger test :

Step 1 : OLS in $y_t = \alpha + \beta x_t + \epsilon_t \rightarrow b$ and residuals e_t

Step 2 : Cointegrated if ADF test on e_t rejects non-stationarity

$$\Delta e_t = \alpha + \rho e_{t-1} + \gamma_1 \Delta e_{t-1} + ... + \gamma_L \Delta e_{t-L} + w_t$$

The residuals from this regression can then be interpreted as deviations from the equilibrium, and if the equilibrium relation actually exists, that is if y_t and x_t actually are cointegrated, the residuals should be stationary. And this can be examined, again, using the ADF test as before.

Because the test is now applied to residuals instead of an actually observed time series, the distribution of the ADF test is different. The 5% critical value for the relevant t-test is now minus 3.4, or even minus 3.8 if the trend term is included.

Engle-Granger test :

$$\Delta e_t = \alpha + (\beta t) + \rho e_{t-1} + \gamma_1 \Delta e_{t-1} + ... + \gamma_L \Delta e_{t-L} + w_t$$

Test (with) or without deterministic trend

Reject H_0 : non-stationarity = cointegrated if $t_{\hat{\rho}} < -3.4$ or (-3.8)

In case of cointegration, the ADL model can be written in the so-called error correction format, which includes lagged difference variables and the stationary linear combination between y_t and x_t. The term error correction follows from the notion that deviations from the long-run equilibrium get corrected by the β_1 parameter in forecasting the changes of y_t.

ECM : if x_t, y_t cointegrated, estimate :

$$\Delta y_t = \alpha + \beta_1 (y_{t-1} - b x_{t-1}) + \beta_2 \Delta y_{t-1} + \beta_3 \Delta x_{t-1} + \epsilon_t$$

Or more lags for $\Delta y_t, \Delta x_t$

Example: Partial adjustment and Adaptive Expectations

Several types of economic behavior lead to relations with time lags because it takes some time before adjustments to changed conditions take place. As an illustration,consider a company producing consumer goods. Let y_t denote the production volume in month t, and let x_t be the total demand volume in that month.

Suppose that the optimal production volume y_t^* is equal to $y_t^* = \gamma + \delta x_t$.

As an example, if the company has a market share of 25%, then $\delta = 0.25$ could make sense for this company. If demand changes, then it may take some time for the company to increase production, because it will need to arrange for extra capital and labor.

In this scenario, we could use the partial adjustment (PA) model, the model postulates that

$$y_t = y_{t-1} + \lambda(y^*_{t-1} - y_{t-1}) + \epsilon_t$$

PA model :

$$y_t = y_{t-1} + \lambda(y^*_{t-1} - y_{t-1}) + \epsilon_t$$
$$\text{where } 0 \leq \lambda \leq 1$$

If the company is pro-active, it can decide to **base its production on the expected demand volume** x^*_t and to produce the corresponding volume $y_t = \gamma + \delta x^*_t$.

The adaptive expectations (AE) model postulates that the expectations are partly adjusted to the previously observed demand volume by means of

AE model :

$$x^*_t = x^*_{t-1} + \lambda(x_{t-1} - x^*_{t-1}) + \epsilon_t$$
$$\text{where } 0 \leq \lambda \leq 1$$

It is assumed that the ϵ_t in the PA and AE models is white noise.

- a) Write the partial adjustment model in terms of only the observed variables y and x, by eliminating y^*_t.What is the type of the resulting model?

First we substitute $y^*_t = \gamma + \delta x_t$ in the PA model.

$$y_t = y_{t-1} + \lambda(y^*_{t-1} - y_{t-1}) + \epsilon_t = y_{t-1} + \lambda(\gamma + \delta x_{t-1} - y_{t-1}) + \epsilon_t$$

Which we can rewrite as:

$$y_t = \lambda\gamma + (1-\lambda)y_{t-1} + \lambda\delta x_{t-1} + \epsilon_t$$

This is an autoregressive distributed lag model (ADL) with AR lag (p=1) and DL lag (r=1).

- b) Write the adaptive expectations model in terms of only the observed variables y and x, by eliminating x^*_t.What is the type of the resulting model?

We first rewrite the AD model as

$$x^*_t = x^*_{t-1} + \lambda(x_{t-1} - x^*_{t-1}) + \epsilon_t = (1-\lambda)x^*_{t-1} + \lambda x_{t-1} + \epsilon_t$$
$$x^*_t - (1-\lambda)x^*_{t-1} = \lambda x_{t-1} + \epsilon_t$$

That means that we should replace the left term in order to eliminate the expectation variable x^*_t. We know $y_t = \gamma + \delta x^*_t$ so we could rewrite $\delta x^*_t = y_t - \gamma$.

We get that $\delta(x^*_t - (1-\lambda)x^*_{t-1}) = y_t - \gamma - (1-\lambda)(y_{t-1} - \gamma)$. And from the previous equation we get:

$$\delta(x^*_t - (1-\lambda)x^*_{t-1}) = \delta\lambda x_{t-1} + \delta\epsilon_t$$

Hence:

$$y_t - \gamma - (1-\lambda)(y_{t-1} - \gamma) = \delta\lambda x_{t-1} + \delta\epsilon_t$$

Equivalently:

$$y_t = \gamma - (1-\lambda)\gamma + (1-\lambda)y_{t-1} + \delta\lambda x_{t-1} + \delta\epsilon_t$$

$$y_t = \gamma\lambda + (1-\lambda)y_{t-1} + \delta\lambda x_{t-1} + \delta\epsilon_t$$

So again we can see that we get an an autoregressive distributed lag model (ADL) with AR lag (p=1) and DL lag (r=1).

- c) What is the condition for stability of the two models in parts a) and b)? Provide an economic interpretation of this condition.

The condition is that $-1 < (1-\lambda) < 1$, that is $0 < \lambda < 2$. We already assumed that $0 < \lambda < 1$, so the condition for stability is that $\lambda \neq 0$.

If $\lambda = 0$ we can see that production is not adjusted in any systematic way in the PA model. Furthermore, we can see that expectations are not adjusted with realized demand in the AE model.

- d) Consider the AE model based on the last two observed sales volumes, where $x_t^* = x_{t-1}^* + \lambda_1(x_{t-1} - x_{t-1}^*) + \lambda_2(x_{t-2} - x_{t-2}^*) + \epsilon_t$. Write this model in terms of only the observed variables y and x, by eliminating x_t^*.What is the type of the resulting model?

Similiar to part b) we first rewrite the AE model:

$$\begin{aligned} x_t^* &= x_{t-1}^* + \lambda_1(x_{t-1} - x_{t-1}^*) + \lambda_2(x_{t-2} - x_{t-2}^*) + \epsilon_t \\ x_t^* &= (1-\lambda_1)x_{t-1}^* - \lambda_2 x_{t-2}^* + \lambda_1 x_{t-1} + \lambda_2 x_{t-2} + \epsilon_t \\ x_t^* - (1-\lambda_1)x_{t-1}^* + \lambda_2 x_{t-2}^* &= +\lambda_1 x_{t-1} + \lambda_2 x_{t-2} + \epsilon_t \end{aligned}$$

from $y_t = \gamma + \delta x_t^* \rightarrow \delta x_t^* = y_t - \gamma$ we get the following two substitutions:

$$-(1-\lambda_1)\delta x_{t-1}^* = -(1-\lambda_1)(y_{t-1} - \gamma)$$

and

$$\lambda_2 \delta x_{t-2}^* = \lambda_2 \delta (y_{t-2} - \gamma)$$

Using these two terms for the above left side we get:

$$\delta(x_t^* - (1-\lambda_1)x_{t-1}^* + \lambda_2 x_{t-2}^*) = y_t - \gamma - (1-\lambda_1)(y_{t-1} - \gamma) + \lambda_2(y_{t-2} - \gamma)$$

We can simplify the left side as follows:

$$\delta(\lambda_1 x_{t-1} + \lambda_2 x_{t-2} + \epsilon_t) = y_t - \gamma - (1-\lambda_1)(y_{t-1} - \gamma) + \lambda_2(y_{t-2} - \gamma)$$

$$\delta\lambda_1 x_{t-1} + \delta\lambda_2 x_{t-2} + \delta\epsilon_t = y_t - \gamma - (1-\lambda_1)y_{t-1} + (1-\lambda_1)\gamma + \lambda_2 y_{t-2} - \lambda_2\gamma$$

We can rewrite so that:

$$\begin{aligned} y_t &= \gamma - (1-\lambda_1)\gamma + \lambda_2\gamma + (1-\lambda_1)y_{t-1} - \lambda_2 y_{t-2} + \delta\lambda_1 x_{t-1} + \delta\lambda_2 x_{t-2} + \delta\epsilon_t \\ &= (\lambda_1 + \lambda_2)\gamma + (1-\lambda_1)y_{t-1} - \lambda_2 y_{t-2} + \delta\lambda_1 x_{t-1} + \delta\lambda_2 x_{t-2} + \delta\epsilon_t \end{aligned}$$

This is an autoregressive distributed lag model (ADL) with AR lag (p=2) and DL lag (r=2). ADL(2,2).

Evaluation of time series

Now we will see how the empirical modeling cycle works in practice.

After checking for stationarity and taking appropriate actions if the series is found to be non-stationary, you start with a first guess model. You estimate the parameters and evaluate the obtained model. The tools for evaluation also provide hints how to modify the initial model.

This modeling cycle usually ends when all diagnostic measures are acceptable, so that you are ready, for example, to make forecasts. Any application of time series modeling should start with examining whether the time series is stationary. If not, then the series should be transformed, for example, by taking the difference until stationarity is achieved.

1. Check for stationarity

- Take difference of time series until stationarity.
- The usual test for stationarity is the Augmented Dickey-Fuller test, which is based on the t-test for rho in the autoregressive model shown in this equation. Take care that the relevant critical values do not correspond to the usual ones of the standard normal distribution.

$$\text{ADF test}: \quad \Delta y_t = \alpha + \beta t + \rho y_{t-1} + \gamma_1 \Delta y_{t-1} + ... + \gamma_L \Delta y_{t-L} + \epsilon_t$$

$$\text{Reject } H_0 : \text{non-stationarity if} \quad t_{\hat{\rho}} < -2.9 \text{ if } \beta = 0, t_{\hat{\rho}} < -3.5 \text{ if } \beta \neq 0$$

Once stationarity is obtained, you can apply least squares to obtain estimates of the coefficients of an autoregression of order p AR(p), or of an autoregressive distributed lag model ADL(p,r), in case you want to involve another variable x.

$$\text{OLS in AR(p) [with trend]: } y_t = \alpha + [\gamma_t t] + \sum_{j=1}^{p} \beta_j y_{t-j} + \epsilon_j$$

$$\text{OLS in ADL(p,r) [with trend]: } y_t = \alpha + [\gamma_t t] + \sum_{j=1}^{p} \beta_j y_{t-j} + \sum_{j=1}^{r} \beta_j x_{t-j} + \epsilon_j$$

Just as in multiple regression, you can apply t-tests and F-tests in the usual way.

2. Check for cointegration

When analyzing two time series x and y jointly, it can occur that they both are non-stationary, but they share the same stochastic trend, meaning that they are co-integrated. As discussed in the previous section, a simple test for co-integration is the Engle-Granger two-step method.

This applies an augmented Dickey-Fuller test to the residuals from the regression of y on x. And if this test indicates that the residuals are stationary, we conclude that x and y are co-integrated.

$$\text{Engle-Granger test}:$$
$$\text{Step 1}: \quad \text{OLS in } y_t = \alpha + \beta x_t + \epsilon_t \rightarrow b \text{ and residuals } e_t$$
$$\text{Step 2}: \quad \text{Cointegrated if ADF test on } e_t \text{ rejects non-stationarity}$$
$$\Delta e_t = \alpha + \beta t + \rho e_{t-1} + \gamma_1 \Delta e_{t-1} + ... + \gamma_L \Delta e_{t-L} + w_t$$
$$\text{Reject } H_0 : \text{non-stationarity} : \text{cointegrated if}$$
$$\text{Critical Value } t_{\hat{\rho}} < -3.4 \text{ if } \beta = 0, t_{\hat{\rho}} < -3.8 \text{ if } \beta \neq 0$$

And if we find co-integration between x and y, you can estimate the parameters in an error correction model (ECM).

ECM : if x_t, y_t cointegrated, estimate :

$$\Delta y_t = \alpha + \beta t + \gamma_0(y_{t-1} - bx_{t-1}) + \sum_{j=1}^{p} \gamma_{y,j}\Delta y_{t-j} + \sum_{j=1}^{r} \gamma_{x,j}\Delta x_{t-j} + \epsilon_t$$

Or with $\beta = 0$

This model has a nice economic interpretation. It describes how deviations from the equilibrium between x_t and y_t, or errors, at time $t-1$ are corrected in the next period. And also here, you can use the t- and F-tests as usual, as all variables in the model are stationary.

3. Diagnostic tests

Once you have initiated the modeling cycle with tests of stationarity, you can propose a first-guess model which usually involves a decision on the number of lags. As an autoregression and an autoregressive distributive lag model basically are multiple regression models, you can use the familiar diagnostic techniques that were discussed in Model Specification.

These include:

- Choice of lag lengths: BIC
- Stability check: Chow tests
- Normal residuals: Jarque-Bera (Notice critical value: 6.0)
- Out-of-sample forecasting (next section)

As you wish to end up with residuals that have no predictive content, it is crucial to examine if the relevant past information is properly captured by the model. You can do so by testing if the residuals are uncorrelated (are white noise).

Two tests for white noise residuals are commonly used:

- A simple test uses the autocorrelations of the residuals. ACF rule-of-thumb: significant if $\mid ACD \mid > 2/\sqrt{n}$ for 95% confidence level.

Let y_t be white noise with variance σ^2. The OLS estimator b in $y_t = \alpha + \beta y_{t-1} + \epsilon_t$ gives the first-order autocorrelation of y_t:

$$b = \frac{\sum_{t=2}^{n}(y_t - \bar{y})((y_{t-1} - \bar{y}))}{\sum_{t=2}^{n}(y_t - \bar{y})^2}$$

$$Var(b) = \frac{\sigma^2}{\sum_{t=2}^{n}(y_t - \bar{y})^2} \text{ where } \sum_{t=2}^{n}(y_t - \bar{y})^2 = \frac{(n-1)\sum_{t=2}^{n}(y_t - \bar{y})^2}{(n-1)} \approx (n-1)\sigma^2 \text{ if n large}$$

$$Var(b) \approx \frac{\sigma^2}{(n-1)\sigma^2} = \frac{1}{(n-1)} = \frac{1}{n}$$

So if n is large, then $b \approx 0$ and $SE(b) \approx \frac{1}{n}$

$$b - 2SE(b) < \beta < b + 2SE(b) = -\frac{2}{n} < \beta < \frac{2}{n}$$

- Another test is the Breusch-Godfrey test.

A catch-all test for autocorrelations is the Breusch-Godfrey test. This test amounts to an auxiliary regression of the residuals on past residuals and on the model variables. And then you take n times the R^2 of that regression. Under the null hypothesis of no autocorrelation, this test has a chi-square distribution with R degrees of freedom.

1. Estimate model and get residuals e_t
2. Regress e_t on all variables of model and r lags of e_t.
3. $BG = nR^2$ of Step 2, and $BG \approx \chi^2(r)$ if e_t is white noise (H_0)

Note that the structure of testing by means of the R squared of a second-step regression closely resembles the tests of Hausman and Sargan test.

So for example, suppose your first guess model includes one-period lags of y_t and x_t to predict y_t.

$$y_t = \alpha + \beta y_{t-1} + \gamma x_{t-1} + \epsilon_t$$

1. OLS residuals $e_t = y_t - a - by_{t-1} + cx_{t-1}$. 2. OLS in $e_t = \alpha + \beta y_{t-1} + \gamma x_{t-1} + \delta_1 e_{t-1} + \delta_2 e_{t-2} + \epsilon_t$. 3. As you have two lags, the chi-squared distribution with two degrees of freedom delivers the proper critical value. $BG = nR^2 \approx \chi^2(2)$ if e_t is white noise.

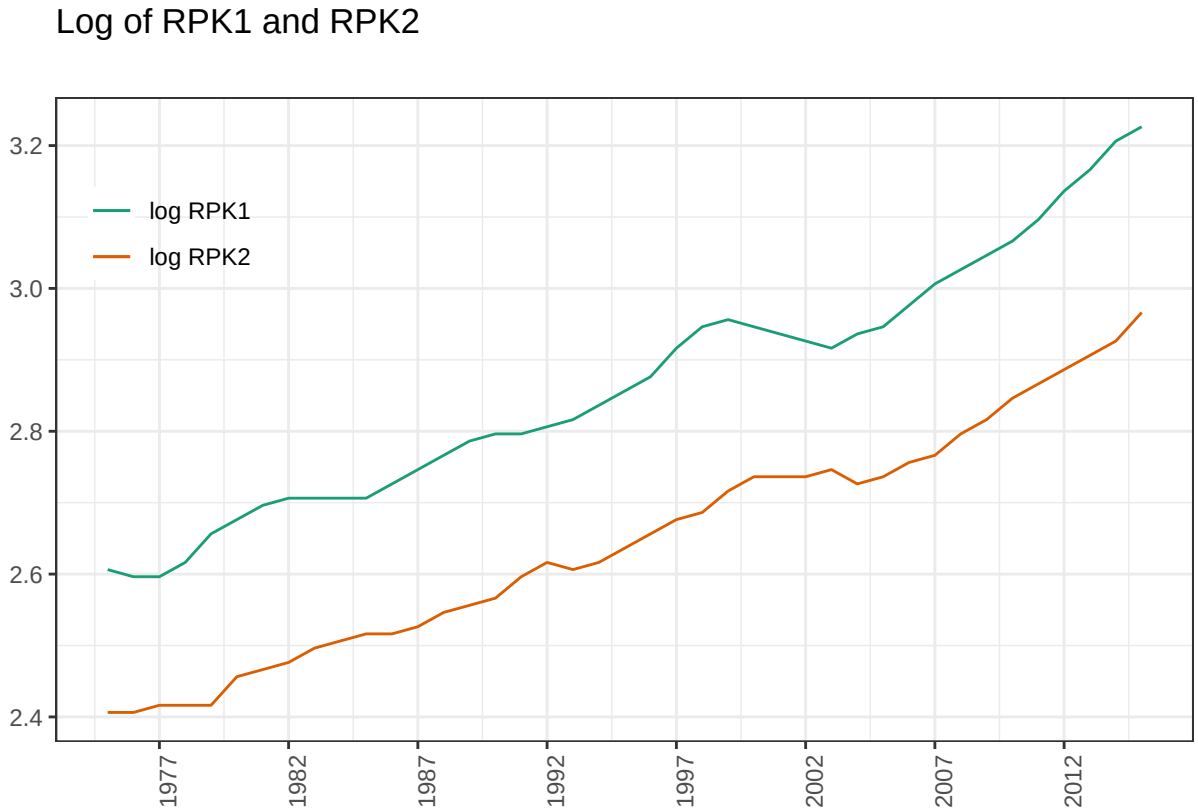
That is, in case $nR^2 > 6$ we reject the null of no autocorrelation, the past still contains some valuable information that should be exploited by adjusting your model. For example, by adding more lags of y and x.

Example of Revenue Airline

```
revenue <- read_csv(
  "https://raw.githubusercontent.com/diego-eco/diego-eco.github.io/master/downloads/dataset61.csv")
revenue$YEAR<- as.Date(paste0(revenue$YEAR, '-01-01'))
```

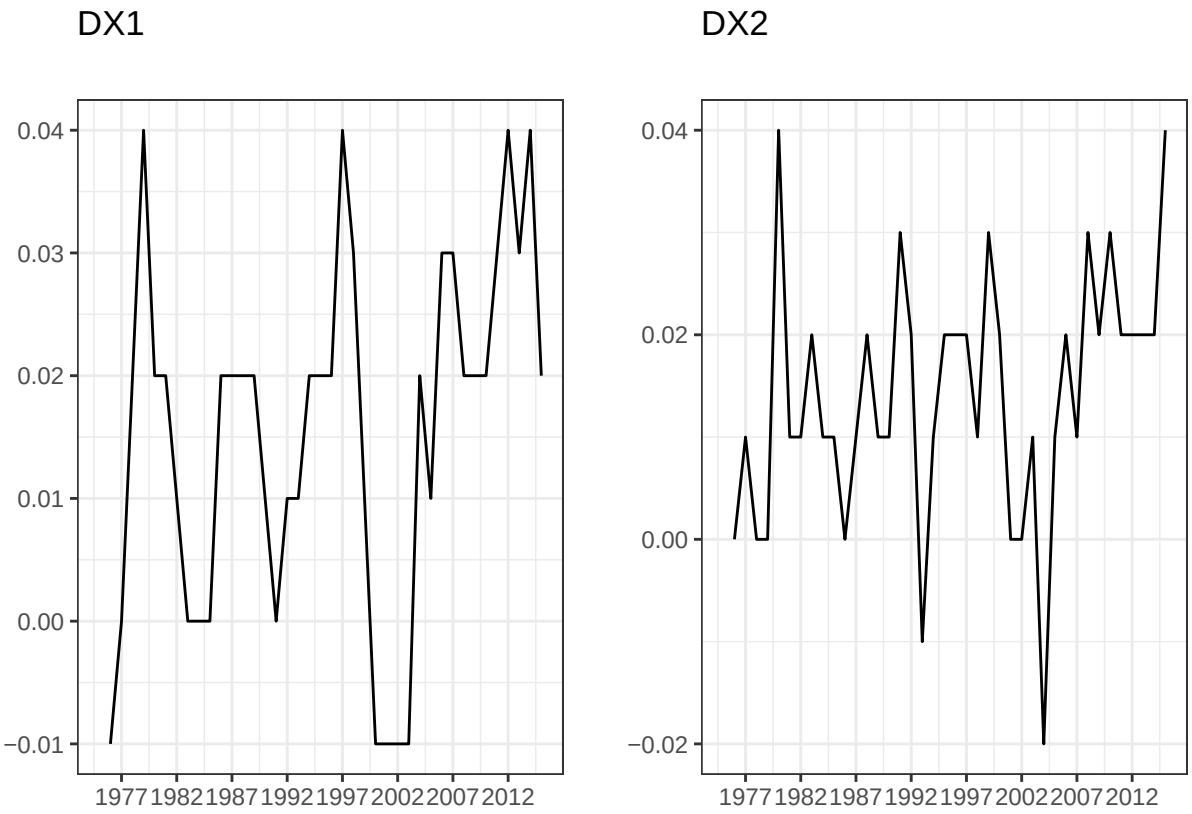
Consider again the two series on revenue passenger kilometers for two airlines. Clearly the two times series show a trending pattern. Note that we have transformed the original series here by taking natural logs, which makes their trends approximately linear.

```
plot_a <- ggplot(data=revenue, aes(x=YEAR)) +
  geom_line(aes(y=X2,col="log RPK2")) +
  geom_line(aes(y=X1,col="log RPK1")) +
  labs(x = "", y = "", title = "Log of RPK1 and RPK2",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.1, .8),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")
plot_a
```



On the other hand, the first differences, which here can be interpreted as annual growth rates, seem stationary because they fluctuate around a constant mean.

```
plot_b <- ggplot(data=revenue, aes(x=YEAR)) +
  geom_line(aes(y=DX1)) +
  labs(x = "", y = "", title = "DX1",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw()
plot_c <- ggplot(data=revenue, aes(x=YEAR)) +
  geom_line(aes(y=DX2)) +
  labs(x = "", y = "", title = "DX2",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw()
grid.arrange(plot_b,plot_c, nrow = 1)
```



To save notation, we use y_t for the logs of each of the series.

1. Check for stationarity

One of the main R packages for time series modeling is tseries

The R fucntion adf.test() performs the Augmented Dickey-Fuller test for the null hypothesis of a unit root of a univarate time series x (equivalently, x is a non-stationary time series).

```
# Set k=1 for the Dickey Fuller test with first lag.
adf.test(revenue$X1,k=1) # contains a unit-root
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  revenue$X1
## Dickey-Fuller = -2.7644, Lag order = 1, p-value = 0.2729
## alternative hypothesis: stationary
```

```
adf.test(revenue$X2,k=1) # contains a unit-root
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  revenue$X2
## Dickey-Fuller = -1.2074, Lag order = 1, p-value = 0.8853
## alternative hypothesis: stationary
```

The ur.df() Augmented Dickey-Fuller test in the urca package gives us a bit more information on and control over the test. he ur.df() function allows us to specify whether to test stationarity around a zero-mean with no trend, around a non-zero mean with no trend, or around a trend with an intercept. This can be useful when we know that our data have no trend, for example if you have removed the trend already. ur.df() allows

us to specify the lags or select them using model selection.

In the ADF test equation, we include a constant (α), a deterministic trend term βt, and a single lag of $\Delta X1_t$

```
# Set k=1 for the Dickey Fuller test with first lag.
test <- urca::ur.df(na.omit(revenue$X1), type = "trend", lags = 1)
summary(test)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.019123 -0.006006 -0.001421  0.004739  0.022106
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4704469  0.1682887   2.795  0.00836 **
## z.lag.1     -0.1814847  0.0656510  -2.764  0.00904 **
## tt           0.0025452  0.0009043   2.815  0.00796 **
## z.diff.lag   0.7606219  0.1342284   5.667 2.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01036 on 35 degrees of freedom
## Multiple R-squared:  0.5137, Adjusted R-squared:  0.472
## F-statistic: 12.32 on 3 and 35 DF,  p-value: 1.174e-05
##
##
## Value of test-statistic is: -2.7644 5.2777 3.962
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

```
# The intercept and tt estimates indicate where there is a non-zero level (intercept) or linear trend (
```

$$\Delta X_{1,t} = \alpha + \beta t + \rho X_{1,t-1} + \gamma \Delta X_{1,t-1} + \epsilon_t$$

Here we can see that $\hat{\rho} = -0.181$ with SE $= 0.065$ and the value of ADF test-statistic is: -2.7644.

We do the same for $\Delta X_{2,t} = \alpha + \beta t + \rho X_{12t-1} + \gamma \Delta X_{2,t-1} + \epsilon_t$

```
# Set k=1 for the Dickey Fuller test with first lag.
test <- urca::ur.df(na.omit(revenue$X2), type = "trend", lags = 1)
summary(test)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.035897 -0.004838 -0.000094  0.004530  0.029455
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.363125   0.294549   1.233    0.226
## z.lag.1     -0.150500   0.124649  -1.207    0.235
## tt           0.002214   0.001627   1.361    0.182
## z.diff.lag   0.187349   0.185459   1.010    0.319
##
## Residual standard error: 0.01206 on 35 degrees of freedom
## Multiple R-squared:  0.1154, Adjusted R-squared:  0.03958
## F-statistic: 1.522 on 3 and 35 DF,  p-value: 0.2259
##
##
## Value of test-statistic is: -1.2074 7.0053 1.8301
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

```
# The intercept and tt estimates indicate where there is a non-zero level (intercept) or linear trend (
```

Here we can see that $\hat{\rho} = -0.150$ with SE $= 0.124$ and the value of ADF test-statistic is: -1.207.

In summary:

- Let y_t denote log(RPK), for $X1_t$ and $X2_t$ with a trend $\beta t \neq 0$ noticed by visual inspection:
- ADF: $\Delta y_t = \alpha + \beta t + \rho y_{t-1} + \gamma \Delta y_{t-1} + \epsilon_t$
- t-value of $\hat{\rho}$: $t = -2.7644$ for $X1_t$ and $t = -1.025$ for $X2_t$

  As both ADF t values are larger than the critical value -3.5 X1 and X2 are not stationary.

Now let's test the first difference DX1 and DX2

```
# Set k=1 for the Augmented Dickey Fuller test with 1 lag.
# Notice that adf.test does not take NA values
adf.test(na.omit(revenue$DX1),k=1) # stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(revenue$DX1)
```

```
## Dickey-Fuller = -3.3213, Lag order = 1, p-value = 0.08298
## alternative hypothesis: stationary
```

```
adf.test(na.omit(revenue$DX2),k=1) # stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(revenue$DX2)
## Dickey-Fuller = -4.032, Lag order = 1, p-value = 0.01854
## alternative hypothesis: stationary
```

When we analyze the growth rates, there is no reason to add any deterministic trend in the test regression.

- Let y_t denote either $\Delta X1$ or $\Delta X2$ with NO trend $\beta t = 0$ noticed by visual inspection.
- ADF: $\Delta y_t = \alpha + \rho y_{t-1} + \gamma \Delta y_{t-1} + \epsilon_t$
- t-value of $\hat{\rho}$: $t = -3.301$ for $X1_t$ and $t = -3.705$ for $X2_t$

And now, the t-ratios in the Dickey-Fuller test become $t_{\Delta X1} = - - 3.301$ and $t_{\Delta X2} = -3.705$, respectively.

As both t values are smaller than the critical value -2.9, DX1 and DX2 are both stationary.

2. Check for causality

Next, we investigate if the two series are mutually linked. For that purpose we estimate two autoregressive distributed lag models, one for the growth rates of X1 (DX1) and the other for those of X2 (DX2).

We include two lags of each, and in the table you see the estimated coefficients, their t-statistics and the associated p-values.

We estimate the two following regressions:

$$\Delta X_{1,t} = \alpha + \beta_1 \Delta X_{1,t-1} + \beta_2 \Delta X_{1,t-2} + \gamma_1 \Delta X_{2,t-1} + \gamma_2 \Delta X_{1,t-2} + \epsilon_t \Delta X_{2,t} = \alpha + \beta_1 \Delta X_{1,t-1} + \beta_2 \Delta X_{1,t-2} + \gamma_1 \Delta X_{2,t-1} + \gamma_2 \Delta X_{1,t-2}$$

```
lm1 <- lm(DX1 ~ lag(DX1,1) + lag(DX1,2) + lag(DX2,1) + lag(DX2,2), data=revenue)
lm2 <- lm(DX2 ~ lag(DX1,1) + lag(DX1,2) + lag(DX2,1) + lag(DX2,2), data=revenue)
summ(lm1,digits = 2)
```

Observations	38 (3 missing obs. deleted)
Dependent variable	DX1
Type	OLS linear regression

F(4,33)	7.51
R^2	0.48
Adj. R^2	0.41

	Est.	S.E.	t val.	p
(Intercept)	0.01	0.00	1.84	0.07
lag(DX1, 1)	0.87	0.18	4.96	0.00
lag(DX1, 2)	-0.42	0.21	-2.02	0.05
lag(DX2, 1)	0.35	0.20	1.74	0.09
lag(DX2, 2)	-0.19	0.15	-1.27	0.21

Standard errors: OLS

```
summ(lm2,digits = 2)
```

Observations	38 (3 missing obs. deleted)
Dependent variable	DX2
Type	OLS linear regression

F(4,33)	10.89
R^2	0.57
Adj. R^2	0.52

	Est.	S.E.	t val.	p
(Intercept)	0.01	0.00	2.86	0.01
lag(DX1, 1)	0.18	0.14	1.29	0.21
lag(DX1, 2)	0.61	0.17	3.68	0.00
lag(DX2, 1)	-0.29	0.16	-1.81	0.08
lag(DX2, 2)	-0.13	0.12	-1.05	0.30

Standard errors: OLS

For $\Delta X_{1,t} = \alpha + \beta_1 \Delta X_{1,t-1} + \beta_2 \Delta X_{1,t-2} + \gamma_1 \Delta X_{2,t-1} + \gamma_2 \Delta X_{1,t-2} + \epsilon_t$ we have:

The null hypothesis that $H_0 : \Delta X_{2,t}$ is not Granger causal for $\Delta X_{1,t}$, that is $H_0 : \gamma_1 = \gamma_2 = 0$

We can see from the results for the first regression that the p-values for the coefficients for lags of $\Delta X_{2,t}$ $t_{\gamma_1} = 1.74$ and $t_{\gamma_2} = -1.27$.

For a joint test (F-test) $F = \frac{(R_1^2 - R_0^2)/g}{(1-R_1^2)/(n-k)} \sim F_{(g,n-k)}$ so we need to model under the null $H_0 : \gamma_1 = \gamma_2 = 0 \rightarrow \Delta X_{1,t} = \alpha + \beta_1 \Delta X_{1,t-1} + \beta_2 \Delta X_{1,t-2} + \epsilon_t$.

The value of $g = 2$ $n = 38$ because we start with 40 observations minus 2 lost due to the two lags in the model. And $R_1^2 = 0.476$, $R_0^2 = 0.476$

```
lm0 <- lm(DX1 ~ lag(DX1,1) + lag(DX1,2), data=revenue) # Restricted mode
lm1 <- lm(DX1 ~ lag(DX1,1) + lag(DX1,2) + lag(DX2,1) + lag(DX2,2), data=revenue) # Unrestricted model
# The rest is calculated automatically.
n <- nobs(lm1)
g <- length(lm1$coefficients)-length(lm0$coefficients)
k <- length(lm1$coefficients)
r2_0 <- summary(lm0)$r.squared
r2_1 <- summary(lm1)$r.squared
F_test <- ((r2_1-r2_0)/g)/((1-r2_1)/(n-k))
F_crit <- qf(0.95,g,(n-k))
print(paste("The F test value is ",round(F_test,3)))
```

```
## [1] "The F test value is  2.15"
```

```
print(paste("The F critical value at 0.95% is ",round(F_crit,3)))
```

```
## [1] "The F critical value at 0.95% is  3.285"
```

```
if(F_test<F_crit){
  print(paste("We do not reject H_0 at 0.95%"))
} else {
  print(paste("We reject H_0 at 0.95% in favor of H_1"))
}
```

```
## [1] "We do not reject H_0 at 0.95%"
```

There's no indication that DX2 is granger causal for DX1

Now we do the same for $\Delta X_{2,t}$ in the model $\Delta X_{2,t} = \alpha + \beta_1 \Delta X_{1,t-1} + \beta_2 \Delta X_{1,t-2} + \gamma_1 \Delta X_{2,t-1} + \gamma_2 \Delta X_{1,t-2} + \epsilon_t$.

The null hypothesis that $H_0 : \Delta X_{1,t}$ is not Granger causal for $\Delta X_{2,t}$, that is $H_0 : \beta_1 = \beta_2 = 0$

```
lm0 <- lm(DX2 ~ lag(DX2,1) + lag(DX2,2), data=revenue) # Restricted mode
lm1 <- lm(DX2 ~ lag(DX1,1) + lag(DX1,2) + lag(DX2,1) + lag(DX2,2), data=revenue) # Unrestricted model
# The rest is calculated automatically.
n <- nobs(lm1)
g <- length(lm1$coefficients)-length(lm0$coefficients)
k <- length(lm1$coefficients)
r2_0 <- summary(lm0)$r.squared
r2_1 <- summary(lm1)$r.squared
F_test <- ((r2_1-r2_0)/g)/((1-r2_1)/(n-k))
F_crit <- qf(0.95,g,(n-k))
print(paste("The F test value is ",round(F_test,3)))
```

```
## [1] "The F test value is  20.978"
```

```
print(paste("The F critical value at 0.95% is ",round(F_crit,3)))
```

```
## [1] "The F critical value at 0.95% is  3.285"
```

```
if(F_test<F_crit){
  print(paste("We do not reject H_0 at 0.95%"))
} else {
  print(paste("We reject H_0 at 0.95% in favor of H_1"))
}
```

```
## [1] "We reject H_0 at 0.95% in favor of H_1"
```

There's evidence that DX1 is granger causal for DX2

Clearly, growth rates of X1 (DX1) can be predicted by their own past, and some of that past is also informative for DX2. So **airline one is Granger causal for airline two, but not the other way around.** And note that you need formal tests to establish this result, as the graphs are in no way informative in this respect.

2. Check for cointegration

Next, let us investigate whether the two series are cointegrated. The linear regression of X2 on X1 gives the **candidate long-run equilibrium relation with a slope coefficient of 0.92.**

```
lm3 <- lm(X2 ~ X1, data=revenue)
summ(lm3, digits = 2)
```

Observations	41
Dependent variable	X2
Type	OLS linear regression

F(1,39)	2777.98
R^2	0.99
Adj. R^2	0.99

$$\text{Step 1 OLS} : X_{2,t} = 0.01 + 0.92 X_{1,t} + e_t$$

	Est.	S.E.	t val.	p
(Intercept)	0.01	0.05	0.23	0.82
X1	0.92	0.02	52.71	0.00

Standard errors: OLS

```
# We add the residuals of lm3 into the dataset
revenue <- revenue %>% mutate(lm3.res = resid(lm3))
# Perform ADF on the residuals.
# Set k=0 for the Dickey Fuller test.
test <- urca::ur.df(na.omit(revenue$lm3.res), type = "none", lags = 1)
summary(test)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.031330 -0.010599  0.000282  0.010458  0.033622
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -0.4962     0.1394  -3.558  0.00104 **
## z.diff.lag   0.3043     0.1637   1.859  0.07104 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01456 on 37 degrees of freedom
## Multiple R-squared:  0.2556, Adjusted R-squared:  0.2154
## F-statistic: 6.353 on 2 and 37 DF,  p-value: 0.004249
##
##
## Value of test-statistic is: -3.5583
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.62 -1.95 -1.61
```

```
# The intercept and tt estimates indicate where there is a non-zero level (intercept) or linear trend (
```

$$\text{Step 2 ADF} : \Delta e_t = 0.00 - 0.50 e_{t-1} + 0.30 \Delta e_{t-1} + res_t$$

The ADF test is based on the estimate of -0.50 for the coefficient of e in period $t-1$ in the relevant auxiliary regression. And it equals $t = -3.558$. And as this is smaller than the 5% critical value -3.4, we can conclude that the residuals from the regression in step one are stationary. **And thus, that X1 and X2**

are co-integrated.

As the t value is smaller than the critical value -3.4, e_t are stationary so X1 and X2 are co-integrated.

When we now include the **error correction term** in the model for the growth rates of $X1$, we get an adjustment parameter of 0.46. And when we include it in the model for the growth rates of $X2$, we get -0.45.

- ECM (after removing insignificant coefficients) and susbtituing Step 1 OLS removing insignificant coefficients: $X_{2,t} = 0.92X_{1,t}$.

$$\Delta X_{1,t} = 0.00 + 1.02\Delta X_{1,t-1} + 0.46(X_{2,t-1} - 0.92X_{1,t-1}) + e_{1,t} \Delta X_{2,t} = 0.02 - 0.45(X_{2,t-1} - 0.92X_{1,t-1}) + e_{2,t}$$

The signs of these parameters imply that X1 and X2 cannot deviate too much from their equilibrium. For example, suppose that there is a positive deviation in period $t-1$. This will have an increasing effect on $X1_t$ in period t and a decreasing effect on $X2_t$. And as a result, the deviation from equilibrium is expected to decline in period t. Note also that deviations from the equilibrium are corrected by changes in both $X1_t$ and $X2_t$

$$\begin{aligned}
&\text{If } D_{t-1} = X_{2,t-1} - 0.92X_{1,t-1} \geq 0 \text{ is positive, then} \\
&0.46 > 0 \rightarrow X_{1,t} \uparrow \rightarrow D_t = X_{2,t-1} - 0.92X_{1,t-1} \downarrow \\
&-0.45 < 0 \rightarrow X_{2,t} \downarrow \rightarrow D_t = X_{2,t-1} - 0.92X_{1,t-1} \downarrow \\
&\text{Error correction mechanism acts on both variables}
\end{aligned}$$

3. ECM: Check for serial correlation and normality

Let us finally employ diagnostic tests on the residual autocorrelation and normality. The error correction models were as follows.

```
ecm1 <- lm(DX1 ~ lag(DX1,1) + I(lag(X2,1)-0.92*lag(X1,1)), data=revenue)
ecm2 <- lm(DX2 ~  I(lag(X2,1)-0.92*lag(X1,1)), data=revenue)
summ(ecm1, digits = 3)
```

Observations	39 (2 missing obs. deleted)
Dependent variable	DX1
Type	OLS linear regression

F(2,36)	27.938
R^2	0.608
Adj. R^2	0.586

	Est.	S.E.	t val.	p
(Intercept)	-0.004	0.003	-1.240	0.223
lag(DX1, 1)	1.022	0.138	7.403	0.000
I(lag(X2, 1) - 0.92 * lag(X1, 1))	0.463	0.107	4.335	0.000

Standard errors: OLS

```
summ(ecm2, digits = 3)
```

- ECM models for log(RPK) of airline companies 1 and 2 (n = 39):

Observations	40 (1 missing obs. deleted)
Dependent variable	DX2
Type	OLS linear regression

F(1,38)	33.604
R^2	0.469
Adj. R^2	0.455

	Est.	S.E.	t val.	p
(Intercept)	0.018	0.002	11.285	0.000
I(lag(X2, 1) - 0.92 * lag(X1, 1))	-0.447	0.077	-5.797	0.000

Standard errors: OLS

$$\Delta X_{1,t} = 0.00 + 1.02\Delta X_{1,t-1} + 0.46(X_{2,t-1} - 0.92X_{1,t-1}) + e_{1,t} \Delta X_{2,t} = 0.02 - 0.45(X_{2,t-1} - 0.92X_{1,t-1}) + e_{2,t}$$

The Jarque-Bera test statistics for normality show that normality of the residuals is not rejected for both X1 and X2. And also the Breusch-Godfrey test values for first-order residual autocorrelation are not significant.

```
# Null of normality
jarque.bera.test(ecm1$residuals) # Null: Normality not rejected
```

```
##
##  Jarque Bera Test
##
## data:  ecm1$residuals
## X-squared = 0.38543, df = 2, p-value = 0.8247
```

```
jarque.bera.test(ecm2$residuals) # Null: Normality not rejected
```

```
##
##  Jarque Bera Test
##
## data:  ecm2$residuals
## X-squared = 1.8226, df = 2, p-value = 0.402
```

Jarque-Bera test : $JB_1 = 0.38 < 6, JB_2 = 1.82 < 6$. Normality not rejected

The function bgtest performs the Breusch-Godfrey test for higher-order serial correlation.

Under the null hypothesis of no autocorrelation, this test has a chi-square distribution with R degrees of freedom

```
bgtest(ecm1, order = 1) # Null: No autocorrelation not rejected
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  ecm1
## LM test = 0.29898, df = 1, p-value = 0.5845
```

```
bgtest(ecm2, order = 1) # Null: No autocorrelation not rejected
```

```
##
```

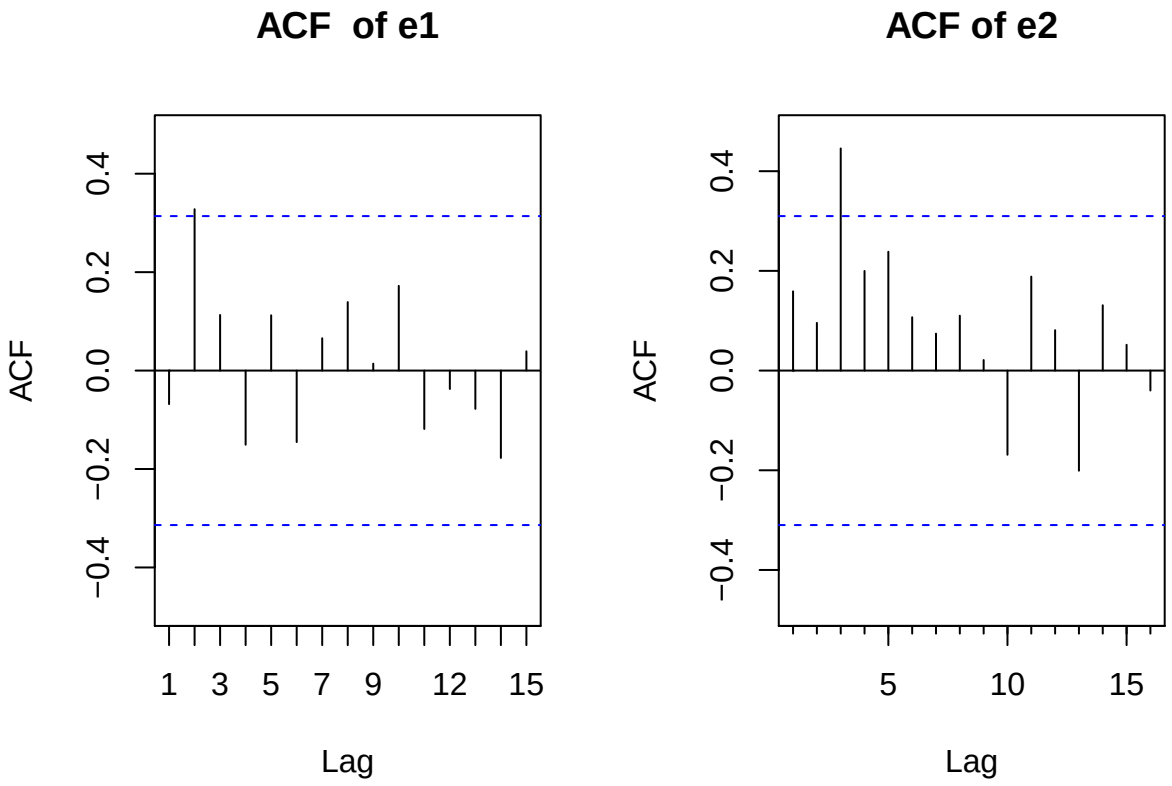
```
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  ecm2
## LM test = 1.2399, df = 1, p-value = 0.2655
```

Breusch-Godfrey test (1 lag) : $BG_1 = 0.29 < 3.9, BG_2 = 1.23 < 3.9$ No autocorrelation

$$ACF : 2/\sqrt{n} = 2/\sqrt{39} = 0.32$$

The absence of such correlation is also visible from the sample autocorrelations of the residuals shown in this graph. Only 1 out of the 20 autocorrelations is outside the 95% confidence bound, and this is not unusual for a test with significance level 5%

```
par(mfrow=c(1,2))
Acf(ecm1$residuals,main = "ACF  of e1")
Acf(ecm2$residuals,main = "ACF of e2")
```



Application on Production and CLI

Look at Dates and Times in R Without Losing Your Sanity to understand how to use correctly date labes in R.

Datasets to be used:

```
production <- read_csv(
  "https://raw.githubusercontent.com/diego-eco/diego-eco.github.io/master/downloads/dataset62.csv")
# We replace the weird "M" before months.
production <- rename(production, date=`YYYY-MM`)
```

```
production$date <- gsub("M","-",production$date)
production$date <- as.Date(as.yearmon(production$date))
```

- **production** Data set on Industrial Production and the Composite Leading Index for the USA, monthly data Jan 1985 - Dec 2007 (Source: Conference Board, USA). Estimation period is Jan 1986 to Dec 2005 (pre sample values in 1985). Forecast evaluation period is Jan 2006 to Dec 2007.
- **production_train** Data set on Industrial Production and the Composite Leading Index for the USA, yearly data 1960 to 2007 (Source: Conference Board, USA). Estimation period is 1960 to 2002. Forecast evaluation period is 2003 to 2007.
- CLI: Composite Leading Index (based on 10 leading indicators)
- IP: Industrial Production (index, seasonally adjusted)
- LOGCLI: logarithm of CLI
- LOGIP: logarithm of IP
- GRCLI: monthly growth rate of CLI,first difference of LOGCLI
- GRIP: monthly growth rate of IP,first difference of LOGIP

In this section we will see a detailed application of the various techniques outlined in the previous four sections. The two time series of interest are the monthly industrial production index for the United States of America and the so-called Composite Leading Index or CLI. The CLI is constructed by The Conference Board based on a set of variables like manufacturers' new orders and consumer expectations.

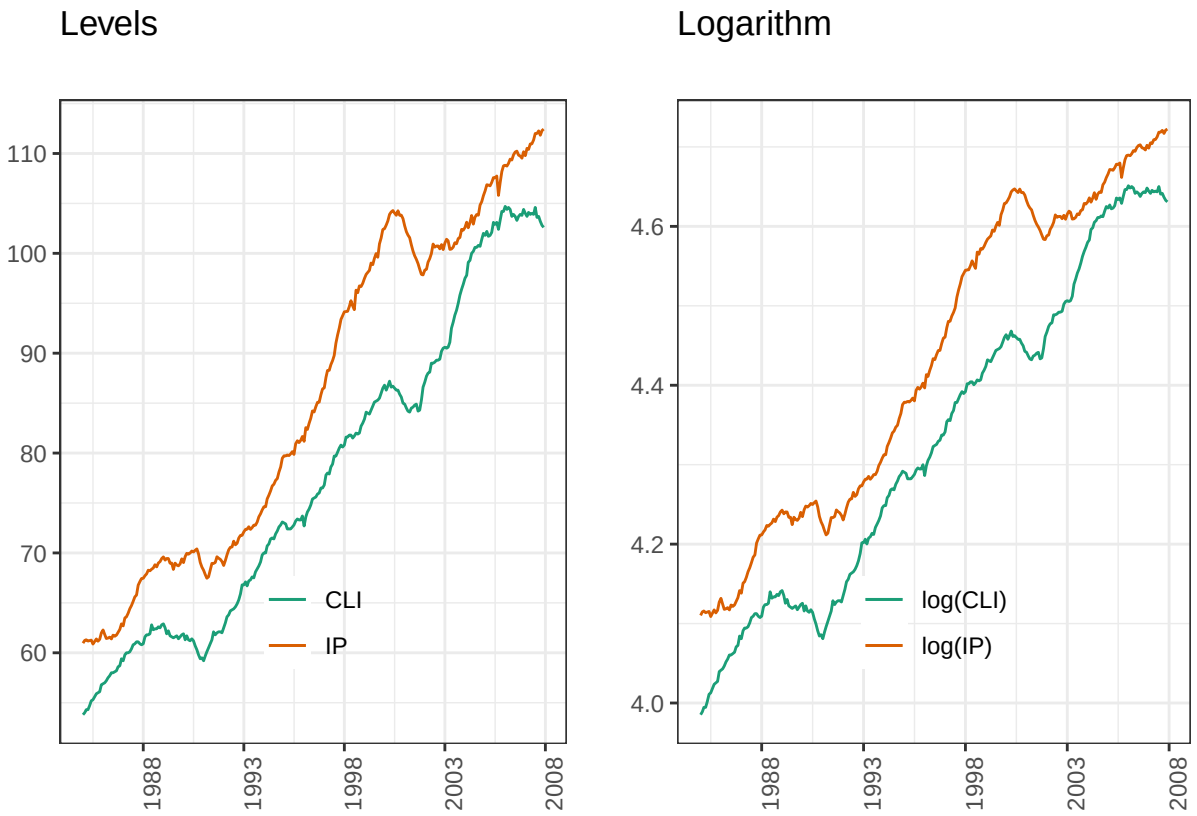
All these variables are forward looking and therefore, they are believed to have predictive value for future macro-economic developments. Here we will investigate whether this index does indeed have predictive power for industrial production. The sample runs from 1986 to 2007, such that we have 264 monthly observations. When needed for models with lags the values for 1985 are also available.

Graphical analysis

As you can see from these graphs, the two time series show an upward trend.

```
plot_a <- ggplot(data=production, aes(x=date)) +
  geom_line(aes(y=IP,col="IP")) +
  geom_line(aes(y=CLI,col="CLI")) +
  labs(x = "", y = "", title = "Levels",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .20),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

plot_b <- ggplot(data=production, aes(x=date)) +
  geom_line(aes(y=LOGIP,col="log(IP)")) +
  geom_line(aes(y=LOGCLI,col="log(CLI)")) +
  labs(x = "", y = "", title = "Logarithm",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .20),
        legend.background = element_rect(fill = "transparent")) +
```

```
  scale_color_brewer(name= NULL, palette = "Dark2")

grid.arrange(plot_a, plot_b, nrow = 1)
```



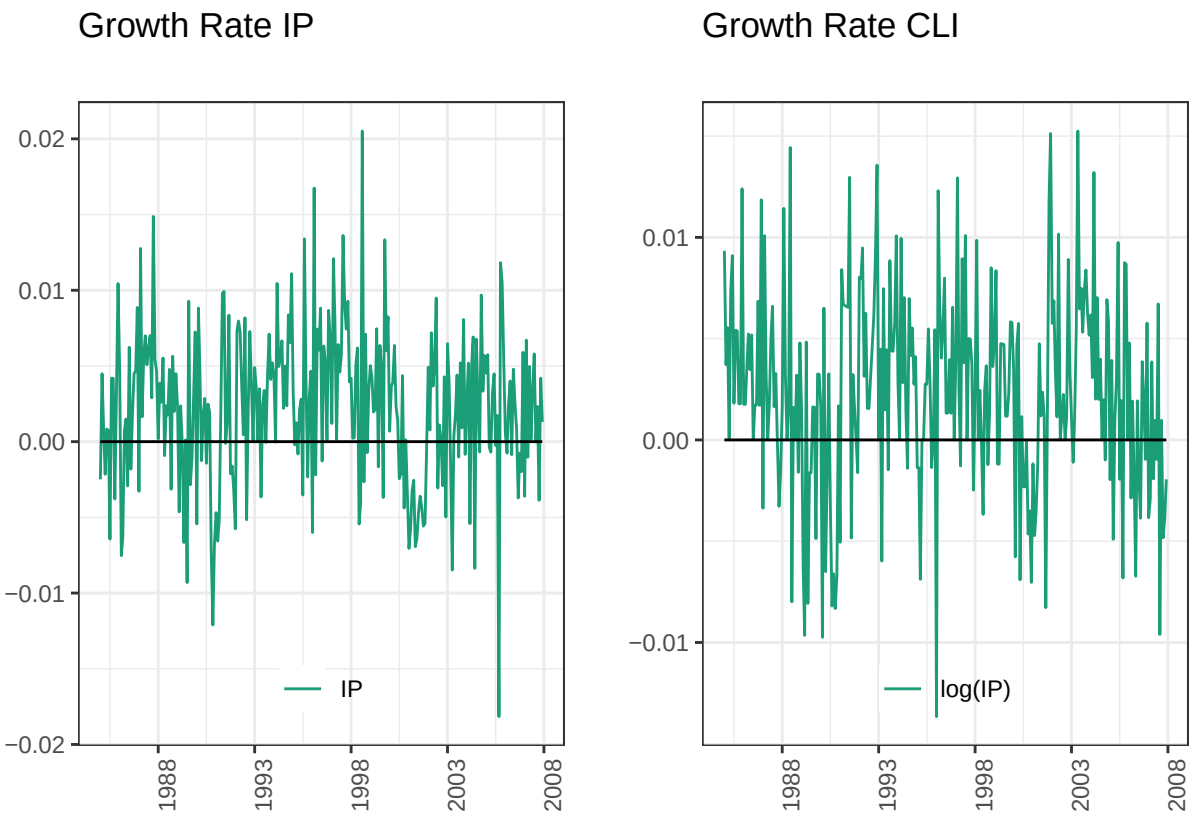
At first glance, the series also seem to obey roughly similar cyclical behavior, which in this case can be associated with the business cycle. If you examine the two series in detail, you may observe that the turning points in this cycle occur earlier in the CLI than in industrial production. And this is exactly the feature that leads to the idea that the CLI may have relevant information for predicting industrial production.

Here we specifically examine if we can forecast industrial production three months ahead. Not only from its own past, but also from the past of the leading index.

Let us consider what the monthly growth rates look like. $GRIP = \Delta log(IP)$ $GRCLI = \Delta log(CLI)$ We will use the acronym GRIP for the monthly growth rate of industrial production, which is the variable we wish to predict.

```
plot_c <- ggplot(data=production, aes(x=date)) +
  geom_line(aes(y=GRIP,col="IP")) +
  geom_line(aes(y=0)) +
  labs(x = "", y = "", title = "Growth Rate IP",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .10),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

plot_d <- ggplot(data=production, aes(x=date)) +
```

```
  geom_line(aes(y=GRCLI,col="log(IP)")) +
  geom_line(aes(y=0)) +
  labs(x = "", y = "", title = "Growth Rate CLI",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .10),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

grid.arrange(plot_c, plot_d, nrow = 1)
```



Separate data set

We take the period from 1986 until 2005 as our estimation sample so that 240 observations are available for estimation. And, we then use the 24 months in 2006 and 2007 as a holdout sample to evaluate our forecasts.

We filter with date data

```
estimation <- production %>% filter(date >  as.Date("1985-12-01") & date <= as.Date("2005-12-01"))
forecast <- production %>% filter(date >    as.Date("2005-12-01"))
```

Test unit root

As always, we start with a test for stationarity, where we use y_t to denote each of the variables. The augmented Dickey-Fuller test equation should include a deterministic trend, as the log data obviously have a trend. In the test equation, we also include three lags of the differenced variable to capture the autocorrelation in the time series.

ADF test :

$$\Delta y_t = \alpha + \beta t + \rho y_{t-1} + \sum_{j=1}^{3} \gamma_j \Delta y_{t-j} + \epsilon_t$$

Reject H_0 : non-stationarity if
$t_{\hat{\rho}} < -2.9$ if $\beta = 0, t_{\hat{\rho}} < -3.5$ if $\beta \neq 0$

```
# Set k=1 for the Dickey Fuller test with first lag and type 'trend'
test_logip <- urca::ur.df(na.omit(estimation$LOGIP), type = "trend", lags = 3)
summary(test_logip)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0207678 -0.0033392  0.0001451  0.0030167  0.0196143
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.314e-02  3.211e-02   1.655 0.099324 .
## z.lag.1     -1.254e-02  7.824e-03  -1.603 0.110308
## tt           3.015e-05  2.079e-05   1.450 0.148416
## z.diff.lag1  1.817e-02  6.394e-02   0.284 0.776582
## z.diff.lag2  2.128e-01  6.263e-02   3.397 0.000802 ***
## z.diff.lag3  2.376e-01  6.419e-02   3.701 0.000268 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005031 on 230 degrees of freedom
## Multiple R-squared:  0.1149, Adjusted R-squared:  0.09565
## F-statistic: 5.971 on 5 and 230 DF,  p-value: 3.249e-05
##
##
## Value of test-statistic is: -1.603 5.0484 1.3956
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

```
test_logcli <- urca::ur.df(na.omit(estimation$LOGCLI), type = "trend", lags = 3)
summary(test_logcli)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0158400 -0.0027337 -0.0000552  0.0027129  0.0136335
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.478e-02  4.051e-02   1.846  0.06617 .
## z.lag.1     -1.840e-02  1.011e-02  -1.820  0.07011 .
## tt           5.112e-05  2.587e-05   1.976  0.04933 *
## z.diff.lag1  9.049e-02  6.544e-02   1.383  0.16806
## z.diff.lag2  1.869e-01  6.485e-02   2.881  0.00434 **
## z.diff.lag3  8.929e-02  6.599e-02   1.353  0.17735
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00486 on 230 degrees of freedom
## Multiple R-squared:  0.07583,    Adjusted R-squared:  0.05574
## F-statistic: 3.774 on 5 and 230 DF,  p-value: 0.002645
##
##
## Value of test-statistic is: -1.8197 7.0627 2.1865
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

The t-ratios of the coefficient $t\hat{\rho} = -1.603$ for log(IP) and $t\hat{\rho} = -1.8197$ for log(CLI) in the augmented Dickey-Fuller regression are both larger than the 5% critical value -3.5.

And hence, not surprisingly, the test confirms that the log series are not stationary.

Running the same auxiliary test regression for the growth rates, now without including a deterministic trend, gives two significant test statistics. And hence, the monthly growth rates are indeed stationary.

```
# Set k=1 for the Dickey Fuller test with first lag.
adf.test(estimation$GRIP,k=3)
```

```
##
##  Augmented Dickey-Fuller Test
##
```

```
## data:  estimation$GRIP
## Dickey-Fuller = -5.228, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(estimation$GRCLI,k=3)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  estimation$GRCLI
## Dickey-Fuller = -5.5533, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

The t-ratios of the coefficient $t\hat{\rho} = -5.228$ for GRIP and $t\hat{\rho} = -5.553$ for GRCLI in the augmented Dickey-Fuller regression are both smaller than the 5% critical value -2.9.

And hence, the monthly growth rates are indeed stationary.

Test for cointegration

Are the two series perhaps, cointegrated?

```
step1 <- lm(LOGIP ~ LOGCLI, data=estimation)
summ(step1, digits = 3)
```

Observations	240
Dependent variable	LOGIP
Type	OLS linear regression

F(1,238)	4540.570
R^2	0.950
Adj. R^2	0.950

	Est.	S.E.	t val.	p
(Intercept)	0.083	0.064	1.288	0.199
LOGCLI	1.005	0.015	67.384	0.000

Standard errors: OLS

$$\text{Step 1 OLS} : log(IP) = 0.083 + 1.005 log(CLI) + e_t$$

Well, a potential long run relation ties the two series with the parameter close to 1.

```
# We add the residuals of step 1 into the dataset
estimation <- estimation %>% mutate(step1.res = resid(step1))
# Perform ADF on the residuals.
test <- urca::ur.df(estimation$step1.res, type = "none", lags = 1)
summary(test)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
```

```
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0191612 -0.0035949  0.0001314  0.0034891  0.0189884
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.006836   0.009528  -0.717    0.474
## z.diff.lag  0.048331   0.065162   0.742    0.459
##
## Residual standard error: 0.005955 on 236 degrees of freedom
## Multiple R-squared:  0.004036,   Adjusted R-squared:  -0.004405
## F-statistic: 0.4781 on 2 and 236 DF,  p-value: 0.6205
##
##
## Value of test-statistic is: -0.7174
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

$$\text{Step 2 ADF} : \Delta e_t = 0.00 - 0.0068 e_{t-1} + 0.048 \Delta e_{t-1} + res_t$$

But in the second step of the Engle-Granger procedure, we find that the relevant parameter is estimated as -0.01 with a t-ratio $t = -0.717$ larger than the critical value -3.8, implying that the two series are not cointegrated.

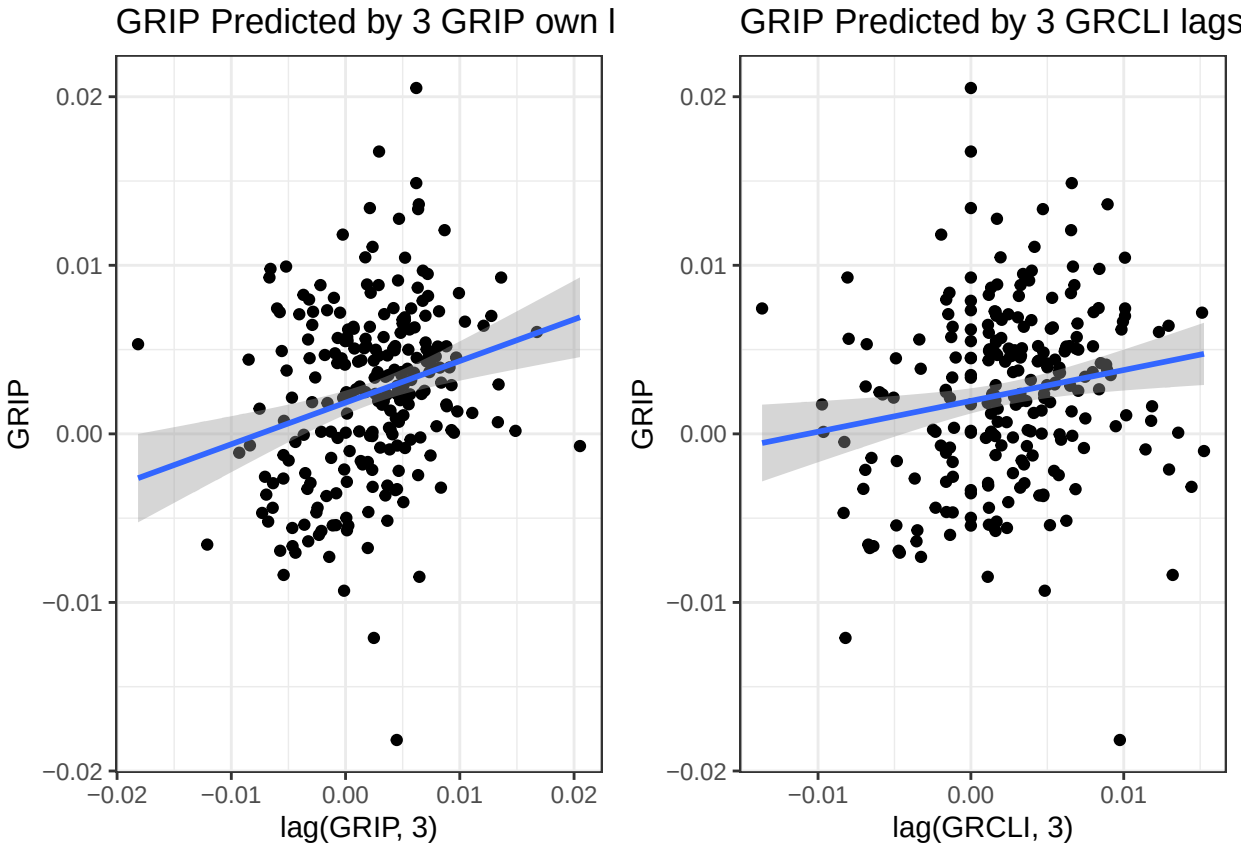
The t value is larger than the critical, implying that the two series are not cointegrated.

So based on this analysis of known stationarity and cointegration, from now on, **we will only consider the monthly growth rates.**

Especify the model

The scatter diagrams with regression lines, suggest that GRIP can be predicted by the growth rates of the leading index three months ago. But the left hand graph also suggests that GRIP can be predicted by its own past.

```
plot_a <- ggplot(data=estimation, aes(x=lag(GRIP,3),y=GRIP)) + geom_point() + geom_smooth(method='lm') +
  labs(title="GRIP Predicted by 3 GRIP own lags") + theme_bw()
plot_b <- ggplot(data=estimation, aes(x=lag(GRCLI,3),y=GRIP)) + geom_point() + geom_smooth(method='lm') +
  labs(title="GRIP Predicted by 3 GRCLI lags") + theme_bw()
grid.arrange(plot_a, plot_b, nrow = 1)
```



Because we wish to forecast three months ahead, only the lags of order three and larger are allowed in the models we are going to consider.

- Forecast $GRIP_t$ with information $GRIP_{t-j}, j = 3, 4, 5, ..$

We will design two models. First, an univariate autoregression for GRIP, and second an autoregressive distributed lag model that also includes lagged CLI growth rates.

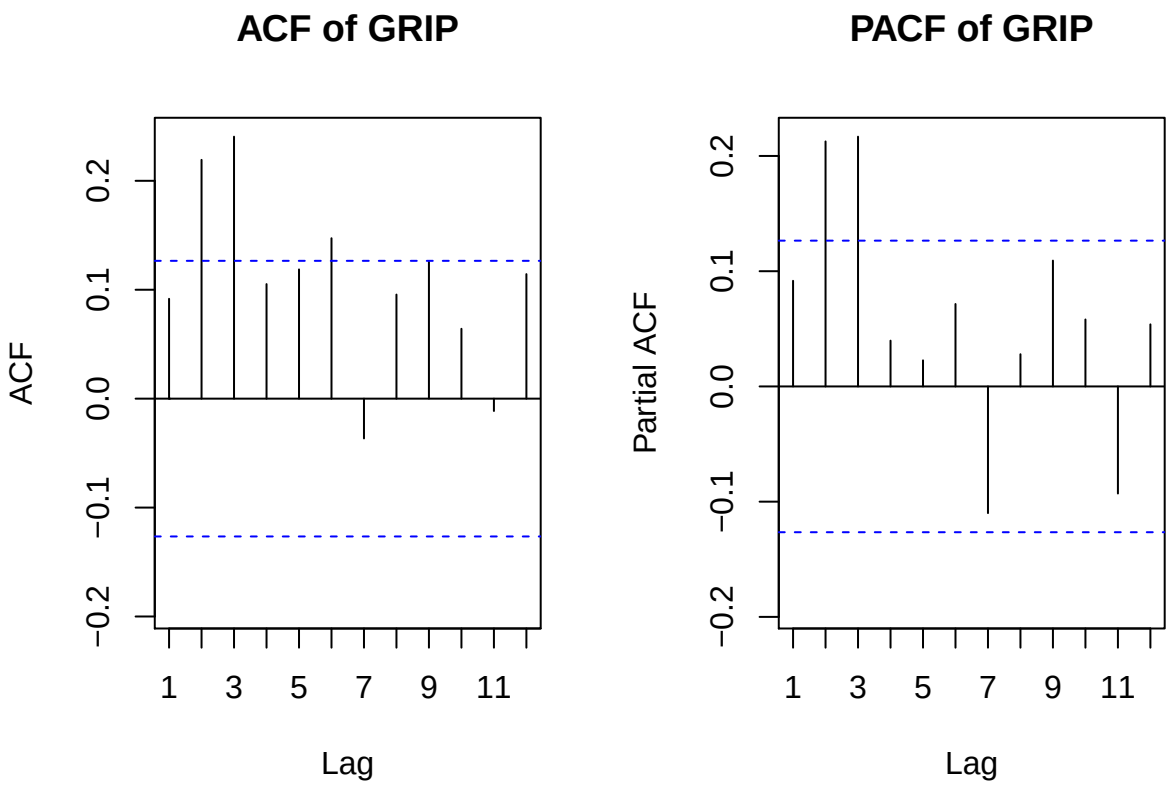
Especify AR(p)

First, we specify the autoregressive model for GRIP.

To get a hint of the appropriate number of lags, we compute the autocorrelations and partial autocorrelations up to lag 12.

Their values are shown in these two graphs.

```
# We create the time series object
par(mfrow=c(1,2))
Acf(estimation$GRIP,main = "ACF of GRIP", lag.max = 12)
Pacf(estimation$GRIP,main = "PACF of GRIP", lag.max = 12)
```



The 95% confidence bounds are equal to plus and minus 0.13. And a first impression is that an AR(3) model may be useful. This is based on the significance of the partial autocorrelations up to order 3.

$$2/\sqrt{n} = 0.13 \longrightarrow AR(3)$$

As $GRIP_{t-1}, GRIP_{t-2}$ may not be used, start with lags 3-12 and reduce (down-testing).

As our model may not use lags 1 and 2, we start with an autoregression that contains all lags from 3 to 12. And then, we delete lag terms that are not significant.

$$GRIP_t = \alpha + \sum_{j=3}^{L} \beta_j GRIP_{t-j} + \epsilon_t$$

```
dynlm(GRIP ~ lag(GRIP,1), data=estimation)
```

```
##
## Time series regression with "numeric" data:
## Start = 1, End = 239
##
## Call:
## dynlm(formula = GRIP ~ lag(GRIP, 1), data = estimation)
##
## Coefficients:
##  (Intercept)  lag(GRIP, 1)
##     0.002119      0.091785
```

Downtesting in R dynlm function

The function dynlm() does not compute information criteria by default. We will therefore write a short function that reports the BIC (along with the chosen lag order p and R^2) for objects of class dynlm.

```
ar12 <- lm(GRIP ~ lag(GRIP,3)+lag(GRIP,4)+lag(GRIP,5)+lag(GRIP,6)+lag(GRIP,7)+
          lag(GRIP,8)+lag(GRIP,9)+lag(GRIP,10)+lag(GRIP,11)+lag(GRIP,12), data=estimation)
summ(ar12,digits = 3)
```

Observations	228 (12 missing obs. deleted)
Dependent variable	GRIP
Type	OLS linear regression

F(10,217)	2.950
R^2	0.120
Adj. R^2	0.079

	Est.	S.E.	t val.	p
(Intercept)	0.001	0.000	2.756	0.006
lag(GRIP, 3)	0.206	0.070	2.951	0.004
lag(GRIP, 4)	0.084	0.072	1.171	0.243
lag(GRIP, 5)	0.087	0.074	1.177	0.240
lag(GRIP, 6)	0.037	0.075	0.498	0.619
lag(GRIP, 7)	-0.169	0.076	-2.233	0.027
lag(GRIP, 8)	0.050	0.075	0.656	0.512
lag(GRIP, 9)	0.132	0.076	1.747	0.082
lag(GRIP, 10)	0.080	0.074	1.089	0.277
lag(GRIP, 11)	-0.098	0.071	-1.374	0.171
lag(GRIP, 12)	0.052	0.072	0.722	0.471

Standard errors: OLS

- Start with L = 12: lags 4-12 individually not significant. They may, however, be jointly significant.

```
ar3 <- lm(GRIP ~ lag(GRIP,3), data=estimation)
summ(ar3,digits = 3)
```

Observations	237 (3 missing obs. deleted)
Dependent variable	GRIP
Type	OLS linear regression

F(1,235)	15.274
R^2	0.061
Adj. R^2	0.057

	Est.	S.E.	t val.	p
(Intercept)	0.002	0.000	5.118	0.000
lag(GRIP, 3)	0.247	0.063	3.908	0.000

Standard errors: OLS

Is a model with only lag three to be preferred over a model with lags three to 12? We use the familiar F-test.

```
lm0 <- lm(GRIP ~ lag(GRIP,3), data=estimation) # Restricted mode
lm1 <- lm(GRIP ~ lag(GRIP,3)+lag(GRIP,4)+lag(GRIP,5)+lag(GRIP,6)+lag(GRIP,7)+
          lag(GRIP,8)+lag(GRIP,9)+lag(GRIP,10)+lag(GRIP,11)+lag(GRIP,12)
          , data=estimation) # Unrestricted model

# H0 the restrictions hold true
# The rest is calculated automatically.
n <- nobs(lm1)
g <- length(lm1$coefficients)-length(lm0$coefficients)
k <- length(lm1$coefficients)
r2_0 <- summary(lm0)$r.squared
r2_1 <- summary(lm1)$r.squared
F_test <- ((r2_1-r2_0)/g)/((1-r2_1)/(n-k))
F_crit <- qf(0.95,g,(n-k))
print(paste("The F test value is ",round(F_test,3)))
```

```
## [1] "The F test value is  1.607"
```

```
print(paste("The F critical value at 0.95% is ",round(F_crit,3)))
```

```
## [1] "The F critical value at 0.95% is  1.923"
```

```
if(F_test<F_crit){
  print(paste("We do not reject H_0 at 0.95%"))
} else {
  print(paste("We reject H_0 at 0.95% in favor of H_1"))
}
```

```
## [1] "We do not reject H_0 at 0.95%"
```

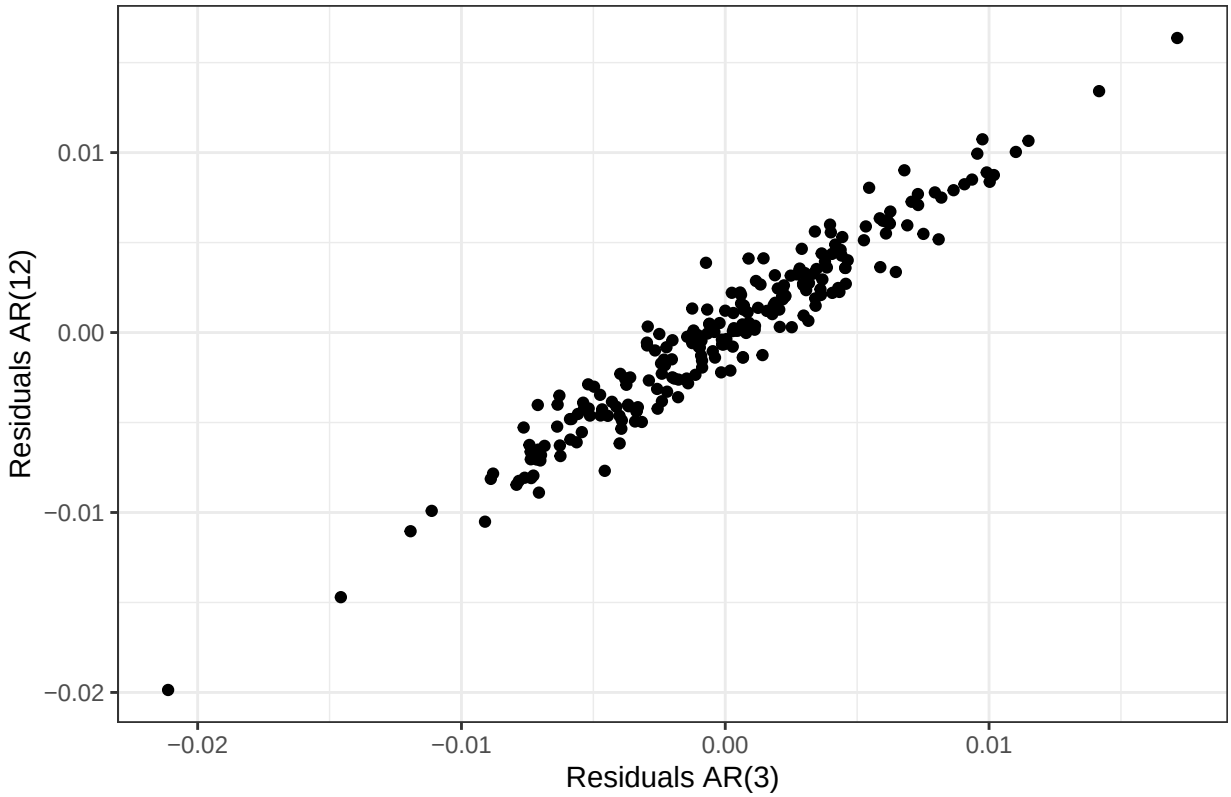
The test value that we obtain is 1.6, and hence, we can stick to just using lag 3 only.

As always, it is good to have a look at the residuals to see if there are any odd values.

```
# We add the residuals into the dataset (notice we need 3 NA values due to the 3 lags used)
estimation <- estimation %>% mutate(ar12.res = c(rep(NA,12),resid(ar12)))
estimation <- estimation %>% mutate(ar3.res = c(rep(NA,3),resid(ar3)))
```

In this scatter of the residuals of the model with 12 lags versus those with the model with three lags, we see that the residuals are very similar.

```
plot_c <- ggplot(data=estimation, aes(x=ar3.res,y=ar12.res)) + geom_point() +
  labs(x="Residuals AR(3)",y="Residuals AR(12)",title="") + theme_bw()
plot_c
```



This is confirmed by the almost same p-values of the diagnostic tests on the residual autocorrelation and on normality.

```
# Null of normality
jarque.bera.test(na.omit(estimation$ar12.res))
```

```
##
##  Jarque Bera Test
##
## data:  na.omit(estimation$ar12.res)
## X-squared = 7.8058, df = 2, p-value = 0.02018
```

```
jarque.bera.test(na.omit(estimation$ar3.res))
```

```
##
##  Jarque Bera Test
##
## data:  na.omit(estimation$ar3.res)
## X-squared = 11.022, df = 2, p-value = 0.004042
```

Normality is rejected for both models, which is due to a few extreme observations.

```
bgtest(ar12, order = 6) # Null: No autocorrelation
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 6
##
## data:  ar12
## LM test = 9.9107, df = 6, p-value = 0.1285
```

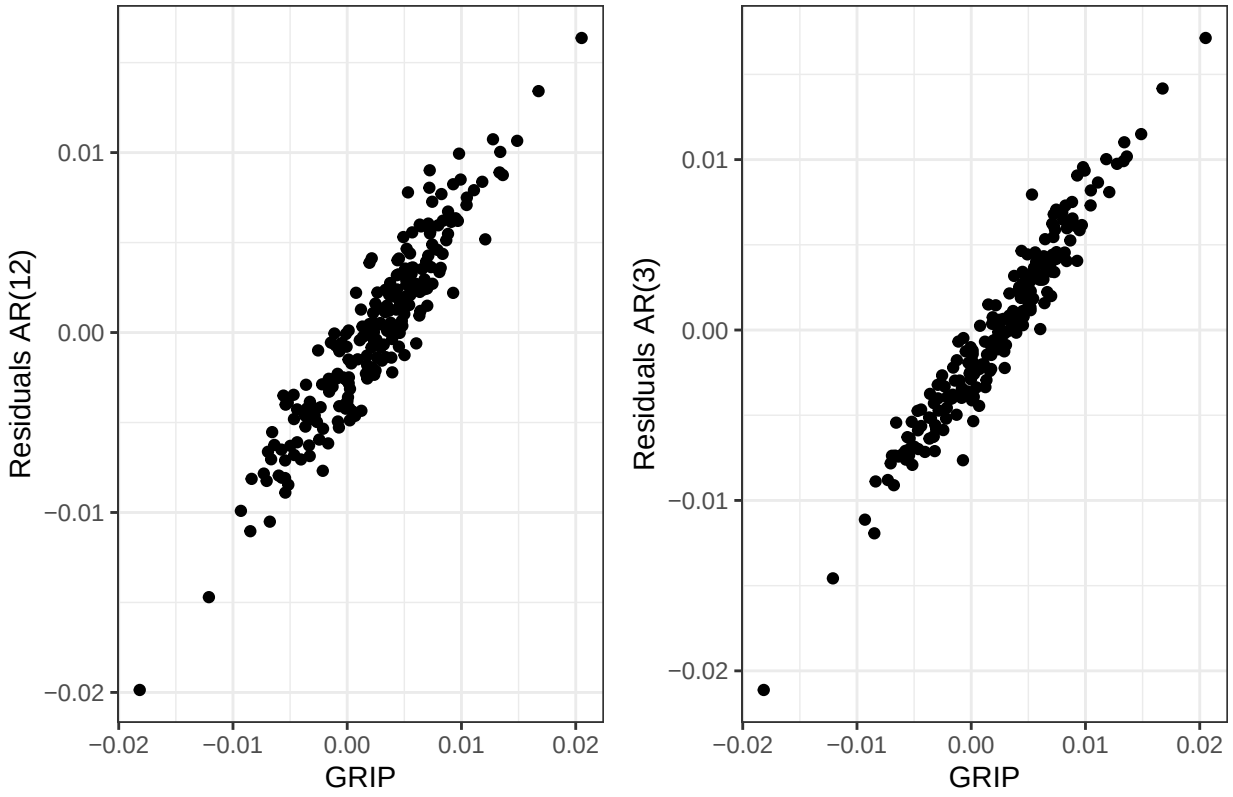
```
bgtest(ar3, order = 6) # Null: No autocorrelation
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 6
##
## data:  ar3
## LM test = 12.719, df = 6, p-value = 0.04772
```

No residual autocorrelation for the AR(3) model is rejected.

This can also be visualized by drawing a scatter of the residuals against the GRIP variable itself, and indeed when GRIP is large positive or large negative the residuals are large too.

```
plot_a <- ggplot(data=estimation, aes(x=GRIP,y=ar12.res)) + geom_point() +
  labs(x="GRIP",y="Residuals AR(12)",title="") + theme_bw()
plot_b <- ggplot(data=estimation, aes(x=GRIP,y=ar3.res)) + geom_point() +
  labs(x="GRIP",y="Residuals AR(3)",title="") + theme_bw()
grid.arrange(plot_a, plot_b, nrow = 1)
```



It seems that there four such large residuals in this case. It seems that there four such large residuals in this case. They associate with four isolated observations, for months with exceptionally positive or negative growth.

- High growth: Feb 1996 (1.7%) and Aug 1998 (2.1%)
- Large negative growth: Nov 1990 (-1.2%) and Sep 2005 (-1.8%)

As we do not see any particular strategy to deal with these four outliers, we simply proceed with our autoregressive model, where the estimated lag 3 parameter is statistically significant.

$$\text{Forecast model 1 : } GRIP_t = 0.002 + 0.247 \cdot GRIP_{t-3} + e_t (t_b = 3.9, R^2 = 0.061)$$

Especify ADL(p,r)

The next question is, can this univariate model be improved by adding the Composite Leading Index? That is, is the variable CLI really leading, and if so, with how many lags?

To find out, we consider the autoregressive distributed lag model ALD(p,r), with p lags for GRIP and r lags for the leading index. Where we again start with lag three, as we wish to forecast three months ahead.

$$\text{ADL(p,r)} : GRIP_t = \alpha + \sum_{j=3}^{p} \beta_j GRIP_{t-j} + \sum_{j=3}^{r} \gamma_j GRICL_{t-j} + \epsilon_t$$

Our modeling cycle starts with p and r both equal to 6 and reduce (down-testing).

Our modeling cycle starts with p and r both equal to 6. And we work our way downwards by deleting insignificant lags.

```
adl66 <- lm(GRIP ~ lag(GRIP,3)+lag(GRIP,4)+lag(GRIP,5)+lag(GRIP,6)+lag(GRCLI,3)+
          lag(GRCLI,4)+lag(GRCLI,5)+lag(GRCLI,6), data=estimation)
summ(adl66,digits = 3)
```

Observations	234 (6 missing obs. deleted)
Dependent variable	GRIP
Type	OLS linear regression

F(8,225)	4.311
R^2	0.133
Adj. R^2	0.102

	Est.	S.E.	t val.	p
(Intercept)	0.001	0.000	1.911	0.057
lag(GRIP, 3)	0.137	0.075	1.817	0.070
lag(GRIP, 4)	-0.044	0.074	-0.594	0.553
lag(GRIP, 5)	0.068	0.069	0.977	0.329
lag(GRIP, 6)	0.074	0.072	1.034	0.302
lag(GRCLI, 3)	0.103	0.072	1.435	0.153
lag(GRCLI, 4)	0.136	0.072	1.891	0.060
lag(GRCLI, 5)	-0.028	0.075	-0.367	0.714
lag(GRCLI, 6)	0.195	0.075	2.592	0.010

Standard errors: OLS

When we do so, we arrive at a model with lag three of GRIP and lag six of the growth rate of the leading index.

```
adl36 <- lm(GRIP ~ lag(GRIP,3)+lag(GRCLI,6), data=estimation)
# We add the residuals into the dataset (notice we need 6 NA values due to the max(p,r))
estimation <- estimation %>% mutate(adl36.res = c(rep(NA,6),resid(adl36)))
summ(adl36,digits = 3)
```

Observations	234 (6 missing obs. deleted)
Dependent variable	GRIP
Type	OLS linear regression

F(2,231)	13.766
R^2	0.106
Adj. R^2	0.099

	Est.	S.E.	t val.	p
(Intercept)	0.001	0.000	3.521	0.001
lag(GRIP, 3)	0.204	0.064	3.182	0.002
lag(GRCLI, 6)	0.238	0.068	3.500	0.001

Standard errors: OLS

```
# Null of normality
jarque.bera.test(na.omit(estimation$adl36.res))
```

```
##
##  Jarque Bera Test
##
## data:  na.omit(estimation$adl36.res)
## X-squared = 7.1705, df = 2, p-value = 0.02773
```

```
# Null: No autocorrelation
bgtest(adl36, order = 6)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 6
##
## data:  adl36
## LM test = 7.5787, df = 6, p-value = 0.2706
```

- Breusch-Godfrey (6 lags): $p - value = 0.27$, no serial correlation
- Jarque-Bera $p - value = 0.027$ (same 4 outliers as before)

Forecast model 2 : $GRIP_t = 0.001 + 0.204 \cdot GRIP_{t-3} + 0.238 \cdot GRCLI_{t-6} + e_t (t_{b_3} = 3.2, t_{b_6} = 3.5, R^2 = 0.106)$

The associated coefficient is significant so we can conclude that, at least in-sample, the index is leading industrial production. Note that the inclusion of the leading index makes the residual autocorrelation that we had in the AR(3) model to disappear. However, the four outliers are still there, such that normality of the residuals is still rejected.

Out sample forecast

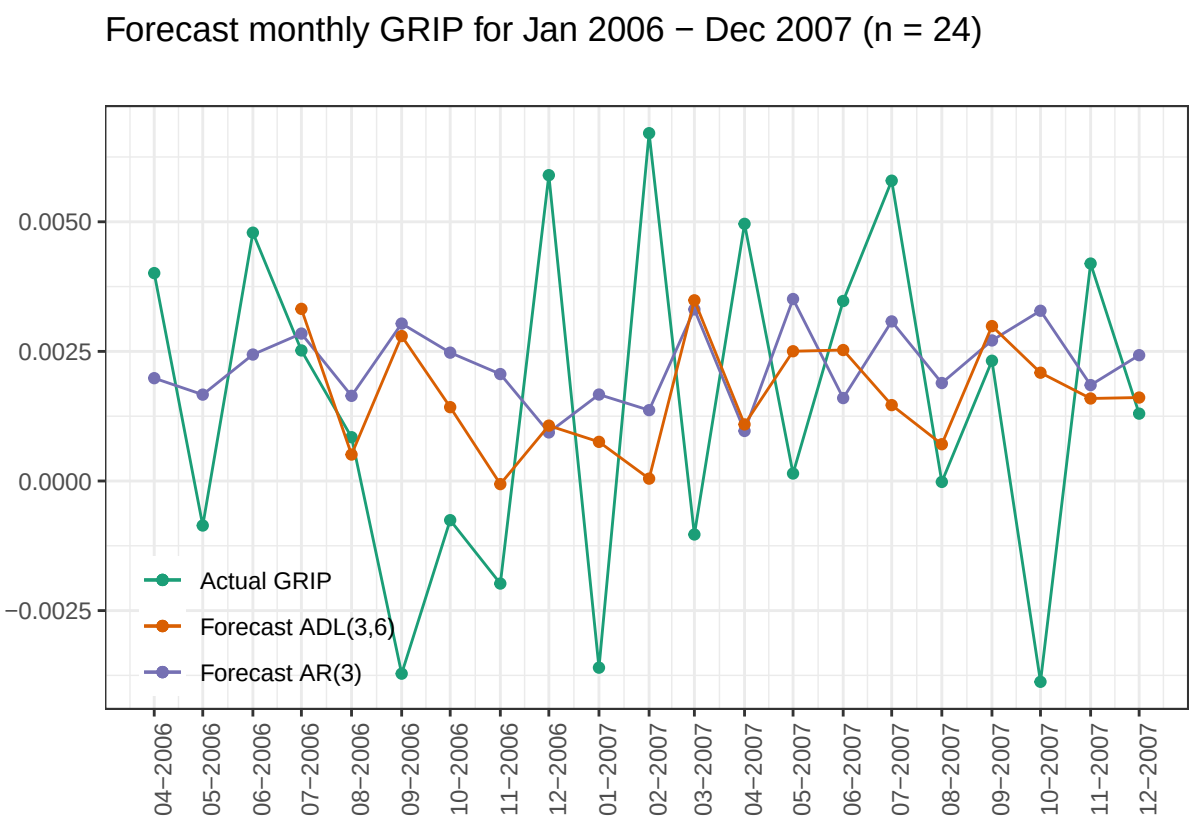
Now let us see whether the in-sample success also carries on to the out-of-sample period.

We use our model each month to forecast IP three months ahead, starting in October 2005 to forecast January 2006, and ending in September 2007 to forecast December 2007. And that gives us two series of 24 forecasts each, which we put here together with the actual monthly growth rates.

- AR (lag 3) **ar3** and ADL (lags 3 and 6) **adl36** estimated from data 1986-2005
- Forecast monthly GRIP for Jan 2006 - Dec 2007 (n = 24) and the annual growth rates of IP for the years 2006 and 2007.

Customize date ranges

```r
# We add the fitted values of both models to the forecast dataset
forecast <- forecast %>% mutate(ar3_fitted=predict(ar3,newdata =  forecast))
forecast <-  forecast %>% mutate(adl36_fitted=predict(adl36,newdata =  forecast))

# Define Start and end times for the subset as R objects that are the time class
startTime <- as.Date("2006-04-01")
endTime <- as.Date("2007-12-01")
# create a start and end time R object
start.end <- c(startTime,endTime)

forecast_plot <- ggplot(data=forecast, aes(x=date)) +
  geom_line(aes(y=GRIP,col="Actual GRIP")) + geom_point(aes(y=GRIP,col="Actual GRIP")) +
  geom_line(aes(y=ar3_fitted,col="Forecast AR(3)")) + geom_point(aes(y=ar3_fitted,col="Forecast AR(3)"))
  geom_line(aes(y=adl36_fitted,col="Forecast ADL(3,6)")) + geom_point(aes(y=adl36_fitted,col="Forecast
  labs(x = "", y = "", title = "Forecast monthly GRIP for Jan 2006 - Dec 2007 (n = 24)",
       subtitle = ("")) +
    scale_x_date(limits = start.end, date_labels = "%m-%Y",date_breaks = "1 month") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.15, .15),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

forecast_plot
```



You can see that even though the difference between the two models is just a single variable, the forecast can be quite different. And you can also observe that the actual growth rates fluctuate considerably per month.

So far, we considered the monthly growth rates. But for longer term decisions, one is often interested in the annual growth rates. That is, the growth rate of IP as measured from January to December.

$$\text{Montly growth rate of y : } g_t^m = \Delta log(y_t) \quad \text{Annual growth rate of y : } g_t^y = log(y_t) - log(y_{t-12})$$

Notice that:

$$\begin{aligned} g_t^y &= log(y_t) - log(y_{t-12}) \\ &= (log(y_t) - log(y_{t-1})) + (log(y_{t-1}) - log(y_{t-2})) + ... + (log(y_{t-11}) - log(y_{t-12})) \\ &= g_t^m + g_{t-1}^m + g_{t-2}^m + g_{t-3}^m + ... + +g_{t-11}^m \end{aligned}$$

We simply add up the monthly growth rates to obtain the annual growth rate.

Returning now to those monthly growth rate forecasts, we have already seen that these monthly rates fluctuate a lot and therefore may not be easy to predict.

To evaluate the forecasts we use the familiar criteria of Root Mean Squared Error and Mean Absolute Error and also the sum of the 24 forecast errors.

Evaluation criteria : RMSE and MAE SUM : sum of forecast errors $\sum_{t=1}^{24}(y_t - \hat{y}_t) RMSE = (\frac{1}{n_f}\sum_{i=1}^{n_f}(y_i - \hat{y}_i)^2)^{\frac{1}{2}} MAE = \frac{1}{n_f}\sum_{i=1}^{n_f}$

The numbers in the last column of this table show that adding the leading index does indeed reduce the out-of- sample forecast errors for the monthly growth rates. So the in-sample success is also confirmed in out-of-sample forecasting.

```
forecast <- forecast %>% mutate(error_ar3=(GRIP-ar3_fitted),error_adl36=(GRIP-adl36_fitted))
RMSE_ar3 <- sqrt((1/length(forecast$GRIP))*sum(forecast$error_ar3^2, na.rm = T))
MAE_ar3 <- (1/length(forecast$GRIP))*sum(abs(forecast$error_ar3), na.rm = T)
SUM_ar3 <- sum(forecast$error_ar3, na.rm = T)
RMSE_adl36 <- sqrt((1/length(forecast$GRIP))*sum(forecast$error_adl36^2, na.rm = T))
MAE_adl36 <- (1/length(forecast$GRIP))*sum(abs(forecast$error_adl36), na.rm = T)
SUM_adl36 <- sum(forecast$error_adl36, na.rm = T)
```

```
tests1 <- matrix(c(RMSE_ar3,RMSE_adl36,MAE_ar3,MAE_adl36,SUM_ar3,SUM_adl36),nrow = 3, byrow = T)
colnames(tests1) <- c("AR(3)","ADL(3,6)")
rownames(tests1) <- c("RMSE","MAE","SUM")
kable(tests1,booktabs = TRUE, digits = 5) %>%
  kable_styling() %>%
  footnote(general = "We prefer the closest to 0")
```

	AR(3)	ADL(3,6)
RMSE	0.00347	0.00318
MAE	0.00279	0.00225
SUM	-0.01563	-0.00674

Note:
We prefer the closest to 0

Selected model ADL(3,6) : $GRIP_t = 0.001 + 0.204 \cdot GRIP_{t-3} + 0.238 \cdot GRCLI_{t-6} + e_t (t_{b_3} = 3.2, t_{b_6} = 3.5, R^2 = 0.106)$

Application on diferent time range

```
prodtrain <- read_csv(
  "https://raw.githubusercontent.com/diego-eco/diego-eco.github.io/master/downloads/trainexer65.csv")
prodtrain$YEAR<- as.Date(paste0(prodtrain$YEAR, '-01-01'))
```

We filter with date data

```
estimation <- prodtrain %>% filter(YEAR <=  as.Date("2002-01-01"))
forecast <- prodtrain %>% filter(YEAR > as.Date("2002-01-01"))
```

- **produtrain** Data set on Industrial Production and the Composite Leading Index for the USA, yearly data 1960 to 2007 (Source: Conference Board, USA). Estimation period is 1960 to 2002. Forecast evaluation period is 2003 to 2007.
- CLI: Composite Leading Index (based on 10 leading indicators)
- IP: Industrial Production (index, seasonally adjusted)
- LOGCLI: logarithm of CLI
- LOGIP: logarithm of IP
- GRCLI: monthly growth rate of CLI,first difference of LOGCLI
- GRIP: monthly growth rate of IP,first difference of LOGIP

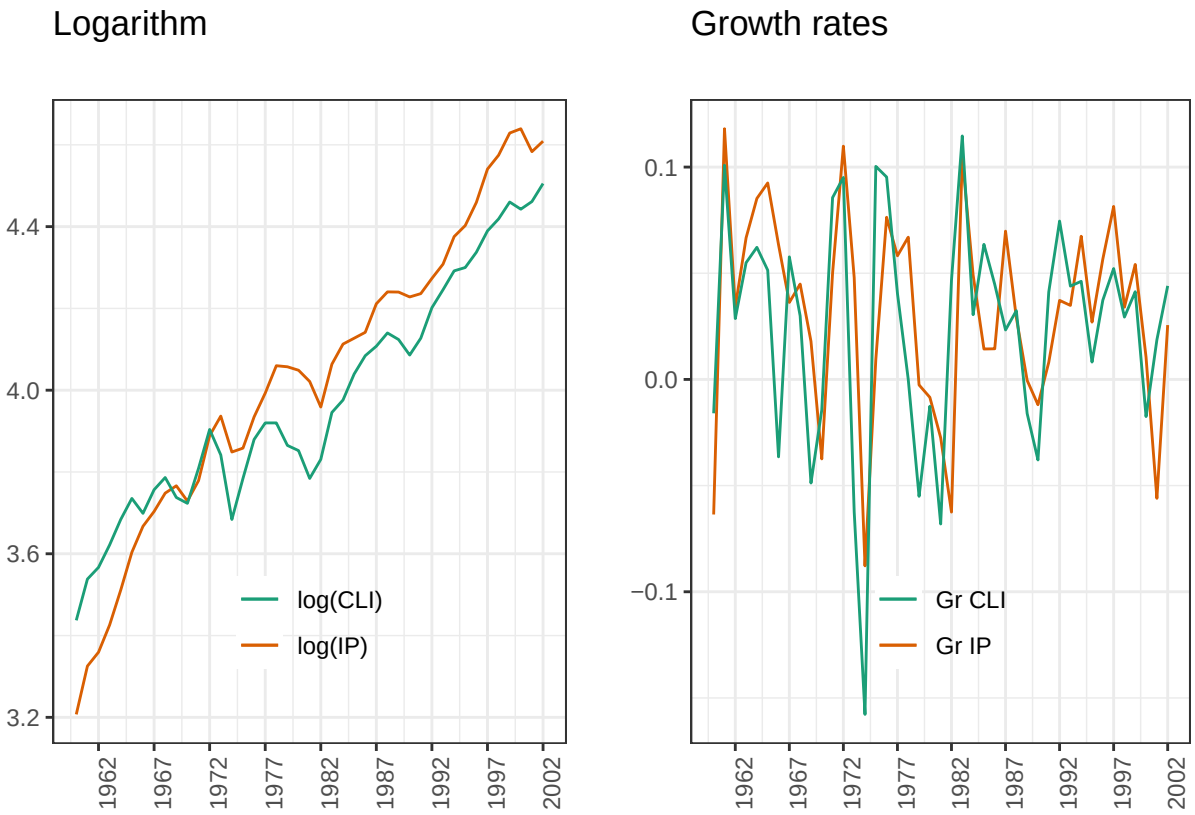
Note: In all questions, use 1960-2002 as estimation and test sample, and use 2003-2007 as hold-out forecast evaluation sample.

a) Graph analysis

Make time series plots of log(IP) and log(CLI), and also of the yearly growth rates GIP and GCLI. What conclusions do you draw from these plots?

```
plot_a <- ggplot(data=estimation, aes(x=YEAR)) +
  geom_line(aes(y=LOGIP,col="log(IP)")) +
  geom_line(aes(y=LOGCLI,col="log(CLI)")) +
  labs(x = "", y = "", title = "Logarithm",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .20),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

plot_b <- ggplot(data=estimation, aes(x=YEAR)) +
  geom_line(aes(y=GIP,col="Gr IP")) +
  geom_line(aes(y=GCLI,col="Gr CLI")) +
  labs(x = "", y = "", title = "Growth rates",
       subtitle = ("")) +
    scale_x_date(date_breaks = "5 year", date_labels = "%Y") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.5, .20),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")
```

```
grid.arrange(plot_a, plot_b, nrow = 1)
```



It seems that both of the log series have a trend, and perhaps cointegrated. The yearly growth rates seems rather stationary and have hevy fluctuations in the 70s.

b) Unit root

(i) Perform the Augmented Dickey-Fuller (ADF) test for log(IP). In the ADF test equation, include (among others) a constant (α), a deterministic trend term (βt), and two lags of $GIP = \Delta log(IP)$. Report the coefficient of $log(IP_{t-1})$ and its standard error and t-value, and draw your conclusion.

(ii) Perform a similar ADF test for $log(CLI)$.

```
# Set k=2 for the Dickey Fuller test with two lags and type 'trend'
adf_logip <- urca::ur.df(estimation$LOGIP, type = "trend", lags = 2)
summary(adf_logip)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -0.088301 -0.020459  0.006329  0.023546  0.060809
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.254164   0.338227   3.708  0.00072 ***
## z.lag.1     -0.354277   0.099364  -3.565  0.00107 **
## tt           0.009322   0.002855   3.265  0.00245 **
## z.diff.lag1  0.381659   0.139618   2.734  0.00976 **
## z.diff.lag2 -0.204374   0.141703  -1.442  0.15811
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03699 on 35 degrees of freedom
## Multiple R-squared:  0.3925, Adjusted R-squared:  0.323
## F-statistic: 5.653 on 4 and 35 DF,  p-value: 0.001283
##
##
## Value of test-statistic is: -3.5654 10.0788 7.2332
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

The value we find for the coefficient of log(GIP) is -0.35 with Std. Error 0.099 and t-value -3.56 $<$ -3.5 so log(IP) is not stationary.

```
# Set k=2 for the Dickey Fuller test with two lags and type 'trend'
adf_logcli <- urca::ur.df(estimation$LOGCLI, type = "trend", lags = 2)
summary(adf_logcli)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.129013 -0.026238  0.003642  0.029110  0.075559
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.880733   0.450014   1.957   0.0583 .
## z.lag.1     -0.245361   0.128956  -1.903   0.0653 .
## tt           0.005395   0.002818   1.914   0.0638 .
## z.diff.lag1  0.325877   0.154288   2.112   0.0419 *
## z.diff.lag2 -0.294723   0.158730  -1.857   0.0718 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04836 on 35 degrees of freedom
## Multiple R-squared:  0.2924, Adjusted R-squared:  0.2115
## F-statistic: 3.615 on 4 and 35 DF,  p-value: 0.01439
##
##
## Value of test-statistic is: -1.9027 4.8721 1.8483
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

The value we find for the coefficient of log(GIP) is -0.24 with Std. Error 0.128 and t-value -1.90 > -3.5 so log(CLI) is not stationary.

c) Cointegration

Perform the two-step Engle-Granger test for cointegration of the time series log(IP) and log(CLI). The secondstep regression is of the type $\Delta e_t = \alpha + -0.0068 e_{t-1} + \beta t + \rho e_{t-1} + \beta_1 \Delta e_{t-1} + \beta_2 \Delta e_{t-2} + res_t$ where e_t are the residuals of step 1. What is your conclusion?

```
step1 <- lm(LOGIP ~ LOGCLI, data=estimation)
summ(step1, digits = 3)
```

Observations	43
Dependent variable	LOGIP
Type	OLS linear regression

F(1,41)	617.020
R^2	0.938
Adj. R^2	0.936

	Est.	S.E.	t val.	p
(Intercept)	-1.021	0.204	-5.004	0.000
LOGCLI	1.271	0.051	24.840	0.000

Standard errors: OLS

```
# We add the residuals of step 1 into the dataset
estimation <- estimation %>% mutate(step1.res = resid(step1))
# Perform ADF on the residuals.
adf_coint <- urca::ur.df(estimation$step1.res, type = "trend", lags = 2)
summary(adf_coint)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
```

```
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.090576 -0.030342 -0.001155  0.036069  0.105169
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0240265  0.0182555   1.316   0.1967
## z.lag.1     -0.1778610  0.1012059  -1.757   0.0876 .
## tt          -0.0008875  0.0007283  -1.219   0.2311
## z.diff.lag1  0.1428094  0.1551645   0.920   0.3637
## z.diff.lag2 -0.3186368  0.1586728  -2.008   0.0524 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05093 on 35 degrees of freedom
## Multiple R-squared:  0.2616, Adjusted R-squared:  0.1772
## F-statistic: 3.099 on 4 and 35 DF,  p-value: 0.02767
##
##
## Value of test-statistic is: -1.7574 1.9343 2.8468
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

The value we find for the coefficient of e_{t-1} is -0.177 with Std. Error 0.101 and t-value -1.75 > -3.8 so > e_t is not stationary: Series are not cointegrated.

d) Granger causality

Perform two F-tests, one for the Granger causality of GIP for GCLI and the other for the Granger causality of GCLI for GIP. Include a constant and two lags of both variables in the test equations. Report the degrees of freedom and the numerical values of the two F-tests, and draw your conclusion.

```
# Test for causality: For CLI
# IP is Granger causal for CLI # Null: Restrictions true, means not granger causal

lm0 <- lm(GCLI ~ lag(GCLI,1) + lag(GCLI,2), data=estimation) # Restricted mode
lm1 <- lm(GCLI ~ lag(GCLI,1) + lag(GCLI,2) + lag(GIP,1) + lag(GIP,2), data=estimation) # Unrestricted m
# The rest is calculated automatically.
n <- nobs(lm1)
g <- length(lm1$coefficients)-length(lm0$coefficients)
k <- length(lm1$coefficients)
r2_0 <- summary(lm0)$r.squared
r2_1 <- summary(lm1)$r.squared
F_test <- ((r2_1-r2_0)/g)/((1-r2_1)/(n-k))
F_crit <- qf(0.95,g,(n-k))
```

```r
print(paste("The degrees of freedom are n = ",n," (n-k) = ",n-k))
```

```
## [1] "The degrees of freedom are n =  41  (n-k) =  36"
```

```r
print(paste("The F test value is ",round(F_test,3)))
```

```
## [1] "The F test value is  2.67"
```

```r
print(paste("The F critical value at 0.95% is ",round(F_crit,3)))
```

```
## [1] "The F critical value at 0.95% is  3.259"
```

```r
if(F_test<F_crit){
  print(paste("We do not reject H_0 at 0.95%"))
} else {
  print(paste("We reject H_0 at 0.95% in favor of H_1"))
}
```

```
## [1] "We do not reject H_0 at 0.95%"
```

For CLI: F-test $= 2.67 < 3.26 \rightarrow H_0$ not rejected, so the Null Hypothesis is not rejected.

IP is NOT Granger causal for CLI

```r
# Test for causality: For IP
#CLI is Granger causal for IP # Null: Restrictions true, means not granger causal

lm0 <- lm(GIP ~ lag(GIP,1) + lag(GIP,2), data=estimation) # Restricted mode
lm1 <- lm(GIP ~ lag(GIP,1) + lag(GIP,2) + lag(GCLI,1) + lag(GCLI,2), data=estimation) # Unrestricted mo
# The rest is calculated automatically.
n <- nobs(lm1)
g <- length(lm1$coefficients)-length(lm0$coefficients)
k <- length(lm1$coefficients)
r2_0 <- summary(lm0)$r.squared
r2_1 <- summary(lm1)$r.squared
F_test <- ((r2_1-r2_0)/g)/((1-r2_1)/(n-k))
F_crit <- qf(0.95,g,(n-k))
print(paste("The degrees of freedom are n = ",n," (n-k) = ",n-k))
```

```
## [1] "The degrees of freedom are n =  41  (n-k) =  36"
```

```r
print(paste("The F test value is ",round(F_test,3)))
```

```
## [1] "The F test value is  11.747"
```

```r
print(paste("The F critical value at 0.95% is ",round(F_crit,3)))
```

```
## [1] "The F critical value at 0.95% is  3.259"
```

```r
if(F_test<F_crit){
  print(paste("We do not reject H_0 at 0.95%"))
} else {
  print(paste("We reject H_0 at 0.95% in favor of H_1"))
}
```

```
## [1] "We reject H_0 at 0.95% in favor of H_1"
```

For IP: F-test $= 11.75 > 3.26 \rightarrow H_0$ rejected

CLI is Granger causal for IP. Indeed, as past CLI helps to predict current IP yearly growth rate. So is indeed a leading index.

e) AR models

Show that the coefficients of both lags in an AR(2) model for GIP are insignificant. Show also that even the slope coefficient in the AR(1) model is insignificant. Make two forecasts for GIP for the five years from 2003-2007, one from the AR(1) model and another from the simple model $GIP_t = \alpha + \epsilon_t$

```
ar2 <- lm(GIP ~ lag(GIP,1) + lag(GIP,2), data=estimation)
summ(ar2, digits = 3)
```

Observations	41 (2 missing obs. deleted)
Dependent variable	GIP
Type	OLS linear regression

F(2,38)	2.431
R^2	0.113
Adj. R^2	0.067

	Est.	S.E.	t val.	p
(Intercept)	0.031	0.009	3.301	0.002
lag(GIP, 1)	0.247	0.147	1.680	0.101
lag(GIP, 2)	-0.235	0.146	-1.612	0.115

Standard errors: OLS

```
ar1 <- lm(GIP ~ lag(GIP,1), data=estimation)
summ(ar1, digits = 3)
```

Observations	42 (1 missing obs. deleted)
Dependent variable	GIP
Type	OLS linear regression

F(1,40)	0.566
R^2	0.014
Adj. R^2	-0.011

	Est.	S.E.	t val.	p
(Intercept)	0.030	0.009	3.511	0.001
lag(GIP, 1)	0.112	0.149	0.752	0.456

Standard errors: OLS

```
ar0 <- lm(GIP ~ 1, data=estimation)
summ(ar0, digits = 3)
```

Observations	43
Dependent variable	GIP
Type	OLS linear regression

```
# We add the fitted values of both models to the forecast dataset
forecast <- forecast %>% mutate(ar1_fitted=predict(ar1,newdata =  forecast))
```

	Est.	S.E.	t val.	p
(Intercept)	0.031	0.007	4.282	0.000

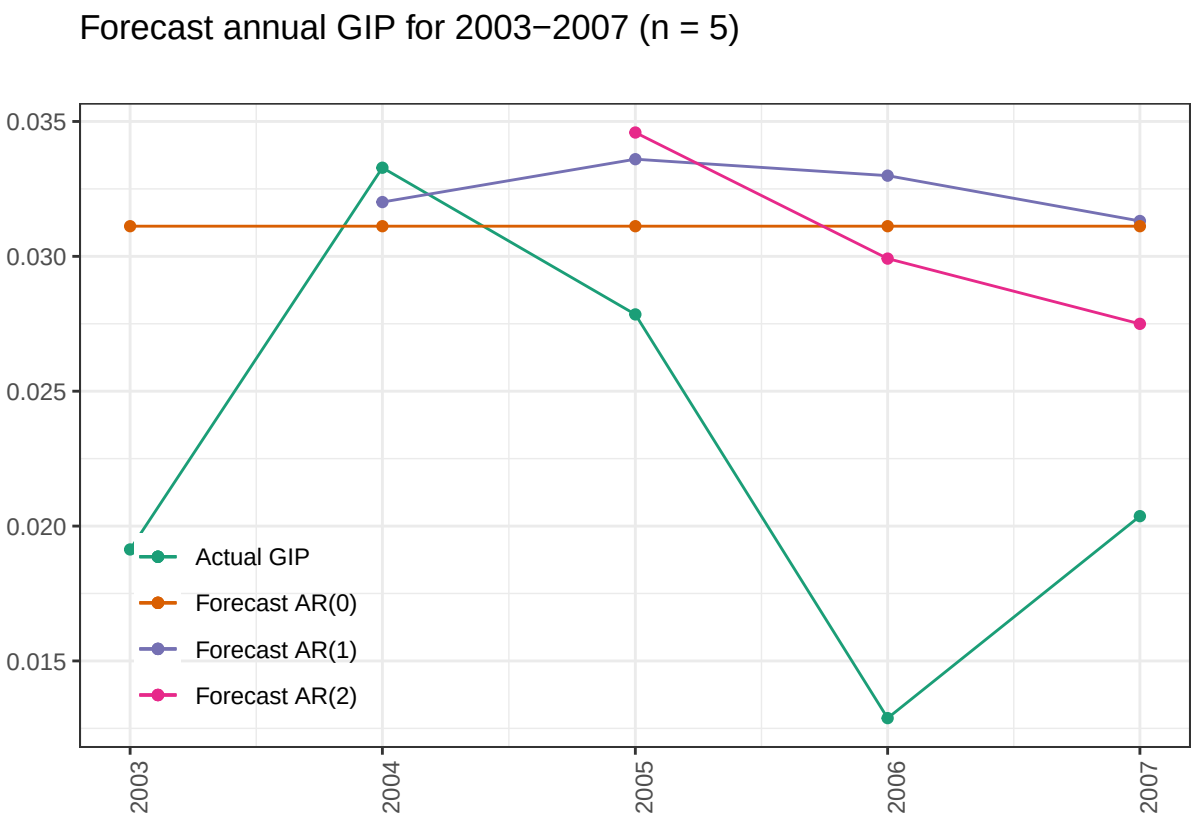
Standard errors: OLS

```
forecast <-  forecast %>% mutate(ar0_fitted=predict(ar0,newdata =  forecast))
forecast <-  forecast %>% mutate(ar2_fitted=predict(ar2,newdata =  forecast))
```

The coefficients in model AR(2) and AR(1) are not signifficant. The last model with only constant term has a mean value 0.031 and it is statistically significant at 99% over the period 1960 to 2002.

We compute the forecast for the models:

```
forecast_plot <- ggplot(data=forecast, aes(x=YEAR)) +
  geom_line(aes(y=GIP,col="Actual GIP")) + geom_point(aes(y=GIP,col="Actual GIP")) +
  geom_line(aes(y=ar1_fitted,col="Forecast AR(1)")) + geom_point(aes(y=ar1_fitted,col="Forecast AR(1)")) +
  geom_line(aes(y=ar0_fitted,col="Forecast AR(0)")) + geom_point(aes(y=ar0_fitted,col="Forecast AR(0)")) +
  geom_line(aes(y=ar2_fitted,col="Forecast AR(2)")) + geom_point(aes(y=ar2_fitted,col="Forecast AR(2)")) +
  labs(x = "", y = "", title = "Forecast annual GIP for 2003-2007 (n = 5)",
       subtitle = ("")) +
  theme_bw() +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.15, .20),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

forecast_plot
```



f) ADL models

Estimate the ADL(2,2) model $GIP_t = \alpha + \beta_1 GIP_{t-1} + \beta_2 GIP_{t-2} + \gamma_1 GCLI_{t-1} + \gamma_2 GCLI_{t-2} + \epsilon_t$, and show by means of an F-test that the null hypothesis that $\beta_1 = \beta_2 = \gamma_2 = 0$. Then estimate the ADL(0,1) model $GIP_t = \alpha + \gamma_1 GCLI_{t-1} + \epsilon_t$ and use this model to forecast GIP for the five years from 2003-2007.

```
adl22 <- lm(GIP ~ lag(GIP,1) + lag(GIP,2) + lag(GCLI,1) + lag(GCLI,2), data=estimation)
summ(adl22)
```

Observations	41 (2 missing obs. deleted)
Dependent variable	GIP
Type	OLS linear regression

F(4,36)	7.78
R^2	0.46
Adj. R^2	0.40

	Est.	S.E.	t val.	p
(Intercept)	0.02	0.01	1.93	0.06
lag(GIP, 1)	-0.22	0.19	-1.16	0.25
lag(GIP, 2)	0.25	0.18	1.37	0.18
lag(GCLI, 1)	0.72	0.15	4.80	0.00
lag(GCLI, 2)	-0.17	0.17	-1.02	0.31

Standard errors: OLS

```
# We add the fitted values of both models to the forecast dataset
forecast <- forecast %>% mutate(adl22_fitted=predict(adl22,newdata =  forecast))
```

We test the null $\beta_1 = \beta_2 = \gamma_2 = 0$

```
# F test
lm0 <- lm(GIP ~ lag(GCLI,1), data=estimation[-1,]) # Restricted mode
lm1 <- lm(GIP ~ lag(GIP,1) + lag(GIP,2) + lag(GCLI,1) + lag(GCLI,2), data=estimation[-1,]) # Unrestrict

### Important: The unrestricted model lm1 has 2 lagas and therefore is estimated 1962-2002
### The restricted model lm0 has 1 lag and therefore can be estimated 1961-2002
### But the F test requires the same sample for both models, thats why we remove the first row

# The rest is calculated automatically.
n <- nobs(lm1)
g <- length(lm1$coefficients)-length(lm0$coefficients)
k <- length(lm1$coefficients)
r2_0 <- summary(lm0)$r.squared
r2_1 <- summary(lm1)$r.squared
F_test <- ((r2_1-r2_0)/g)/((1-r2_1)/(n-k))
F_crit <- qf(0.95,g,(n-k))
print(paste("The degrees of freedom are n = ",n," (n-k) = ",n-k))
```

```
## [1] "The degrees of freedom are n =  40  (n-k) =  35"
```

```
print(paste("The F test value is ",round(F_test,3)))
```

```
## [1] "The F test value is  1.585"
```

```r
print(paste("The F critical value at 0.95% is ",round(F_crit,3)))
```

```
## [1] "The F critical value at 0.95% is  2.874"
```

```r
if(F_test<F_crit){
  print(paste("We do not reject H_0 at 0.95%"))
} else {
  print(paste("We reject H_0 at 0.95% in favor of H_1"))
}
```

```
## [1] "We do not reject H_0 at 0.95%"
```

As the null hypothesis is not rejected, we keep the simple model ADL(0,1) model $GIP_t = \alpha + \gamma_1 GCLI_{t-1} + \epsilon_t$.

We estimate the ADL(0,1) model $GIP_t = \alpha + \gamma_1 GCLI_{t-1} + \epsilon_t$

```r
adl01 <- lm(GIP ~ lag(GCLI,1), data=estimation)
summ(adl01)
```

Observations	42 (1 missing obs. deleted)
Dependent variable	GIP
Type	OLS linear regression

F(1,40)	18.64
R^2	0.32
Adj. R^2	0.30

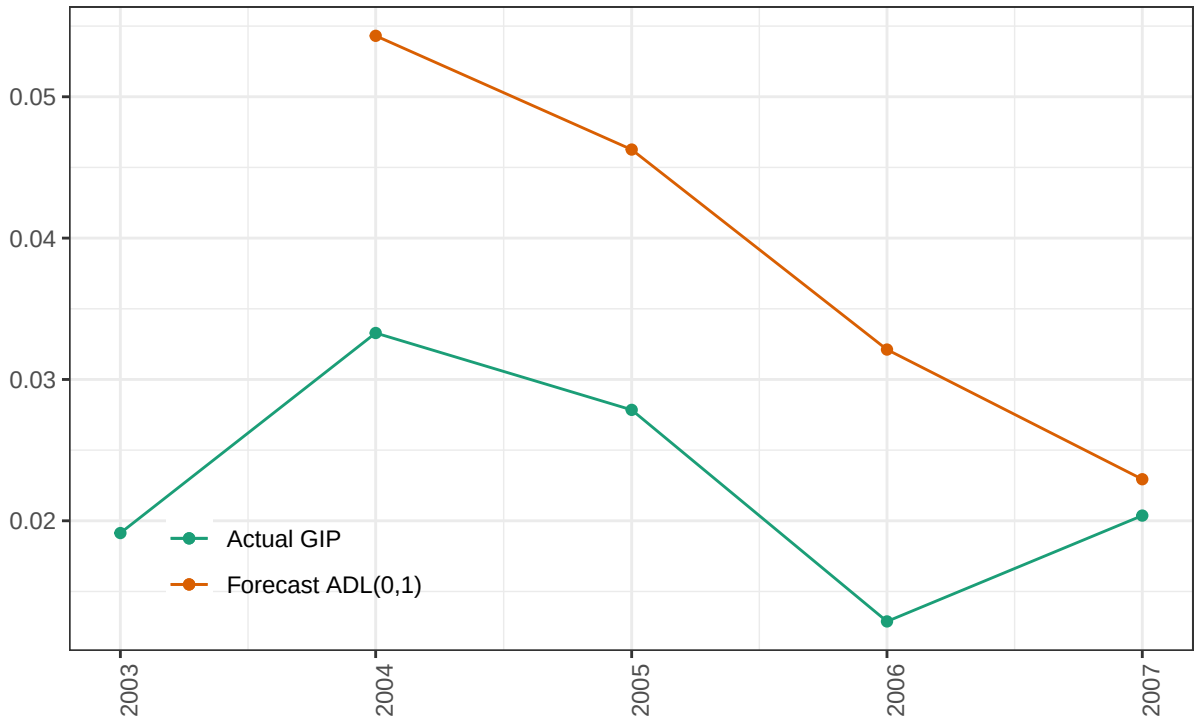
	Est.	S.E.	t val.	p
(Intercept)	0.02	0.01	3.40	0.00
lag(GCLI, 1)	0.47	0.11	4.32	0.00

Standard errors: OLS

```r
# We add the fitted values of both models to the forecast dataset
forecast <-  forecast %>% mutate(adl01_fitted=predict(adl01,newdata =  forecast))
```

We use this model to forecast GIP for the five years from 2003-2007.

```r
forecast_plot <- ggplot(data=forecast, aes(x=YEAR)) +
  geom_line(aes(y=GIP,col="Actual GIP")) + geom_point(aes(y=GIP,col="Actual GIP")) +
  geom_line(aes(y=adl01_fitted,col="Forecast ADL(0,1)")) +
  geom_point(aes(y=adl01_fitted,col="Forecast ADL(0,1)")) +
  #geom_line(aes(y=adl22_fitted,col="Forecast ADL(2,2)")) +
  #geom_point(aes(y=adl22_fitted,col="Forecast ADL(2,2)")) +
  labs(x = "", y = "", title = "Forecast annual GIP for 2003-2007 (n = 5)",
       subtitle = ("")) +
  theme_bw() +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.20, .15),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")

forecast_plot
```



g) Out of sample forecast

Compare the three series of forecasts of parts (e) and (f) by computing their values of the root mean squared error (RMSE), mean absolute error (MAE), and the sum of the forecast errors (SUM). Check that it seems quite difficult to forecast the IP growth rates for 2003 to 2007 from models estimated from 1960 to 2002. Can you think of possible reasons why this is the case?

```
forecast <- forecast %>% mutate(error_ar2=(GIP-ar2_fitted),
                                error_ar1=(GIP-ar1_fitted),
                                error_ar0=(GIP-ar0_fitted),
                                error_adl22=(GIP-adl22_fitted),
                                error_adl01=(GIP-adl01_fitted))

RMSE_ar2 <- sqrt((1/length(forecast$GIP))*sum(forecast$error_ar2^2, na.rm = T))
RMSE_ar1 <- sqrt((1/length(forecast$GIP))*sum(forecast$error_ar1^2, na.rm = T))
RMSE_ar0 <- sqrt((1/length(forecast$GIP))*sum(forecast$error_ar0^2, na.rm = T))
RMSE_adl22 <- sqrt((1/length(forecast$GIP))*sum(forecast$error_adl22^2, na.rm = T))
RMSE_adl01 <- sqrt((1/length(forecast$GIP))*sum(forecast$error_adl01^2, na.rm = T))


MAE_ar2 <- (1/length(forecast$GIP))*sum(abs(forecast$error_ar2), na.rm = T)
MAE_ar1 <- (1/length(forecast$GIP))*sum(abs(forecast$error_ar1), na.rm = T)
MAE_ar0 <- (1/length(forecast$GIP))*sum(abs(forecast$error_ar0), na.rm = T)
MAE_adl22 <- (1/length(forecast$GIP))*sum(abs(forecast$error_adl22), na.rm = T)
MAE_adl01 <- (1/length(forecast$GIP))*sum(abs(forecast$error_adl01), na.rm = T)

SUM_ar3 <- sum(forecast$error_ar3, na.rm = T)
SUM_ar2 <- sum(forecast$error_ar2, na.rm = T)
```

```
SUM_ar1 <- sum(forecast$error_ar1, na.rm = T)
SUM_ar0 <- sum(forecast$error_ar0, na.rm = T)
SUM_adl22 <- sum(forecast$error_adl22, na.rm = T)
SUM_adl01 <- sum(forecast$error_adl01, na.rm = T)
```

```
tests1 <- matrix(c(RMSE_ar2,RMSE_ar1,RMSE_ar0,RMSE_adl22,RMSE_adl01,
                   MAE_ar2,MAE_ar1,MAE_ar0,MAE_adl22,MAE_adl01,
                   SUM_ar2,SUM_ar1,SUM_ar0,SUM_adl22,SUM_adl01),nrow = 3, byrow = T)
colnames(tests1) <- c("AR(2)","AR(1)","AR(0)","ADL(2,2)","ADL(0,1)")
rownames(tests1) <- c("RMSE","MAE","SUM")
kable(tests1,booktabs = TRUE, digits = 5) %>%
  kable_styling() %>%
  footnote(general = "We prefer the closest to 0")
```

	AR(2)	AR(1)	AR(0)	ADL(2,2)	ADL(0,1)
RMSE	0.00879	0.01057	0.01102	0.00758	0.01522
MAE	0.00618	0.00762	0.00928	0.00516	0.01225
SUM	-0.03091	-0.03554	-0.04207	-0.02175	-0.06125

Note:
We prefer the closest to 0

Let's see how our fitted models perform for the whole sample period:

In 2003-2007 the GIP has relatevely flat as GCLI had still high variance. We can see that it is not possible to predict the growth rate of IP in 2003-2007, simply the mean over 1960-2002 is best: It seems that the economic structure has changed over the millenium term. And after 2003 there has been a turmoil worldwide, specially in the USA. The good performance over the previous years breakdown in 2003-2007 period.

```
prodtrain <-  prodtrain %>% mutate(adl01_fitted=predict(adl01,newdata =  prodtrain))
prodtrain <- prodtrain %>% mutate(adl22_fitted=predict(adl22,newdata =  prodtrain))
prodtrain <- prodtrain %>% mutate(ar1_fitted=predict(ar1,newdata =  prodtrain))
prodtrain <-  prodtrain %>% mutate(ar0_fitted=predict(ar0,newdata =  prodtrain))
prodtrain <-  prodtrain %>% mutate(ar2_fitted=predict(ar2,newdata =  prodtrain))

forecast_plot <- ggplot(data=prodtrain, aes(x=YEAR)) +
  geom_line(aes(y=GIP,col="Actual GIP")) + geom_point(aes(y=GIP,col="Actual GIP")) +
  geom_line(aes(y=adl01_fitted,col="Forecast ADL(0,1)")) +
  geom_point(aes(y=adl01_fitted,col="Forecast ADL(0,1)")) +
  geom_line(aes(y=adl22_fitted,col="Forecast ADL(2,2)")) +
  geom_point(aes(y=adl22_fitted,col="Forecast ADL(2,2)")) +
  #geom_line(aes(y=ar1_fitted,col="Forecast AR(1)")) + geom_point(aes(y=ar1_fitted,col="Forecast AR(1)"
  #geom_line(aes(y=ar0_fitted,col="Forecast AR(0)")) + geom_point(aes(y=ar0_fitted,col="Forecast AR(0)"
  #geom_line(aes(y=ar2_fitted,col="Forecast AR(2)")) + geom_point(aes(y=ar2_fitted,col="Forecast AR(2)"
  geom_vline(xintercept = as.numeric(as.Date("2003-01-10"))) +
  labs(x = "", y = "", title = "All models Forecast annual GIP for 1960-2007 (n = 5)",
       subtitle = ("Estimation / Forecast cut off in 2003")) +
  theme_bw() +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        legend.position = c(.70, .20),
        legend.background = element_rect(fill = "transparent")) +
  scale_color_brewer(name= NULL, palette = "Dark2")
```

```
forecast_plot
```