Time series analysis

Stochastic processes

Concepts

- Models of time series
  - Moving averages (MA) and autoregressive (AR) processes
  - Mixed models (ARMA/ARIMA)
- The Box-Jenkins model building process
  - Model identification
    - Autocorrelations
    - Partial-autocorrelations
  - Model estimation
    - The objective is to minimize the sum of squares of errors
  - Model validation
    - Certain diagnostics are used to check the validity of the model
  - Model forecasting

Autoregressive (AR) and moving average (MA)

- So, what's the big difference?
  - The AR model includes lagged terms of the time series itself
  - The MA model includes lagged terms on the noise or residuals

- How do we decide which to use?
  - ACF and PACF

Autocorrelation functions (ACFs) and Partial-autocorrelation functions (PACFs)

- The autocorrelation function (ACF) is a set of correlation coefficients between the series and lags of itself over time

- The partial autocorrelation function (PACF) is the partial correlation coefficients between the series and lags of itself over time
  - Amount of correlation between a variable and a lag of itself that is not explained by correlations at all *lower-order*-lags
    - Correlation at lag 1 "propagates" to lag 2 and presumably to higher-order lags
    - PA at lag 2 is difference between the actual correlation at lag 2 and expected correlation due to propagation of correlation at lag 1

Autoregressive (AR) models

- An autoregressive model of order "p"
  - AR(p)

$$X_t = \beta_1 X_{t-1} + \beta_2 X_{t-2} + \ldots + \beta_p X_{t-p} + \mathrm{e}_t$$

  - Current value of X_t can be found from past values, plus a random shock e_t

- Like a multiple regression model, but X_t is regressed on past values of X_t

The AR(1) Model

- A simple way to model dependence over time is with the "autoregressive model of order 1"

- This is a OLS model of X_t regressed on lagged X_{t-1}

$$X_t = \beta_0 + \beta_1 X_{t-1} + \mathrm{e}_t$$

  - What does the model say for the $t+1$ observation?

$$X_{t+1} = \beta_0 + \beta_1 X_t + e_{t+1}$$

- The AR(1) model expresses what we don't know in terms of what we do know at time t

The AR(1) Model

$$X_t = \beta_0 + \beta_1 X_{t-1} + \mathrm{e}_t$$

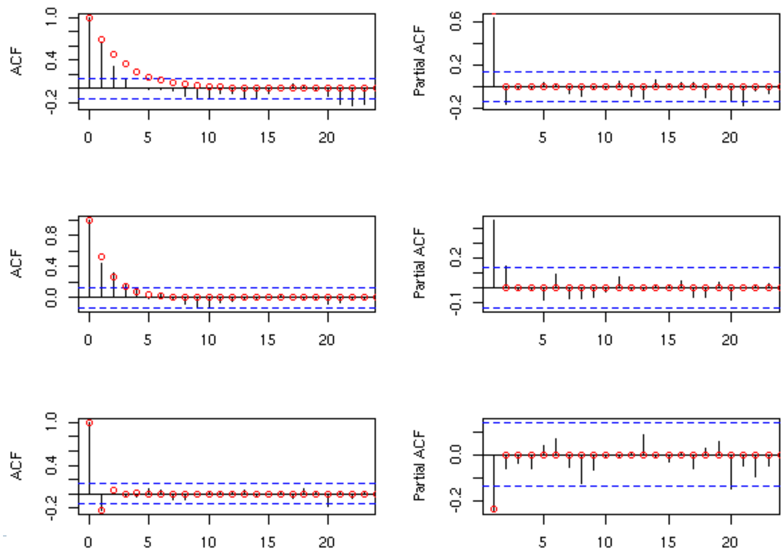
- If β_1 is zero, X depends purely on the random component (e), and there is no temporal dependence
- If β_1 is large, previous values of X influence the value of X_t

- If our model successfully captures the dependence structure in the data then the residuals should look random
  - There should be no dependence in the residuals!
- So to check the AR(1) model, we can check the residuals from the regression for any "left-over" dependence

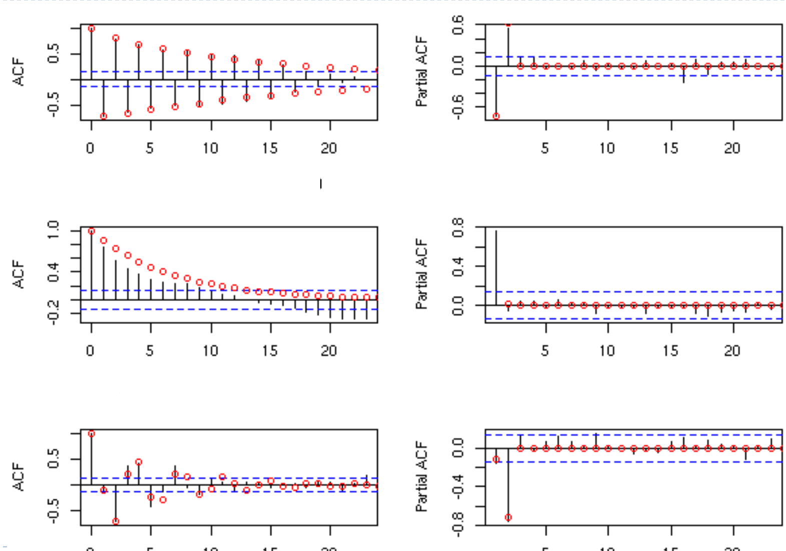
Identifying an AR process

- If the PACF displays a sharp cutoff while the 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

ACF and PACF for an AR(1) process



ACF and PACF for an AR(2) process



Moving-average (MA) models

- A moving-average model of order "q"
  - MA(q)

$$X_t = e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q}$$

  - Current value of X_t can be found from past shocks/error (e), plus a new shock/error (e_t)

- The time series is regarded as a moving average (unevenly weighted, because of different coefficients) of a random shock series e_t

The MA(1) model

- A first order moving average model would look like:

$$X_t = e_t + \beta_1 e_{t-1}$$

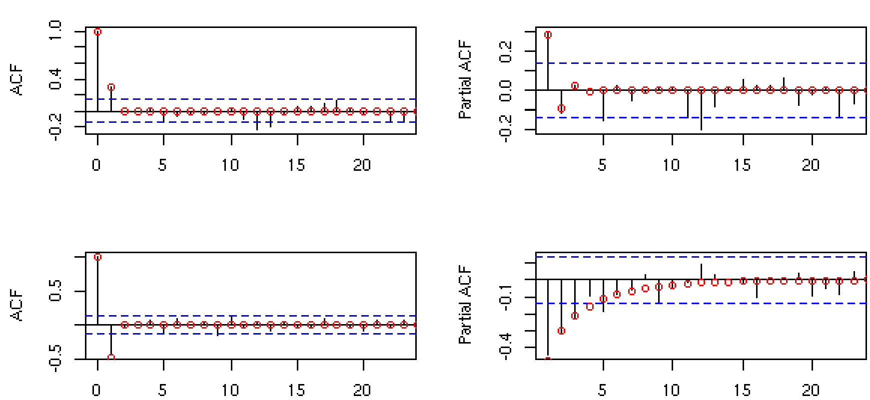
  - If β_1 is zero, X depends purely on the error or shock (e) at the current time, and there is no temporal dependence
  - If β_1 is large, previous errors influence the value of X_t

- If our model successfully captures the dependence structure in the data then the residuals should look random

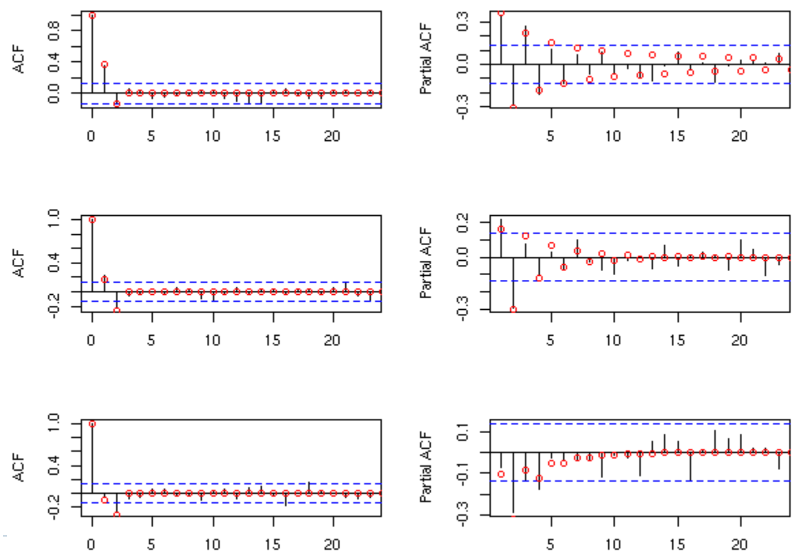
Identifying a MA process

- If the ACF of the differenced series displays a sharp cutoff and/or the lag-1 autocorrelation is negative then consider adding an MA term to the model
  - The lag at which the ACF cuts off is the indicated number of MA terms

ACF and PACF for an MA(1) process



ACF and PACF for an MA(2) process



Just to reiterate one more time…

- The diagnostic patterns of ACF and PACF for an AR(1) model are:
  - ACF: declines in geometric progression from its highest value at lag 1
  - PACF: cuts off abruptly after lag 1
- The opposite types of patterns apply to an MA(1) process:
  - ACF: cuts off abruptly after lag 1
  - PACF: declines in geometric progression from its highest value at lag 1

Mixed ARMA models

- An ARMA process of the order (p, q)

$$X_t = \beta_1 X_{t-1} + \ldots + \beta_p X_{t-p} + e_t + \alpha_1 e_{t-1} + \ldots + \alpha_q e_{t-q}$$

  - Just a combination of MA and AR terms

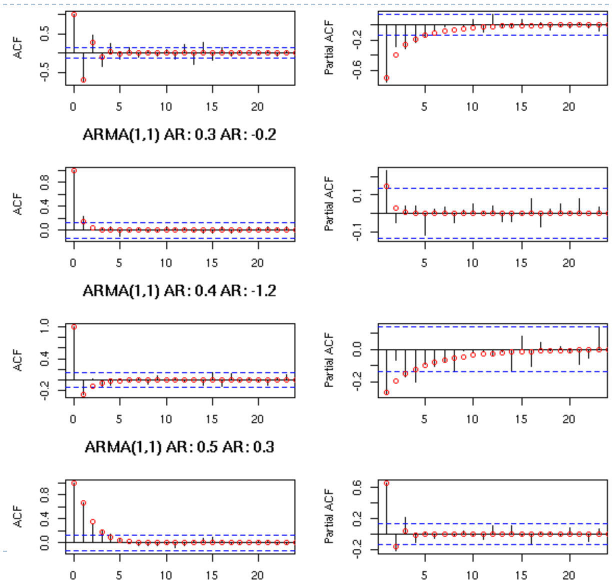
- Sometimes you can use lower-order models by combining MA and AR terms
  - ARMA(1,1) vs. AR(3,0)
  - Lower order models are better!

Identifying a ARMA process

- For the ARMA(1,1), both the ACF and the PACF exponentially decrease

- Much of fitting ARMA models is guess work and trial-and-error!

ACF and PACF for an ARMA(1,1) process



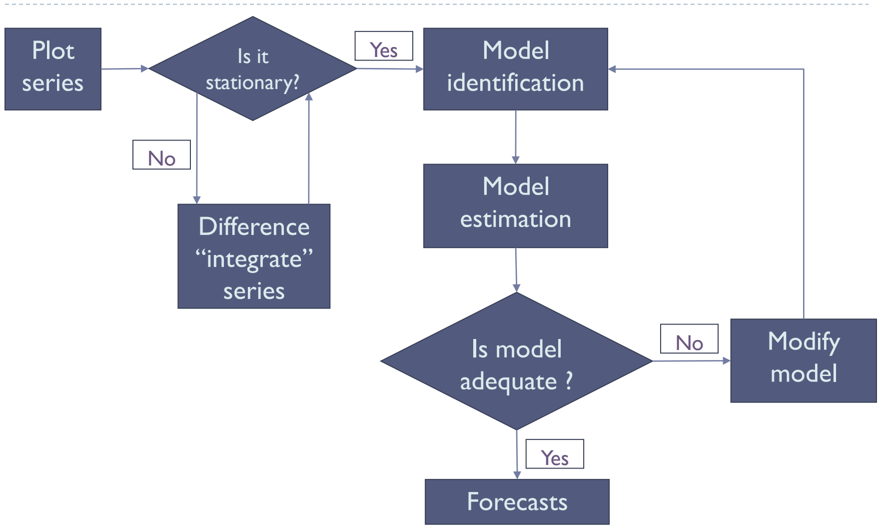
How do we choose the best model?

- In most cases, the best model turns out a model that uses either only AR terms or only MA terms
- It is possible for an AR term and an MA term to cancel each other's effects, even though both may appear significant in the model
  - If a mixed ARMA model seems to fit the data, also try a model with one fewer AR term and one fewer MA term
- As with OLS, simpler models are better!

The Box-Jenkins model building process

- Model identification
  - Autocorrelations
  - Partial-autocorrelations
- Model estimation
  - The objective is to minimize the sum of squares of errors
- Model validation
  - Certain diagnostics are used to check the validity of the model
    - Examine residuals, statistical significance of coefficients
- Model forecasting
  - The estimated model is used to generate forecasts and confidence limits of the forecasts

The Box-Jenkins model building process



What is an ARIMA model?

- Type of ARMA model that can be used with *some kinds* of non-stationary data
  - Useful for series with stochastic trends
    - First order or "simple" differencing
  - Series with deterministic trends should be differenced first then an ARMA model applied

- The "I" in ARIMA stands for integrated, which basically means you're differencing
  - *Integrated* at the order *d* (e.g., the dth difference)

The ARIMA Model

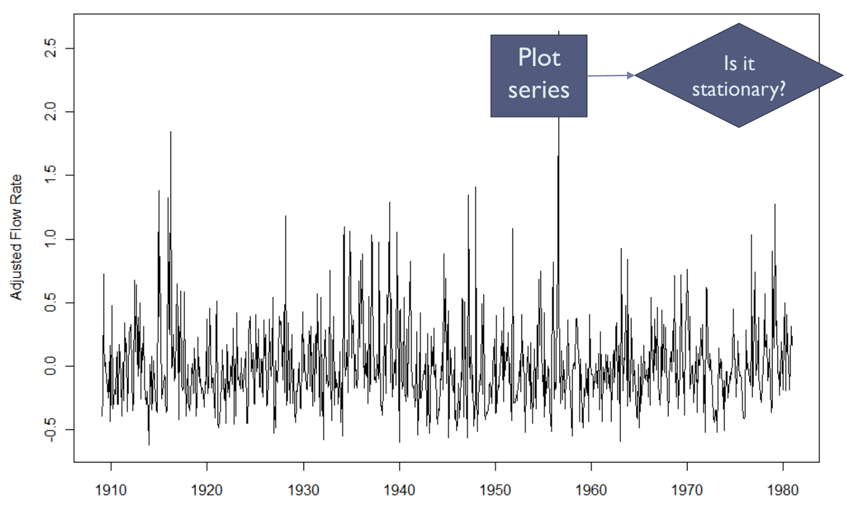
- Typically written as ARIMA(*p, d, q*) where:
  - *p* is the number of autoregressive terms
  - *d* is the order of differencing
  - *q* is the number of moving average terms
    - ARIMA(1,1,0) is a first-order AR model with one order of differencing

- You can specify a "regular" AR, MA or ARMA model using the same notation:
  - ARIMA(1,0,0)
  - ARIMA(0,0,1)
  - ARIMA(1,0,1)
    - etc, etc, etc.

The Box-Jenkins approach

- Model identification – two methods:
  - Examine plots of ACF and PACF
  - Automated iterative procedure
    - Fitting many different possible models and using a goodness of fit statistic (AIC) to select "best" model
- Model adequacy – two methods:
  - Examine residuals
  - AIC or SBC

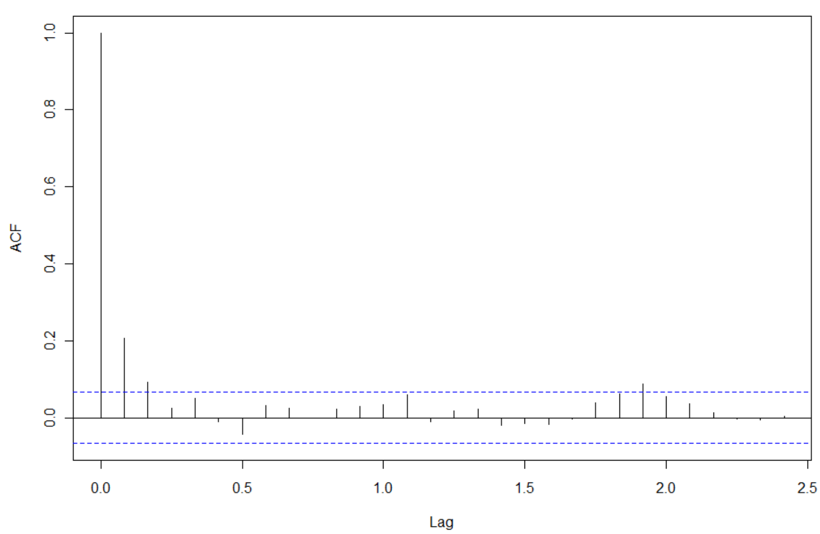
Example: stationary time series



Example: stationary series

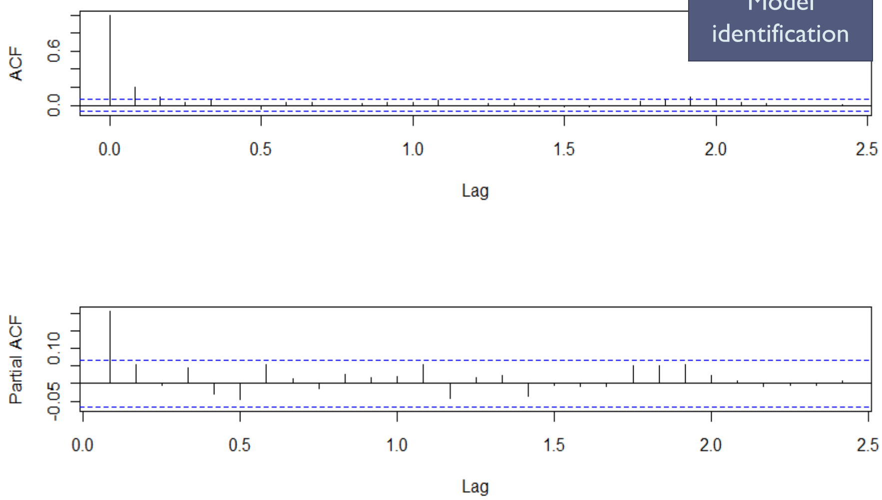
```
> adf.test(res.ts, alternative = "stationary")

        Augmented Dickey-Fuller Test

data:  res.ts
Dickey-Fuller = -8.4678, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary

> acf(res.ts)
```

Example: stationary series



Example: stationary series

Model identification



Example

- An ARIMA(1,0,0)?

Model identification

Model estimation

```
> res.ar<-arima(res.ts, order=c(1,0,0))

Call:
arima(x = res.ts, order = c(1, 0, 0))
Coefficients:
         ar1  intercept
      0.2057    -0.0001
s.e.  0.0333     0.0144

sigma^2 estimated as 0.11: log likelihood = -286.34, aic = 578.67
```

This is not really the intercept, it's the mean

An R caution

- When fitting ARIMA models, R calls the estimate of the mean, the estimate of the intercept
  - This is ok if there's no AR term, but not if there is an AR term
- For example, suppose we have a stationary TS:

$$X_t = \alpha + \beta x_{t-1} + e_t$$

- We can calculate the mean/intercept of the series as:

$$\mu = \alpha + \beta\mu \text{ or } \alpha = \mu(1-\beta)$$

  - So, the intercept (α) is not equal to the mean (μ) unless β=0
- In general, the mean and the intercept are the same only when there is no AR term

Fixing R's output

- To covert the mean into the true intercept we need to subtract the mean from all values of x
  - X_t and x_{t-1}, x_{t-p}

$$X_t - \mu = \beta(x_{t-1} - \mu) + e_t$$

or

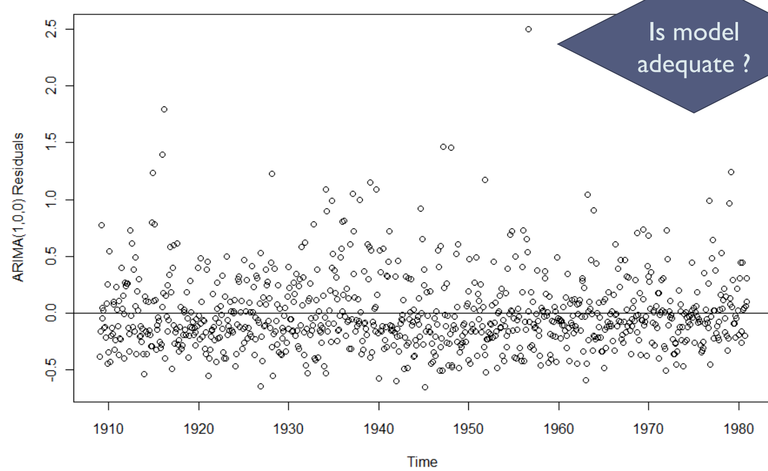
$$\alpha = \mu(1 - \sum_{t=1}^{p} \beta)$$

$$\begin{aligned} \alpha &= -.0001(1 - .2057) \\ &= -.000079 \\ X_t &= -.00008 + .2057 x_{t-1} + e_t \end{aligned}$$

Example

```
> plot(resid(res.ar),type="p", ylab="ARIMA(1,0,0) Residuals")
> abline(h=0)
```

Is model adequate ?



Example

```
> Box.test(res.ar$resid, lag = 10, type = "Ljung", fitdf=1)

        Box-Ljung test

data:  res.ar2$resid
X-squared = 9.1067, df = 9, p-value = 0.3334
```

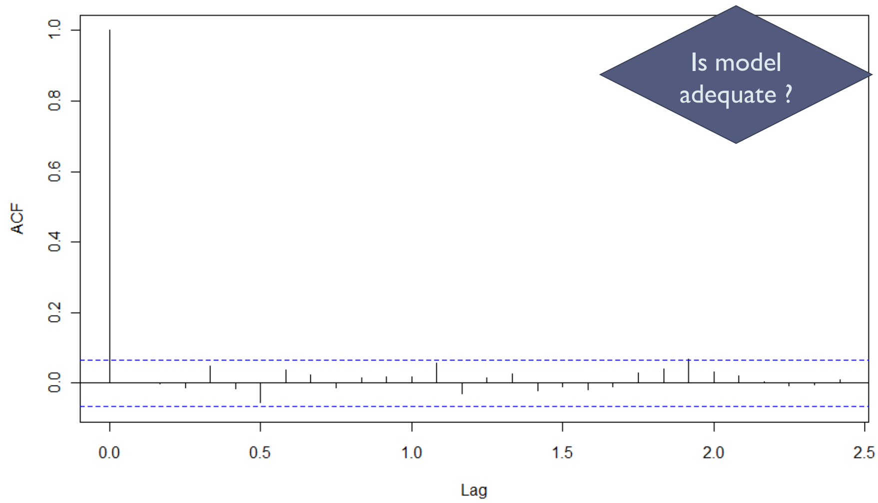
Where `fitdf=p+q` (in this case 1)

Is model adequate ?

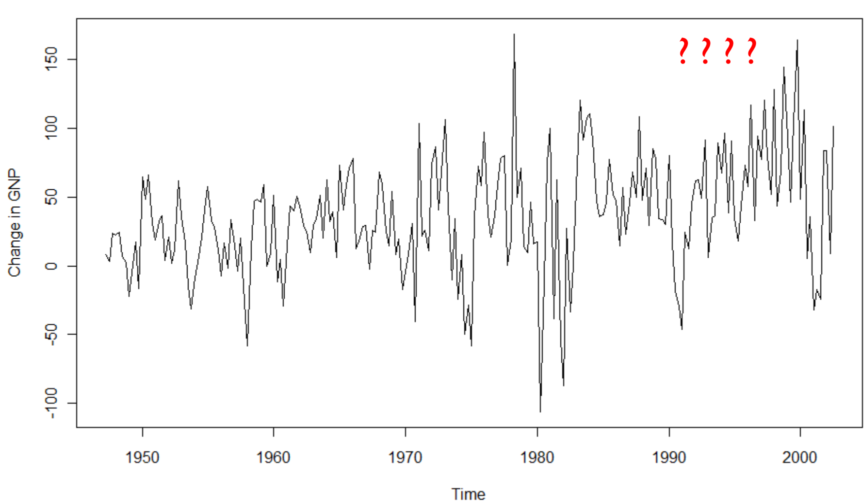
- The Ljung–Box test is defined as follows:
  - H_0:The residuals are random
  - H_a:The residuals are not random
- Tells you if your residuals are random using the ACF at a specified number of lags
- Tests the "overall" randomness of the residuals, even at high lags

Example

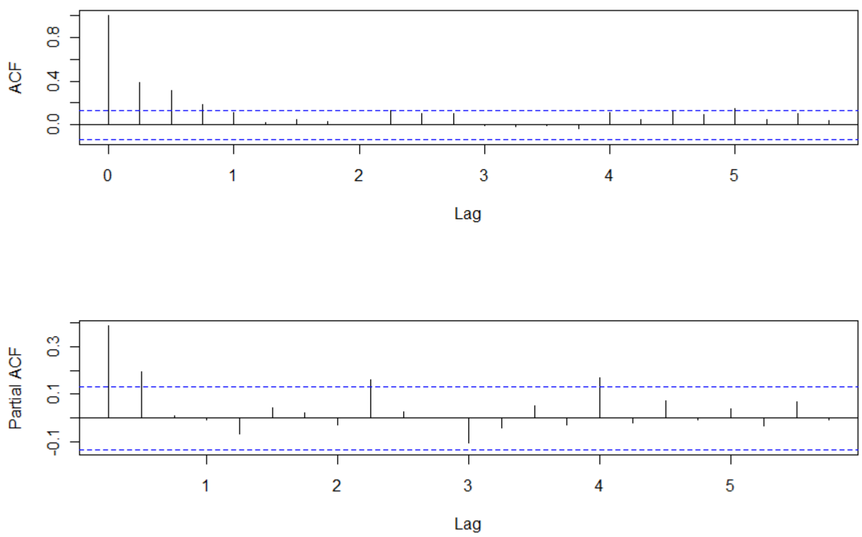
```
> acf(resid(res.ar))
```



Example: what if it's not so clear what model type to use



Example



Example

```
> adf.test(gnpgr, alternative = "stationary")

        Augmented Dickey-Fuller Test

data:  gnpgr
Dickey-Fuller = -5.7372, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

Example

```
> gnpgr.ar = arima(gnpgr, order = c(1, 0, 0))
> gnpgr.ma = arima(gnpgr, order = c(0, 0, 1))

> library(forecast)
> gnp.fit <- auto.arima(gnpgr, stationary=T)

> gnpgr.ar
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1  intercept
       0.390    36.0930
s.e.   0.062     4.2681

sigma^2 estimated as 1513:  log likelihood = -1127.83
AIC = 2261.66   AICc = 2261.77   BIC = 2271.87
```

$$\begin{aligned}\alpha &= 36.093(1-.39)\\ &= 22.0167\\ X_t &= 22.0167 + .39x_{t-1} + e_t\end{aligned}$$

Example

```
> gnpgr.ma
ARIMA(0,0,1) with non-zero mean

Coefficients:
          ma1  intercept
       0.2777    36.0524
s.e.   0.0534     3.4228

sigma^2 estimated as 1596:  log likelihood = -1133.72
AIC = 2273.43   AICc = 2273.54   BIC = 2283.64
```

(compared to AIC 2261.66 for ARIMA(1,0,0))

$$X_t = 36.052 + .2777e_{t-1} + e_t$$

Example

```
> best.order<-c(0,0,0)
> best.aic<-Inf
> for (i in 0:2) for (j in 0:2) {
  fit.aic<-AIC(arima(resid(flu.shgls), order=c(i,0,j)))
  if (fit.aic < best.aic) {
       best.order <- c(i,0,j)
       best.arma <- arima(resid(flu.shgls), order=best.order)
       best.aic <-fit.aic }}
```

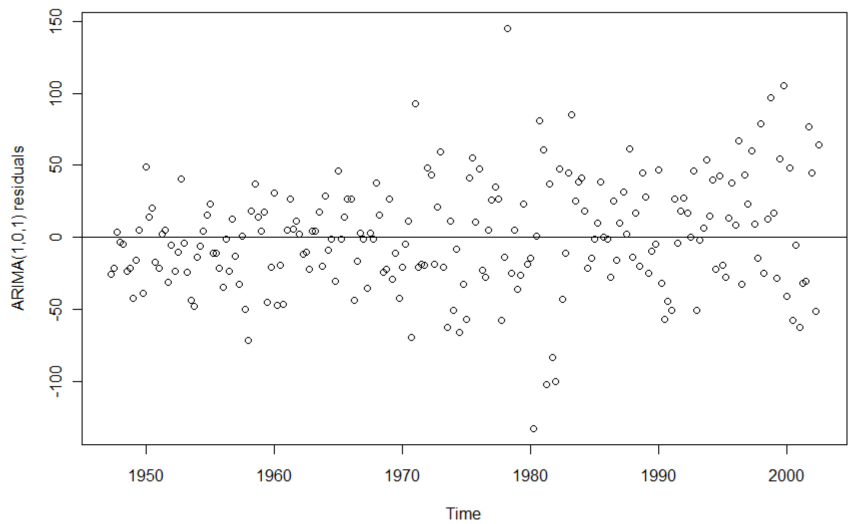
Example

```
> best.arma
Series: gnpgr
ARIMA(2,0,0) with non-zero mean

Coefficients:
         ar1     ar2  intercept
      0.3136  0.1931    36.0519
s.e.  0.0662  0.0663     5.1613

sigma^2 estimated as 1457:  log likelihood = -1123.67
AIC = 2255.34   AICc = 2255.52   BIC = 2268.95
```

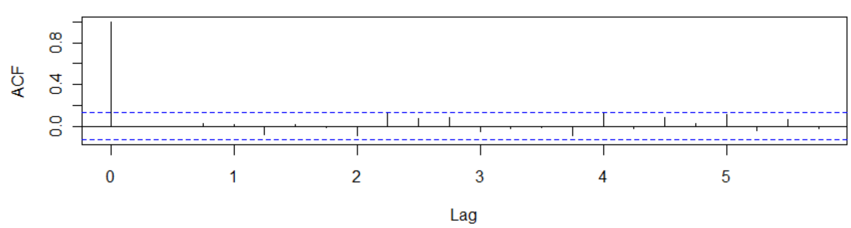
(compared to AIC 2261.7 for ARIMA(1,0,0) and 2273.4 for ARIMA(0,0,1))

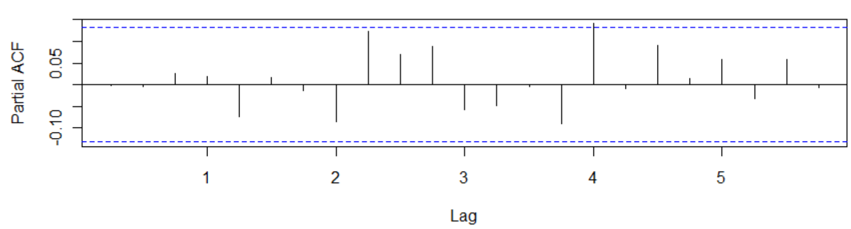
$$\alpha = 36.052(1-(.3136+.1931))$$
$$= 17.784$$
$$X_t = 17.784 + .3136x_{t-1} + .1931x_{t-2} + e_t$$

Example



Example





Example

```
> Box.test(gnp.fit$resid, lag = 10, type = "Ljung", fitdf=2)

        Box-Ljung test

data:  best.arma$resid
X-squared = 8.0113, df = 8, p-value = 0.4324
```

Forecasting

```
> gnp.pred<-predict(best.arma ,n.ahead=24)

$pred
         Qtr1     Qtr2     Qtr3     Qtr4
2002                            51.29776
2003 53.41565 44.44204 42.03675 39.54928
2004 38.30461 37.43384 36.92036 36.59114
2005 36.38872 36.26166 36.18271 36.13341

$se
         Qtr1     Qtr2     Qtr3     Qtr4
2002                            38.17185
2003 40.00510 41.52367 41.92702 42.11442
2004 42.18078 42.20773 42.21797 42.22200
2005 42.22355 42.22416 42.22439 42.22449

> plot(gnpgr,xlim=c(1945, 2010))
> lines(gnp.pred$pred, col="red")
> lines(gnp.pred$pred+2*gnp.pred$se, col="red", lty=3)
> lines(gnp.pred$pred-2*gnp.pred$se, col="red", lty=3)
```

Forecasting