Lecture 8: Introduction to Time Series
Statistics 509 – Winter 2022
Reference: Chapter 12 of Ruppert/Matteson

Brian Thelen
258 West Hall
bjthelen@umich.edu

Overview of Lecture 8

- Introduction
- Standard (stationary) time series models
  - AR, MA, ARMA, and ARIMA
- Estimation and Model Selection
- Examples
- Bringing in deterministic trends (e.g., seasonal patterns)

Time Series: Introduction

Background. A significant portion of financial data corresponds to sequence(s) of

$$X_1, X_2, \ldots, X_n$$

representing prices/contracts/yields at times

$$t_1 < t_2 < \cdots < t_n$$

Objective (of time series statistical analysis)
Develop probability/statistical machinery that can

- accurately/rigorously model real-life time series
- provide a framework for model estimation/selection
- provide a model for prediction
    - including quantification of prediction errors

First Time Series Models: Overview

Model. First model time series taken at regular intervals of length $\triangle > 0$, i.e.,

$$\begin{aligned} t_2 &= 2\triangle \\ t_3 &= 3\triangle \\ \vdots & \quad \vdots \\ t_n &= n\triangle \end{aligned}$$

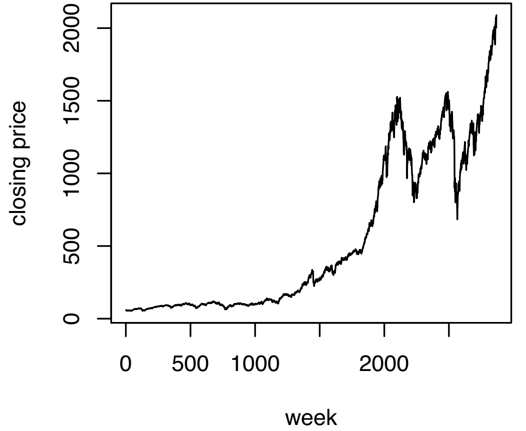
Definition. The (time) sampling interval $\triangle$ can be thought of as the (constant) sampling density parameter.

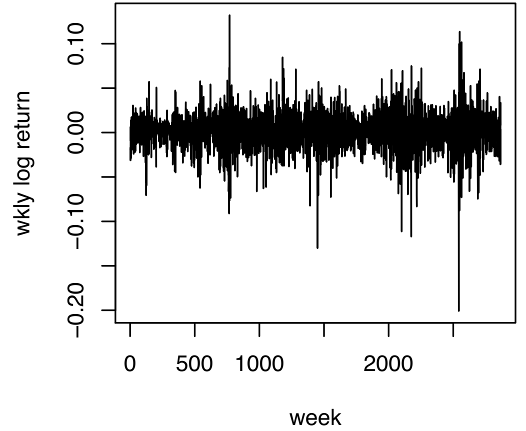
Examples. Stock price

- Weekly
- Day
- Hourly intervals
- 5-minute intervals
- 1-minute intervals

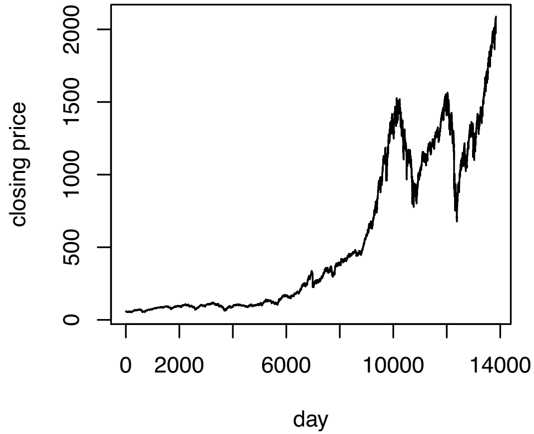
Question. Which of these is accurately-modeled as a time series with sampling at a regular interval?

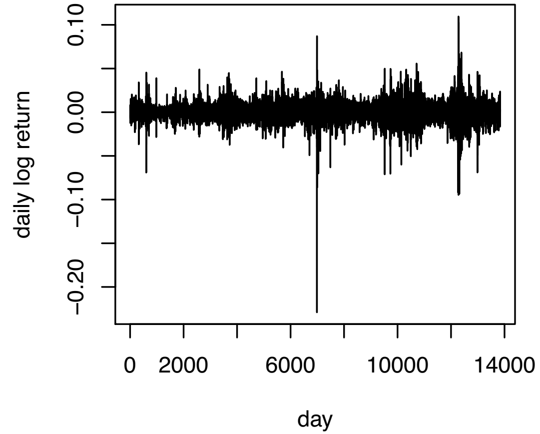
Example: SP500 Closing Prices





price is not stationary but return seems is





Covariances for Random Vectors

Recall. Suppose $\mathbf{X}$ and $\mathbf{Y}$ are a p-dimensional and q-dimensional random vectors, respectively. Then the covariance matrices

$$(p \times p) \;\; \mathbf{\Sigma_X} \quad \text{with} \quad [\mathbf{\Sigma_X}]_{mn} = Cov(X_m, X_n)$$

$$(p \times q) \;\; \mathbf{\Sigma_{X,Y}} \quad \text{with} \quad [\mathbf{\Sigma_{X,Y}}]_{mn} = Cov(X_m, Y_n)$$

$$(q \times p) \;\; \mathbf{\Sigma_{Y,X}} \quad \text{with} \quad \mathbf{\Sigma_{Y,X}} = \Sigma_{XY}^{T}$$

Second-Order Time Series: Statistical Moments

Definition. We say a time series $X_1, X_2, \ldots, X_n, \ldots$ is second-order if the means and variances of all rvs are finite.

Definition For second-order time series $X_1, X_2, \ldots, X_n, \ldots$ can define the mean function, variance function, auto-covariance function, and auto-correlation function as

$$\begin{aligned} \mu(n) &= E(X_n) \\ \text{Var}(n) &= \text{Var}(X_n) \\ \Sigma(m,n) &= \text{Cov}(X_m, X_n) \\ \rho(m,n) &= \text{Corr}(X_m, X_n) \end{aligned}$$

Remark. Note that

$$\Sigma(n,n) = \text{Var}(n)$$

$$\rho(n,n) = 1$$

$$\rho(m,n) = \frac{\Sigma(m,n)}{\sqrt{\text{Var}(m)\,\text{Var}(n)}}$$

Example. Suppose have an independent sequence of rvs $W_1, W_2, \dots$ with a common mean of μ and common variance of σ^2, and suppose that

$$\begin{array}{rcl} X_1 & = & W_1 \\ X_2 & = & X_1 + W_2 \\ X_3 & = & X_2 + W_3 \\ & \vdots & \end{array}$$

$X_n = \sum W_i$

Derive the mean, variance, auto-covariance, and auto-correlation functions of

(a) the time series $W_1, W_2, \dots$

(b) the time series $X_1, X_2, \dots$

Remark. The time series $X_1, X_2, \dots$ is referred to as a random walk –

Answer.

Answer.

(a) $\mu_n = E[W_n] = \mu$

$Var[W_n] = \sigma^2$

$$\Sigma = \begin{cases} \sigma^2 & m=n \\ 0 & m\neq n \end{cases}$$

$$P = \begin{cases} 1 & m=n \\ 0 & m\neq n \end{cases}$$

(b) $\mu_x(n) = n\mu$

$Var_x(n) = n\sigma^2$

$$\Sigma(m,n) = Cov\left[\sum_{1}^{m} W_i, \sum_{1}^{m} W_i + \sum_{m+1}^{n} W_i\right)$$

$$= \min(m,n)\,\sigma^2$$

$$\rho_x(m,n) = \frac{\min(m,n)\,\sigma^2}{mn\sigma^2} = \frac{\min(m,n)}{mn}$$

Answer.

Stationarity - Strict

Definition. A time series $X_1, X_2, \ldots, X_n, \ldots$ is said to be stationary if for any k and n, we have that the multivariate random vector

$$(X_1, X_2, \ldots, X_n)$$

has same joint distribution as 任意n个都一样

$$(X_{1+k}, X_{2+k}, \ldots, X_{n+k})$$

Stationarity: Statistical Properties. Suppose $X_1, X_2, \ldots, X_n, \ldots$ is second-order stationary process. Then

$$X_1 \quad \text{has the same distribution as} \quad X_n$$

$$(X_1, X_n) \quad \text{has the same distribution as} \quad (X_{1+k}, X_{n+k})$$

Thus

$$\mu_X(n) = \text{E(X) for all n}$$

$$\cdot$$

$$\Sigma_X(m, n) = \text{Cov(X\_k, X\_\{n-m+k\}) for all k}$$

Stationarity - Weak

Definition A second-order process $X_1, X_2, \ldots$ is said to be weakly stationary if the mean and covariance functions have same form as a strictly stationary process, i.e.,

$$\begin{aligned} \mu_X(n) &= \mathrm{E(X)} && \text{for all } n \\ \Sigma_X(m,n) &= \gamma(n-m) && \text{for all } m,n \end{aligned}$$

ACF是对称的

where we are (slightly) abusing notation in the last equation.

Terminology. Using the term stationary typically will refer to "weakly stationary"

Stationary Time Series: Mean/Covariance Estimation

Estimation. For a second-order stationary time series $X_1, X_2, X_3, \ldots$, it is natural to estimate the mean

$$\hat{\mu} = \frac{1}{N}\sum_{1}^{N} X_i$$

and the autocovariances $\gamma(0)$ and $\gamma(1)$ are estimated as

$$\hat{\gamma}(0) = \frac{1}{N}\sum_{1}^{N}(X_i - \hat{\mu})^2$$

$$\hat{\gamma}(1) = \frac{1}{N}\sum_{1}^{N-1}(X_i - \hat{\mu})(X_{i+1} - \hat{\mu}) \quad .$$

- For a general lag h value, 除以N可以让covariance正定

$$\hat{\gamma}(h) = \frac{1}{N}\sum_{1}^{N-h}(X_i - \hat{\mu})(X_{i+h} - \hat{\mu})$$

- Estimate correlation $\rho(h)$ via $\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}$.

Sample autocovariance/autocorrelation functions

Stationary Time Series: Mean/Covariance Est - Comments

Remarks.

- Need to keep $n - |h|$ of a reasonable value
- Some use divisor of $n - |h|$ instead of n - for larger n does not matter
- For $\mathbf{X}_m = (X_1, X_2, \ldots, X_m)$,

n越大准确度越高

$$\hat{\mathbf{\Sigma}}_{\mathbf{X}_m} = \begin{bmatrix} \hat{\gamma}(0) & \hat{\gamma}(1) & \hat{\gamma}(2) & \cdots & \hat{\gamma}(m-1) \\ \hat{\gamma}(1) & \hat{\gamma}(0) & & \ddots & \hat{\gamma}(m-2) \\ \cdots & & \hat{\gamma}(0) & & \cdots \\ \hat{\gamma}(m-1) & \hat{\gamma}(m-2) & \cdots & & \hat{\gamma}(0) \end{bmatrix}$$

constant along all diagonals and sub diagonals

Toeplitz matrix

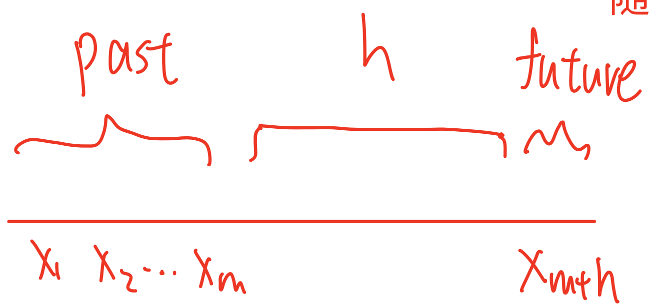
Stationary Time Series: Ergodicity

Question. What are sufficient conditions ensuring some level of accuracy in the mean and covariance estimates?

Answer. One such condition is **ergodicity** want consistent estimate

Definition We way that a time series is ergodic if all events in the past and events in the future become approximately independent when there is a large enough separation.

Picture 随着h增大,future和past无关



Remark. Implication of ergodicity is that long term averages converge to their expected value, e.g.,

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \to E[X_1] \quad , \quad \hat{\gamma}(j) \to r(j)$$

 用ts先把数据转化为时间序列对象

Commands/Objects for Time Series in R

The function ts creates a time-series object

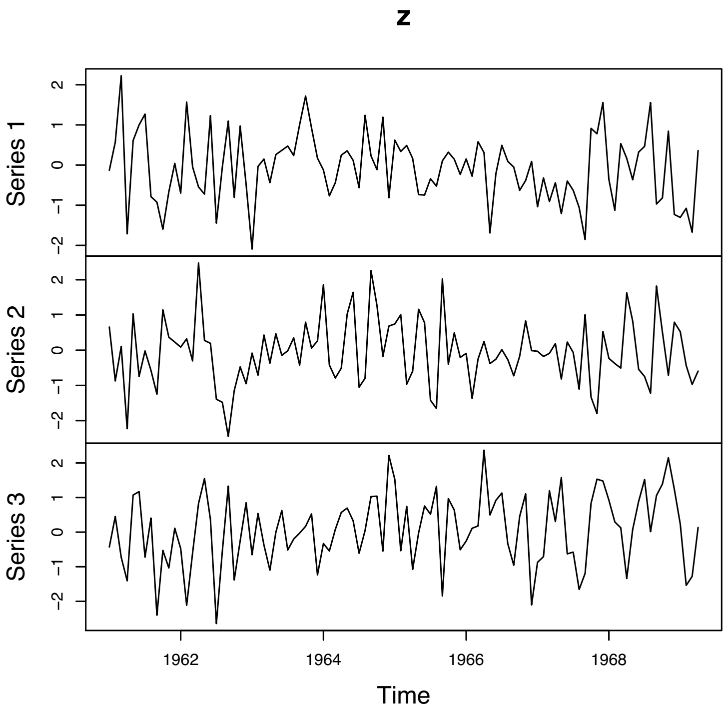
```
Usage
ts(data = NA, start = 1, end = numeric(0), frequency = 1,
   deltat = 1, ts.eps = getOption("ts.eps"), class = , names = )

Arguments
data -- a numeric vector or matrix of the observed time-series values.
A data frame will be coerced to a numeric matrix via data.matrix.
Start the time of the first observation. Either a single number or a
vector of two integers, which specify a natural time unit and a
(1-based) number of samples into the time unit.

frequency -- the number of observations per unit of time.

deltat -- the fraction of the sampling period between successive
observations; e.g., 1/12 for monthly data. Only one of frequency
or deltat should be provided

class -- class to be given to the result, or none if NULL or "none".
The default is "ts" for a single series, c("mts", "ts") for multiple
series. names a character vector of names for the series in a multiple
series: defaults to the colnames of data, or Series 1, Series 2, ....
```

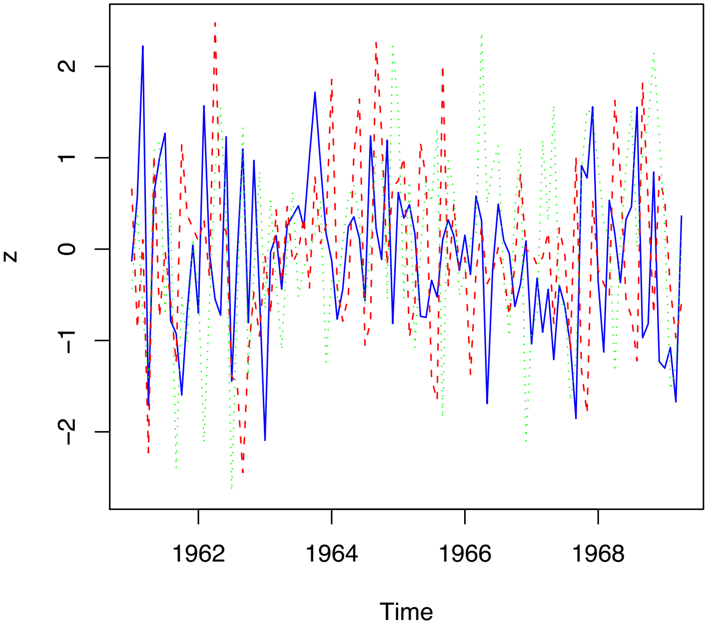
R-Commands for Time Series - Example

Remark. Below are sample commands for generating time series object and plotting variables within this object.

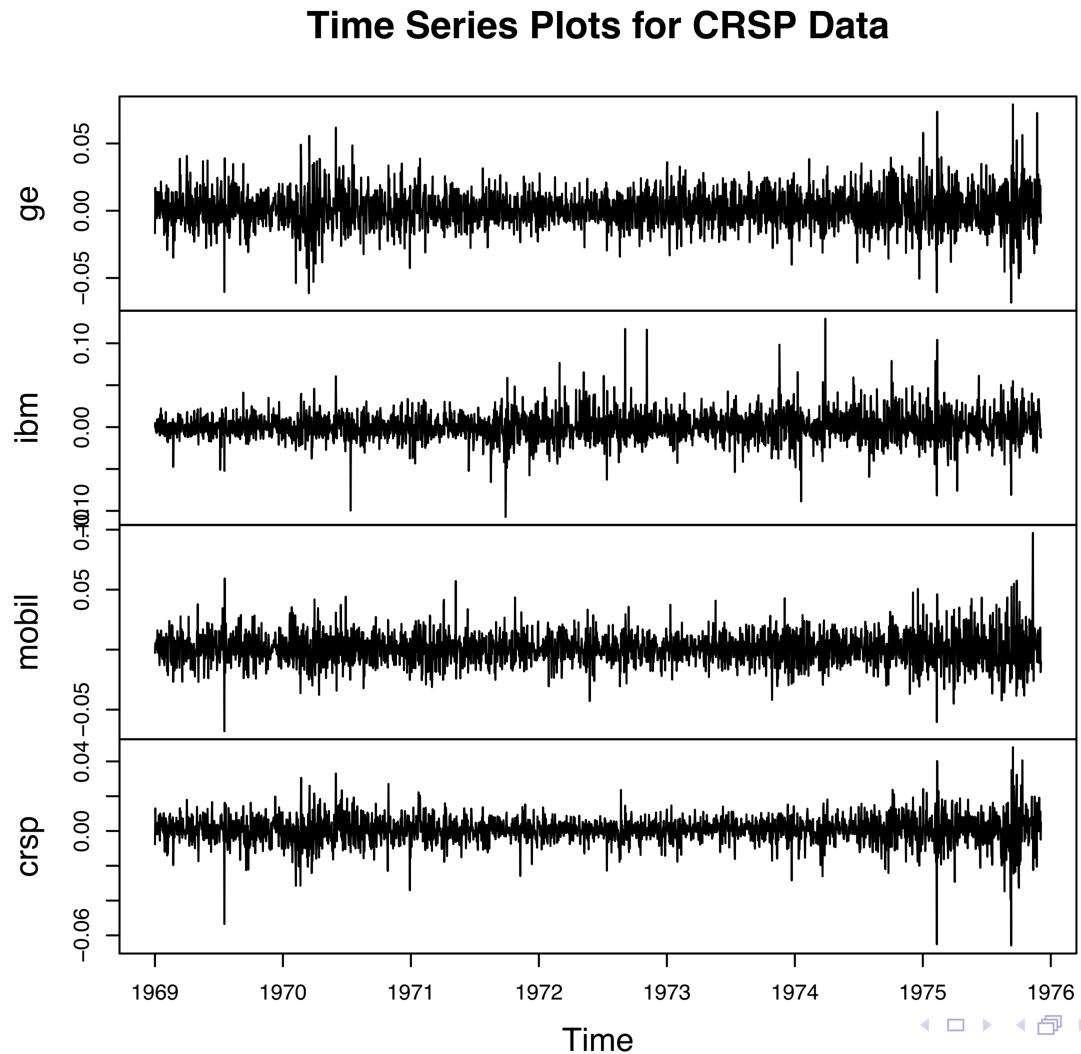
```
> z <- ts(matrix(rnorm(300), 100, 3), start=c(1961, 1), frequency=12)
> class(z)
[1] "mts" "ts"
> windows()
> plot(z)

> windows()
> plot(z, plot.type="single", lty=1:3,col=c('blue','red','green'))

> z
            Series 1    Series 2    Series 3
Jan 1961 -0.12878292  0.65764810 -0.42841724
Feb 1961  0.56953797 -0.87244583  0.45189490
Mar 1961  2.22384399  0.10234520 -0.72916674
Apr 1961 -1.71256530 -2.22925944 -1.40732626
May 1961  0.61079122  1.03242131  1.07167504
Jun 1961  0.99965640 -0.74642505  1.17053742
Jul 1961  1.26909493 -0.02074828 -0.72352484
```

100行3列的时序数据


z
Series 1
Series 2
Series 3
Time
1962
1964
1966
1968


z
Time
1962
1964
1966
1968

Time Series: Another Example



Example: R-Commands for Time Series Plotting

```
> library(Ecdat)
> data(CRSPday)
> CRSPday[c(1:4),]
     year month day        ge       ibm     mobil      crsp
[1,] 1989     1   3 -0.016760  0.000000 -0.002747 -0.007619
[2,] 1989     1   4  0.017045  0.005128  0.005510  0.013016
[3,] 1989     1   5 -0.002793 -0.002041  0.005479  0.002815
[4,] 1989     1   6  0.000000 -0.006135  0.002725  0.003064
> is.ts(CRSPday)
[1] TRUE
> windows()
> plot(CRSPday[,c(4:7)],main='Time Series Plots for CRSP Data')
```

ACF Plots and Testing Autocorrelations

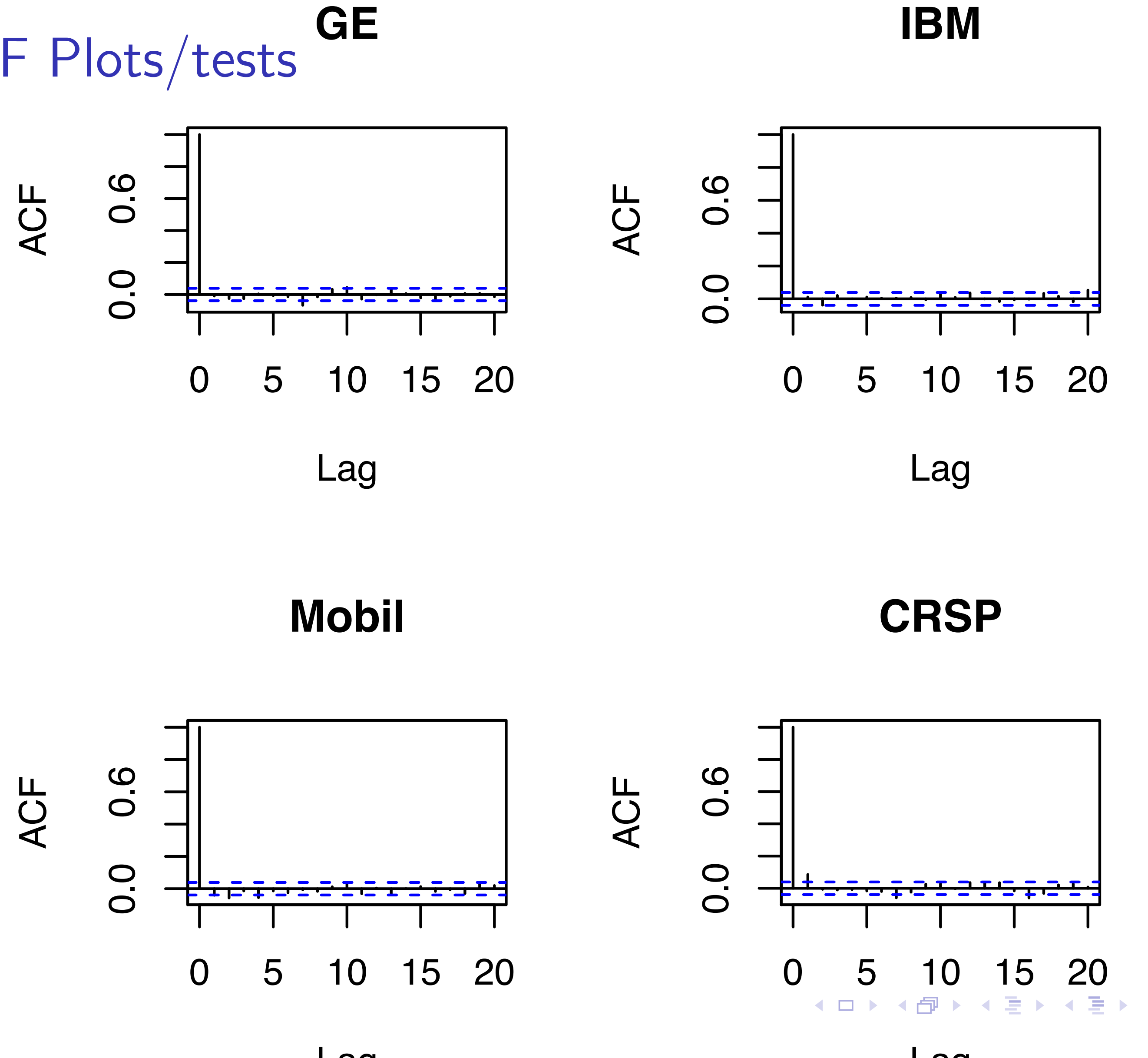
The data is n

- Useful to do plots of autocorrelation function – a type of EDA
  - Looking for patterns/seasonality
  - Looking for if there is significant autocorrelation or not
  - Use R-function acf()
- Use R-function of Box.test to test hypothesis of

$$H_o : \gamma(i) = 0 \text{ for } i = 1 \cdots N \qquad H_1 : \gamma(i) \neq 0 \text{ for some } i$$

  - Simultaneous tests - Box-Pierce and Box-Ljung

ACF Plots/tests



Autocorrelation Function - R code

```
> par(mfrow=c(2,2))
> acf(as.vector(CRSPday[,4]),lag=20,main='GE')
> acf(as.vector(CRSPday[,5]),lag=20,main='IBM')
> acf(as.vector(CRSPday[,6]),lag=20,main='Mobil')
> acf(as.vector(CRSPday[,7]),lag=20,main='CRSP')
```

Box-Ljung Tests on Autocorrelation

```
> Box.test(CRSPday[,4],lag=10,type='Ljung')

data:  CRSPday[, 4]
X-squared = 23.7418, df = 10, p-value = 0.008316
> Box.test(CRSPday[,5],lag=10,type='Ljung')

data:  CRSPday[, 5]
X-squared = 8.0944, df = 10, p-value = 0.6196
```

not significant,因此数据平稳

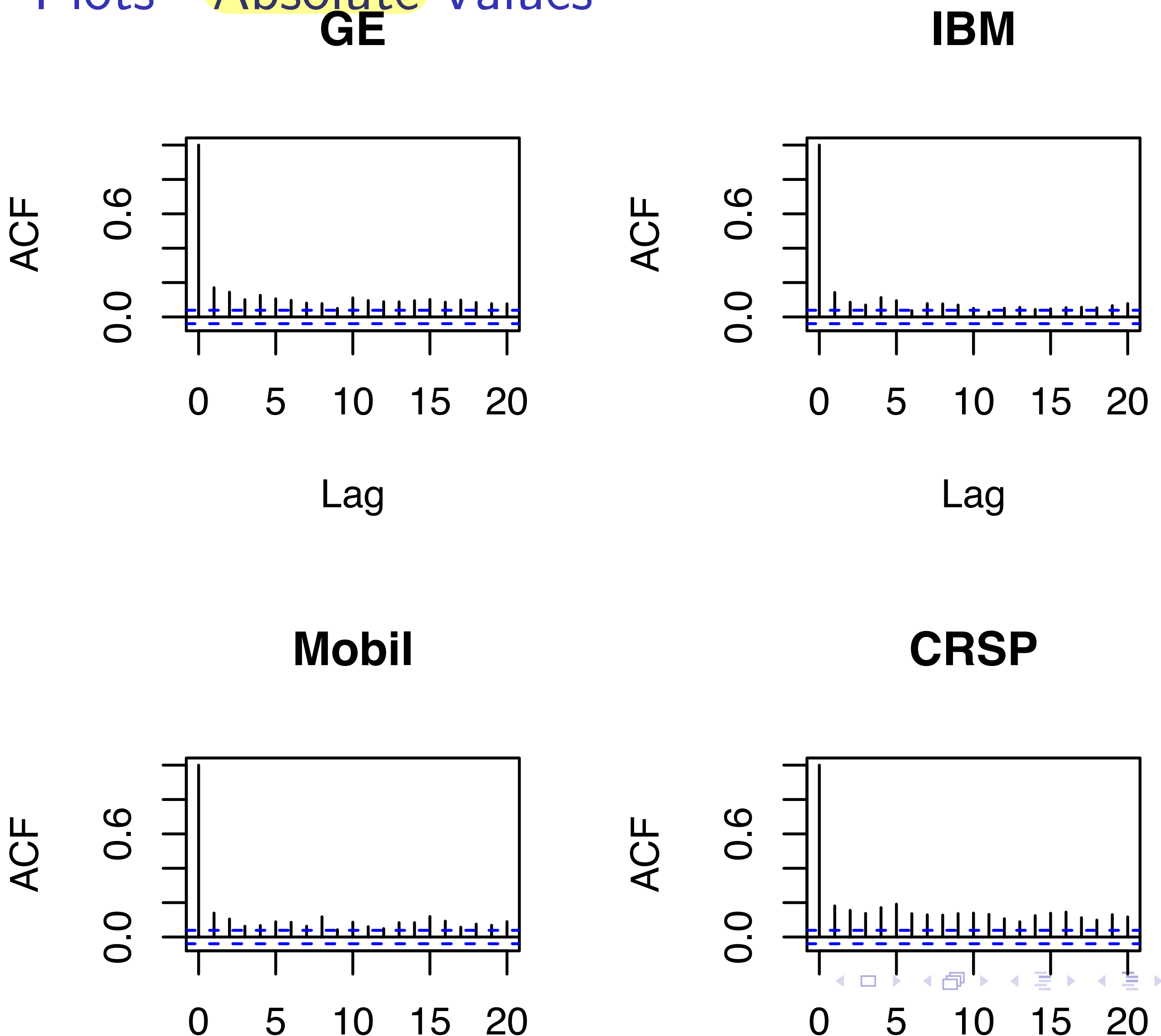
```
> Box.test(CRSPday[,6],lag=10,type='Ljung')

data:  CRSPday[, 6]
X-squared = 26.7899, df = 10, p-value = 0.002811
```

significant

```
> Box.test(CRSPday[,7],lag=10,type='Ljung')

data:  CRSPday[, 7]
X-squared = 34.3991, df = 10, p-value = 0.000158
```

significant

ACF Plots - Absolute Values



Autocorrelation Abs Values - R code

```
> par(mfrow=c(2,2))
> acf(as.vector(abs(CRSPday[,4])),lag=20,main='GE')
> acf(as.vector(abs(CRSPday[,5])),lag=20,main='IBM')
> acf(as.vector(abs(CRSPday[,6])),lag=20,main='Mobil')
> acf(as.vector(abs(CRSPday[,7])),lag=20,main='CRSP')
```

Partial Autocorrelation

Definition For a stationary time series $X_1, X_2, \ldots$, the partial autocorrelation function is given denoted by $\phi_{k,k}$, and it is the correlation between X_1 and X_{k+1} conditioned on all the data in between, i.e., on $X_2, \ldots, X_k$.

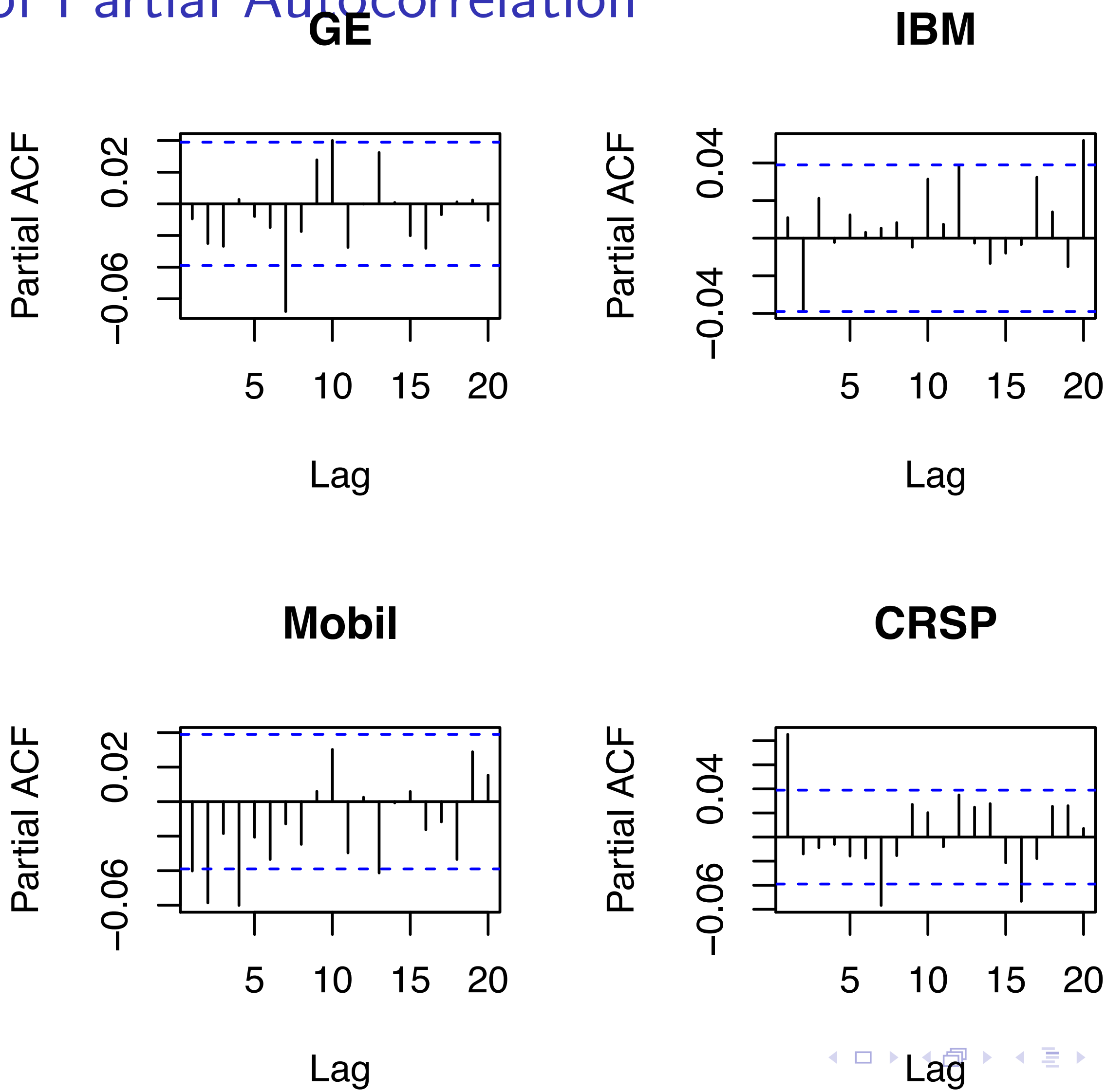
- It can be estimated by doing linear regression analysis of

$$X_m = \phi_{0,k} + \phi_{1,k} \cdot X_{m-1} + \ldots + \phi_{k,k} \cdot X_{m-k} \quad m = k+1, \ldots, n$$

- Useful in identifying order (diagnostic) for identifying order of Autoregressive processes

之前的phi已经去掉了其他X的作用

Plots of Partial Autocorrelation



Partial Autocorrelation - R-code

```
> par(mfrow=c(2,2))
> pacf(as.vector(CRSPday[,4]),lag=20,main='GE')
> pacf(as.vector(CRSPday[,5]),lag=20,main='IBM')
> pacf(as.vector(CRSPday[,6]),lag=20,main='Mobil')
> pacf(as.vector(CRSPday[,7]),lag=20,main='CRSP')
```

Refresher on Geometric Series 等比数列级数

Background. Suppose $|\alpha| < 1$. Then

$$S_n = \sum_{k=0}^{n} \alpha^k$$

Result 1. For n,

$$S_n = \frac{1 - \alpha^{n+1}}{1 - \alpha}$$

Proof. Multiply both sides by $(1 - \alpha)$ and verify that they are equal.

Result 2.

$$\sum_{k=0}^{\infty} \alpha^k = \frac{1}{1 - \alpha}.$$

Proof. Invoke **Result 1** and note that as $n \to \infty$, have $\alpha^{n+1} \to 0$.

Examples for Geometric Series

Examples

$$\sum_{k=0}^{10} \left[\frac{2}{3}\right]^k = \frac{1-\left(\frac{2}{3}\right)^{11}}{1-\frac{2}{3}}$$

$$\sum_{k=0}^{\infty} \left[\frac{2}{3}\right]^k = \frac{1}{1-\frac{2}{3}} = 3$$

Example Problem

Example. Suppose have an independent sequence of rvs $W_1, W_2, \ldots$ with a common mean of μ and common variance of σ^2 and $X_1, X_2, \ldots$ is the associated random walk

weakly

(a) Under what conditions is time series of $W_1, W_2, \ldots$ stationary?

(b) If $\sigma^2 > 0$, is it possible for the random walk $X_1, X_2, \ldots$ to be stationary?

(c) Suppose that W_0 is rv independent of $W_1, W_2, \ldots$ with the same mean μ_o and variance σ_o^2, and suppose

$$\begin{aligned} Y_0 &= W_0 \\ Y_j &= \alpha Y_{j-1} + W_j \qquad\qquad j = 1, 2, \ldots \end{aligned}$$

where α is a real number. Derive the mean and auto-covariance function of this time series. mean and variance

(d) If $|\alpha| < 1$, what does variance of W_0 need be to ensure that $Y_0, Y_1, Y_2, \ldots$ is weakly stationary? What is the final form of the auto-covariance in this case?

Answer. (a) Mean and variance are constants

(b) No, because $Var(X_n) = n\sigma^2$, not constant

(c) $Y_j = \alpha Y_{j-1} + W_j = \alpha(\alpha Y_{j-2} + W_{j-1}) + W_j \cdots$

$$= \sum_{i=0}^{j} \alpha^{j-i} W_i$$

$$E[Y_j] = \sum_{i=0}^{j} \alpha^{j-i} E[W] = \frac{1-\alpha^{j+1}}{1-\alpha}\mu$$

$$Cov(Y_m, Y_n) = Cov(\sum_{i=0}^{m} \alpha^{m-i} W_i, \sum_{i=0}^{n} \alpha^{n-i} W_i)$$

$$= \sum_{i=0}^{m} \sum_{j=0}^{n} \alpha^{m-i} \alpha^{n-j} Cov(W_i, W_j)$$

$$= \sum_{i=0}^{m} \alpha^{m-i} \alpha^{n-i} Var(W_i)$$

$$= \frac{\alpha^{n-m} - \alpha^{n+m}}{\alpha^2 - 1} \sigma^2$$

Answer.

(d) $m \to \infty$ $n-m$ stay fixed, $\alpha^{n+m} = 0$

$$\mu_y(n) \to \frac{\mu}{1-\alpha}$$

$$\Sigma_Y(m,n) \to \frac{\sigma^2 \alpha^{n-m}}{1-\alpha^2}$$

Time series Y is asymptotically converting to stationary process with $\mu_y = \frac{\mu}{1-\alpha}$, $\Sigma = \frac{\sigma^2 \alpha^{n-m}}{1-\alpha^2}$

Answer.

Answer.

Examples: Simulations

Suppose we have

- iid rvs $W_0, W_1, W_2, \ldots$ with mean 0 and std. deviation of σ
- random walk sequence $X_0, X_1, X_2, \ldots$ where

$$X_j = \sum_{i=0}^{j} W_i$$

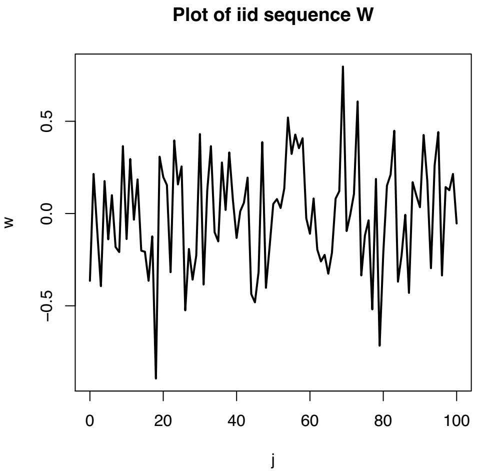
- Autoregressive sequence $Y_0, Y_1, \ldots$ where

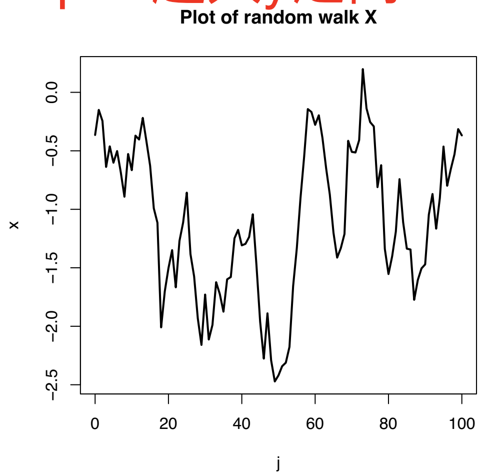
$$Y_j = \alpha Y_{j-1} + W_j \qquad j = 1, 2, \ldots$$

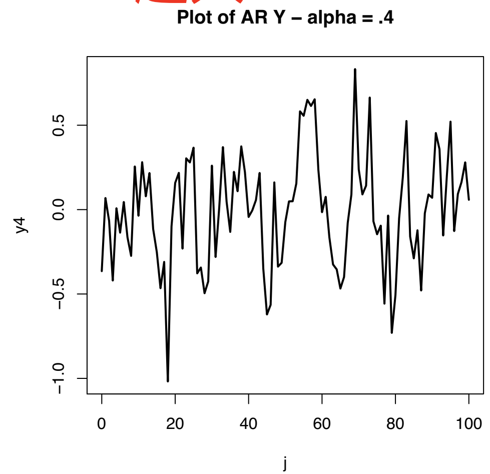
$E(Y) = 0 \qquad r(Y) = \frac{\sigma^2 \alpha^{|h|}}{1-\alpha^2}$

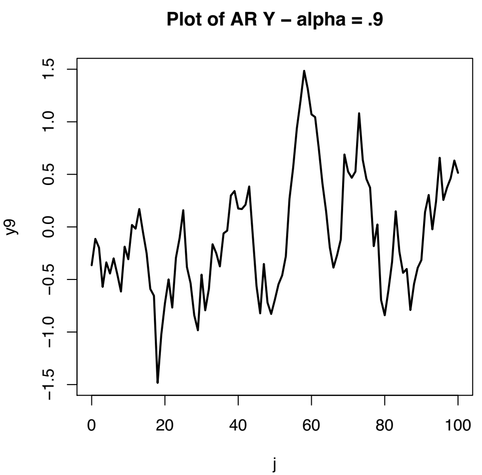
Examples: Simulations Cont'd

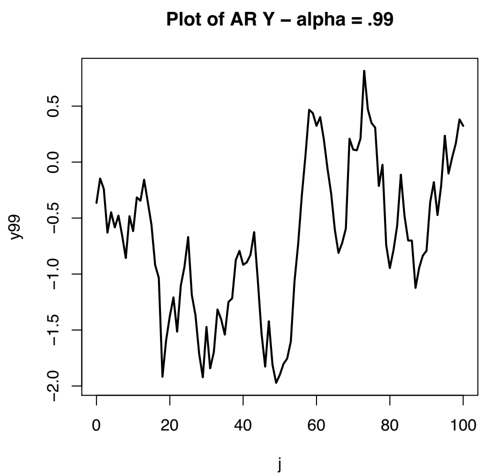
alpha越大y之间correlation越大

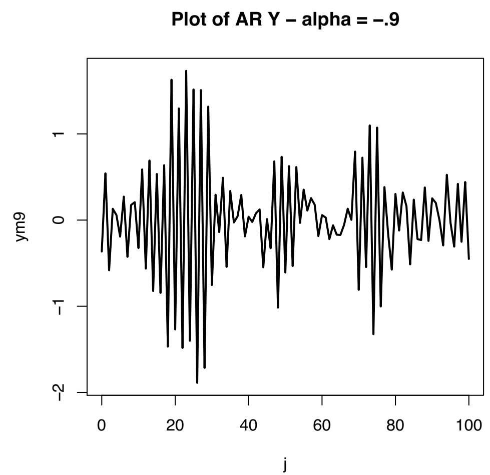












Autoregressive Time Series - AR(1)

Definition. For iid sequence of $\epsilon_0, \epsilon_1, \ldots$ with mean zero and variance σ^2 and parameter α, μ, the time series defined as

$$X_n - \mu = \alpha(X_{n-1} - \mu) + \epsilon_n \qquad n \geq 1$$

corresponds to an AR(1) model with mean μ and coefficient α.

Remarks.

- Note that the AR(1) model can be rewritten as

$$X_n = \mu + \alpha(X_{n-1} - \mu) + \epsilon_n = (1-\alpha)\mu + \alpha X_{n-1} + \epsilon_n .$$

- if an AR(1) process is stationary, then $|\alpha| < 1$, and in this case

ACF以几何速度下降,short memory

Add more term to enlarge the memory

$$\begin{aligned}
\mu &= E(X_n) = \mu \\
\gamma(0) &= \text{Var}(X_n) = \frac{\sigma^2}{1-\alpha^2} \\
\gamma(m) &= \text{Cov}(X_n, X_{n+m}) = \frac{\sigma^2 \alpha^{|m|}}{1-\alpha^2} && \text{for all } m \\
\rho(m) &= \alpha^{|m|} && \text{for all } m
\end{aligned}$$

Pictures and Implications

Time Series Models - Prediction

Remark. One of the primary motivation for time series modeling is that of prediction, i.e., if have a model with known (estimated) paramters, then can do an estimation of a future value.

Goal is to give predicition interval/error and std

The best predictor is the condition expectation on previous value

Example See $X_1, X_2, \ldots, X_n$ and want to predict X_{n+1}

Remark. Will focus on best linear prediction, i.e, the linear estimators which minimize the residual mean-squared prediction error, i.e., want

$$\hat{X}_{n+1} = \alpha_o + \sum_{i=1}^{n} \alpha_i X_i$$

which satisfies that it minimizes

$$E\left[\left(\hat{X}_{n+1} - X_{n+1}\right)^2\right]$$

over all choices of $\alpha_o, \alpha_1, \alpha_2, \ldots, \alpha_n$.

Time Series Models - Prediction – cont'd

Remarks.

- Best (general) predictor, in terms of MSE, is the conditional expectation

使用线性回归模型最小化MSE

$$\hat{X}_{n+1} = E(X_{n+1}|X_1, X_2, \ldots, X_n).$$

- For special case where $X_1, X_2, \ldots, X_{n+1}$ are multivariate normal 正太分布时线性回归和MSE结果一致
  - the best predictor is a linear predictor, i.e., the best linear predictor is the best predictor
  - this is a special case – not true in general
- Can use results from multivariate normal to derive the optimal linear predictor (in general)

Best Linear Prediction: General Case

Exercise. Suppose Y is a random variable with mean μ_Y and variance of σ_Y^2, $\mathbf{X}$ is a d-dimensional random vector with mean $\boldsymbol{\mu}_\mathbf{X}$ and covariance $\boldsymbol{\Sigma}_\mathbf{X}$, and they have cross-covariance of

$$\text{Cov}(Y, \mathbf{X}) = \boldsymbol{\Sigma}_{Y,\mathbf{X}}.$$

(a) Derive an expression for the coefficients $\alpha_o, \alpha_1, \ldots, \alpha_d$ for the best linear predictor $\hat{Y} = \alpha_o + \sum_{i=1}^{d} \alpha_i X_i$. *Hint:* Let $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_d]^\top$ and note that $\hat{Y} = \alpha_o + \boldsymbol{\alpha}^\top \mathbf{X}$. Also, do this first for the case where $\mu_Y = 0, \boldsymbol{\mu}_\mathbf{X} = 0$ and then generalize.

(b) Derive an expression for the mean-squared prediction error of the prediction in **(a)** .

$a^T X = Y$

$a^T X X^T = Y X^T$

$a^T = Y X^T (X X^T)^{-1} = \Sigma_{YX} \Sigma_X^{-1}$

Answer.

(a) $\hat{y} = a^T X = (\Sigma_X^{-1}\Sigma_{YX})^T X = \Sigma_{YX}\Sigma_X^{-1} X$ (Special Case)

$MSE = \sigma_Y^2 - \Sigma_{YX}\Sigma_X^{-1}\Sigma_{XY}$

$\hat{y}' = \hat{y} - \mu_y \quad \hat{x}' = \hat{x} - \mu_x$

$\hat{y}' = \mu_y + \Sigma_{YX}\Sigma_X^{-1}(X - \mu_X)$

(b) $\mathrm{Cov}(X_i, \hat{Y} - Y) = 0$

Answer.

Example - Showing linear predictors can be very sub-optimal.

Suppose have Y is rv taking values of $-2, 0, 1$ each with probability of $\frac{1}{3}$, and X is rv defined by $X = |Y|$.

(a) Derive the best predictor of Y based on X.

(b) Derive the best linear predictor of Y based on X.

Answer. 非正态分布下linear predictor不是best predictor

(a) $\hat{Y} = \begin{cases} -2 & X=2 \\ X & X \neq 2 \end{cases}$

$E[(\hat{Y}-Y)^2] = 0$

(b) $\hat{Y} = \mu_Y + \Sigma_{YX}\Sigma_X^{-1}(X-\mu_X)$

$= \mu_Y + \frac{\sigma_{YX}(X-\mu_X)}{\sigma_X^2} = \frac{2}{3} - X$

Answer.

Assume $\mu_y, \mu_x = 0$

$$E[(Y-\hat{Y})^2] = Var(\Sigma_{YX}\Sigma_X^{-1}X - Y)$$

$$= \sigma_Y^2 - \Sigma_{YX}\Sigma_X^{-1}\Sigma_{XY}$$

The prediciton error is no bigger than original variance

预测误差比原始误差小

X和Y相关性越大,预测误差越小

Optimal Linear Prediction of Time Series

Exercise. Suppose that $X_1, X_2, \ldots$ is a second order time series.

(a) Derive a formula for coefficients which correspond to the best linear predictor of X_{n+1} based on $X_1, X_2, \ldots, X_n$. With $\mathbf{X}_n = [X_1 \cdots X_n]^\top$, assume you have

$$\mathbf{\Sigma}_{X_{n+1},\mathbf{X}_n},\ \mathbf{\Sigma}_{\mathbf{X}_n},\ \boldsymbol{\mu}_{\mathbf{X}_n},\ \mu_{X(n+1)}$$

(b) Derive a general expression for the mean-squared prediction error (MSPE).

Answer.

直接带入线性回归

Linear Prediction: Stationary Processes

Remark. For a stationary process $X_1, X_2, \ldots, X_n, \ldots$, the best linear predictor is given by 本质上和之前相同

$$\hat{X}_{n+1} = \boldsymbol{\Sigma}_{X_{n+1},\mathbf{X}_n} \boldsymbol{\Sigma}_{\mathbf{X}_n}^{-1} (\mathbf{X}_n - \boldsymbol{\mu}_n) + \mu$$

where $\boldsymbol{\mu}_n$ is $n \times 1$ vector with all entries being μ and

$$\begin{aligned}
\boldsymbol{\Sigma}_{\mathbf{X}_n} &= \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \mathrm{Cov}(X_1, X_3) & \cdots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \mathrm{Cov}(X_2, X_3) & \cdots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \mathrm{Cov}(X_n, X_3) & \cdots & \mathrm{Cov}(X_n, X_n) \end{bmatrix} \\
&= \begin{bmatrix} \gamma(0) & \gamma(1) & \gamma(2) & \cdots & \gamma(n-1) \\ \gamma(1) & \gamma(0) & \gamma(1) & \cdots & \gamma(n-2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma(n-1) & \gamma(n-2) & \gamma(n-3) & \cdots & \gamma(0) \end{bmatrix}
\end{aligned}$$

and 因为是stationary,可以直接用协方差表示

$$\boldsymbol{\Sigma}_{X_{n+1},\mathbf{X}_n} = [\gamma(n) \;\; \gamma(n-1) \;\; \cdots \;\; \gamma(1)]$$

Autoregressive Time Series - AR(p)

Remark. AR(1) models are a first-order model for the memory of the proces – can generalize so as to have longer-term memory.

Definition. With iid sequence of $\epsilon_0, \epsilon_1, \ldots$ having mean zero and variance σ^2, the time series defined as

$$X_n - \mu \quad = \quad \sum_{i=1}^{p} \alpha_i (X_{n-i} - \mu) + \epsilon_i$$

where $\alpha_1, \alpha_2, \ldots, \alpha_p$ is said to follow an **AR(p) process.** Note

$$X_n \quad = \quad \alpha_o + \sum_{i=1}^{p} \alpha_i X_{n-i} + \epsilon_i$$

where $\alpha_o = \mu - \sum_{i=1}^{p} \alpha_i \mu = (1 - \sum_{i=1}^{p} a_i) \mu$.

Remark. There are restrictions on coefficients if this process is going to be stationary. One sufficient condition is ... $\sum_{i=1}^{p} |a_i| < 1$

Not simple

Not necessarily

Prediction for AR Processes

Assume an AR(p) process of

$$X_n - \mu = \sum_{i=1}^{p} \alpha_i (X_{n-i} - \mu) + \epsilon_i$$

- Prediction of $X_{n+1}, X_{n+2}, \ldots$ based on data up to time $X_1, X_2, \ldots, X_n$ is given by

$$\begin{aligned}
\hat{X}_{n+1} &= \mu + \sum_{i=1}^{p} \alpha_i (X_{n+1-i} - \mu) \\
\hat{X}_{n+2} &= \mu + \alpha_1 (\hat{X}_{n+1} - \mu) + \sum_{i=2}^{p} \alpha_i (X_{n+2-i} - \mu) \\
\vdots &= \vdots
\end{aligned}$$

$MSE = \sigma^2$

- Optimal linear predictor if errors are iid mean-zero (assuming second order)

Statistical Inference for AR Model

Objectives of Statistical Inference for AR Models

Suppose that have time series data $X_1, X_2, \ldots, X_n$ – we would like to

- Based on time series data, estimate the parameters for an AR(p) model
- Do some goodness-of-fit tests/diagnostics for the estimated AR(p) model
- Derive/estimate the appropriate model order p

Remark. With AR(p) parameter estimates of $\hat{\mu}, \hat{\alpha}_1, \ldots, \hat{\alpha}_p$, and data $X_1, X_2, \ldots, X_n$, the predictions are:

$$\begin{aligned}
\hat{X}_{n+1} &= \hat{\mu} + \sum_{i=1}^{p} \hat{\alpha}_i (X_{n+1-i} - \hat{\mu}) \\
\hat{X}_{n+2} &= \hat{\mu} + \hat{\alpha}_1 (\hat{X}_{n+1} - \hat{\mu}) + \sum_{i=2}^{p} \hat{\alpha}_i (X_{n+2-i} - \hat{\mu}) \\
\vdots &= \vdots
\end{aligned}$$

Estimation of AR Parameters: Least Squares

Least Squares Method: For AR model,

$$X_n \quad = \quad \alpha_o + \sum_{i=1}^{p} \alpha_i X_{n-i} + \epsilon_i$$

- Similarity with linear regression equations
- Natural to least-squares as basis for estimation –

MSE

$$(\hat{\alpha}_o, \hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_p) = \underset{\alpha_o, \alpha_1, \alpha_2, \ldots, \alpha_p}{\arg\min} \sum_{i=p+1}^{n} \left(X_i - \alpha_o - \sum_{i'=1}^{p} \alpha_{i'} X_{i-i'} \right)^2$$

and

$$\hat{\sigma}^2 = \frac{1}{n - 2p - 1} \sum_{i=p+1}^{n} \left(X_i - \hat{\alpha}_o - \sum_{i'=1}^{p} \hat{\alpha}_{i'} X_{i-i'} \right)^2$$

For the case of iid normally distributed errors, this is approximately maximum-likelihood.

Estimation of AR Parameters: Yule-Walker

Remark. For AR(p) model, there are linear equations that the auto-covariances and variance must satisfy. With $\mu = 0$, recall the model is

通过解方程找到参数

$$X_n = \sum_{i=1}^{p} \alpha_i X_{n-i} + \epsilon_n$$

so

$$\begin{aligned}
\gamma(0) = \mathrm{Var}(X_n) &= E\left[X_n \cdot \left(\sum_{i=1}^{p} \alpha_i X_{n-i} + \epsilon_n\right)\right] \\
&= \sum_{i=1}^{p} \alpha_i E(X_n X_{n-i}) + E(X_n \epsilon_n) \\
&= \sum_{i=1}^{p} \alpha_i \gamma(i) + \sigma^2 \qquad .
\end{aligned}$$

Remark. Generate equations for $\gamma(0), \gamma(1), \ldots, \gamma(p)$

Estimation of AR Parameters: Yule-Walker Equations

By calulations as on previous slide,

矩量法求参数

$$
\begin{aligned}
\gamma(0) = \; &= \sum_{i=1}^{p} \alpha_i \gamma(i) + \sigma^2 \\
\gamma(1) \; &= \sum_{i=1}^{p} \alpha_i \gamma(i-1) = \sum_{i=1}^{p} \alpha_i \gamma(|i-1|) \quad . \\
\gamma(2) \; &= \sum_{i=1}^{p} \alpha_i \gamma(|i-2|) \\
\vdots \quad & \vdots \quad \vdots \\
\gamma(p) \; &= \sum_{i=1}^{p} \alpha_i \gamma(|i-p|) \quad .
\end{aligned}
$$

- Have $p+1$ equations and $p+1$ unknowns
- To estimate AR coefficients and σ^2, simply plug in the covariance estimates – these are the **Yule-Walker estimates**

Example. Write out the Yule-Walker equations for case of AR(2).

Answer.

$$\gamma(0) = \alpha_1 \gamma(1) + \alpha_2 \gamma(2) + \sigma^2$$

$$\gamma(1) = \alpha_1 \gamma(0) + \alpha_2 \gamma(1)$$

$$\gamma(2) = \alpha_1 \gamma(1) + \alpha_2 \gamma(0)$$

AR Estimation: Diagnostics and Model Selection

Background. Suppose that have time series data $X_1, X_2, \ldots, X_n$ and have done estimates for an AR(p) model.

- Model selection – selecting the AR order p
  - AIC criteria
    - Assumes that errors are normally distributed and MLE estimation method
    - Also used with estimation methods that are close to MLE (e.g., Yule-Walker)
  - Partial auto-correlation function: correlation coefficient between X_n and X_{n+m} adjusted for $X_{n+1}, \ldots, X_{n+m-1}$
- Regression-type diagnostics with residuals

$$\hat{\epsilon}_i \quad = \quad X_i - \hat{X}_i = X_i - \left[\hat{\alpha}_o + \sum_{i'=1}^{p} \hat{\alpha}_{i'} X_{i-i'}\right]$$

  - Analyze
    - normal distribution assumption – QQ plots
    - serial correlations ($\hat{\epsilon}_{i+1}$ vs. $\hat{\epsilon}_i$)
    - patterns (e.g., increasing/decreasing variance in time)

R-commands: Estimation for AR Processes

Remark. Estimation with time series object x.

```
ar(x, aic = TRUE, order.max = NULL,
   method=c("yule-walker", "burg", "ols", "mle", "yw"),
   na.action, series, ...)

## S3 method for class 'ar':
predict(object, newdata, n.ahead = 1, se.fit = TRUE, ...)

Arguments
x -- A univariate or multivariate time series.
```

使用AIC决定阶数

```
aic -- Logical flag. If TRUE then the Akaike Information Criterion
is used to choose the order of the autoregressive model. If FALSE,
the model of order order.max is fitted.

order.max -- Maximum order (or order) of model to fit. Defaults to
10*log10(N) where N is the number of observations except for
method="mle" where it is the minimum of this quantity and 12.

method -- Character string giving the method used to fit the model.
Must be one of the strings in the default argument (the first few
characters are sufficient). Defaults to "yule-walker".
```

```
na.action -- function to be called to handle missing values.

demean -- should a mean be estimated during fitting?
series -- names for the series. Defaults to deparse(substitute(x)).

var.method -- the method to estimate the innovations variance
(see Details).

object -- a fit from ar.

newdata -- data to which to apply the prediction.

n.ahead -- number of steps ahead at which to predict.

se.fit -- logical: return estimated standard errors of the
prediction error?

*****************************************************

predict(object, newdata, n.ahead = 1, se.fit = TRUE, ...)

Arguments
object -- a fit from ar.
newdata -- data to which to apply the prediction.
n.ahead -- number of steps ahead at which to predict.
se.fit -- logical (TRUE or FALSE): return estimated standard errors of
the prediction error?
```

Estimation Example: Simulation AR(1) Process

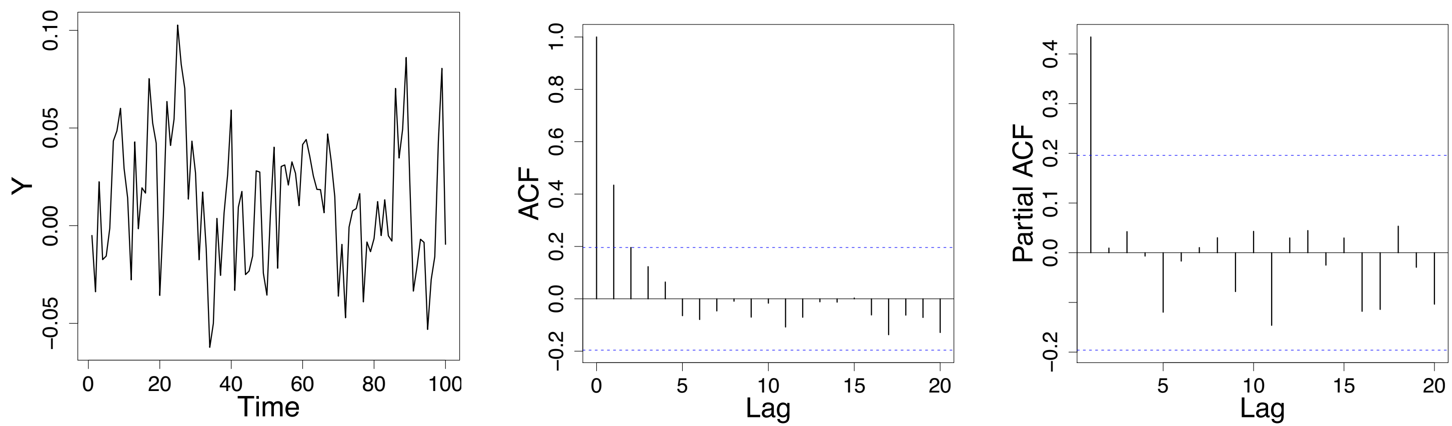
Example. Simulated an AR(1) process with $Y_0 = 0$

$$Y_i = \underline{.4} \cdot Y_{i-1} + \epsilon_i \qquad\qquad i = 1, 2, \ldots, 100$$

where $\epsilon_1, \epsilon_2, \ldots \sim \mathcal{N}(0, \underline{(.03)}^2)$.

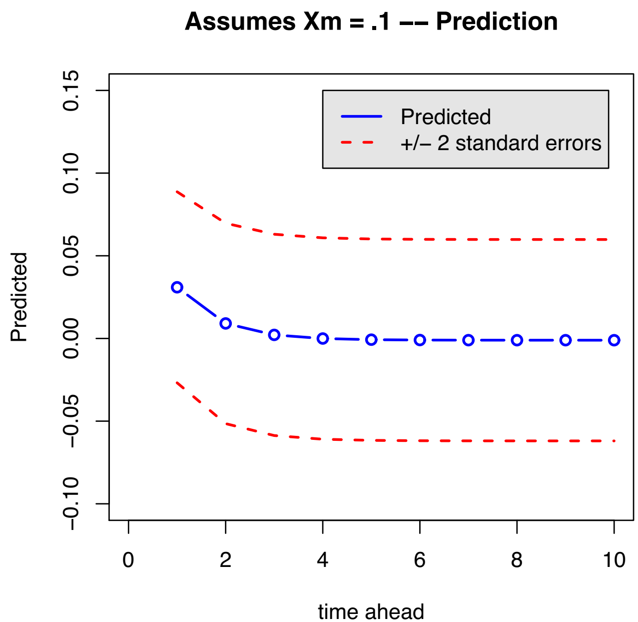
Plots of Time series, ACF, and PACF

以alpha指数下降



Applied R-Function "ar" and "predict" Generated

- $\hat{\mu} = 0.0013$, $\hat{\alpha}_1 = 0.4341, \hat{\sigma} = 0.0309$ 估计得挺准
- std. error of $\hat{\alpha}_1$ is 0.0933 and est. st. dev. is 0.0310
- Assuming a value of $X_n = .10$, the plot below is the plot of prediction vs. lag along with ± 2 standard errors



R-commands for Simulation

```
> n = 101
> xt = c(0:(n-1))
> eps = rnorm(n,0,.03)
> x = cumsum(eps)
> y = eps
> alpha = .4
>
> for (i in c(2:n)){
+       y[i] = alpha*y[i-1]+eps[i]
+    }
> plot(y, type="l", xlab="Time", ylab="Y")
> acf(y)
> acf(y, type="partial")
>
> y_AR1 <- ar(y,aic=FALSE,order.max=1,method=c("mle"))
> names(y_AR1)
 [1] "order"        "ar"           "var.pred"     "x.mean"       "aic"
 [6] "n.used"       "order.max"    "partialacf"   "resid"        "method"
[11] "series"       "frequency"    "call"         "asy.var.coef"
```

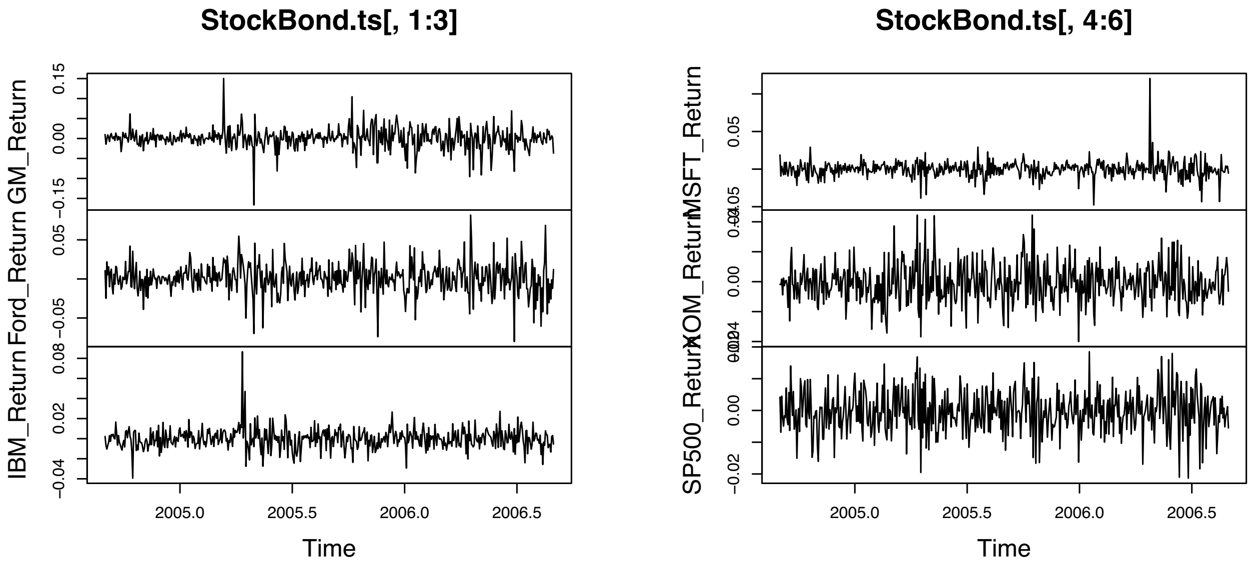
```
> y_AR1
Coefficients:
     1
0.4341
Order selected 1
sigma^2 estimated as  0.0009531
> sqrt(0.0009531)
[1] 0.03087232
> y_AR1$x.mean
[1] 0.00125588
> sqrt(y_AR1$asy.var.coef)
           [,1]
[1,] 0.09099952

%%%%%%%%%  Prediction  %%%%%%%%%%%%%%
> xnew = c(.10)
> y_ARpred10 <- predict(y_AR1,newdata=xnew, n.ahead = 10, se.fit = TRUE)
>
> windows()
> plot(c(1:10),y_ARpred10$pred,type='b',lty=1,xlab='time ahead',
+ ylab='Predicted',xlim=c(0,10),ylim=c(-.1,.15),lwd=2,col='blue',
+ main = 'Assumes Xm = .1 -- Prediction')
> lines(c(1:10),y_ARpred10$pred+2*y_ARpred10$se,lty=2,lwd=2,col='red')
> lines(c(1:10),y_ARpred10$pred-2*y_ARpred10$se,lty=2,lwd=2,col='red')
> legend(4,.15, c("Predicted","+/- 2 standard errors"), lty=c(1,2),
+ lwd=c(2,2),col=c("blue","red"), bg="gray90")
```

General AR Estimation Example.

Example. Estimation of AR model for

- IBM daily log returns Sept 1 2004 - Sept 1 2006
- SP500 daily log returns Sept 1 2004 - Sept 1 2006
- Used AIC criteria to select order
  - Order 0 suggests a random walk model is most appropriate
- Carried out prediction (and standard error on prediction) for case where order was not 0

Time Series Plots R-code for this is on next page



R-Code for Plots

```
> Ny = 2
> Nd = Ny*252
> X = read.csv("Data\\Stock_Data_87-06.csv",header=TRUE)
> GM_lret <- diff(log(X$GM_AC))
> F_lret <- diff(log(X$F_AC))
> IBM_lret <- diff(log(X$IBM_AC))
> MSFT_lret <- diff(log(X$MSFT_AC))
> XOM_lret <- diff(log(X$XOM_AC))
> SP500_lret <- diff(log(X$SP500_AC))
> Stock_lret <- cbind(rev(GM_lret[1:Nd]),rev(F_lret[1:Nd]),
+ rev(IBM_lret[1:Nd]),rev(MSFT_lret[1:Nd]),rev(XOM_lret[1:Nd]),
+ rev(SP500_lret[1:Nd]))
>
> StockBond.ts <- ts(data=Stock_lret,start=c(2004,169),frequency=252,
+ names=c("GM_Return","Ford_Return","IBM_Return","MSFT_Return",
+ "XOM_Return","SP500_Return"))
>
> plot(StockBond.ts[,1:3])
> plot(StockBond.ts[,4:6])
```

Results for IBM daily log returns: AR(1) and AR(p)/AIC

```
> IBM_AR1 <- ar(StockBond.ts[,3], aic=FALSE,order.max=1,method=c("mle"))
> IBM_AR1

Call:
ar(x = StockBond.ts[, 3], aic = FALSE, order.max = 1, method = c("mle"))

Coefficients:
     1
0.0465

Order selected 1  sigma^2 estimated as  0.0001022

> IBM_AR <- ar(StockBond.ts[,3], aic=TRUE,order.max=6,method=c("mle"))
> IBM_AR

Call:
ar(x = StockBond.ts[, 3], aic = TRUE, order.max = 6, method = c("mle"))

Coefficients:
     1       2       3
0.0356  0.0559  0.1271

Order selected 3  sigma^2 estimated as  0.0001002
```

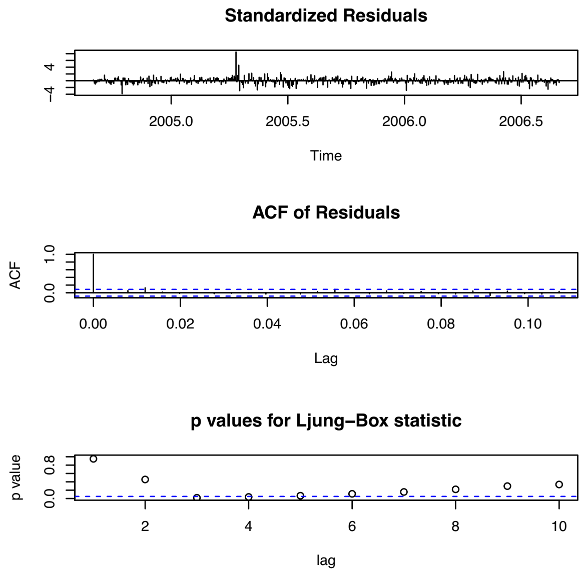
Results for SP500 daily log returns: AR(p) and AIC

AIC give us p=2

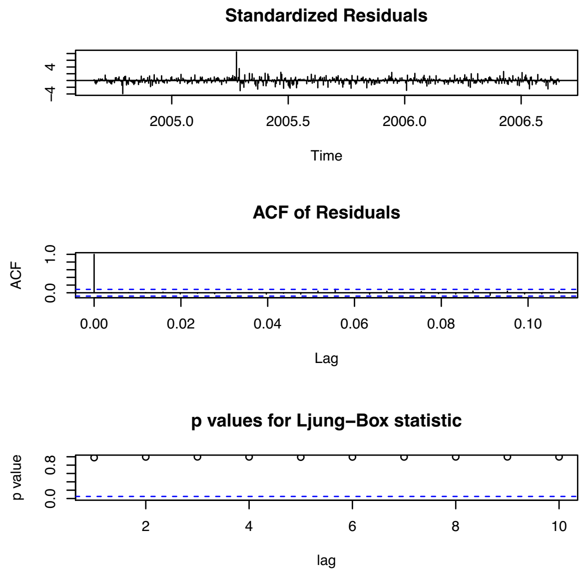
```
> SP500_AR <- ar(StockBond.ts[,6], aic=TRUE,order.max=6,method=c("mle"))
> SP500_AR

Call:
ar(x = StockBond.ts[, 6], aic = TRUE, order.max = 6, method = c("mle"))

Coefficients:
      1        2
-0.0477  -0.0863

Order selected 2  sigma^2 estimated as  4.325e-05
```

Example: Diagnostic Analysis on Residuals I

- If model is accurate, diagnostic plots of residuals $\{\hat{\epsilon}\}$ should reflect distribution/properties of white noise process $\{\epsilon_i\}$
  - Uncorrelated between lags - ACF and Ljung-Box test p-values
  - No visible patterns over time
  - Distribution (QQ plots)



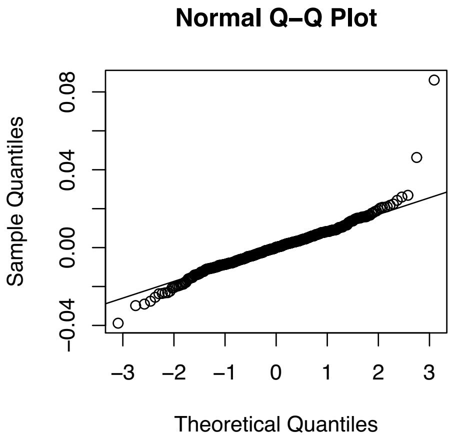
IBM - AR(1) model



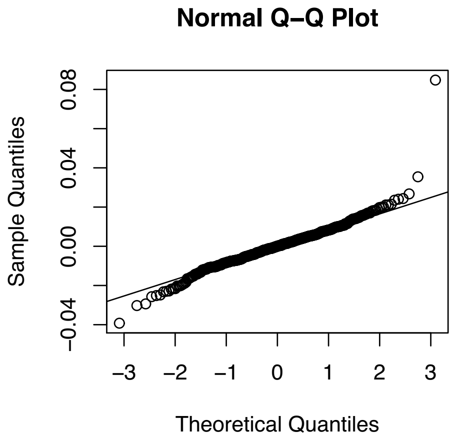
IBM - AR(3) model

This is not a good fitting because residual has covariance

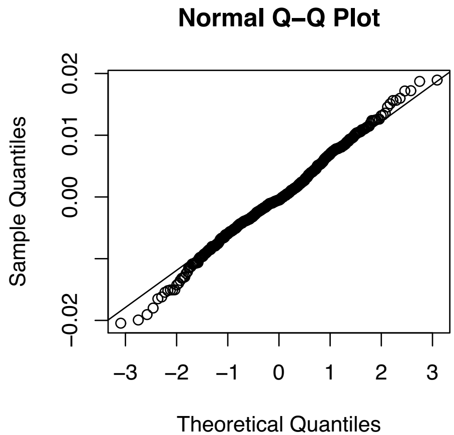
Example: Diagnostic Analysis on Residuals: QQ Plots



IBM and AR(1)



IBM and AR(3)



SP500 and AR(2)

R-commands

```
> # Diagnostics plots from arima
> IBM_AR_arima1 <- arima(StockBond.ts[,3], order = c(1,0,0), method = "ML")
> tsdiag(IBM_AR_arima1)
> IBM_AR_arima <- arima(StockBond.ts[,3], order = c(3,0,0), method = "ML")
> tsdiag(IBM_AR_arima2)

> # QQ plots of residuals
> myqqnorm(IBM_AR1$resid)
> myqqnorm(IBM_AR$resid)
> myqqnorm(SP500_AR$resid)
```

Moving Average Processes

Remark. Recall the AR(p) model

$$X_n = \alpha_1 X_{n-1} + \alpha_2 X_{n-2} + \cdots + \alpha_p X_{n-p} + \epsilon_n$$

where $\epsilon_1, \epsilon_2, \ldots$ are iid with mean 0 and variance σ^2. This introduces a certain type of correlation where

- Typically $\mathrm{Cov}(X_n, X_{n+k}) \neq 0$ for any n, k.
- For the process $X = \{X_n\}$ to be stationary, there are restrictions on coefficients $\{\alpha_i\}$

Using MA term to get rid of correlation error

Definition. A time series $X = \{X_n\}$ is said to be a moving average process of order q (and write MA(q)) if it can be written as

$$X_n = \epsilon_n + \theta_1 \epsilon_{n-1} + \cdots + \theta_q \epsilon_{n-q}$$

where $\epsilon_1, \epsilon_2, \ldots$ are iid with mean 0 and variance σ^2, and $\theta_1, \theta_2, \ldots, \theta_q$ are real parameters with $\theta_q \neq 0$. With the convention that $\theta_0 = 1$, the MA process can be written as

$$X_n = \theta_0 \epsilon_n + \theta_1 \epsilon_{n-1} + \cdots + \theta_q \epsilon_{n-q}.$$

Exercise. Suppose $X = \{X_n\}$ is a MA process of order q as in the definition on the preceding page. Derive the autocovariance function and show that the process is (weakly) stationary (i.e., MA processes are always stationary).

Answer. $E[X_n] = E\left[\sum \theta_i \epsilon_{n+1-i}\right] = 0$

$\mathrm{Var}(X_n) = \sigma^2 \sum_{0}^{q} \theta_i^2$

$\gamma(h) = \sigma^2 \sum_{0}^{q-h} \theta_i \theta_{i-h}$

For h>q no dependence, ACF=0

Answer.

Exercise. Suppose $X' = \{X'_n\}$ is a process with

$$\begin{aligned} E(X'_n) &= 0 && \text{for all } n \\ \text{Var}(X'_n) &= \sigma^2 && \text{for all } n \\ \text{Cov}(X'_n, X'_{n+1}) &= .5\,\sigma^2 && \text{for all } n \\ \text{Cov}(X'_n, X'_{n+k}) &= 0 && \text{for all } k > 1 \end{aligned}$$

Show that there is a MA process $X = \{X_n\}$ with the same first/second moments (means, variances, and covariances).

Answer.

MA(1)

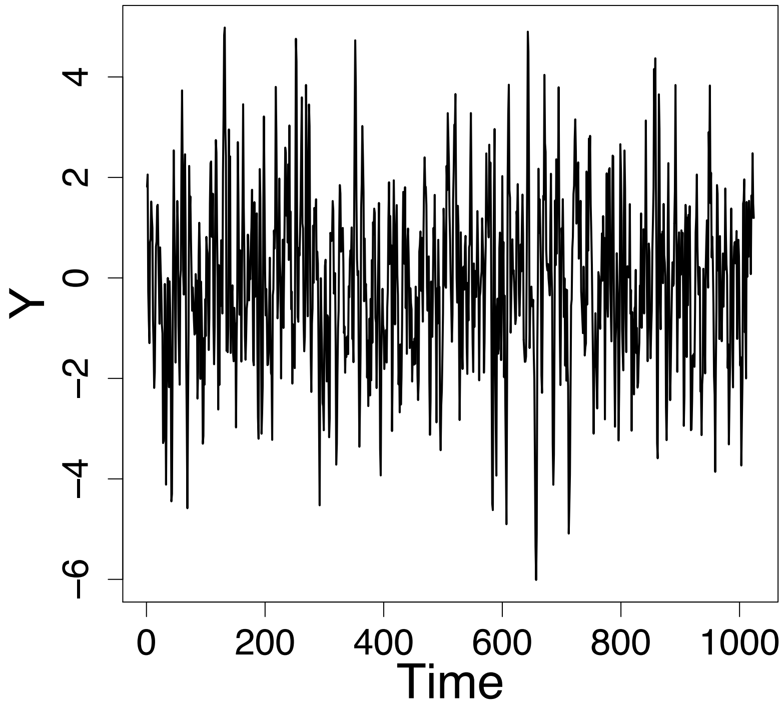
$\theta_1 = \sqrt{0.5}$

Answer.

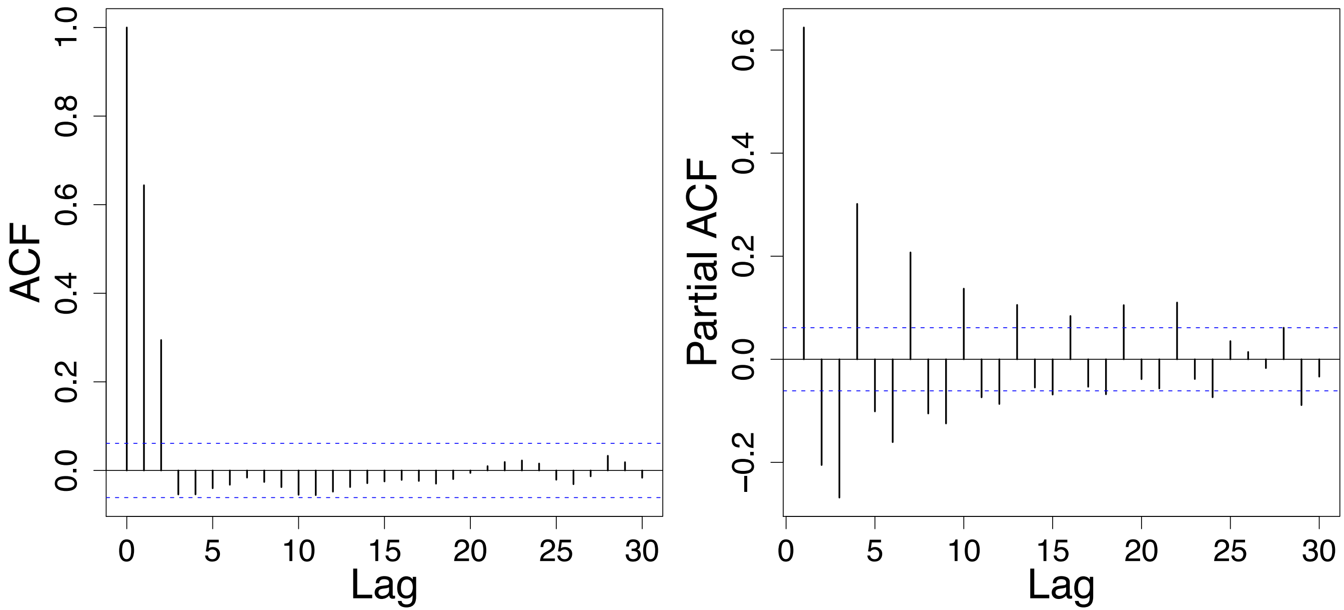
Simulation Example

```
> wnoise <- rnorm(1026)
> y.ma <- wnoise[1:1024] + wnoise[2:1025] + wnoise[3:1026]
> plot(y.ma, type="l", xlab="Time", ylab="Y")
> acf(y.ma)
> acf(y.ma, type="partial")
```

MA(2)



Typically we only add MA term to AR to make it better



PACF拖尾

Backward Shift Operator

- $BX_n = X_{n-1}$, $B^2 X_n = X_{n-2}$, ...
- AR(p) can be written as

$$\alpha(B)X_n = \epsilon_n$$

  where

$$\alpha(B) = 1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p$$

- Similarly, MA(q) can be written as

$$X_n = \theta(B)\epsilon_n$$

  where

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$$

AR vs MA

- Linear process: a mean zero time series X_n that can be written as

$$X_n = \sum_{j=-\infty}^{\infty} \psi_j \epsilon_{n-j}$$

with $\sum_j |\psi_j| < \infty$. J大于0则是casualu

- Duality between AR and MA
  - AR(1): $X_n = \epsilon_n + \alpha_1 \epsilon_{n-1} + \alpha_1^2 \epsilon_{n-2} + \cdots$
  - MA(1): $\epsilon_n = X_n + \theta_1 X_{n-1} + \theta_1^2 X_{n-2} + \cdots$

互相之间可以转换

ARMA Process – ARMA(p, q)

Can approximate almost any stationary process

- Mix AR and MA processes by replacing the error process in AR with MA, i.e.

$$X_n = \alpha_1 X_{n-1} + \cdots + \alpha_p X_{n-p} + \epsilon_n + \theta_1 \epsilon_{n-1} + \cdots + \theta_q \epsilon_{n-q}$$

  or alternatively,

$$X_n - \alpha_1 X_{n-1} - \cdots - \alpha_p X_{n-p} = \epsilon_n + \theta_1 \epsilon_{n-1} + \cdots + \theta_q \epsilon_{n-q}$$

- Using the shift operator: $\alpha(B) X_n = \theta(B) \epsilon_n$.

$$X_n = \frac{\theta(B)}{\alpha(B)} \epsilon$$

root必须在单位圆外

ARIMA Process – ARIMA(p, d, q)

Remark. The ARMA process applies directly to the time series – a more general class of processes can use the ARMA process for the differential process.

Definition. The differential operator $\nabla = 1 - B$. Then

$$\nabla X_n = \left[(1-B)X\right]_n = X_n - X_{n-1}.$$

$$\nabla^2 X_n = \left[(1-B)^2 X\right]_n = (X_n - X_{n-1}) - (X_{n-1} - X_{n-2}) = X_n - 2X_{n-1} + X_{n-2}.$$

$$\nabla^d X = (1-B)^d X$$

Definition. A process $X = \{X_n\}_n$ is said to be an ARIMA(p, d, q) process if $\nabla^d X \sim$ ARMA(p, q).

ARIMA Processes: Examples

ARIMA$(0,1,0)$ Note ARMA(0,0) process corresponds to a white noise process $\epsilon_1, \epsilon_2, \ldots$ So ARIMA(0,1,0) process assumes that

$$\nabla X_n = X_n - X_{n-1} = \epsilon_n$$

i.e. have the random walk model of

$$X_n = X_{n-1} + \epsilon_n \quad .$$

ARIMA$(p,1,q)$ An ARIMA$(p,1,q)$ process $X = \{X_n\}$ is one where $Y = \nabla X$ is ARMA(p,q).

ARIMA$(p,2,q)$ process $X = \{X_n\}$ is one where $Y = \nabla^2 X$ is ARMA(p,q) – note that

$$\begin{aligned} Y_n &= \nabla(X_n - X_{n-1}) \\ &= [X_n - X_{n-1}] - [X_{n-1} - X_{n-2}] \\ &= X_n - 2X_{n-1} + X_{n-2} \end{aligned}$$

Simulation of AR/ARMA/ARIMA Processes

```
arima.sim(model, n, rand.gen = rnorm, innov = rand.gen(n, ...),
          n.start = NA, start.innov = rand.gen(n.start, ...), ...)

Arguments
model -- A list with component ar and/or ma giving the AR and MA
coefficients respectively. Optionally a component order can be used.
An empty list gives an ARIMA(0, 0, 0) model, that is white noise.

n -- length of output series, before un-differencing.

rand.gen optional: -- a function to generate the innovations.

innov -- an optional times series of innovations. If not provided,
rand.gen is used.

n.start -- length of \burn-in" period. If NA, the default, a reasonable
value is computed.

start.innov -- an optional times series of innovations to be used for
the burn-in period. If supplied there must be at least n.start values
(and n.start is by default computed inside the function).

... additional arguments for rand.gen. Most usefully, the
standard deviation of the innovations generated by rnorm can be
specified by sd.
```

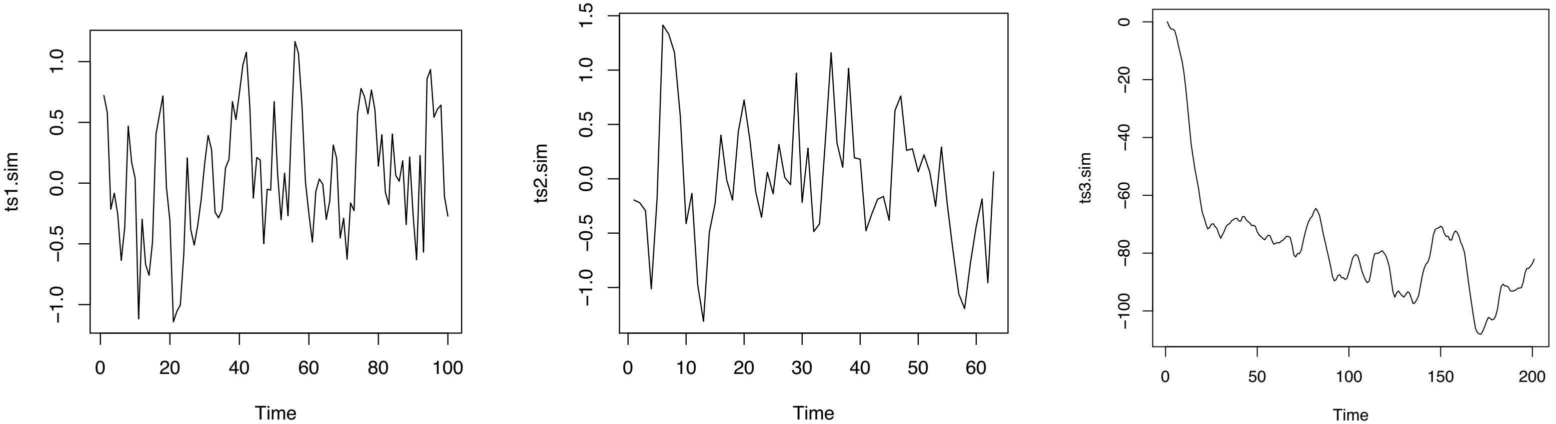
Simulation of AR/ARMA/ARIMA Processes

```
arima.sim(model, n, rand.gen = rnorm, innov = rand.gen(n, ...),
          n.start = NA, start.innov = rand.gen(n.start, ...), ...)
```

使用MLE来估计每个order的参数

```
Arguments
model -- A list with component ar and/or ma giving the AR and MA
coefficients respectively. Optionally a component order can be used.
An empty list gives an ARIMA(0, 0, 0) model, that is white noise.

n -- length of output series, before un-differencing.

rand.gen optional: -- a function to generate the innovations.

innov -- an optional times series of innovations. If not provided,
rand.gen is used.

n.start -- length of \burn-in" period. If NA, the default, a reasonable
value is computed.

start.innov -- an optional times series of innovations to be used for
the burn-in period. If supplied there must be at least n.start values
(and n.start is by default computed inside the function).

... additional arguments for rand.gen. Most usefully, the
standard deviation of the innovations generated by rnorm can be
specified by sd.
```

Simulations




```
> # Simulation ARIMA processes
> ts1.sim <- arima.sim(n = 100, list(ar = c(0.8897, -0.4858),
+ ma = c(-0.2279, 0.2488)),sd = sqrt(0.1796))
> ts.plot(ts1.sim)
>
> # An ARIMA simulation
> ts2.sim <- arima.sim(list(order = c(1,1,0), ar = 0.7), n = 200)
> ts.plot(ts2.sim)
>
> # An ARIMA simulation
> ts3.sim <- arima.sim(list(order = c(1,1,1), ar = 0.7,ma=.7), n = 200)
> ts.plot(ts3.sim)
```

R-Functions for ARIMA Estimation

```
arima(x, order = c(0, 0, 0),
      seasonal = list(order = c(0, 0, 0), period = NA),
      xreg = NULL, include.mean = TRUE, transform.pars = TRUE,
      fixed = NULL, init = NULL, method = c("CSS-ML", "ML", "CSS"),
      n.cond, optim.control = list(), kappa = 1e6)

Arguments
x -- a univariate time series
order -- A specification of the non-seasonal part of the ARIMA model:
the three components (p, d, q) are the AR order, the degree of differencing,
and the MA order.

seasonal -- A specification of the seasonal part of the ARIMA model,
plus the period (which defaults to frequency(x)). This should be a list
with components order and period, but a specification of just a numeric
vector of length 3 will be turned into a suitable list with the
specification as the order.

xreg -- Optionally, a vector or matrix of external regressors, which must
have the same number of rows as x.

include.mean -- Should the ARIMA model include a mean term? The default is
TRUE for undifferenced series, FALSE for differenced ones (where a mean
would not affect the fit nor predictions).
```

transform.pars -- Logical. If true, the AR parameters are transformed to ensure that they remain in the region of stationarity. Not used for method = "CSS".

fixed -- optional numeric vector of the same length as the total number of parameters. If supplied, only NA entries in fixed will be varied. transform.pars = TRUE will be overridden (with a warning) if any AR parameters are fixed. It may be wise to set transform.pars = FALSE when fixing MA parameters, especially near non-invertibility.

init -- optional numeric vector of initial parameter values. Missing values will be filled in, by zeroes except for regression coefficients. Values already specified in fixed will be ignored.

method -- Fitting method: maximum likelihood or minimize conditional sum-of-squares. The default (unless there are missing values) is to use conditional-sum-of-squares to find starting values, then maximum likelihood.

n.cond -- Only used if fitting by conditional-sum-of-squares: the number of initial observations to ignore. It will be ignored if less than the maximum lag of an AR term.

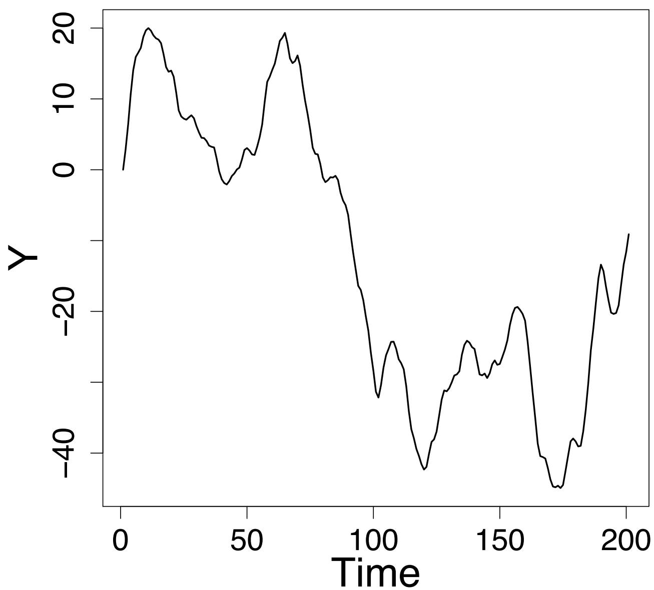
optim.control List of control parameters for optim.

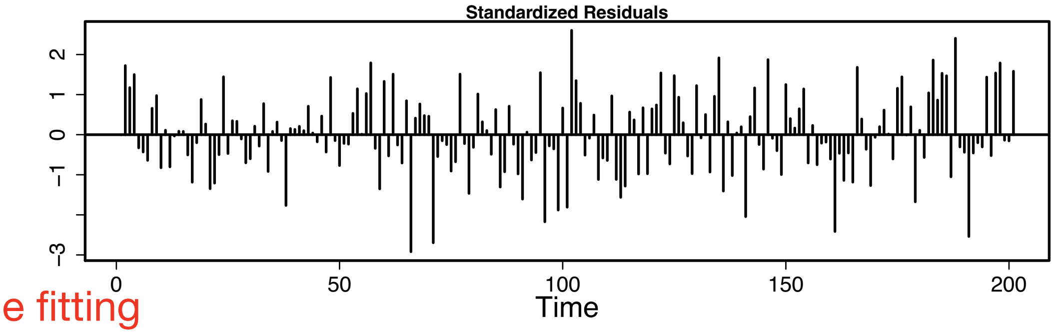
kappa the prior variance (as a multiple of the innovations variance) for the past observations in a differenced model. Do not reduce this.

Simulation Example

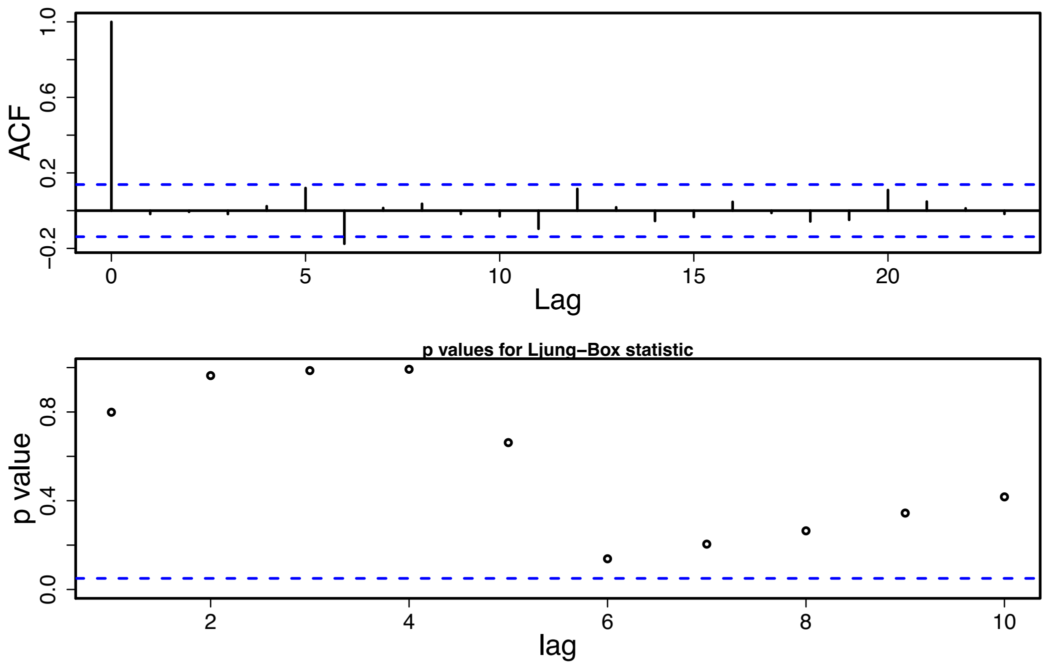
- ARIMA(1,1,1) process with AR and MA coefficients being 0.7, $\sigma = 1$ and $n = 200$.
- Diagnostics – residuals (test for white noise)
  - No obvious pattern and relatively constant variance.
  - Auto-correlations should be approximately zero.

```
> ts.sim <- arima.sim(model=list(order=c(1,1,1),
+   ar=0.7, ma=0.7), n=200)
> plot(ts.sim, type="l", xlab="Time", ylab="Y")
> ts.est111 <- arima(ts.sim, order=c(1,1,1))
> ts.est111
Coefficients:
         ar1     ma1
      0.7052  0.5777
s.e.  0.0571  0.0726
sigma^2 estimated as 0.663
> tsdiag(ts.est111)
```



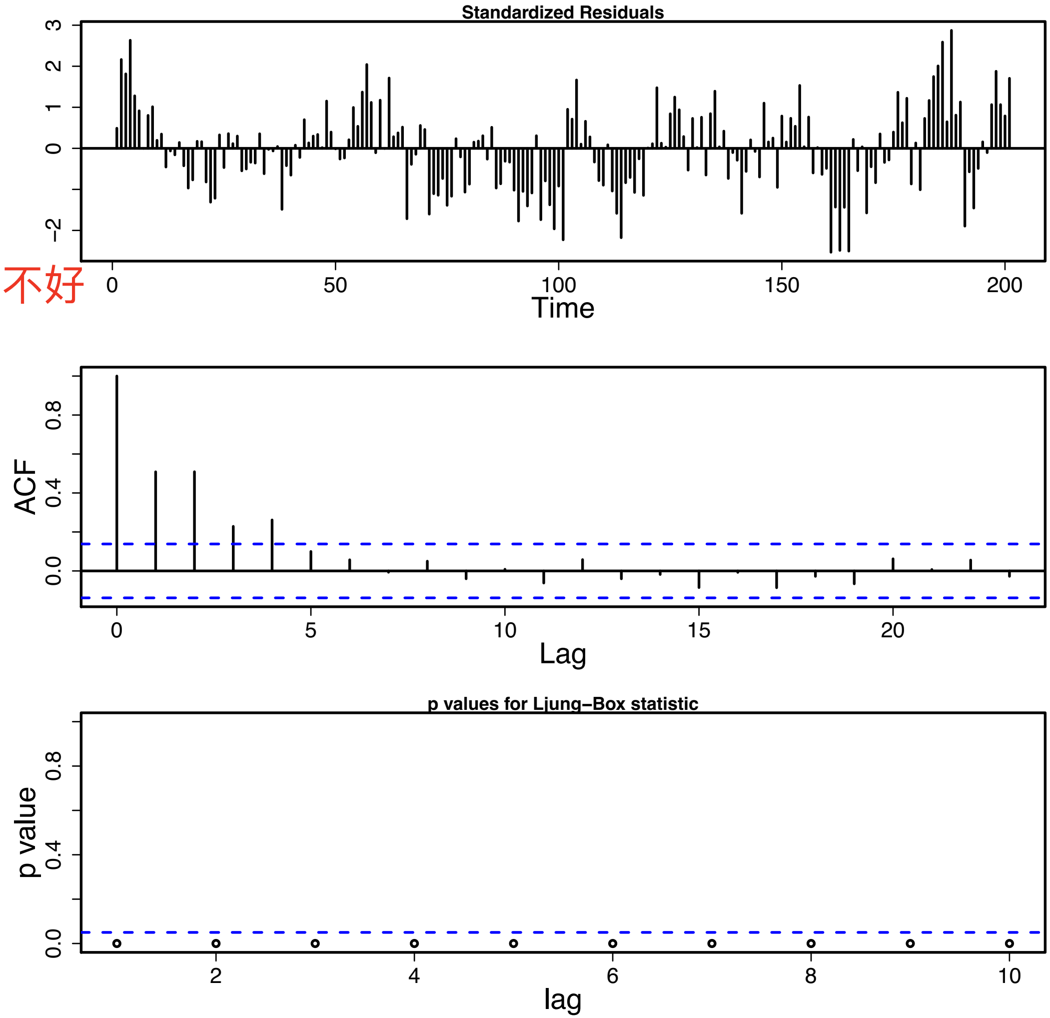


This is fine fitting



```
> ts.est101 <- arima(ts.sim, order=c(1,0,1))
> ts.est101
Coefficients:
         ar1     ma1  intercept
      0.9928  0.8568    -7.7923
s.e.  0.0056  0.0257    12.2415
sigma^2 estimated as 1.060
> tsdiag(ts.est101)
```

ARMA(1,1)表现不好


Standardized Residuals
Time
ACF
Lag
p values for Ljung–Box statistic
p value
lag

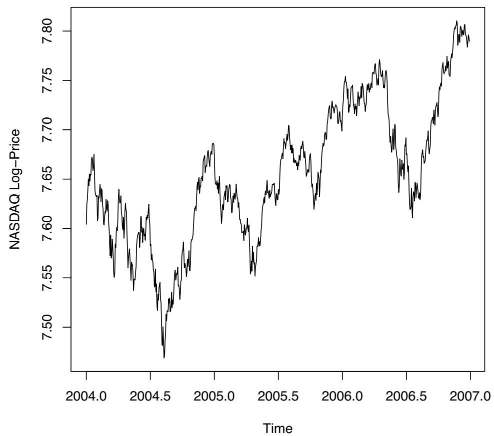
Model Selection

- Need to choose ARIMA model (p, d, q) - natural to use AIC or BIC
- function "arima" puts out AIC
- function "auto.arima" (from forecast package) automatically selects based on selected information criteria

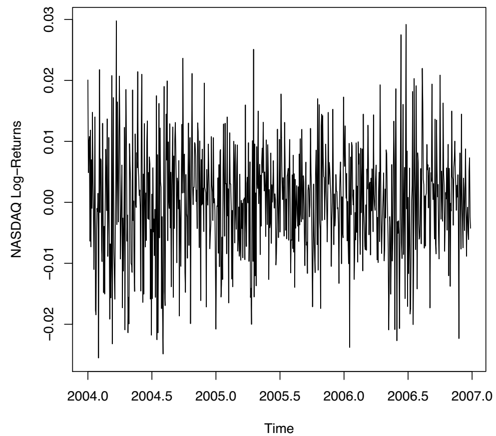
```
auto.arima(x, d=NA, D=NA, max.p=5, max.q=5,
    max.P=2, max.Q=2, max.order=5, start.p=2, start.q=2,
    start.P=1, start.Q=1, stationary=FALSE, seasonal=TRUE,
    ic=c("aicc","aic", "bic"), stepwise=TRUE, trace=FALSE,
    approximation=(length(x)>100 | frequency(x)>12), xreg=NULL,
    test=c("kpss","adf","pp"), seasonal.test=c("ocsb","ch"),
    allowdrift=TRUE, lambda=NULL, parallel=FALSE, num.cores=NULL)
Arguments:
x a univariate time series
d Order of first-differencing. If missing, will choose based on KPSS test.
D Order of seasonal-differencing. If missing, will choose based on OCSB test.
max.p Maximum value of p
max.q Maximum value of q
max.P Maximum value of P
max.Q Maximum value of Q
```

Nasdaq Daily Price

```
> X = read.csv("Data\\NASDAQ-2004-06.csv",header=TRUE)
> NASDAQ_logprice <- log(X$AClose)
> NASDAQ_logret <- diff(log(X$AClose))
> NASDAQ_logprice.ts <- ts(data=NASDAQ_logprice,start=c(2004,1),frequency=252,
+ names=c("NASDAQ_LogPrice"))
> NASDAQ_logret.ts <- ts(data=NASDAQ_logret,start=c(2004,1),frequency=252,
+ names=c("NASDAQ_LogRet"))
>
> plot(NASDAQ_logprice.ts,ylab='NASDAQ Log-Price')
> plot(NASDAQ_logret.ts,ylab='NASDAQ Log-Returns')
```



Log Price - NASDAQ



Relative Log Returns - NASDAQ

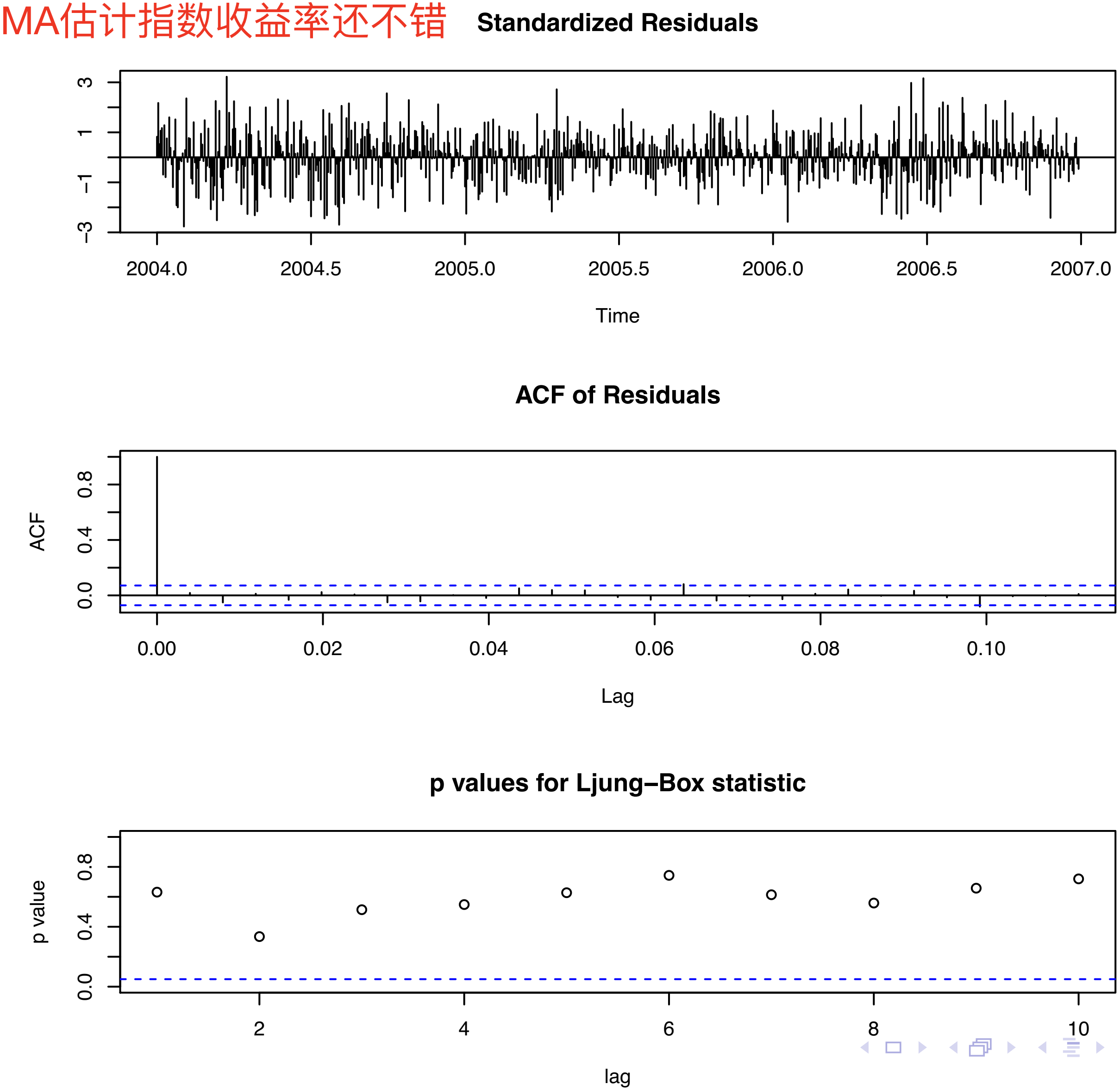
Nasdaq Daily Price

```
> library(forecast)
> auto.arima(NASDAQ_logret.ts,max.p=4,max.q=4,ic="bic")
Series: NASDAQ_logret.ts
ARIMA(0,0,0) with zero mean
```

white noise

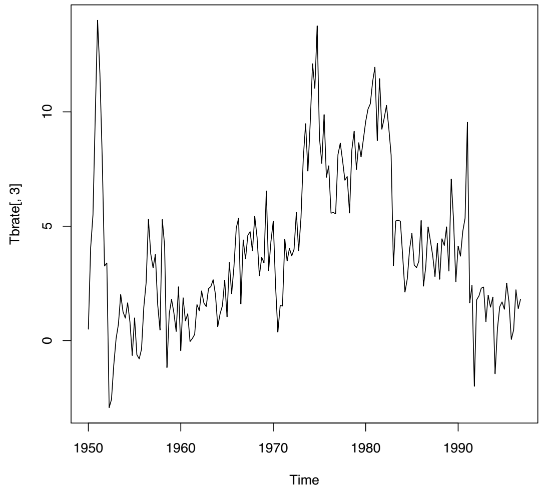
```
sigma^2 estimated as 8.523e-05:  log likelihood=2462.69
AIC=-4923.38   AICc=-4923.37   BIC=-4918.75
> NS_est010 =  arima(NASDAQ_logprice.ts, order = c(0,1,0), method = "ML")
> NS_est010
Series: NASDAQ_logprice.ts
ARIMA(0,1,0)

sigma^2 estimated as 8.523e-05:  log likelihood=2462.69
AIC=-4923.38   AICc=-4923.37   BIC=-4918.75
> windows()
> tsdiag(NS_est010)
```

ARIMA估计指数收益率还不错

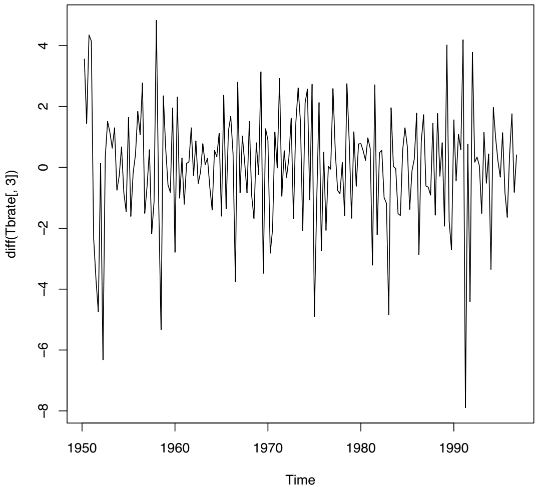


Treasury Bill Rates - Example pg 251 of Ruppert

```
> data(Tbrate,package="Ecdat")
> # r = the 91-day Treasury bill rate
> # y = the log of real GDP
> # pi = the inflation rate
> # fit the nonseasonal ARIMA model found by auto.arima
> plot(Tbrate[,3])
> plot(diff(Tbrate[,3]))
```



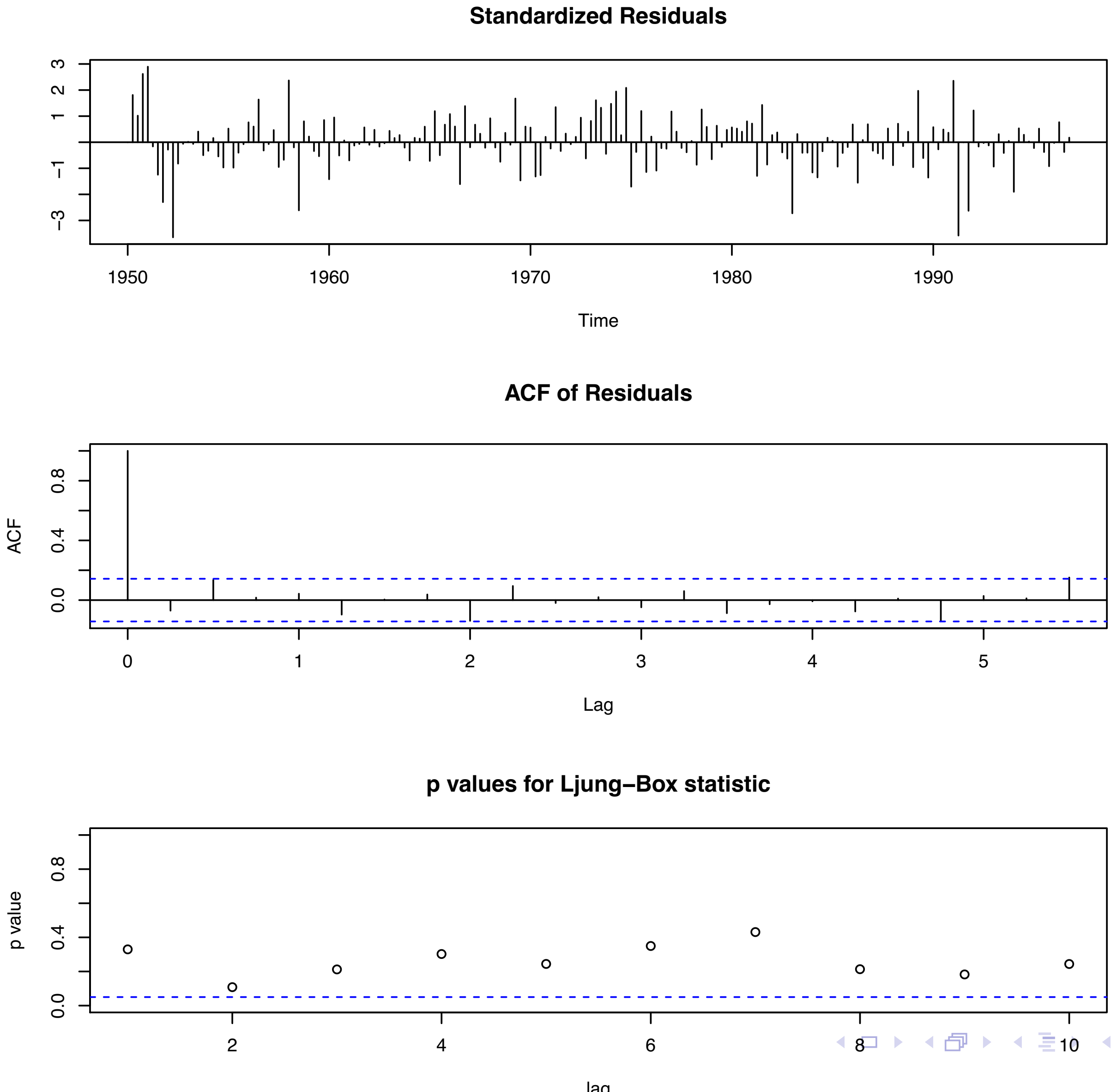
Treasury Bill Rates



Treasury Bill Rates - Diff

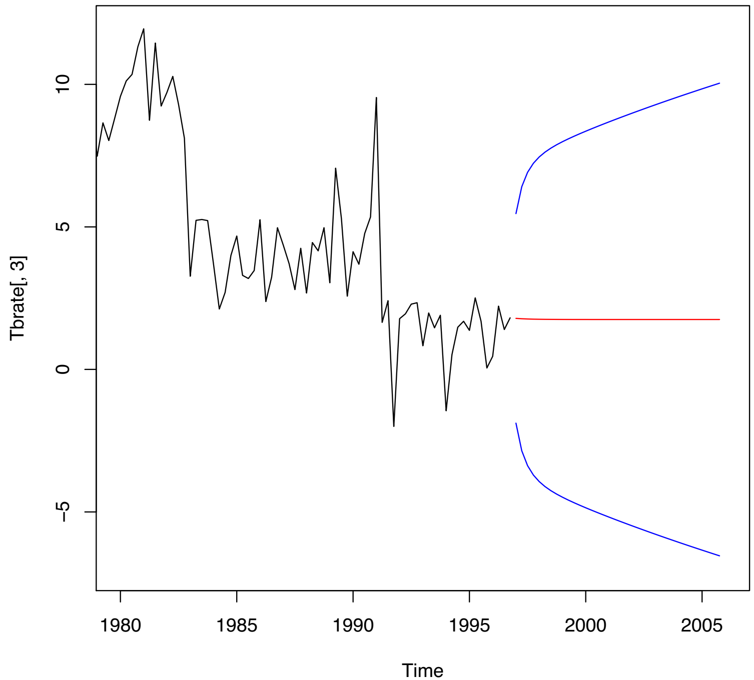
TBill Rates - ARIMA Model Selection

```
> auto.arima(Tbrate[,3],max.P=0,max.Q=0,ic="bic")
Series: Tbrate[, 3]
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
      0.6749  -0.9078
s.e.  0.0899   0.0501

sigma^2 estimated as 3.516:  log likelihood=-383.12
AIC=772.24   AICc=772.37   BIC=781.94
> fit = arima(Tbrate[,3],order=c(1,1,1))
> tsdiag(fit)
```


Standardized Residuals
3
2
1
−1
−3
1950
1960
1970
1980
1990
Time
ACF of Residuals
ACF
0.0
0.4
0.8
0
1
2
3
4
5
Lag
p values for Ljung−Box statistic
p value
0.0
0.4
0.8
2
4
6
8
10
lag

TBill Rates - Forecast

```
> fit = arima(Tbrate[,3],order=c(1,1,1))
> forecasts = predict(fit,36)
> windows()
> plot(Tbrate[,3],xlim=c(1980,2006),ylim=c(-7,12))
> lines(seq(from=1997,by=.25,length=36),forecasts$pred,col="red")
> lines(seq(from=1997,by=.25,length=36),forecasts$pred + 1.96*forecasts$se,
+ col="blue")
> lines(seq(from=1997,by=.25,length=36),forecasts$pred - 1.96*forecasts$se,
+ col="blue")
```



置信区间扩散的原因是在价格上计算

Seasonal ARIMA Models

- Section 10.1 of Ruppert provides a brief overview
- An ARIMA $(p, d, q) \times (p_s, d_s, q_s)_s$ model corresponds to
- An $ARIMA((1,1,0) \times (1,1,0)_s)$ model – then it turns out that the model is

$Y_t \rightarrow Y_t - Y_{t-s}$

$$
\begin{aligned}
(Y_t - Y_{t-s}) - (Y_{t-1} - Y_{t-1-s}) \\
= \quad & \phi\{(Y_{t-1} - Y_{t-1-s}) - (Y_{t-2} - Y_{t-2-s})\} \\
& + \phi^*\{(Y_{t-s} - Y_{t-2s}) - (Y_{t-1-s} - Y_{t-1-2s})\} \\
& + \phi\phi^*\{(Y_{t-1-s} - Y_{t-1-2s}) - (Y_{t-2-s} - Y_{t-2-2s})\} + \epsilon_t
\end{aligned}
$$

or in shorthand notation

$$\triangle\triangle_s Y = \phi \cdot B\{\triangle\triangle_s Y\} + \phi^* B_s \{\triangle\triangle_s Y\} + \phi\phi^* B B_s \{\triangle\triangle_s Y\} + \epsilon$$

先做S阶差分,再做常规差分

where ϵ_t is white noise process.

General Remarks

- Trends
- Seasonality/periodicity
- Differentiation (parameter d in ARIMA)
- L1 type of fitting (not so easy/available)
- Causal representation
- Checking normality of residuals