ARIMA

Simulation and Fitting

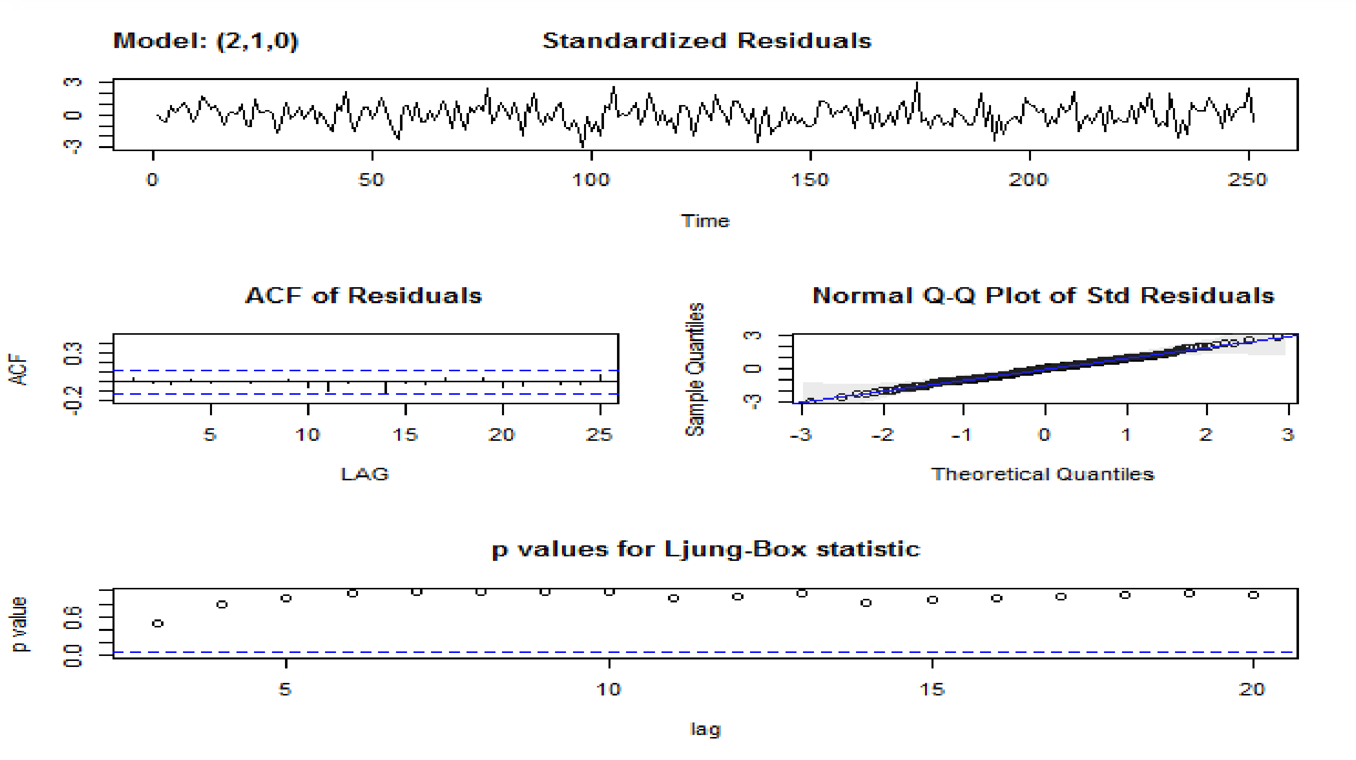
Identifying ARIMA Model

- We can try differencing and then fitting the ARMA model
- If it fits after one or more differencing, then it can be ARIMA model

Example: Simulated ARIMA

```r
library(astsa)
# Simulating ARIMA(2,1,0) model with drift 2
x <- arima.sim(model = list(order = c(2, 1, 0),
                     ar=c(1.5,-0.75)), n = 250, mean = 2)

x.fit <- sarima(x, p = 2, d = 1, q = 0)
```



Sane's STATS Academy of Statistics

Example: Global mean land-ocean temperature deviations

```
> globtemp.fit <- sarima(globtemp,2,1,0,details = F)
> globtemp.fit$ttable
         Estimate     SE t.value p.value
ar1       -0.3027 0.0834 -3.6298  0.0004
ar2       -0.2678 0.0831 -3.2239  0.0016
constant   0.0073 0.0056  1.3047  0.1942
>
> globtemp.fit <- sarima(globtemp,3,1,0,details = F)
> globtemp.fit$ttable
         Estimate     SE t.value p.value
ar1       -0.3669 0.0842 -4.3592  0.0000
ar2       -0.3421 0.0849 -4.0319  0.0001
ar3       -0.2363 0.0838 -2.8186  0.0056
constant   0.0071 0.0044  1.6168  0.1083
>
> globtemp.fit <- sarima(globtemp,3,1,1,details = F)
> globtemp.fit$ttable
         Estimate     SE t.value p.value
ar1        0.2162 0.3897  0.5549  0.5799
ar2       -0.1546 0.1553 -0.9956  0.3213
ar3       -0.0344 0.1787 -0.1926  0.8475
ma1       -0.6007 0.3838 -1.5653  0.1199
constant   0.0071 0.0035  2.0114  0.0463
```

Fit

Overfit

ARIMA – Automated Option

Package **forecast**

Package **forecast**

- Package forecast by Rob Hyndman provides an automated function auto.arima() which identifies the optimal ARIMA fit
- It also provides a function named checkresiduals() which analyzes the residuals

Function `auto.arima()`

Syntax :

auto.arima(y, …)

where y : a univariate time series

- All the other parameters like p, d, q etc. are taken based on the automated algorithms programed with this function
- Return type of this function is object of class ARIMA

Function `checkresiduals()`

Syntax :

checkresiduals(ojbTS, …)

Where objTS : forecast object created from any time series modelling function

- Displays the result of Ljung-Box test which tests for white noise
- Small p-value of Ljung-Box test indicates the data probably are not white noise. In other words, larger the p-value there is more possibility of being a white noise
- Function also produces three diagnostic plots namely, line graph, ACF and histogram for examining the residuals

Example 1: Global mean land-ocean temperature deviations

```
> library(forecast)
> fit <- auto.arima(globtemp)
> checkresiduals(fit)

        Ljung-Box test

data:  residuals
Q* = 8.7308, df = 6, p-value = 0.1893

Model df: 4.   Total lags used: 10
```

- P-value > 0.05 indicates that it is a white noise most probably

```
>      fit
Series: globtemp
ARIMA(3,1,0) with drift

Coefficients:
          ar1      ar2      ar3   drift
      -0.3669  -0.3421  -0.2363  0.0071
s.e.   0.0842   0.0849   0.0838  0.0044

sigma^2 estimated as 0.01003:  log likelihood=120.93
AIC=-231.85   AICc=-231.39   BIC=-217.32
```

Example 2: International visitors to Australia

```
> fit <- auto.arima(austa)
> checkresiduals(fit)

        Ljung-Box test

data:  residuals
Q* = 3.2552, df = 8, p-value = 0.9173

Model df: 2.   Total lags used: 10
```

- P-value > 0.05 indicates that it is a white noise most probably

```
> fit
Series: austa
ARIMA(0,1,1) with drift

Coefficients:
         ma1   drift
      0.3006  0.1735
s.e.  0.1647  0.0390

sigma^2 estimated as 0.03376:  log likelihood=10.62
AIC=-15.24   AICc=-14.46   BIC=-10.57
```

Seasonal ARIMA

Fitting

Seasonal ARIMA

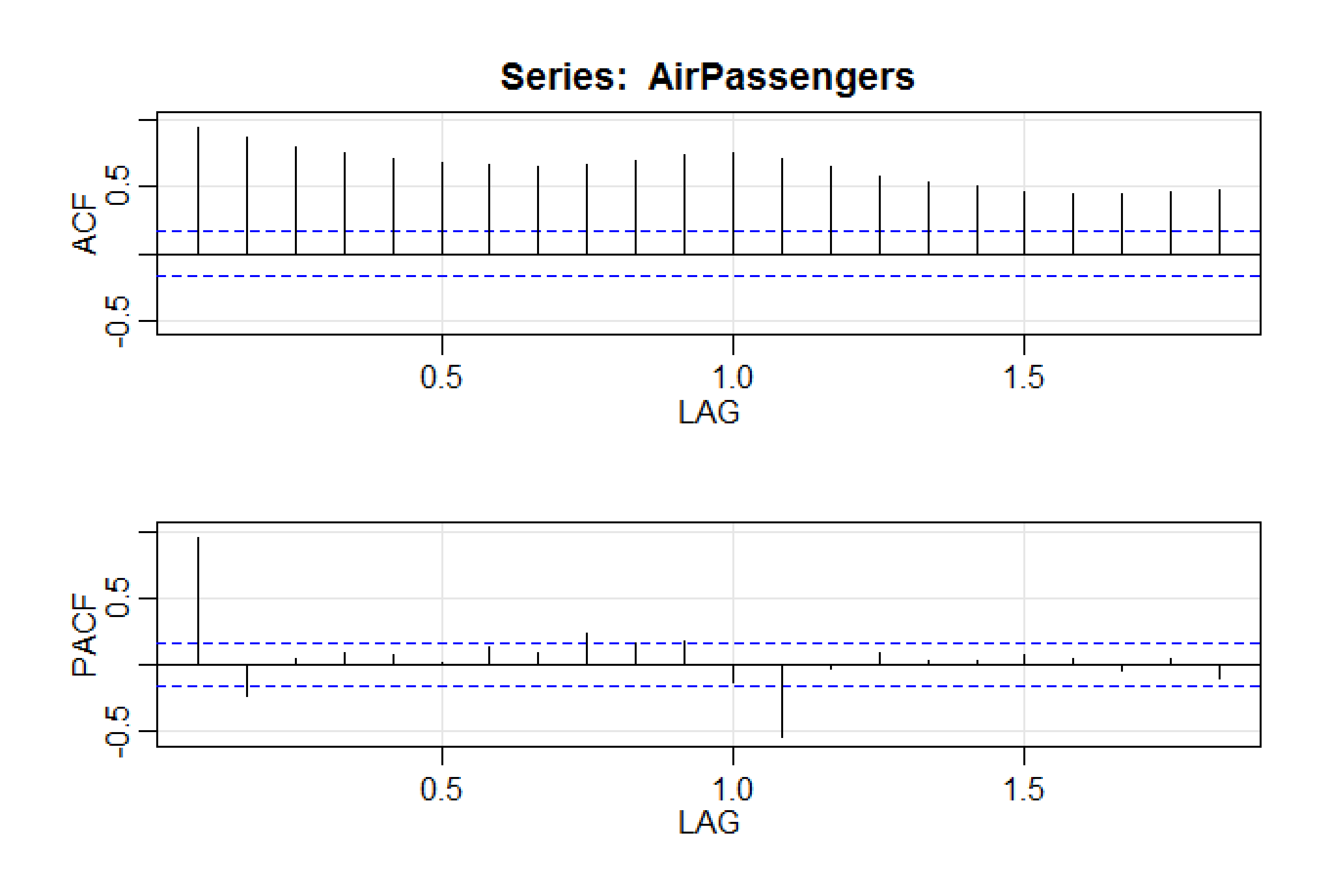
- It can be sometimes observed that the current values are influenced by some seasonal lag *s*
- There can some regular lags with which a pattern can be observed
- For seasonal ARIMA, we require four more parameters,
    - P : Seasonal AR order
    - D : Seasonal Difference
    - Q : Seasonal MA order
    - S : Seasonal period

Thumb Rules for Identifying the Seasonal Models

- We can guess the SARMA models also with the help of ACF and PACF graphs
- Consider P = order of SAR , Q = order of SMA , S : Seasonal Period in the table below
- Also note that
  - “Tails off after g” means Starting to decrease at g
  - “Cuts off after g” means Disappears or becomes very small at g

Graph	$AR(P)_S$	MA(Q)S	$ARMA(P,Q)_S$
ACF	Tails off after seasonal lags	Cuts off after lag QS	Tails off after seasonal lags
PACF	Cuts off after lag PS	Tails off after seasonal lags	Tails off after seasonal lags

Example: AirPassengers



Fitting the Model to AirPassengers

```
> air.fit <- auto.arima(AirPassengers)
> air.fit
Series: AirPassengers
ARIMA(0,1,1)(0,1,0)[12]

Coefficients:
          ma1
      -0.3184
s.e.   0.0877

sigma^2 estimated as 138.3:  log likelihood=-508.32
AIC=1020.64   AICc=1020.73   BIC=1026.39
```

- Model identified here is ARIMA(0,1,1)(P=0,D=1,Q=0)S=12

Forecasting

Generating Forecasts

- Forecasts can be calculated based on the built model using various functions. But we are going to cover forecasting using
  - Function `sarima.for()` from package **astsa**
  - Function `forecast()` from package **forecast**

Function `sarima.for()` from package **astsa**

Syntax :

sarima.for(xdata, n.ahead, p, d, q, P = 0, D = 0, Q = 0, S = -1,…)

Where

xdata : Univariate time series

n.head : forecast horizon

p : AR order

d : difference order

q : MA order

P : Seasonal AR order

D : Seasonal Difference order

Q : Seasonal MA order

S : Seasonal period cycle