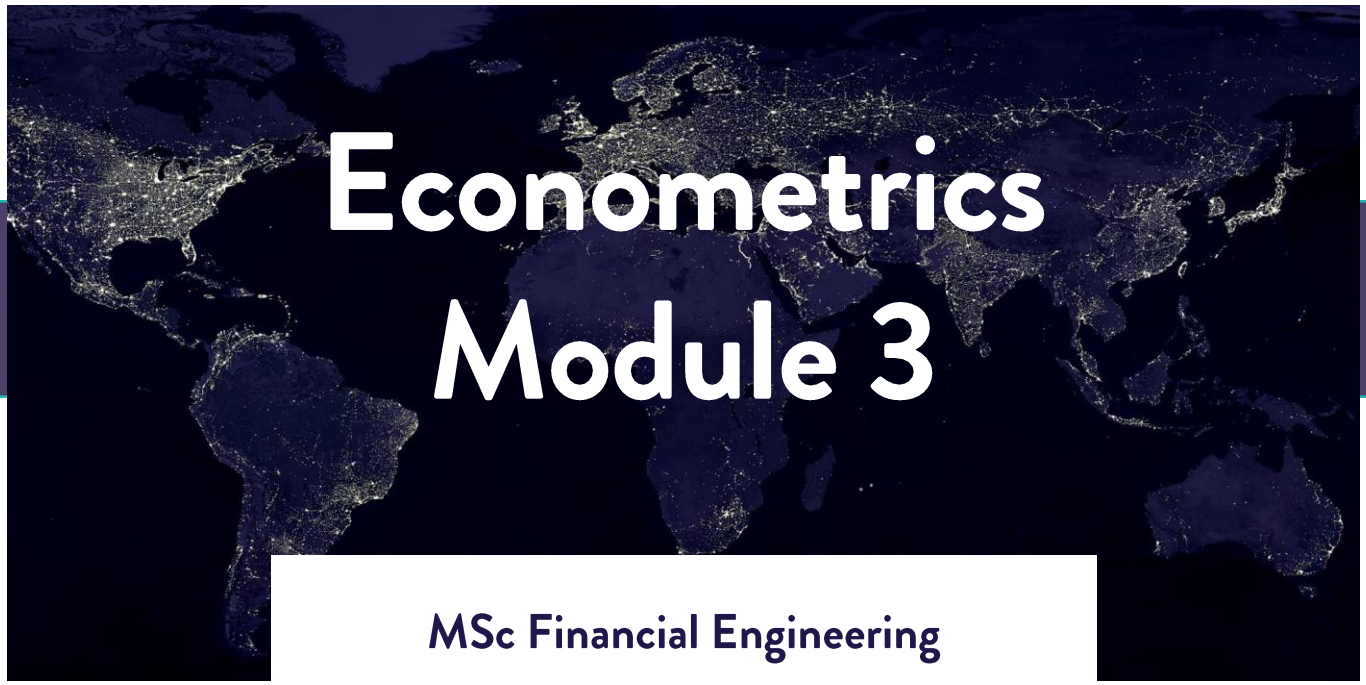
WORLDQUANT UNIVERSITY



# Econometrics Module 3

## MSc Financial Engineering

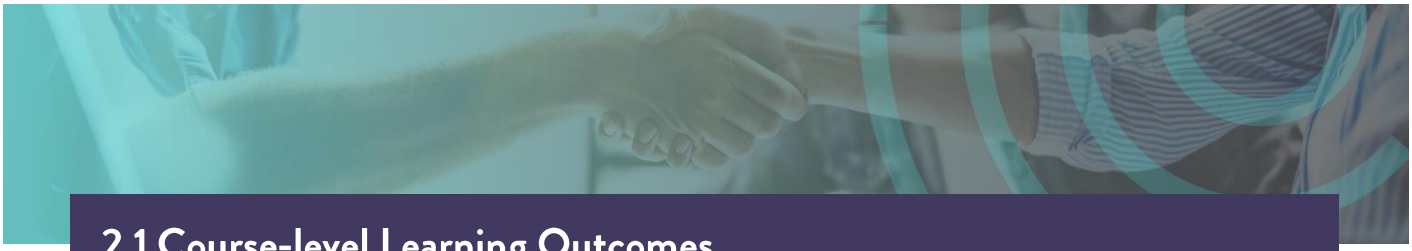
# Table of Contents

# 1. Brief

This document contains the core content for Module 3 of Econometrics, entitled Univariate Time Series Models. It consists of four sets of supplementary notes and two lecture video transcripts.

# 2. Course Context

Econometrics is the second course presented in the WorldQuant University (WQU) Master of Science in Financial Engineering (MScFE) program. In this course, you will apply statistical techniques to the analysis of econometric data. The course starts with an introduction to the R statistical programming languages that you will use to build econometric models, including multiple linear regression models, time series models, and stochastic volatility models. You will learn to develop programs using the R language, solve statistical problems, and understand value distributions in modeling extreme portfolio and basic algorithmic trading strategies. The course concludes with a review on applied econometrics in finance and algorithmic trading.



## 2.1 Course-level Learning Outcomes

**Upon completion of the Econometrics course, you will be able to:**

1. Write programs using the R language.
2. Use R packages to solve common statistical problems.
3. Formulate a generalized linear model and fit the model to data.
4. Use graphic techniques to visualize multidimensional data.
5. Apply multivariate statistical techniques (PCA, factor analysis, etc.) to analyze multidimensional data.
6. Fit a time series model to data.
7. Fit discrete-time volatility models.
8. Understand and apply filtering techniques to volatility modeling.
9. Understand the use of extreme value distributions in modeling extreme portfolio returns.
10. Define some common risk measures like VaR and Expected Shortfall.
11. Define and use copulas in risk management.
12. Implement basic algorithmic trading strategies.

## 2.2 Module Breakdown

**The Econometrics course consists of the following one-week modules:**

1. Basic Statistics
2. Linear Models
3. Univariate Time Series Models
4. Univariate Volatility Modeling
5. Multivariate Time Series Analysis
6. Introduction to Risk Management
7. Algorithmic Trading

# 3. Module 3

# Univariate Time Series Models

This module studies the applications of univariate, single equation time series models. Using these applications, this module attempts to explain and forecast a variable such as a stock price, stock returns, or inflation using only its own history.

## 3.1 Module-level Learning Outcomes

**After completing this module, you will be able to:**

1. Describe the main components of a time series model.
2. Fit AR, MA, and ARMA models to data.
3. Perform unit root tests.
4. Fit ARIMA models.
5. Fit other time series models.

# 3.2 Transcripts and Notes

## 3.2.1 Notes: AR Process

The **Autoregressive Model** of order 1, or AR(1) is a stochastic process in discrete time that is usually equally spaced. This process is used in statistical calculations in which future values are estimated based on a weighted sum of past values. An autoregressive process operates under the premise that past values have an effect on current values. A process considered AR(1) is the first order process, meaning that the current value is based on the immediately preceding value. An AR(2) process has the current value based on the previous two values.

Autoregressive processes are used by investors in technical analysis. Trends, moving averages and regressions consider past prices in an effort to create forecasts of future price movement. One drawback to this type of analysis is the past prices won't always be the best predictor of future movements, especially if the underlying fundamentals of a company have changed.

### Forecasting an AR(1)

The estimated AR(1) model for a series with $T$ observations is

$$y*(t) = \alpha* + \beta* y(t-1).$$

We use the value of $y$ in $T+1$ period to predict

$$y*(T+1) = \alpha* + \beta* y(T).$$

In the first instance we already know $y(T)$.

Example:

Suppose $\alpha* = .2\ \beta* = .5\ and\ Y(t) = 1.2$

$y*(t) = .2 + .5\,y(t-1)$ then

$$y*(T+1) = \alpha* + \beta*$$
$$y(T+1) = .2 + .5(1.2) = .8.$$

## Desirable forecasting

- Unbiased: We want $E[y*(t)] = y(t)$ or the expected value of the forecast error

$$E(y(t) - y*(t)) = E[u*(t)] = 0.$$

- Models with small average errors are preferred.
- Models with small sums of squared errors are preferred.
- No pattern in $u*(t)$ series. Error terms are just random according to the plot.

**Type of forecast errors**

- Mean error

$$\frac{\sum u*(t)}{T}$$

  where $T$ is the number of periods forecast.

- Mean absolute (value) error: $\sum$

$$\frac{\sum|u*(t)|}{T}$$

- Mean squared error of forecast:

$$\frac{\sum u*(t)^2}{T.}$$

## Thiel's U statistic

**Theil's U statistic** is a relative accuracy measure that compares the forecasted results with the results of forecasting naively based on last period. It also squares the deviations to give more weight to large errors:

Theil's U Statistic:

$$\sqrt{\frac{\sum_{t=1}^{n-1}\left(\frac{Y_{t+1}^{*}-Y_t}{Y_t}\right)^2}{\sum_{t=1}^{n-1}\left(\frac{Y_{t+1}-Y_t}{Y_t}\right)^2}},$$

where $Y_t$ is the actual value of a point for a given time period, $t$ is the forecast period, and $Y_t*$ is the forecasted value.

**Theil's U statistic interpretation:**

- Less than 1 – The forecasting technique is better than guessing.
- Equal to 1 – The forecasting technique is about as good as guessing.
- More than 1 – The forecasting technique is worse than guessing.

## 3.2.2 Notes: Moving Average (MA) Process

The **Moving Average process**, or MA process, is used in time series analysis for modeling univariate time series. An MA process shows that the current values of the time series depends on the previous (unobserved) random shocks.

MA(q) process can be written as following:

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q},$$

where

- $\mu$ is the mean of the series,
- $\theta_1, \ldots, \theta_q$ are the model parameters,
- $\varepsilon_t, \varepsilon_{t-1}, \ldots, \varepsilon_{t-q}$ is the white noise error terms, and
- $q$ is the order of the MA model.

If we consider $B$ as the lag operator, then MA($q$) process can be written as the following:

$$X_t = \mu + (1 + \theta_1 B + \ldots + \theta_q B^q)\varepsilon_t.$$

## 3.2.3 Notes: ARIMA

**ARIMA** stands for autoregressive integrated moving average model. ARIMA $(p, d, q)$ has both the AR and MA components in the equation.

- $p$ – the order of the autoregressive model.
- $d$ – the order required to make the variable stationary.
- $q$ – the order of the moving average model.

$$y_t = a_0 + \sum_{i=1}^{p} a_i \, y_{t-i} + \sum_{i=0}^{q} \beta_i \, \varepsilon_{t-i} \quad \text{ARMA}(p, q)$$

$$\psi(L)(y_t - \mu) = \Theta(L)\varepsilon_t \text{ - ARMA}(m, n)$$

$$\psi(L) = 1 - \sum_{i=1}^{m} \psi_i \, L^i$$

$$\Theta(L) = 1 - \sum_{i=1}^{n} \theta_i \, L^i.$$

### Lag operators

$L$ – lag operator

$$L^i y_t = y_{t-i}$$

**Properties:**

$$Lc = c$$

$$(L^i + L^j) y_t = L^I y_t + L^j y_t = y_{t-j}$$

$$L^i L^j y_t = L^i (L^j y_t) = L^i L^j y_{t-j} = y_{t-i-j}$$

$$L^{-i}y_t = y_{t+i}$$

For $|a| > 1$ the infinite sum $(1 + aL + a^2L^2 + \cdots)y_t = \frac{y_t}{(1-aL)}$

For $|a| > 1$ the infinite sum $(1 + aL + a^2L^2 + \cdots)y_t = \frac{-aLy_t}{(1-aL)}$

## Simulating an ARMA(1,1) process

Books and research articles often use ARMA and ARIMA concepts to describe this type of univariate model. Are ARMA and ARIMA the same model? Well, they refer to the same concept, the single difference being the information regarding the order of integration of time series.

Example:
ARIMA(1,1,1)

In this case $p = 1, d = 1$ (the order of integration) and $q = 1$.

$d = 1$ means that we have to apply the first difference to the time series in order to obtain stationarity. We introduce the time series in the model.

So basically ARIMA(1,1,1) is the same with ARMA(1,1), the difference being the fact that the additional parameter in ARIMA ($d$ which shows the order of integration) indicates what type of updates we have to perform to the data before being introduced in the ARMA model. It offers us additional information regarding the data.

We simulate the following process:

- $x_t = \alpha x_{t-1} + \varepsilon_t + \beta\varepsilon_{t-1}$
- $x_t$ – simulated financial variable
- $\varepsilon_t,\ \varepsilon_{t-1}$ – random shocks
- $\alpha, \beta$ – parameters

R code:

```
set.seed(1)
simulated_x<- arima.sim(n=2000, model=list(ar=0.45, ma=-0.45))
plot(simulated_x)
```

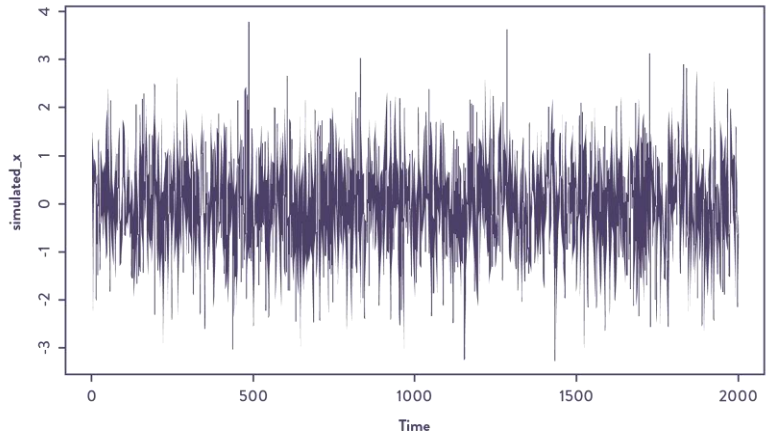



Figure 1: Simulated ARMA(1,1) process for $\alpha = 0.45$ and $\beta = 0.45$

R code:

```
acf(simulated_x)
```

ACF chart is the following:

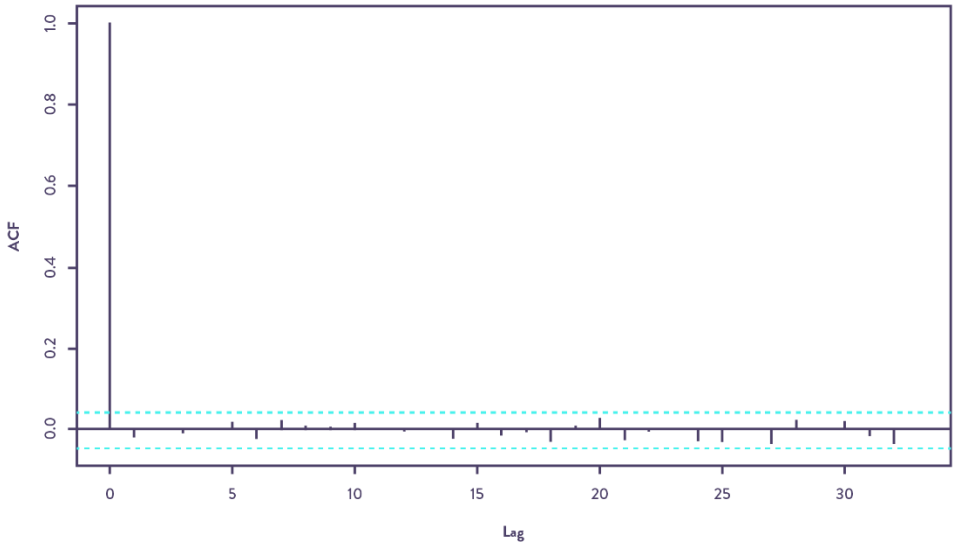



Figure 2: Series simulated_x

R code:

```
pacf(simulated_x)
```

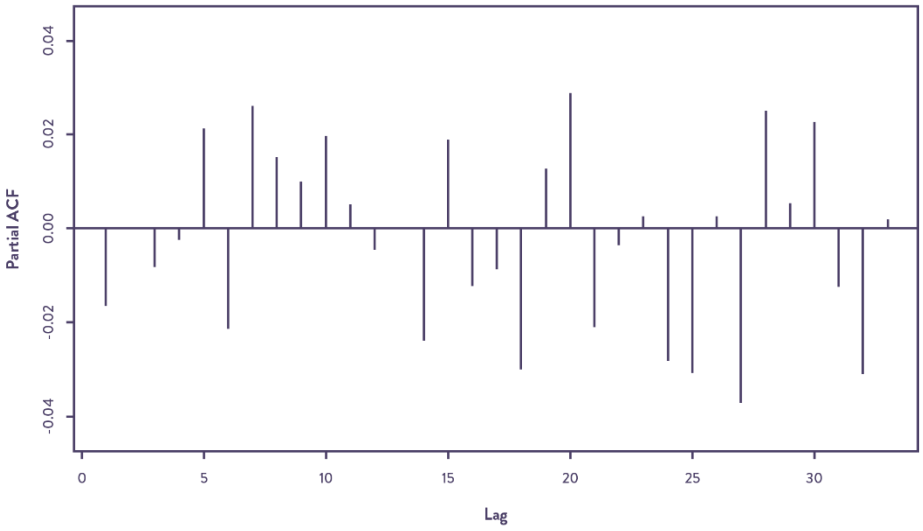



Figure 3: Series simulated_x

## 3.2.4 Notes: The Box-Jenkins Method

Suppose we want to forecast house prices in the largest cities of Australia using an ARMA model. The million-dollar question, is what model should we use? Are house prices in Australia following an AR process, an MA process or an ARMA process? The **Box-Jenkins method** can provide us answers to these questions. The Box-Jenkins method is applied in ARMA/ARIMA models for finding the best fit.

### Auto-correlations and partial auto-correlations

We use the auto-correlation and partial auto-correlation function to determine $p$ orders and $q$ orders and confirm the differencing that was suggested by the plot. The auto-correlation at lag $k$, $ACF(k)$, is the linear Pearson correlation between observations $k$ time periods (lags) apart. If the $ACF(k)$ differs significantly from zero, the serial dependence among the observations must be included in the ARIMA model. Like the $ACF(k)$, the partial auto-correlation at lag $k$, or $PACF(k)$, measures the correlation among observations $k$ lags apart. However, the $PACF(k)$ removes, or "partials out," all intervening lags.

In general, for *autoregressive* processes $[ARIMA(p, 0, 0)]$

- The ACF declines exponentially
- The PACF spikes on the first $p$ lag

By contrast, for *moving average* processes $[\text{ARIMA}(0, 0, q)]$

- The ACF spikes on the first $q$ lag
- The PACF declines exponentially

Mixed processes $[ARIMA(p, d, q)]$ decline on both ACF and the PACF. Finally, if ACF or PACF declines slowly (i.e., has auto-correlations with $t$ values$> 1.6$ for more than five or six consecutive lags), the process is probably not stationary and

therefore should be differenced. Usually, a large number of significant auto-correlations or partial auto-correlations indicates non-stationarity and a need for further differencing.

### Identifying an AR process

- If PACF displays a sharp cut-off while ACF decays more slowly (i.e., has significant spikes at higher lags), we say that the series displays an "AR signature".
- The lag at which the PACF cuts off is the indicated number of AR terms.

Correlogram – plot of the auto-correlation function (ACF).
Partial correlogram – plot of the partial auto-correlation function (PACF).

| Steps in Box-Jenkins methodology | Description |
|---|---|
| 1. Identification of the model | Choosing $p$, $d$ and $q$ parameters using the correlogram and the partial correlogram. |
| 2. Parameter estimation of the chosen model | The parameters are estimated using Maximum Likelihood Estimation (MLE) or non-linear least squares. |
| 3. Diagnostic checking | Are the rsiduals normally disturbed? If they are, we go to step 4. If not, we go back to step 1. |
| 4. Forecasting | Forecast house prices in Australia's largest cities using the ARMA model. |

## Theoretical patterns of ACF and PACF

| Type of model | Typical pattern of ACF | Typical pattern of PACF |
|---|---|---|
| AR($p$) | Decays exponentially or with a damped sine wave pattern or both | Significant spikes through lags $p$ |
| MA($q$) | Significant spikes through lags $q$ | Declines exponentially |
| ARMA($p$, $q$) | Exponential decay | Exponential decay |

**Source: Damodar Gujarati – "Basic Econometrics"**

## ACF and PACF examples

**Case 1**

The following ACF and PACF graphs suggests ARIMA(1,0,0).

The lag at which the PACF cuts off is the indicated number of AR terms.

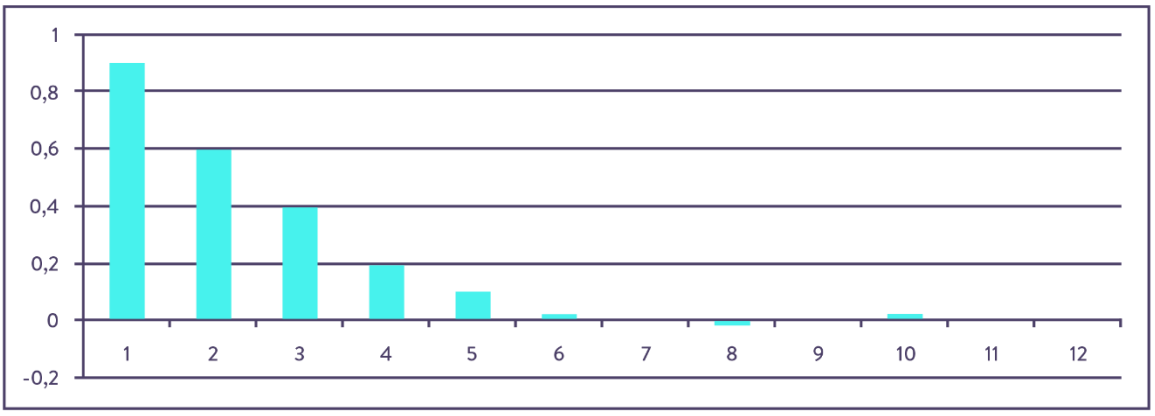



Figure 4: ACF

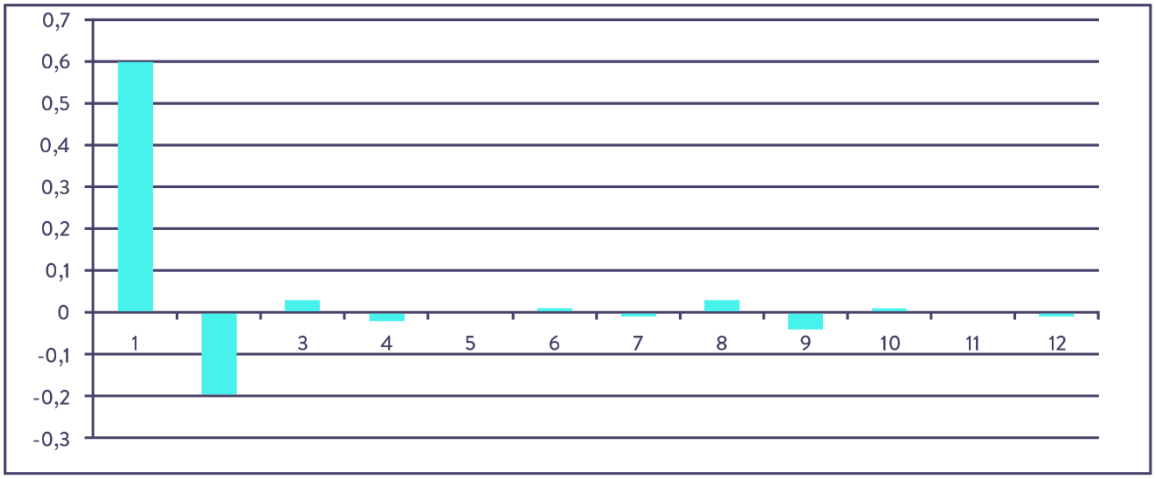



Figure 5: PACF

**Case 2**

The following ACF and PACF graphs suggest ARIMA(0,0,1), which looks a lot like the obverse of case 1.

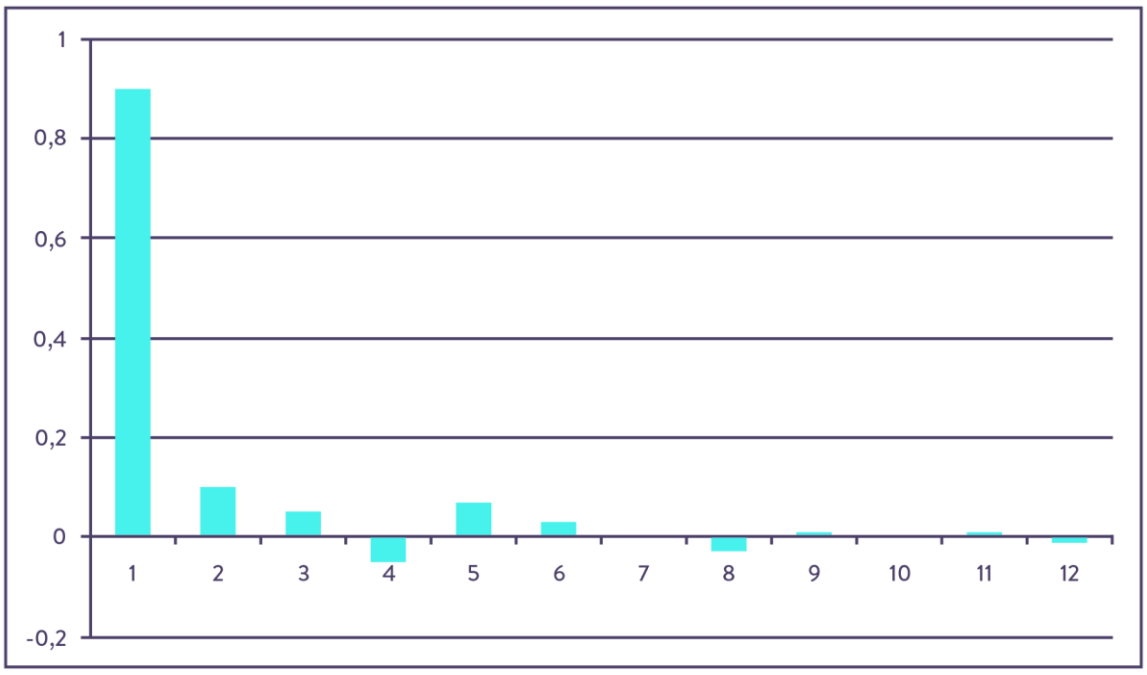



Figure 6: ACF

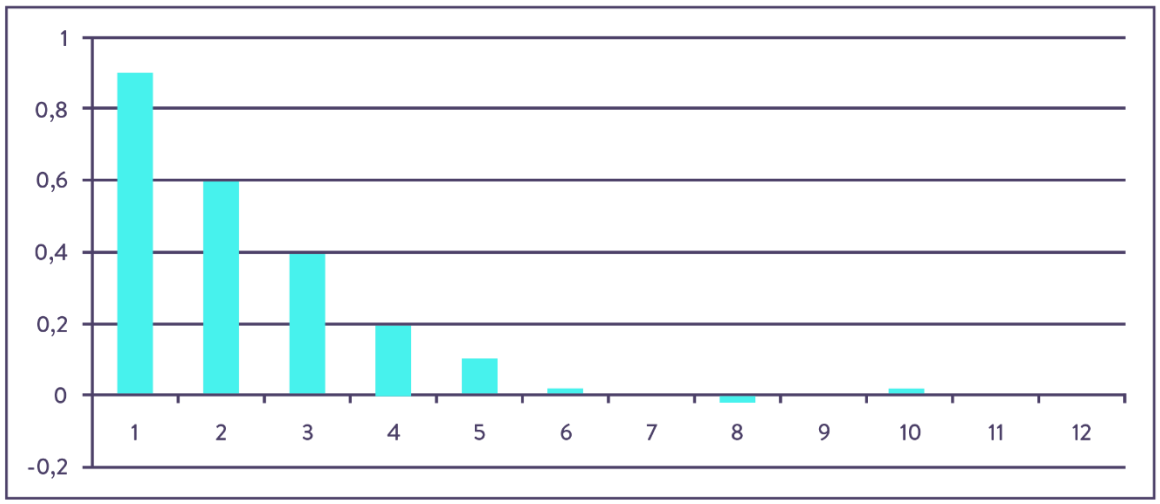



Figure 7: PACF

**Case 3**

The following ACF and PACF graphs suggests ARIMA(2,0,0).

The lag at which the PACF cuts off (at 2) is the indicated number of AR terms.

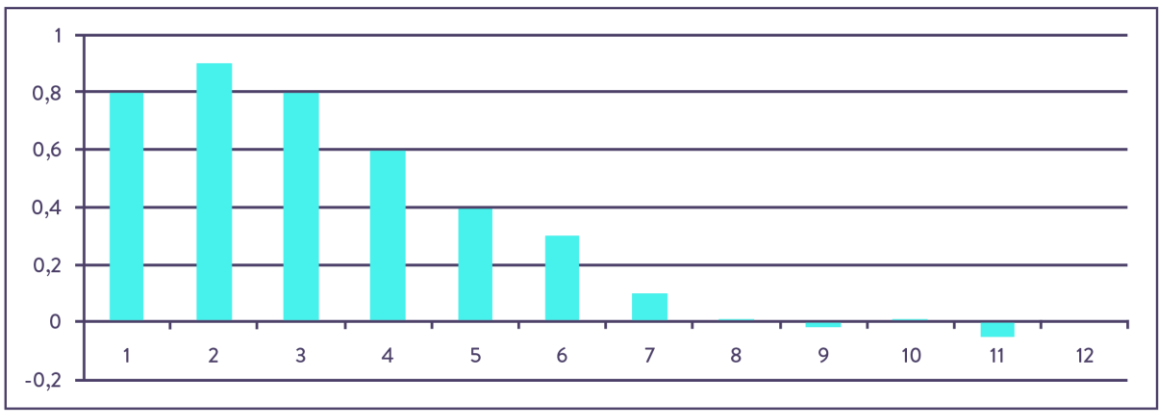



Figure 8: ACF

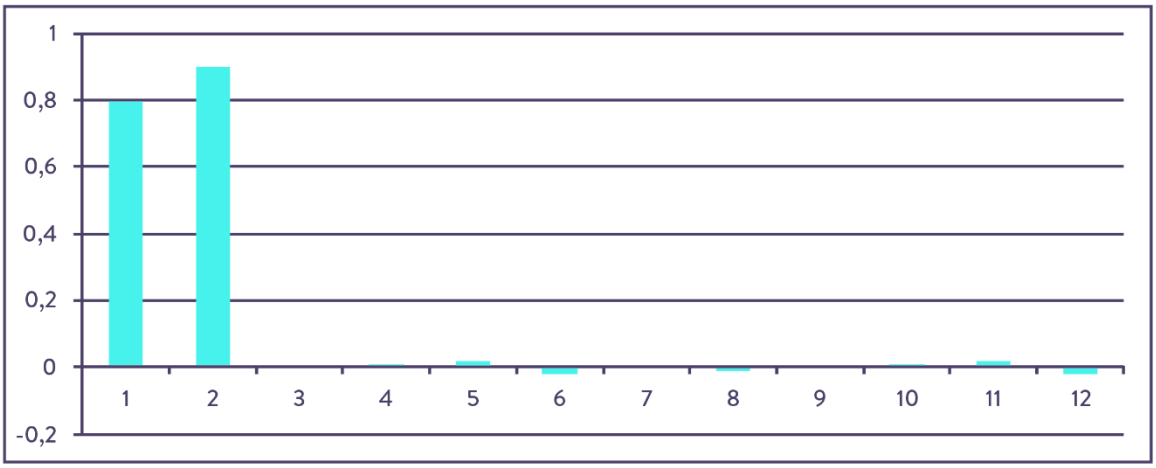



Figure 9: PACF

**Case 4**

The following ACF and PACF graphs suggests ARIMA(p,0,q).

Both ACF and PACF look as follows:

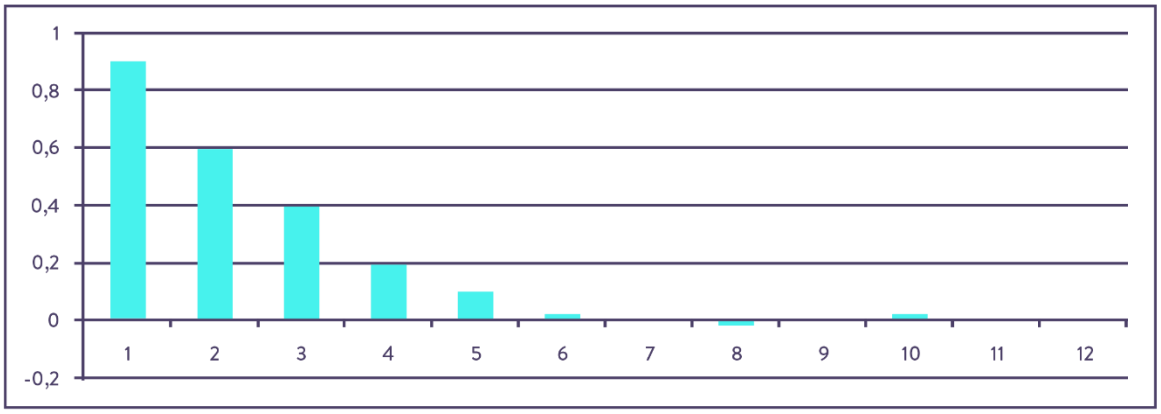



Figure 10: ACF and PACF

**Case 5**

The following ACF and PACF graphs suggests ARIMA(0,1,0).

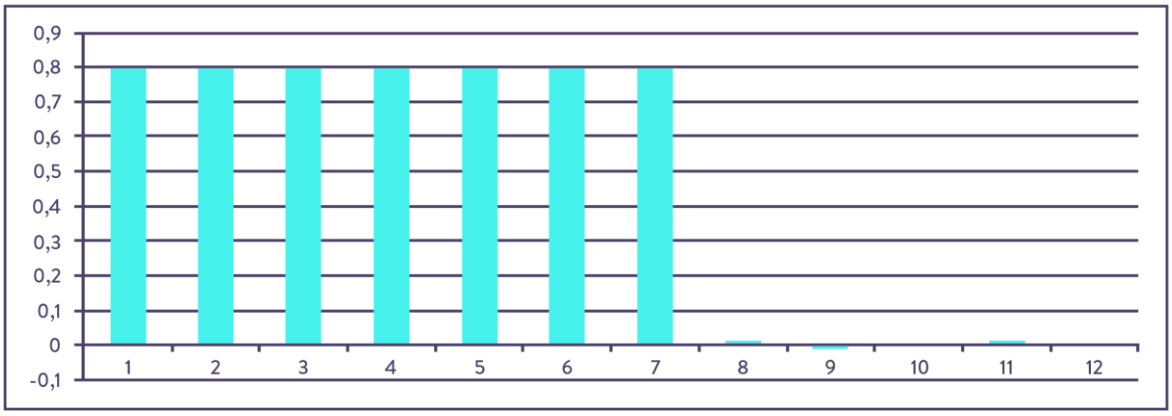



Figure 11: ACF

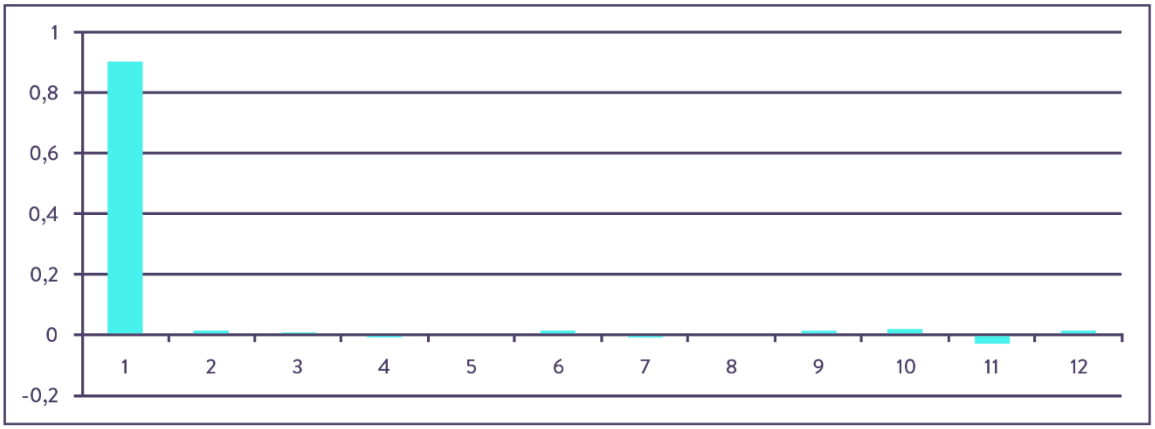



Figure 12: PACF

**Case 6**

The following, called a *damp sine wave*, ACF and PACF graph may also suggest ARIMA(0,1,0).

The PACF is the same as in the last case but the ACF looks as follows:

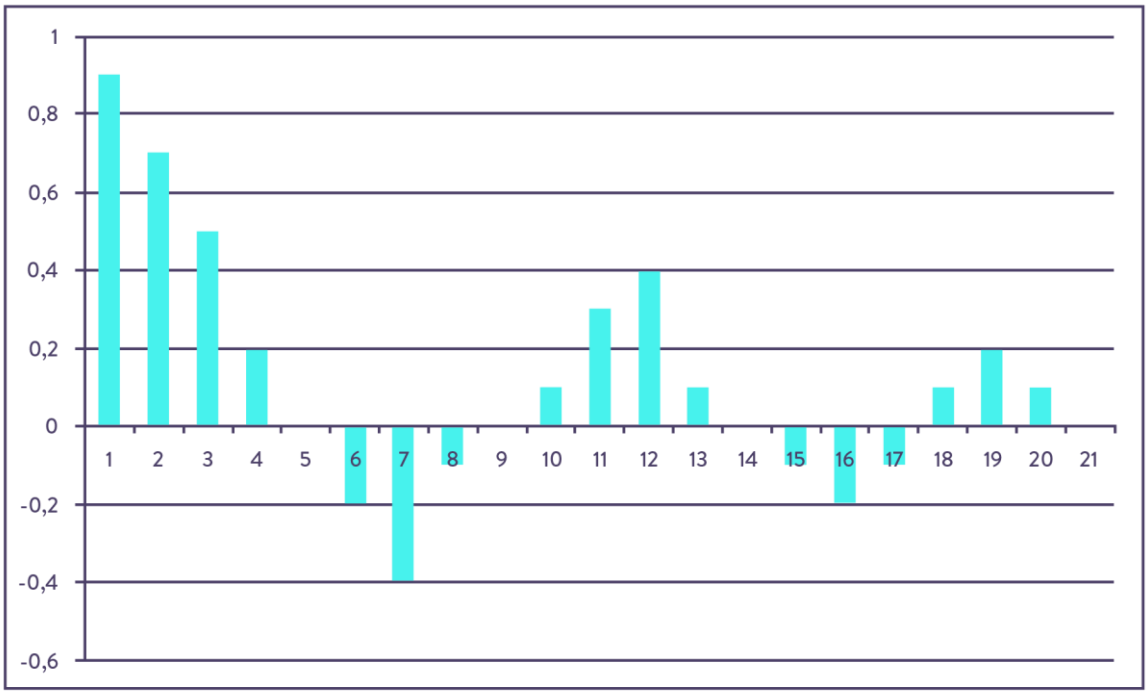



Figure 13: ACF

- Choosing the right $p$ and $q$ parameters is an art. Real ACFs and PACFs do not match exactly with the theoretical patterns as indicated in the table.
- ARMA modeling requires practice as choosing the right $p$ and $q$ is not an easy task.
- Quant analysts combine the information provided by ACF and PACF with the one provided by information criteria, such as the Akaike Information Criterion, Bayesian Information Criterion and Hannan-Quinn Information Criterion.

## Modeling house prices in Australia

Let's see what type of processes we must follow to model the house prices in Australia. The steps are:

1 Collect the data.

Source: Australian Bureau of Statistics.

http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/6416.0Sep%202014?OpenDocument#Time

Download Table 1. Residential Property Price Index, Index Numbers and Percentage Changes (xls document).

**2** Select the indicator.

| | A | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|
| 1 | | Residential Property Price Index ; Brisbane ; | Residential Property Price Index ; Adelaide ; | Residential Property Price Index ; Perth ; | Residential Property Price Index ; Hobart ; | Residential Property Price Index ; Darwin ; | Residential Property Price Index ; Canberra ; | Residential Property Price Index ; Weighted average of eight capital cities ; |
| 2 | **Unit** | Index Numbers | Index Numbers | Index Numbers | Index Numbers | Index Numbers | Index Numbers | Index Numbers |
| 3 | **Series Type** | Original | Original | Original | Original | Original | Original | Original |
| 4 | **Data Type** | INDEX | INDEX | INDEX | INDEX | INDEX | INDEX | INDEX |
| 5 | **Frequency** | Quarter | Quarter | Quarter | Quarter | Quarter | Quarter | Quarter |
| 6 | **Collection Month** | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 7 | **Series Start** | Sep-2003 | Sep-2003 | Sep-2003 | Sep-2003 | Sep-2003 | Sep-2003 | Sep-2003 |
| 8 | **Series End** | Sep-2014 | Sep-2014 | Sep-2014 | Sep-2014 | Sep-2014 | Sep-2014 | Sep-2014 |
| 9 | **No. Obs** | 45 | 45 | 45 | 45 | 45 | 45 | 45 |
| 10 | **Series ID** | A83728401F | A83728410J | A83728419C | A83728428F | A83728437J | A83728446K | A83728455L |
| 11 | Sep-2003 | 64.2 | 62.2 | 48.3 | 61.2 | 40.5 | 68.3 | 69.0 |
| 12 | Dec-2003 | 69.4 | 63.9 | 50.6 | 66.5 | 43.3 | 70.9 | 71.6 |
| 13 | Mar-2004 | 70.7 | 64.8 | 52.5 | 68.7 | 45.5 | 69.9 | 71.3 |
| 14 | Jun-2004 | 71.7 | 65.9 | 53.7 | 71.8 | 45.2 | 70.1 | 70.6 |

Figure 14: Source: Australian Bureau of Statistics, our analysis

Consider the following indicator in the analysis – Residential Price Index; Weighted average of eight capital cities (column J).

**3** Make a plot.

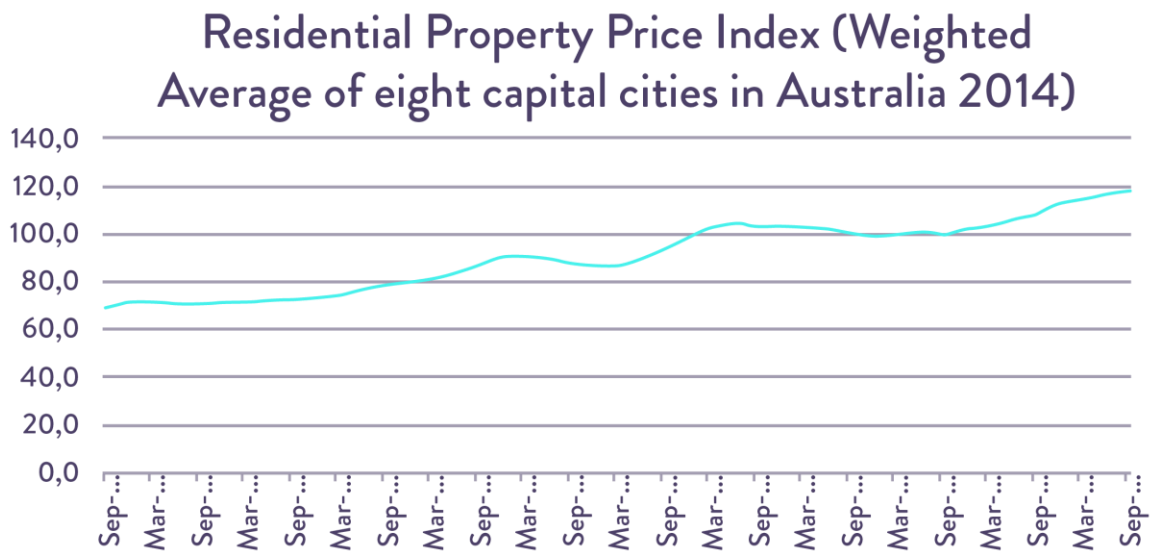



Figure 15

Source: Australian Bureau of Statistics, our analysis

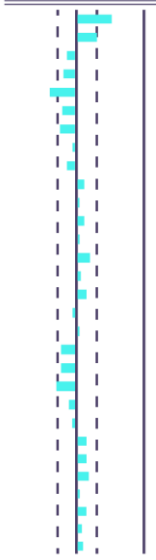
Cities considered in the analysis: Sydney, Melbourne, Brisbane, Adelaide, Perth, Hobart, Darwin and Canberra.

**4** Apply natural logarithm to the data.

**5** Apply the first difference in order to obtain stationarity.

**6** Plot ACF and PACF.

Date: 06/11/15 Time: 15:49
Sample: 2003Q3 2014Q3
Included observations: 44



| | AC | PAC | Q-Stat | Prob |
|---|---|---|---|---|
| 1 | 0.535 | 0.535 | 13.462 | 0.000 |
| 2 | 0.304 | 0.025 | 17.907 | 0.000 |
| 3 | -0.169 | -0.477 | 19.313 | 0.000 |
| 4 | -0.207 | 0.100 | 21.476 | 0.000 |
| 5 | -0.441 | -0.315 | 31.574 | 0.000 |
| 6 | -0.234 | 0.120 | 34.485 | 0.000 |
| 7 | -0.276 | -0.178 | 38.648 | 0.000 |
| 8 | -0.068 | -0.088 | 38.908 | 0.000 |
| 9 | -0.157 | -0.145 | 40.326 | 0.000 |
| 10 | 0.075 | 0.089 | 40.663 | 0.000 |
| 11 | -0.019 | -0.147 | 40.686 | 0.000 |
| 12 | 0.103 | -0.083 | 41.357 | 0.000 |
| 13 | -0.024 | -0.032 | 41.394 | 0.000 |
| 14 | 0.199 | 0.181 | 44.068 | 0.000 |
| 15 | 0.063 | -0.129 | 44.346 | 0.000 |
| 16 | 0.135 | -0.069 | 45.665 | 0.000 |
| 17 | -0.046 | 0.008 | 45.825 | 0.000 |
| 18 | -0.023 | -0.153 | 45.866 | 0.000 |
| 19 | -0.221 | -0.025 | 49.821 | 0.000 |
| 20 | -0.214 | -0.297 | 53.692 | 0.000 |
| 21 | -0.287 | -0.031 | 60.951 | 0.000 |
| 22 | -0.083 | 0.003 | 61.585 | 0.000 |
| 23 | -0.049 | -0.071 | 61.815 | 0.000 |
| 24 | 0.154 | -0.100 | 64.215 | 0.000 |
| 25 | 0.136 | -0.095 | 66.189 | 0.000 |
| 26 | 0.176 | -0.148 | 69.658 | 0.000 |
| 27 | 0.037 | -0.033 | 69.820 | 0.000 |
| 28 | 0.134 | 0.046 | 72.103 | 0.000 |
| 29 | 0.078 | -0.079 | 72.924 | 0.000 |
| 30 | 0.095 | -0.119 | 74.222 | 0.000 |

Figure 16: Plotting ACF and PCF

**7** Comment on the results:

- According to the results obtained, ACF and PACF decline with some spikes.
- ACF tends to have a damp sine wave evolution.
- A slight damp sine wave evolution might be seen at PACF.

According to Professor Walter Enders, if ACF and PACF decay (either direct or oscillatory) then we have an ARMA process. According to our graph, ACF and PACF decline after lag 1, which may indicate that the house prices in the large Australian cities follow an ARMA(1,1) process. The damp sine wave evolutions of ACF and PACF might also indicate an ARIMA(0,1,0) process.

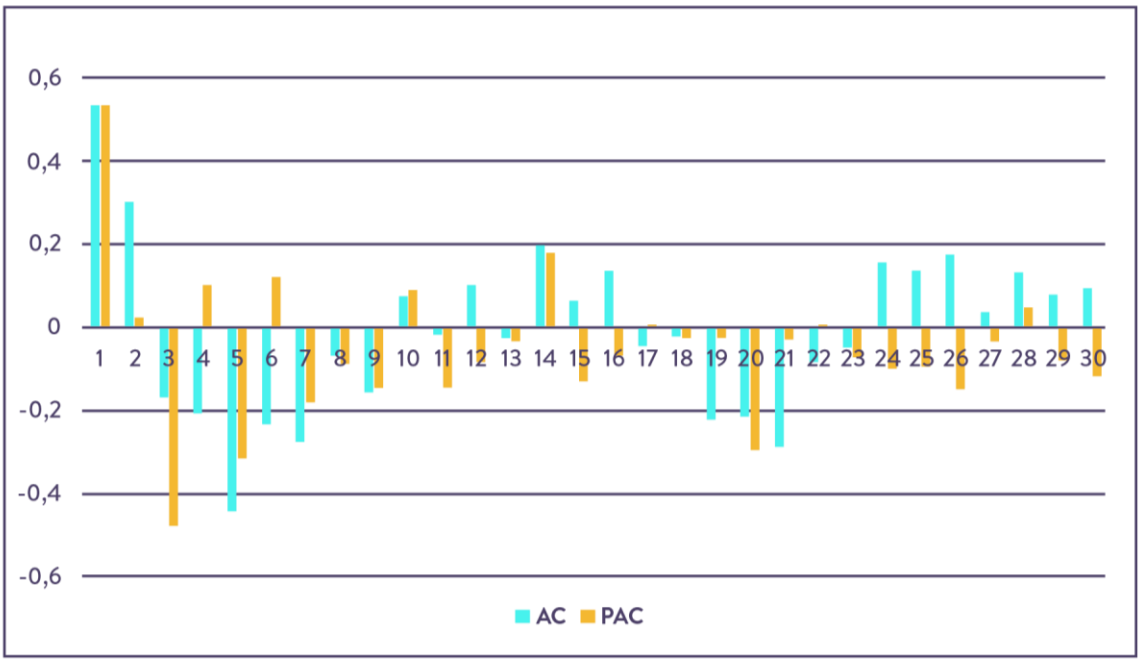



Figure 17: ACF and PCF evolution

## Selecting the optimal number of lags using Information Criteria

Information Criteria represent another solution to obtain the optimal number of lags ($p$ and $q$) for ARMA models. The most popular indicators are Akaike Information Criterion (AIC) and Schwartz Bayesian Criterion (SBC). AIC and SBC represent a direct way to obtain the optimal number of lags as compared to ACF and PACF charts that may provide subjective patterns in many cases.

**Formulas:**

$AIC = Tln(sum\ of\ squared\ residuals) + 2n$

$SBS = Tln(sum\ of\ squared\ residuals) + nln(T)$

$AIC$ - Akaike Information Criterion

$SBC$ - Schwartz Bayesian Criterion

- Goal: We select the model with $p$ and $q$ parameters that provides the lowest AIC and SBC (the two can also be negative).

We will not go into many details regarding IC. However, there are some core concepts we have to take into consideration when using IC in real situations, such as the following:

- ACF/PACF and Information Criterion can indicate different models. So, we must decide on the final model selected.
- SBC has superior large sample properties.
- AIC performs better in small samples than SBC.
- AIC tends to indicate the overparameterized model. In this case, you should check the $t$-statistics of all coefficients to see if they are significant.
- We suggest comparing results/forecasts from different ARMA models (different $p$ and $q$ values).
- It is easier to determine $p$ and $q$ parameters using an Information Criterion as compared to the Box Jenkins Method. The interpretation of ACF and PACF is an art and can be subjective.

## Trend, cycles and ARMA

How can we apply an ARMA model to a nonstationary series? Most real series show a trend, an average increase or decrease over time. For example, inflation generally increases over time, and deaths due to measles have dropped. Series also show cyclic behavior.

For example, influenza deaths rise in winter and dip in summer. We can remove trends and cycles from a series through differencing. For example, suppose $t$ is in months and $Yt$ is a series with a linear trend. That is, every month the series increases on average (in expectation) by some constant amount $C$.

Since $Yt = C + Yt - I + N$, where $Nt$ is a random "noise" component with expectation zero, then the differences, $Yt - Yt - 1$ are equal to $C + Nt$. Thus, $Nt + C$ (or just $Nt$) is a stationary series with the linear trend removed. We could now apply the ARMA model to $Zt = C + Nt$, even though it is not appropriate to do so to $Yt$ directly.

By analogy, higher order polynomial trends, (i.e., quadratic trends, cubic trends) can be removed by differencing more than once. Cycles can be removed by differencing at different lags. For example, a yearly cycle in a monthly series of stock prices can often be removed by forming the difference $Zt = Yt - Yt - 12$, a 12th order (lag) difference.

Once an ARMA model for $Zt$ is known, we could reverse the differencing to form the original $Yt$ from the $Zt$. We term this process *integration* (although the word summation might seem more appropriate). The combined model for the original series $Yt$, which involves auto regression, moving average, and integration of elements, is termed the *ARIMA* $(p, d, q)$ model (model of orders $p, d,$ and $q)$ with $p$ AR terms, $d$ differences, and $q$ MA terms.

## ARMA model in finance and algorithmic trading

ARMA is considered one of the core econometric models that are easy to understand and implement. However, when we think of algorithmic trading, we generally consider sophisticated trading strategies and complex models. The ARMA approach should not be looked down in this context as it can be a good starting point for complex strategies.

Arnold Zellner, one of the fathers of Bayesian econometrics, once said that we should "keep things sophisticated simple". Therefore, not everything that is complicated works in real situations and not everything that looks too basic should be dropped instantly when designing an algorithmic strategy. Below are some core findings regarding ARMA in finance.

- ARMA is a popular tool in predicting macroeconomic variables like GDP, inflation, industrial production or unemployment
- Provides poor stock return/value forecasts
- Provides better results in spread forecasting
- The forecasts are highly dependent on the latest registered values

- ARMA forecasts can be improved by combining them with results from other models and algorithms. For example, Netflix data analysts obtained better predictions after combining results from different algorithms (hybrid approach)
- Adding exogenous variables can improve ARMA predictions (ARMAX model)

## 3.2.5 Transcript: Evaluating a Univariate Time Series Model (Part 1)

Now we will perform an empirical application of what we've learnt in the module by fitting and evaluating a univariate time series model using R. This evaluation is divided into two videos. In this first video we will be visualizing and constructing the ACF and PACF of data using R.

### Data information

We will attempt to model the US CPI inflation rate in compounded annual rate of change. You can download this data in Microsoft Excel from https://fred.stlouisfed.org/series/CPIAUCSL. Use "EDIT GRAPH" and change the "Units" to "Percent Change from Year Ago". We will use the monthly data from the 1st of January 1970 to the 1st of January 2018. (This can be adjusted on the main graph window before clicking "DOWNLOAD".)

The following packages will be used in this R:

```
library(tidyverse)
library(stats)
library(readxl)
library(tseries)
library(forecast)
```

The packages forecast, tidyverse and *t* series need to be downloaded and installed. Load the data using the tidyverse data management package, which will look like this:

```
Inflation <- read_excel("C:/ <your path>/<your file name>.xls ", sheet = "FRED
Graph",
                        col_names = T,
                        col_types = c("date", "numeric"),
                        skip = 10)
```

## Visualization

The first step in any analysis is to know what types of time series models we may need to consider. Let's plot the full sample data:

```
ggplot(data = Inflation, mapping = aes(x=observation_date,y=CPIAUCSL_PC1))+
geom_line(color ="darkblue") + labs(x="",y="CPI Inflation - compounded annual rate of
change")
```

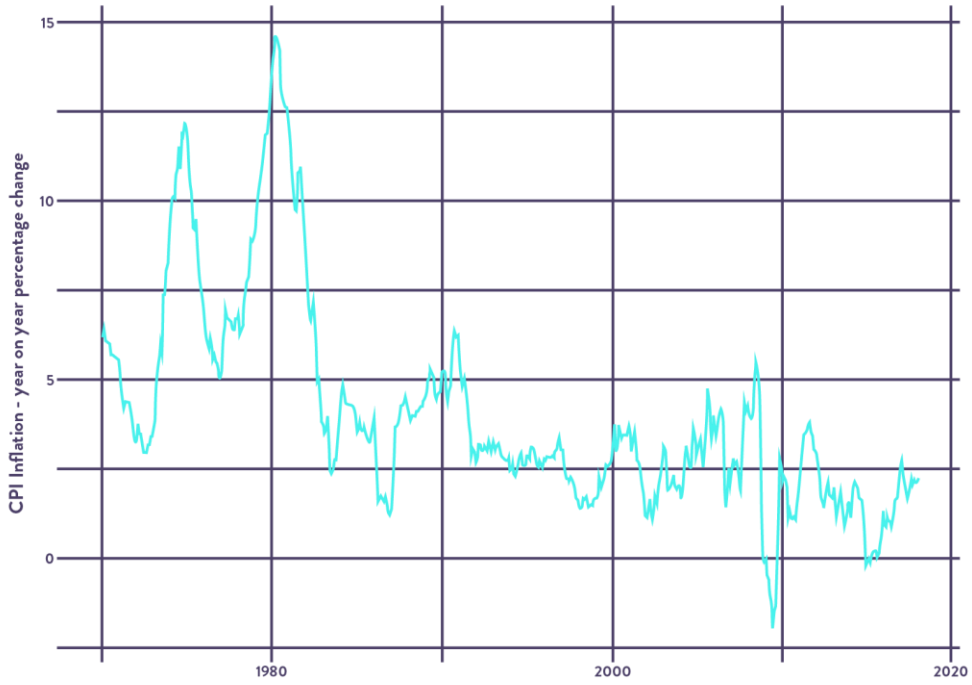



Figure 18: Visualization

This process, viewed as a whole, appears quite non-stationary. There is no obvious mean over the whole sample, and it seems to be wandering downwards.

However, if we use our real-world knowledge of what we intend to model, we know that the 1970s, which saw high and variable inflation, partly induced by the oil shocks, was a very different period from the last few decades. We also know that the US Federal Reserve Bank has become quite good at keeping the inflation rate at around 2%. Thus,

we can consider that there may be two different data generating processes here. We will see how this affects our results.

## Auto-correlation

To show how different the process is in its early and later phases, we truncate the data at the beginning of 1992. Then we construct the ACF for both the full and truncated samples.

R code:

```
# Convert the data frame to a time series object
CPI_percent_change = <- ts(Inflation$CPIAUCSL_PC1, start=1970,
frequency=12)
# Truncate the time series object
CPI_percent_change_truncated = <- window(CPI_percent_change, start =
1992)# construct the ACF for the full and truncated sample
acf(CPI_percent_change)
acf(CPI_percent_change_truncated)
```

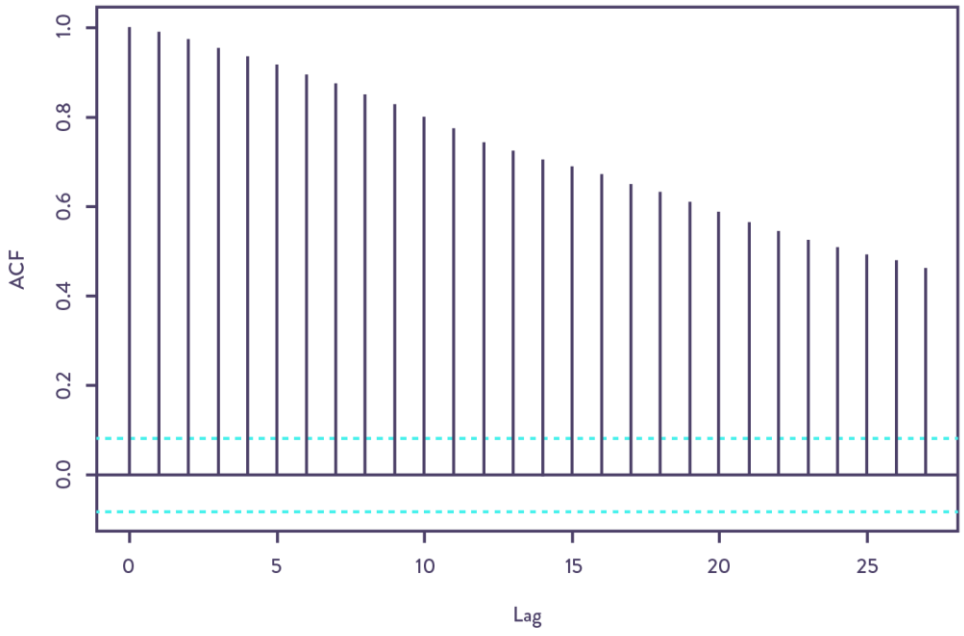



Figure 19: Series CPI_percentage_change

Note that the zero-$th$ auto-correlation is always 1 – a variable is always perfectly correlated with its own value. The ACF of the full-time path of the inflation variable is typical of a non-stationary process: auto-correlations are very close to 1 and slowly fade.

In this example, a shock in some months will still have an impact on the current level more than two years into the future.

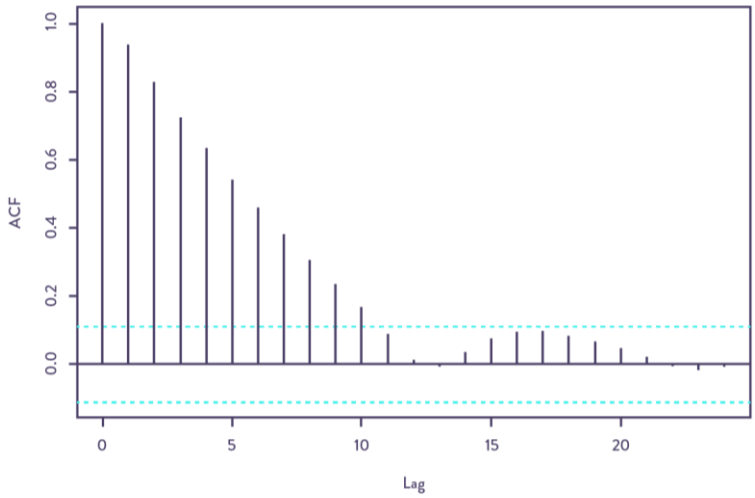



Figure 20: Series CPI_percentage_change_truncated

The ACF of the truncated time path is far more typical of a somewhat persistent, but fundamentally stationary process: a rapidly decaying ACF. This suggests further evidence that the time series behavior of this process changed over time.

## Partial auto-correlation

Next, we construct the PACF for both the full and truncated sample.

R code:

```
pacf(CPI_percent_change)
pacf(CPI_percent_change_truncated)
```

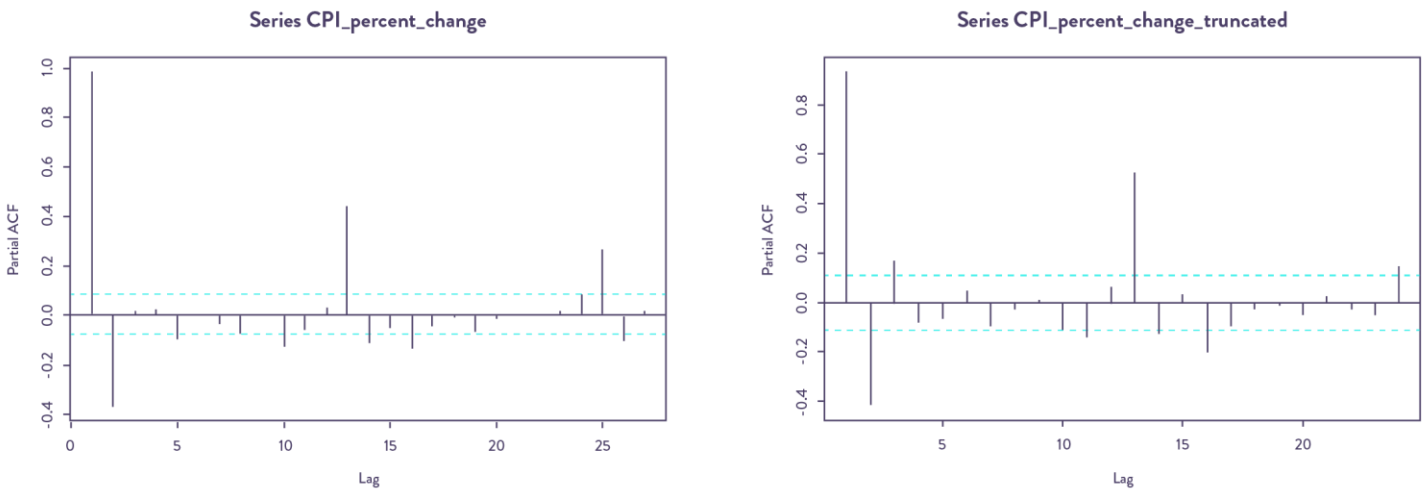



Figure 21: Side by side view

In this case, there seems to be no substantive difference between the two PACFs on the different samples. There is a strong impact of the first lag – notice that pacf() by default drops the zero-$th$ lag. There is a strong negative impact on the second lag, which is conditional on the first. Then there is a strong impact of the 13$^{th}$ lag.
Several other lags are just barely significant.

There is no clear conclusion we can draw from this, besides that an autoregressive model may be useful for this data and that it must have sufficient lags to capture the time series properties of the data.

In this video we visualized and constructed the ACF and PACF of a data set to evaluate a univariate time series model. In the next video we will be evaluating this model further.

## 3.2.6 Transcript: Evaluating a Univariate Time Series Model (Part 2)

In the previous video we visualized the ACF and PACF of data using R to evaluate a univariate time series model. However, eye-balling the univariate time series model and its auto- and partial auto-correlation functions is not sufficient for evaluating it.

In this video we will continue the evaluation of the univariate time series model by conducting formal tests, namely the Augmented Dickey-Fuller (ADF) Test, to test for a null of non-stationarity against an alternative of stationarity, and the KPSS Test for a null stationarity against an alternative of non-stationarity.

### Formal stationary tests

**Augmented Dickey-Fuller Tests:**

The R code is:

```
# conduct formal stationarity tests:
# augmented dickey fuller tests:
adf.test(CPI_percent_change)
adf.test(CPI_percent_change_truncated)
```

The R output is:

```
      Augmented Dickey-Fuller Test
data:  CPI_percent_change
Dickey-Fuller = -3.8252, Lag order = 8, p-value = 0.01774
alternative hypothesis: stationary
      Augmented Dickey-Fuller Test
data:  CPI_percent_change_truncated
Dickey-Fuller = -4.6883, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
Warning message:
In adf.test(CPI_percent_change_truncated) :
  p-value smaller than printed p-value
```

Interpretation:

The ADF test of the full sample of the series cannot, at a 1% level of significance, reject the hypothesis of a unit root. However, it is close to rejection, as we would have rejected the hypothesis at a 2% level. At a 1% level, the ADF test of the truncated sample rejects the null of the non-stationarity. Notice the warning: the actual p-value is smaller than the reported 1%.

From these results we would confirm our initial hypothesis: that the early part of the sample is non-stationary enough to make the whole series non-stationary, but if we focus on the last part of the sample it tests as stationary.

**KPSS tests:**

The R code is:

```
# KPSS tests:
kpss.test(CPI_percent_change)
kpss.test(CPI_percent_change_truncated)
```

The R output is:

```
KPSS Level = 4.9003, Truncation lag parameter = 5, p-value = 0.01
Warning message:
In kpss.test(CPI_percent_change) : p-value smaller than printed p-value
KPSS Test for Level Stationarity
data:  CPI_percent_change_truncated
KPSS Level = 1.1072, Truncation lag parameter = 4, p-value = 0.01
Warning message:
In kpss.test(CPI_percent_change_truncated) :
  p-value smaller than printed p-value
```

Interpretation:

The KPSS tests for both the full and truncated sample reject the hypothesis of stationarity. It often happens that different tests give different conclusions. You should work to understand how much you can trust each by knowing how each test works.

For now, we will fit both a stationary model and non-stationary model to the truncated series.

## Fitting and evaluating models

Here we fit a number of models to the inflation process from 1990 to 2013 to allow some room to explore its forecasting performance. We saw in the ACF and PACF in the previous video that there is significant autocorrelation for many lags.

## High order AR

Let's estimate a high order AR:

R code:

```
# fit a simple AR model with 12 lags, no differencing, no moving
average terms - i.e. an ARIMA(12,0,0) model:
AR_model1 <- arima(window(CPI_percent_change,start=1990,end = 2013),
order=c(12,0,0), method = "ML")
summary(AR_model1)

# Output:
Call:
arima(x = window(CPI_percent_change, start = 1990, end = 2013),
order = c(12,
    0, 0), method = "ML")
Coefficients:
         ar1      ar2     ar3     ar4      ar5     ar6      ar7
ar8     ar9    ar10
      1.4345  -0.6725  0.1596  0.1365  -0.2740  0.2170  -0.0029  -
0.1530  0.1495  0.0384
s.e.  0.0600   0.1046  0.1123  0.1126   0.1126  0.1139   0.1149
0.1142  0.1144  0.1142
         ar11    ar12  intercept
      -0.1853  0.0543     2.7197
```

```
s.e.   0.1059  0.0617     0.2168
sigma^2 estimated as 0.1249:  log likelihood = -106.59,  aic =
241.17
```

Note that none of the coefficients are statistically significant, so this is not a parsimonious model. Let's check if it has white noise errors:

R code:

```
Box.test(AR_model1$residuals, lag = 12)

# Output:
     Box-Pierce test
data:  AR_model1$residuals
X-squared = 80.582, df = 12, p-value = 3.197e-12
```

The Box-Pierce test is one of a range of tests for serial correlation. If, for example, all auto-correlations of a process up to lag 12 (i.e. the `lag = 12` argument in the code above) were zero, then, for a sample auto-correlations $\hat{\rho}_k$ of $n$ draws from this process, the quantity:

$$\hat{Q}_{12} = n \sum_{k-1}^{12} \hat{\rho}_k$$

will have a $\chi^2$ distribution with 12 degrees of freedom. The $p$-value gives you the usual information: the probability that we would have obtained the observed value $\hat{Q}_{12}$ if its true value were zero.

The intuition is simple: if the residuals are white noise, their auto-correlations should be close to zero in any sample. This test is one way to assign probability values to any observed series of auto-correlations. If a test statistic yields a too low probability value, we reject its hypothesis.

In this case, the probability value is practically zero. Thus, we clearly reject the hypothesis that the errors are white noise – there is too much persistence (or,

equivalently, too much auto-correlation) in inflation to be captured by a simple AR model.

## ARIMA

Next, we turn to an ARIMA (12, 2, 2) [12 lags, 2 times differences, 2 moving average terms] model to see if differencing will improve the fit of the model:

R code:

```
ARIMA_model <- arima(window(CPI_percent_change,start=1990,end =
2013), order=c(12,2,2), method = "ML")
summary(ARIMA_model)
Box.test(ARIMA_model$residuals, lag = 12)

# Output:
Call:
arima(x = window(CPI_percent_change, start = 1990, end = 2013),
order = c(12,
    2, 2), method = "ML")
Coefficients:
         ar1      ar2      ar3     ar4      ar5     ar6      ar7
ar8     ar9     ar10
      0.3000  -0.0914  -0.0405  0.0346  -0.1287  0.0506  -0.0356  -
0.0464  0.0192  -0.0340
s.e.  0.0876   0.0618   0.0525  0.0502   0.0509  0.0534   0.0514
0.0506  0.0513   0.0509
        ar11     ar12      ma1      ma2
      0.1763  -0.5968  -0.8132  -0.1868
s.e.  0.0502   0.0475   0.1173   0.1161
sigma^2 estimated as 0.08541:  log likelihood = -57.76,  aic =
145.52
      Box-Pierce test
data:  ARIMA_model$residuals
X-squared = 20.191, df = 12, p-value = 0.06355
```

This model seems a bit better. Some of the coefficients are significant – the standard error being two times smaller. The parameter estimate is a rough guide, but you can check the $p$-values by investigating the other elements in the model object.

The residuals tests as white noise at a 5% level. This is still not a parsimonious model, as there are many insignificant coefficients to zero. In practice, it can be very hard to find a good single equation model for a time series this persistent with such different behavior over time.

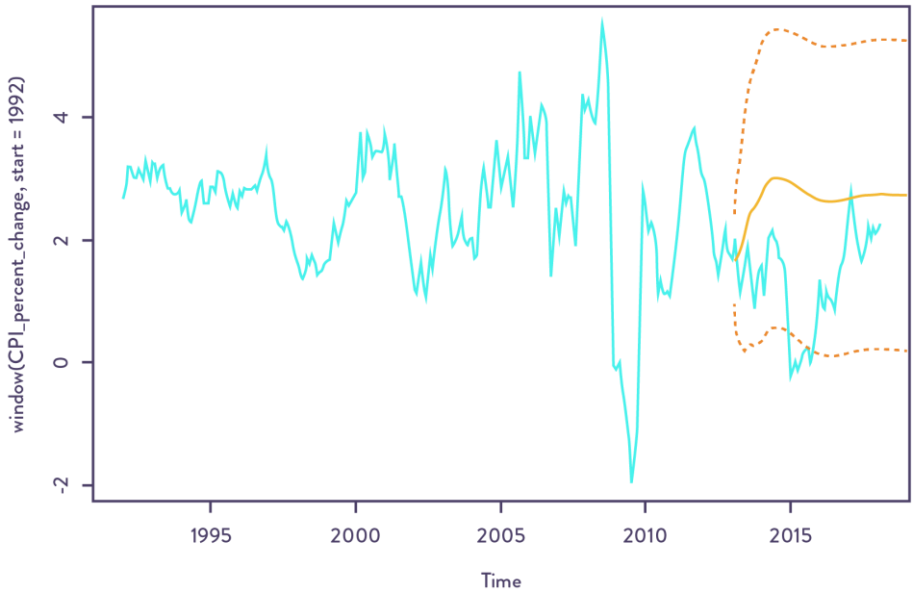
## Forecasting

To illustrate how to do an h-step ahead forecast without re-estimation in R, we use a simpler model: an AR(12) model.

```
AR_model <- ar(window(CPI_percent_change,start=1990,end = 2013),
order=12,method = "mle")
AR_forecast <- predict(AR_model,n.ahead= 72,se.fit=TRUE)
```

The object AR_forecast contains two elements: AR_forecast$pred is the predicted value of the series, and AR_forecast$se is the estimated standard error of the forecast. We will use this to plot the predicted values along with our uncertainty about those forecasts.

```
plot(window(CPI_percent_change,start=1992)
lines(AR_forecast$pred,col="blue")
lines(AR_forecast$pred+2*AR_forecast$se,col="cornflowerblue",lty =
"dashed")
lines(AR_forecast$pred-2*AR_forecast$se,col="cornflowerblue",lty =
"dashed")
```




The result of our forecast is plotted here. As expected, this is not a very strong model. It seems to get the mean right, but with a very large amount of uncertainty around it. With a series as persistent as this, containing periods of such large volatility, we cannot expect simple time series models to work well.

In this video we evaluated a univariate time series model by conducting formal tests. In the peer review question you will apply the methods that we have discussed in the video to the growth rate in GDP before the 2008 crisis, which is less volatile.

## References

Ahoniemi K. (2006). Modeling and Forecasting Implied Volatility – an Econometric Analysis of the VIX Index, Helsinki Center of Economic Research, Discussion Paper. Forecasting with ARMA Models, University of Leicester Courses.

Dong K. (2015). Liquidity Prediction in Limit Order Book Markets, University of Liverpool.

Greene, W. (2000). *Econometric Analysis, Prentice-Hall*, NY.

Guerrón-Quintana P. and Zhong M. (2017). Macroeconomic Forecasting in Times of Crises. Federal Reserve Board, Washington D.C., Staff working paper.

Gujarati, D. (2004). *Basic Econometrics*, McGraw-Hill.

Iqbal M. and Naveed A. (2016). Forecasting Inflation: Autoregressive Integrated Moving Average Model, European Scientific Journal.

Monticini A. and Ravazzolo F. (2011). Forecasting the intraday market price of money.

Norges Bank Working Paper.

Xu, S. Y. Stock Price Forecasting Using Information from Yahoo Finance and Google Trend.