ARMA Models

Regression: $Y_i = \beta X_i + \epsilon_i$, where ϵ_i is **white noise**

White Noise:

- independent normals with common variance
- is basic building block of time series

AutoRegression: $X_t = \phi X_{t-1} + \epsilon_t$ (ϵ_t is **white noise**)

Moving Average: $\epsilon_t = W_t + \theta W_{t-1}$ (W_t is **white noise**)

ARMA: $X_t = \phi X_{t-1} + W_t + \theta W_{t-1}$

Stationary Time Series

A time series is stationary when it is "stable", meaning:

- the mean is constant over time (no trend)
- the correlation structure remains constant over time

Dealing with Trend and Heteroscedasticity

Here, we will coerce nonstationary data to stationarity by calculating the return or growth rate as follows.

Often time series are generated as

$$X_t = (1 + p_t)X_{t-1}$$

meaning that the value of the time series observed at time t equals the value observed at time $t - 1$ and a small percent change p_t at time t.

A simple deterministic example is putting money into a bank with a fixed interest p. In this case, X_t is the value of the account at time period t with an initial deposit of X_0.

Typically, p_t is referred to as the *return* or *growth rate* of a time series, and this process is often stable.

Typically, p_t is referred to as the *return* or *growth rate* of a time series, and this process is often stable.

For reasons that are outside the scope of this course, it can be shown that the growth rate p_t can be approximated by

$$Y_t = \log X_t - \log X_{t-1} \approx p_t.$$

In R, p_t is often calculated as `diff(log(x))` and plotting it can be done in one line `plot(diff(log(x)))`.

Stationary Time Series: ARMA

Wold Decomposition

Wold proved that any stationary time series may be represented as a linear combination of white noise:

$$X_t = W_t + a_1 W_{t-1} + a_2 W_{t-2} + \dots$$

For constants $a_1, a_2, \dots$

Any ARMA model has this form, which means they are suited to modeling time series.

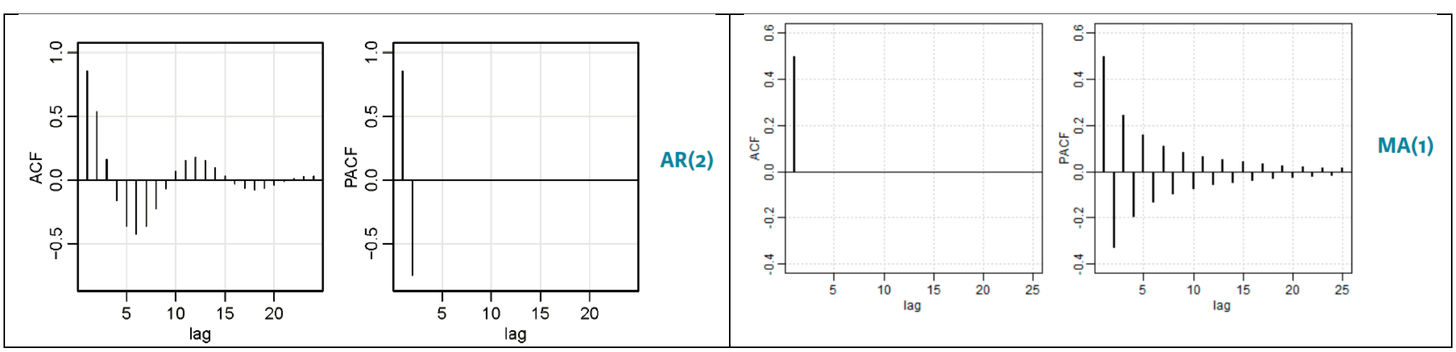
ARMA simulation

MA(1) : $X_t = W_t + 0.9\ W_{t-1}$ => arima.sim(list(order=c(0,0,1), ma=0.9), n=100)

AR(2) : $X_t =$ **0**X_{t-1} - 0.9 $X_{t-2} + W_t$ => arima.sim(list(order=c(2,0,0), ar=c(**0**,-0.9)), n=100)

ACF and PACF

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



Estimation with **astsa:** sarima(x, p, d, q)

Model Choice and Residual Analysis

AIC and BIC

$$\overset{\text{Error}}{average(observed - predicted)^2} \quad + \quad \overset{\text{Number of Parameters}}{k(p+q)}$$

AIC and BIC measure the error and penalize (differently) for adding parameters

For example, AIC has $k = 2$ and BIC has $k = log(n)$

Goal: find the model with the smallest AIC or BIC

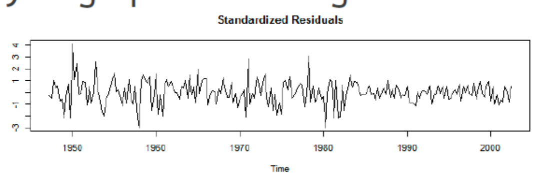
If AIC selects one models and BIC selects the other one, go for the simpler model, the one with the minimum value of p, d, q

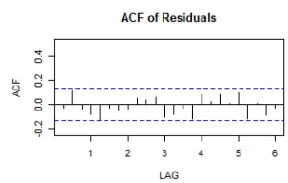
Residual Analysis

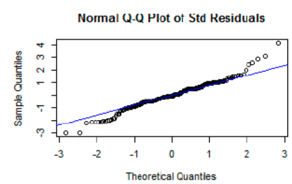
Make sure residual are white Gaussian noise

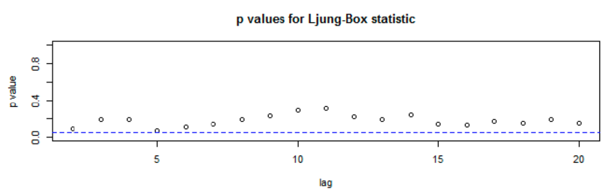
sarima() includes residual analysis graphic showing:

1. Standardized residuals
2. Sample ACF of residuals
3. Normal Q-Q plot
4. Q-statistic p-values





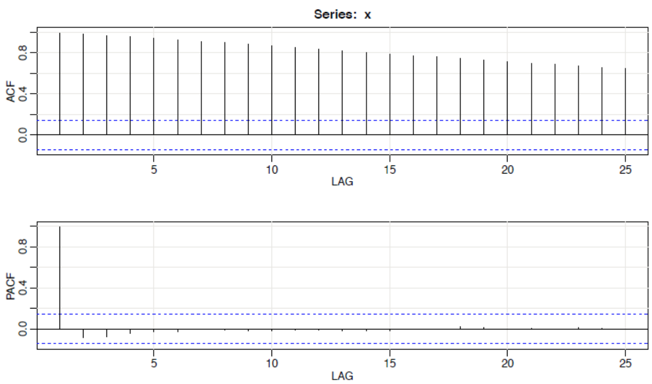




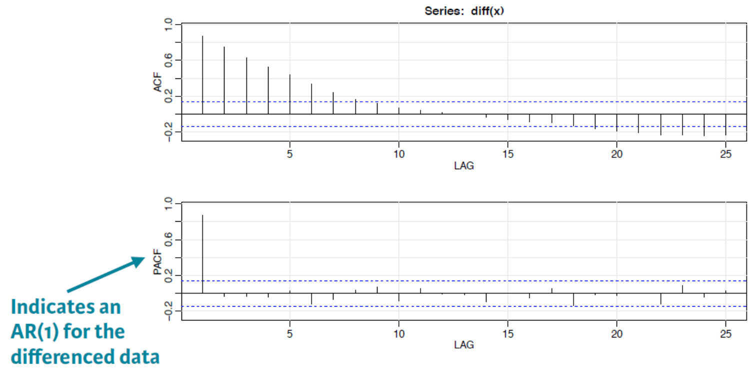
ARIMA

Identifying ARIMA: a time series exhibits ARIMA behavior if the ***differenced data*** has ARMA behavior.

```
> x <- arima.sim(list(order = c(1, 1, 0),  ar = .9), n = 200)
> acf2(x)
```




```
> x <- arima.sim(list(order = c(1, 1, 0),  ar = .9), n = 200)
> acf2(diff(x))
```



Forecasting ARIMA

sarima.for(x, n.ahead, p, d, q)

Pure Seasonal Models

Consider pure seasonal models such as an SAR($P = 1$)$_{S=12}$

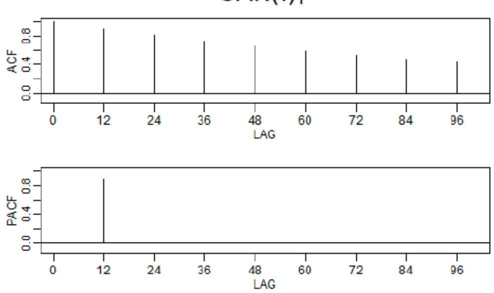
$$X_t = \Phi X_{t-12} + W_t$$

ACF and PACF

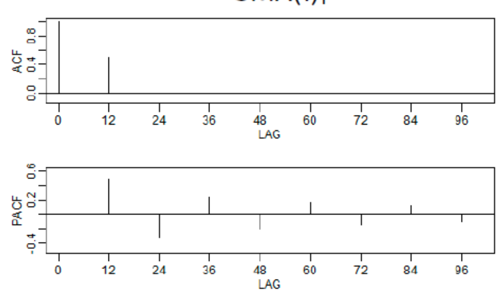
	SAR(P)$_S$	SMA(Q)$_S$	SARMA(P, Q)$_S$
ACF*	Tails off	Cuts off lag QS	Tails off
PACF*	Cuts off lag PS	Tails off	Tails off

* The values at the nonseasonal lags are zero

SAR(1)$_1$



SMA(1)$_1$



SAR(P)$_S$, **SMA(Q)$_S$**, or **SARMA(P,Q)$_S$** for the pure seasonal AR, MA or ARMA with seasonal period S

Fitting a pure seasonal model

To get a feeling of how pure seasonal models work, it is best to consider simulated data. We generated 250 observations from a pure seasonal model given by

$$X_t = .9X_{t-12} + W_t + .5W_{t-12} \,,$$

which we would denote as a SARMA($P = 1$, $Q = 1$)$_{S=12}$. Three years of data and the model ACF and PACF are plotted for you.

```
sarima(x, p = 0, d = 0, q = 0, P = 1, D = 0, Q = 1, S = 12)
```

Mixed Seasonal Models

Mixed model: SARIMA(p, d, q) x (P, D, Q)$_s$ model

Consider a SARIMA(0, 0, 1) x (1, 0, 0)$_{12}$ model

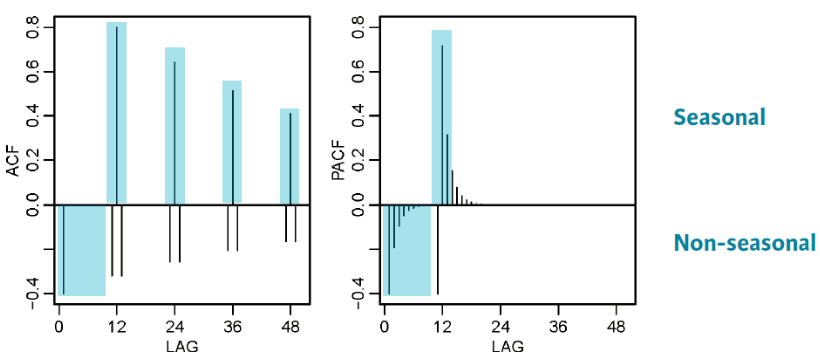
$$X_t = \Phi X_{t-12} + W_t + \theta W_{t-1}$$

SAR(1): Value this month is related to last year's value X_{t-12}

MA(1): This month's value related to last month's shock W_{t-1}

- The ACF and PACF for this mixed model:

$$X_t = .8X_{t-12} + W_t - .5W_{t-1}$$



Seasonal: ACF tails off and PACF cuts off, suggesting a SAR model.

Non-seasonal: ACF cuts off and PACF tails off, suggesting a MA model.

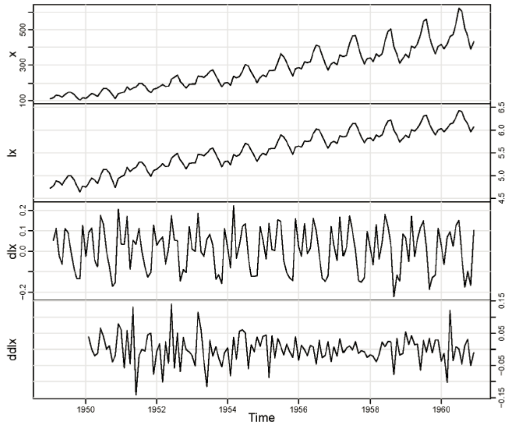
Example: Air Passenger

x: `AirPassengers`

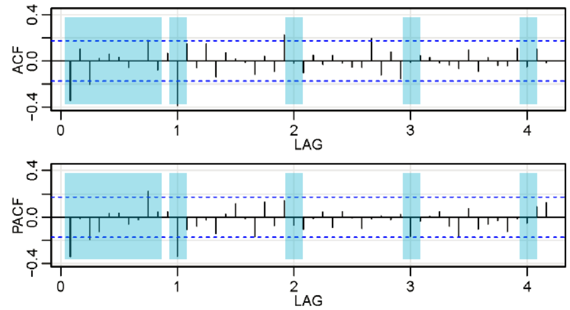
lx: `log(x)`

dlx: `diff(lx)`

ddlx: `diff(dlx, 12)`



dlx has seasonal component
ddlx is stationary



Seasonal: ACF cutting off at lag 1s (s = 12); PACF tailing off at lags 1s, 2s, 3s...

Non-Seasonal: ACF and PACF both tailing off

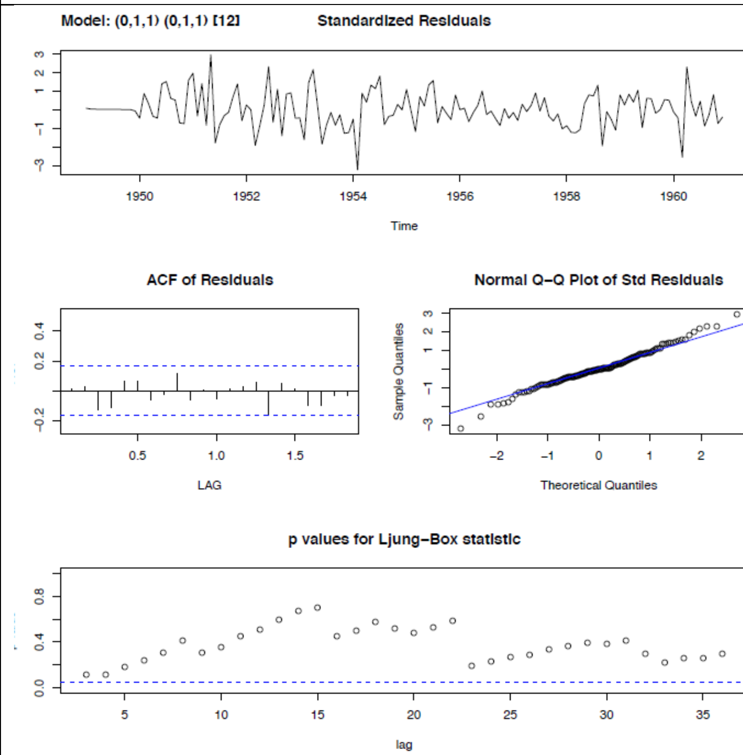
```
> airpass_fit1 <- sarima(log(AirPassengers), p = 1,
        d = 1, q = 1, P = 0,
        D = 1, Q = 1, S = 12)
> airpass_fit1$ttable

       Estimate     SE t.value p.value
ar1      0.1960 0.2475  0.7921  0.4296
ma1     -0.5784 0.2132 -2.7127  0.0075
sma1    -0.5643 0.0747 -7.5544  0.0000

> airpass_fit2 <- sarima(log(AirPassengers),
        0, 1, 1, 0, 1, 1, 12)
> airpass_fit2$ttable

       Estimate     SE t.value p.value
ma1     -0.4018 0.0896 -4.4825       0
sma1    -0.5569 0.0731 -7.6190       0
```

ar1 isn't significant, so remove it.



Also perform **residual analysis** to check if they are white noise.