Parameter Estimation and Model Identification for Stationary Models

Introduction and Setting

Parameter Estimation and Model Identification for Stationary Models

Model-free methods

We have studied stationary and non-stationary processes and techniques for their analysis. Some techniques are ***model-free.*** That is, we do not need to fit a specific model to the data in order to perform the analysis.

Examples of "model-free" techniques

- Windowed spectral estimators
- Sample autocorrelations
- Filtering procedures

Model-Dependent Techniques

- In Units 4,5, and 6 we studied a variety of models.
  - AR models
  - Signal-plus-noise models
  - ARMA models
  - ARIMA models
  - Seasonal models
- Units 4,5 and 6 were focused on learning the ***properties*** of these models.
- Unit 7 focused on how to generate **forecasts** from them.

Key point

The forecasting techniques were based on the ***assumption*** that ***we know the model.***

Typical Situation

We are given a realization of time series data for which we may want to forecast future values.

- But, we don't know the model.

Good news

If your data set is a realization from an ARMA, ARIMA, Seasonal, etc. model, good techniques exist for estimating the model.

Bad news

It is highly unlikely that any data set you encounter will actually be a realization from one of these "nice" models.

Good news… remember…

"Essentially all models are wrong but some are useful."
—G. E. P. Box ("father" of modern time series analysis)

Setting

We have a $X_1, X_2, \ldots X_n$ realization of time series data.

Goal

Identify an appropriate model

This includes:

- Decision about ***type of model***
  - Does the model need to be stationary or non-stationary?
  - Is there an underlying signal?
- After the type of model is selected, we must estimate the model parameters.

First – We consider stationary models

Suppose that we have decided to fit a ***stationary ARMA(p,q) model*** to the data $X_1, X_2, \ldots X_n$.

In this case, we need to estimate:

- p and q—*called model identification*
- $\varphi_1, \varphi_2, \ldots, \varphi_p$
- $\theta_1, \theta_2, \ldots, \theta_q$
- μ, σ_a^2

} *called parameter estimation*

Strategy

Although identification (ID) of p and q is the first step in the analysis, many model ID techniques involve parameter estimation.

- So, we will cover parameter estimation techniques before discussing model ID.
- In practice, we use computer packages to perform the calculations.

DataScience@SMU

Estimation Methods for Stationary Models

Note about tswge estimation functions:

- est.arma.wge
  - Maximum likelihood estimation only
- est.ar.wge
  - Maximum likelihood
  - Yule-Walker
  - Burg

That is:

- For ARMA(p,q) models with $q > 0$, tswge only produces maximum likelihood estimates
- For AR(p) models, tswge provides ML, YW. and Burg estimates

DataScience@SMU

Maximum Likelihood Estimation

Maximum Likelihood (ML) Estimation

- Widely used throughout statistical analysis
- Involves maximizing the likelihood function $L = f(x_1, x_2, \ldots, x_n)$, which is the joint distribution of the time series realization
- Iterative, computationally intensive procedure
- Involves distributional assumptions about white noise (we will assume normal white noise)
- Is applicable to AR(p), MA(q), and ARMA(p,q) models
- Will typically use ML estimates for our final models

DataScience@SMU

tswge

Maximum Likelihood Estimation

We will generate realizations from AR and ARMA models, and find ML estimates (assuming we know *p* and *q*).

- The key function for ML estimation in an ARMA model is **est.arma.wge(data,p,q).**
  - Type is ‘mle’
- If you know that the model is a AR(*p*) model, then you can also use **est.ar.wge(data,p,q,type).**
  - Type can be ‘mle’ (default), ‘yw’, or ‘burg’

ARMA(2,1) $(1 \quad 1.6B + .8B^2)(X_t \quad 50) = (1 \quad .8B)a_t, \qquad {}^2_a = 5$

```
x21=gen.arma.wge(n=100,phi=c(1.6,-.8),theta=.8,vara=5,sn=55)
x21=x21+50 # gen.arma generates data from a zero
# mean model. We use this strategy to generate the
# AR(2) model with mean 50.
est.arma.wge(x21,p=2,q=1)
mean(x21)
```

Lots of output is generated. Among the output, you will see

```
$phi
[1]  1.6443723 -0.8146435

$theta
[1] 0.8699951

$avar
[1] 5.186772

> mean(x21)
[1] 49.9592
```

The final estimated model is

$$(1-1.64B+.81B^2(X_t-49.96)=(1-.87B)a_t \quad \hat{\sigma}_a^2 = 5.19$$

AR(4) $(1+.7B^2 \quad .1B^3+.72B^4)(X_t \quad 20)=a_t, \qquad {}^2_a=10$

```
x40=gen.arma.wge(n=100,phi=c(0,-.7,.1,-.72),vara=10,sn=72)
x40=x40+20
est.ar.wge(x40,p=4,type='mle')
# or you could use est.arma.wge(x40,p=4)
mean(x40)
```

Among the output, you will see

```
$phi
[1]  0.05656755 -0.68759141  0.11266029 -0.68184277

$avar
[1] 8.776949

> mean(x40)
[1] 19.98281
```

The final estimated model is:

$(1 \quad .06B+.69B^2 \quad .11B^3+.68B^4)(X_t \quad 19.98)=a_t, \qquad \hat{}^2_a=8.78$

DataScience@SMU

Yule-Walker Estimation

Alternative Estimators for AR(*p*) Models

Yule-Walker estimators

Consider the (zero mean) AR(*p*) model

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + a_t$$

So, for $k > 0$, we have

$$E[X_{t-k} X_t] = E[X_{t-k} (\phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + a_t)]$$

$$\underbrace{E[X_{t-k} X_t]}_{\gamma_k} = \phi_1 \underbrace{E[X_{t-k} X_{t-1}]}_{t-1-(t-k) = k-1} + \phi_2 E[X_{t-k} X_{t-2}] + \ldots + \phi_p E[X_{t-k} X_{t-p}] + \underbrace{E[X_{t-k} a_t]}_{=0 \text{ since } X_{t-k} \text{ and } a_t \text{ are uncorrelated when } k>0}$$

So,

$$\gamma_k = \phi_1 \gamma_{k-1} + \phi_2 \gamma_{k-2} + \ldots + \phi_p \gamma_{k-p}, \quad k > 0$$

Also, recall that for a zero mean process, $E[X_t X_{t+k}] = E[X_{t-k} X_t] = \gamma_k$.

Light board

Yule-Walker estimators

$$\gamma_k = \quad _1 \quad \gamma_{k\ 1} + \quad _2 \quad \gamma_{k\ 2} + \ldots + \quad _p \quad \gamma_{k\ p}, \qquad k > 0$$

Light board

Dividing both sides by γ_0, we get

$$\rho_k = \quad _1 \quad \rho_{k\ 1} + \quad _2 \quad \rho_{k\ 2} + \ldots + \quad _p \quad \rho_{k\ p}, \qquad k > 0$$

Letting $k = 1, 2, \ldots, p$, we get the well-known

Yule-Walker equations

$$\rho_1 = \varphi_1 + \varphi_2 \rho_1 + \ldots + \varphi_p \rho_{p-1}$$

$$\rho_2 = \varphi_1 \rho_1 + \varphi_2 + \ldots + \varphi_p \rho_{p-2}$$

$$\vdots$$

$$\rho_p = \varphi_1 \rho_{p-1} + \varphi_2 \rho_{p-2} + \ldots + \varphi_p$$

Estimate γ_k by $\hat{\gamma}_k$, where

$$\hat{\gamma}_k = \frac{\sum_{t=1}^{n-k}(X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^{n}(X_t - \bar{X})^2}, \quad k = 1, 2, \ldots, n-1$$

$$= 0, k \geq n$$

$$= \hat{\gamma}_{-k}, k < 0$$

Solve the resulting equations for $\gamma_1, \gamma_2, \cdots, \gamma_p$ and denote these estimates $\hat{\gamma}_1, \hat{\gamma}_2, \cdots, \hat{\gamma}_p$.

- Called ***Yule-walker estimates***

DataScience@SMU

Burg Estimation

Burg estimates

$$X_t = {}_1X_{t\ 1} + \ldots + {}_pX_{t\ p} + a_t$$

$$X_{p+1} = {}_1X_p + \ldots + {}_pX_1 + a_{p+1}$$

$$X_{p+2} = {}_1X_{p+1} + \ldots + {}_pX_2 + a_{p+2}$$

$$\vdots$$

$$X_n = {}_1X_{n\ 1} + \ldots + {}_pX_{n\ p} + a_n$$

Light board

Use least squares to find $\hat{\varphi}_1,\ldots,\hat{\varphi}_p$ that minimize

$$S_c = \sum_{t=p+1}^{n} \hat{a}_t^2 = \sum_{t=p+1}^{n} \left(X_t \quad \hat{}_1 \quad \hat{}_1X_{t\ 1} \quad \cdots \hat{}_pX_{t\ p} \right)^2$$

DataScience@SMU

Burg Estimation

Burg estimates

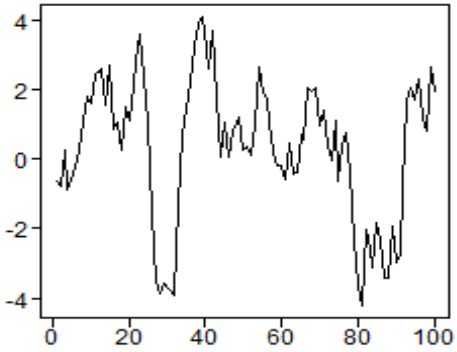
- Note, we had to begin the equations at X_{p+1} because $X_1,\ldots,X_p$ involve values such as X_0, X_{-1}, and so on that are not observed
- This is unfortunate because we're nearly "throwing away" $X_1,\ldots,X_p$
- To adjust for this consider the "backwards" model

$$X_t = \ _1X_{t+1} + \ _2X_{t+2} + \cdots + \ _pX_{t+p} + \ _t$$

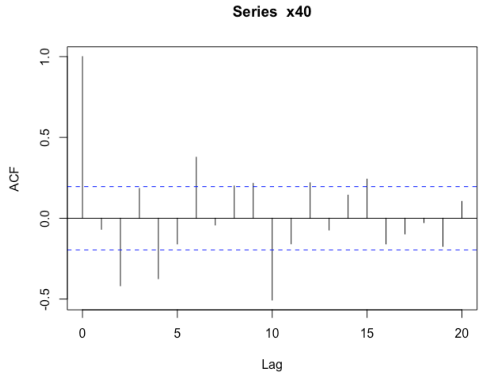
  - In AR models, there is no preference in time direction (recall $\rho_k = \rho_{-k}$)
  - Process "driving process forward" also drives it backward

Burg estimates

AR(1) Realization

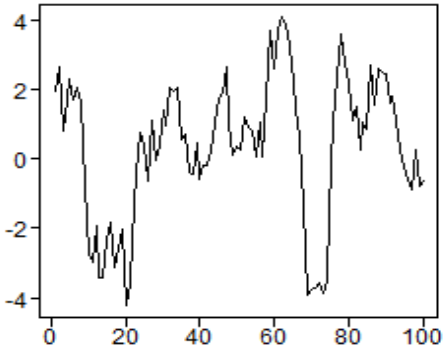


x40

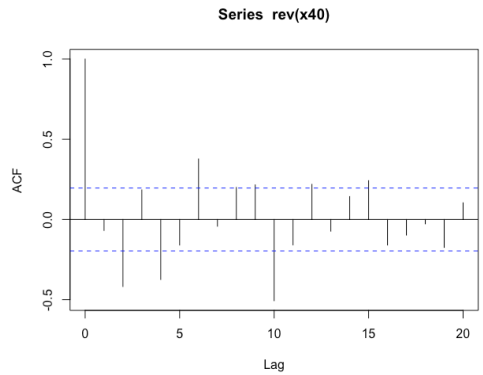


acf(x40)

Realization “reversed” in time



rev(x40)



acf(rev(x40))

“Flipping the time (horizontal axis) does not change the general appearance or apparent correlation structure of the realization.”

DataScience@SMU

Burg Estimation

The Backward Model

Burg estimates

"Backward" model

$$X_t = {}_1X_{t+1} + \ldots + {}_pX_{t+p} + {}_t$$

$$X_1 = {}_1X_2 + \ldots + {}_pX_p + {}_1$$

$$X_2 = {}_1X_3 + \ldots + {}_pX_{p+1} + {}_2$$

$$\vdots$$

$$X_{n\ p} = {}_1X_{n\ p+1} + \ldots + {}_pX_n + {}_{n\ p}$$

Light board

Forward-backward least squares (FBLS) estimates found by minimizing

$$\sum_{t=p+1}^{n} a_t^2 + \sum_{t=1}^{n-p} {}_t^2 = FBSS$$

- FBLS makes better use of **all** of the data.

Burg estimates

Burg used the Durbin-Levinson algorithm to minimize FBSS under the constraint that the model is stationary.

- These estimates are called “Burg” estimates
- Burg algorithm given in Woodward et al. (2017)

DataScience@SMU

tswge and Estimation

tswge demo

`tswge` function `est.ar.wge` computes the following types of estimates:

- yw
- burg
- mle

est.ar.wge(x, p, factor = TRUE, type)

```
x=gen.arma.wge(n=200,phi=c(1.6,-.9),vara=2,sn=33)
x.yw=est.ar.wge(x,p=2,type='yw')
x.yw
x.burg=est.ar.wge(x,p=2,type='burg')
x.mle=est.ar.wge(x,p=2,type='mle')
```

All three estimation types (yw, burg, and mle) gave good estimates of the parameters $\varphi_1 = 1.6$ and $\varphi_2 = -.9$.

- Note that x.yw, x.burg and x.mle displays details about the respective estimates.
- We see that all three methods yield white-noise variance estimates close to 2.12

DataScience@SMU

Comparing Estimates

tswge demo

$$X_t \quad 2.195B + 1.994B^2 \quad .796B^3)X_t = a_t$$

Real root very close to unit circle

Factor	Abs recip	System freq
1-0.995*B*	0.9950	0.0
1-1.2B+0.8B^2	0.8944	0.133

```
x=gen.arma.wge(n=100,phi=c(2.195,-1.994,.796), sn=53)
x.yw=est.ar.wge(x,p=3,type='yw')
```

```
Coefficients of Original polynomial:
1.3413 -0.5734 0.1059
```

```
x.burg=est.ar.wge(x,p=3,type='burg')
```

```
Coefficients of Original polynomial:
2.1204 -1.9000 0.7531
```

```
x.mle=est.ar.wge(x,p=3,type='mle')
```

```
Coefficients of Original polynomial:
2.1411 -1.9164 0.7616
```

tswge demo

$$(1 \quad 2.195B + 1.994B^2 \quad .796B^3)X_t = a_t$$

Real root very close to unit circle

Factor	Abs recip	System freq
$1 - 0.995B$	0.9950	0.0
$1 - 1.2B + 0.8B^2$	0.8944	0.133

	φ_1	φ_2	φ_3
True	2.195	–1.994	.796
YW	1.34	–0.57	.11
Burg	2.12	–1.90	.75
MLE	2.14	–1.92	.76

Roots close to unit circle causes problems with YW!

Recommendation:

Don't use Yule-Walker estimates.

DataScience@SMU

Estimation Summary

Estimation Summary

- The functions est.arma.wge and est.ar.wge are for finding ***stationary*** models when you ***know p and q.***
- For ARMA models with $q > 0$, we will always find ML estimates using est.arma.wge.
  - Maximum likelihood techniques are computationally intensive in the ARMA and AR setting and must be found iteratively
  - est.arma.wge may sometimes find models with roots inside the unit circle
    - These models should not be used as final models.

Estimation Summary

- For AR models, we have three choices available through est.ar.wge:
  1. ML
     - Has the best mathematical properties
     - But it may sometimes find models with roots inside the unit circle
  2. Yule-Walker
     - A very popular estimation technique
     - Fast/non-iterative in nature
     - Will always find a stationary AR model
     - However, it should not be used due to limitations that will be discussed
  3. Burg
     - Fast/non-iterative in nature
     - Will always find a stationary AR model
     - Is preferable over Yule-Walker

DataScience@SMU

Estimation of White-Noise Variance

Estimating σ_a^2

Given a model (assuming $\mu = 0$)

$$X_t \quad {}_1X_{t\ 1} \quad \cdots \quad {}_pX_{t\ p} = a_t \quad {}_1a_{t\ 1} \quad \cdots \quad {}_qa_{t\ q}$$

and solving for a_t, we get

$$a_t = X_t \quad {}_1X_{t\ 1} \quad \cdots \quad {}_pX_{t\ p} + {}_1a_{t\ 1} + \cdots + {}_qa_{t\ q}$$

Given an *estimated* model, we can estimate the noise

$$\hat{a}_t = X_t \quad \hat{}_1X_{t\ 1} \quad \cdots \quad \hat{}_pX_{t\ p} + \hat{}_1\hat{a}_{t\ 1} + \cdots + \hat{}_q\hat{a}_{t\ q}$$

Finding the estimates $\hat{a}_t$ involves issues of starting values, etc.

tswge uses a procedure called backcasting (see Woodward et al. (2017) to find estimates)

$$\hat{a}_t, t = 1, ..., n$$

Estimating σ_a^2

tswge estimates σ_a^2 using the "backcast" residuals.

Letting $\hat{a}_t, t = 1, ..., n$ denote the estimated residuals, then tswge estimates $\hat{\sigma}_a^2$ using the formula

$$\hat{\sigma}_a^2 = \frac{1}{n} \sum_{t=1}^{n} \hat{a}_t^2$$

tswge example

```
x=gen.arma.wge(n=100,phi=c(2.195,-1.994,.796), sn=53)
x.mle=est.ar.wge(x,p=3,type='mle')
x.mle
```

This code produces the ML estimates in the previous AR(3) example. Among the output is the white-noise variance estimate, denoted as $avar. In this case, $avar = 1.007455. That is, $\hat{\sigma}_a^2 = 1.007$.

Note: By default, σ_a^2 = 1 in gen.arma.wge.

Summary

- In this course, we will estimate μ by $\bar{X}$ and σ_a^2 by

$$\hat{\sigma}_a^2 = \frac{1}{n}\sum_{t=1}^{n} \hat{a}_t^2 \text{ (based on backcasting)}$$

- And will use these estimates in our final models

Important notes:

- We have obtained the parameter for stationary ARMA models **assuming we know *p* and *q***
- When analyzing an actual data set, we will not know *p* and *q*
- We address the identification of *p* and *q* for stationary ARMA models ***next***

DataScience@SMU

Model Identification | Introduction

ARMA Model Identification

Given $X_1,\ldots, X_n$—a realization from an ARMA (*p*,*q*) model—we want to **estimate *p* and *q*.**

- "Estimating" *p* and *q* is called *model identification.*
  A number of techniques are available. Among these, the two most popular are:
  - **AIC-type model identification**
  - **Box-Jenkins model identification**
    - A classical approach that is still used by some time series analysts
- We recommend the AIC-type model identification, although we will address both for completeness.

DataScience@SMU

First Step

Check for White Noise!

First step: Test for white noise.

- A common (and embarrassing/costly) mistake is to spend time and resources fitting an ARMA(p,q) model to a set of data when actually the data are white noise—and do not require modeling.
- Before modeling further, ***check for white noise.***

Tests for white noise: $H_0: X_t$ is white noise

$H_a: X_t$ is not white noise

For $k \neq 0,\ \rho_k = 0$ if X_t is white noise

If X_t is white noise: $E(\hat{\rho}_k) \approx 0 \quad k > 0$

$$var(\hat{\rho}_k) \approx \frac{1}{n}$$

$$\text{cov}(\hat{\rho}_k, \hat{\rho}_{k+r}) \approx 0, \quad r \neq 0$$

$\hat{\rho}_k$'s approximately normal

Test the hypotheses: $H_0: \rho_k = 0$

$H_a: \rho_k \neq 0$

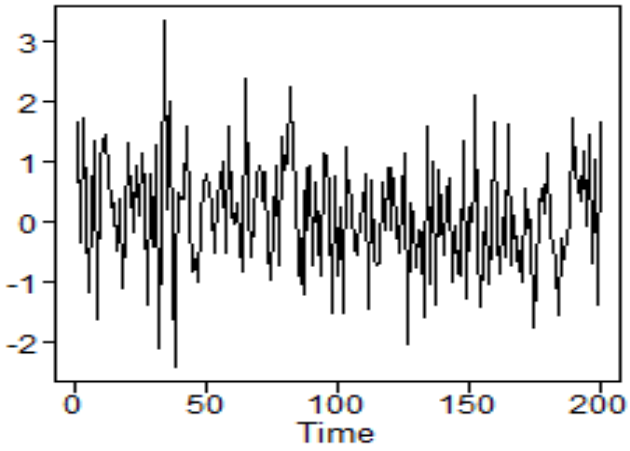
Reject H_0 at 5% level if $|\hat{\rho}_k| > 2\left(1/\sqrt{n}\right)$

Notes:

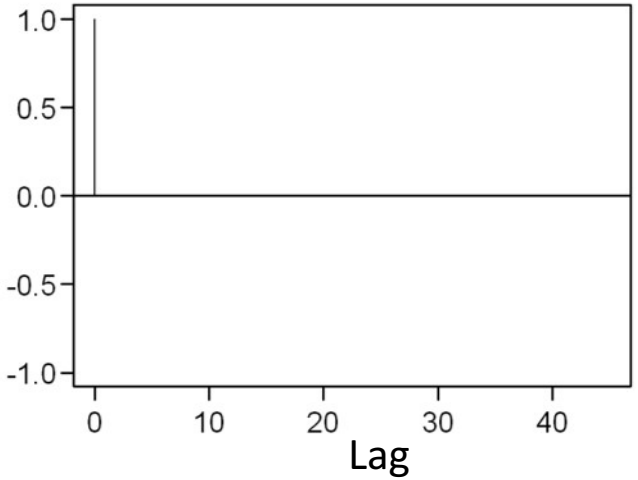
- 5% applies separately for each k and so approximately 5% of the sample autocorrelations may be outside the limits when data are white noise.
- If these occurrences are for small lags, continue modeling.

White Noise

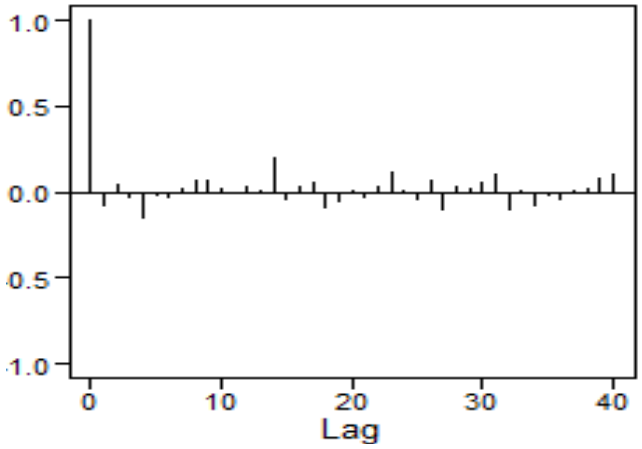
Realization



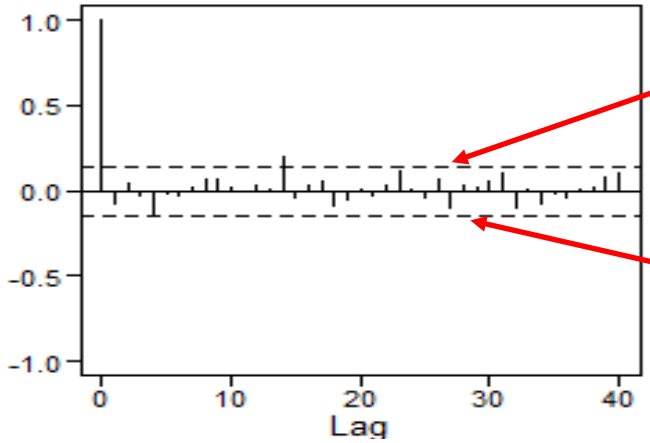
True autocorrelations



Sample autocorrelations



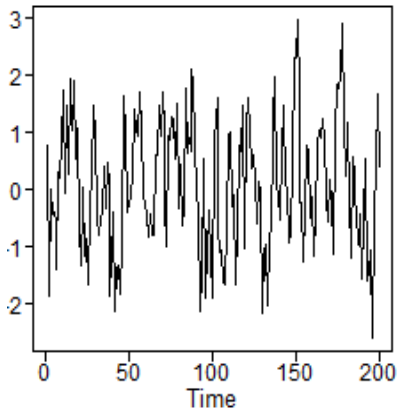
Sample autocorrelations



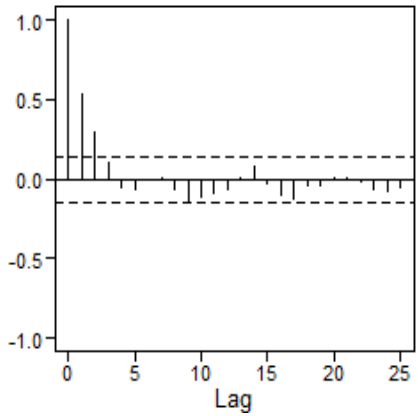
95% limit lines

$$(1 \quad .6B)X_t = a_t$$

Realization

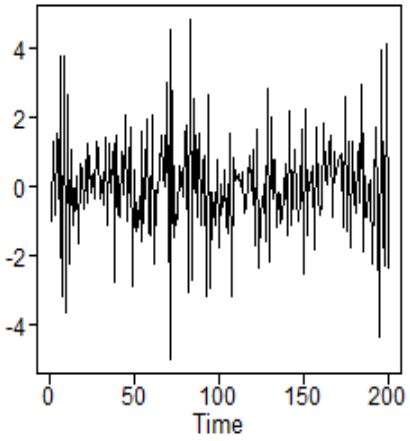


Sample autocorrelations

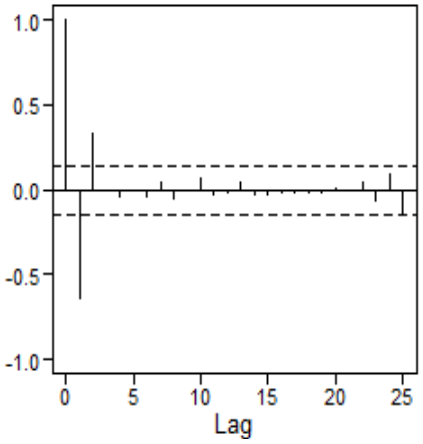


$$X_t = (1 \quad .9B \quad .99B^2)a_t$$

Realization



Sample autocorrelations



DataScience@SMU

Observations

- Only 2 out of 25 sample autocorrelations lie outside limit lines
- Sample autocorrelations look correlated
- Damped exponential pattern at early lags
- ***Not white noise***
- ***What model?***

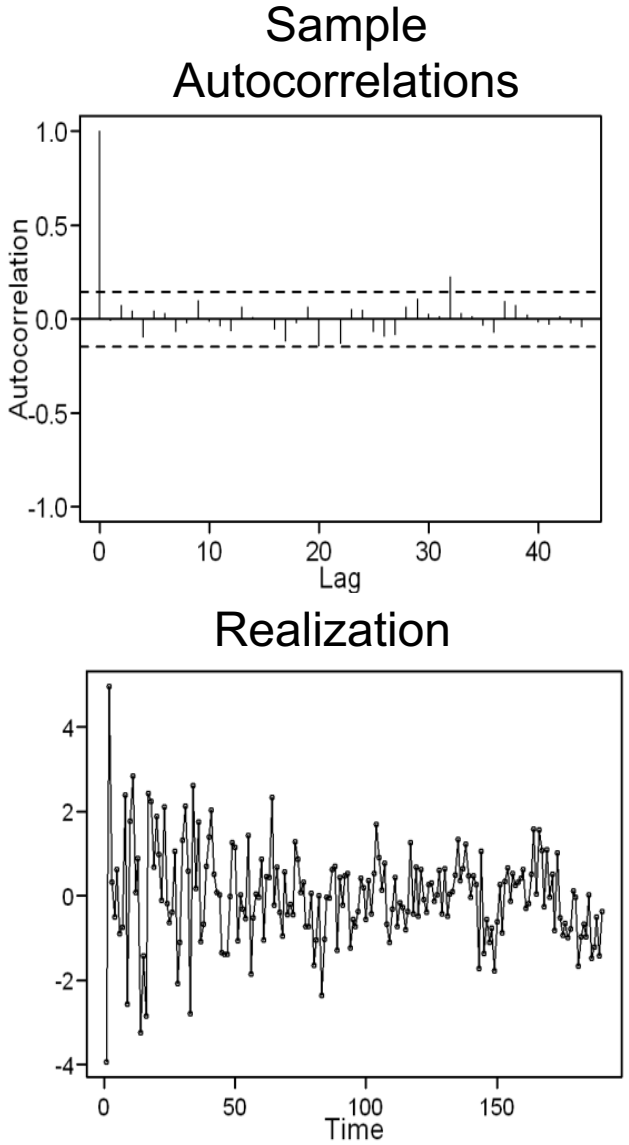
Observations

- Only 2 out of 25 sample autocorrelations lie outside limit lines
- There is a distinct drop in magnitude of sample autocorrelations after lag 2
- ***Not white noise***
- ***What model?***

Other tests for white noise

- Portmanteau tests (discussed in Chapter 9, Woodward et al. (2017) and in a Unit to come.)
- Cumulative spectrum
- Tests for randomness

Another Example



Observations

- Only 1 of 40 sample autocorrelations lie outside limit lines
- Sample autocorrelations look uncorrelated
- ***Sample autocorrelations indicate white noise***

Observations

- Higher variability early in realization
- Definitely not simply random noise
- ***Realization indicates data are NOT white noise***
  - Even though sample autocorrelations suggested white noise

Lesson

Always plot the data!

DataScience@SMU

Model Identification | AIC-Type

Introduction

Suppose that we have decided that the data are not white noise, and we wish to fit an appropriate AR/ARMA model to the data.

First step: Check for White Noise

Second step: Identify/estimate *p* and *q*.

Method 1: AIC

Method 2: Box-Jenkins

Suppose that we have decided that the data are not white noise, and we wish to fit an appropriate AR/ARMA model to the data.

Second step: Identify/estimate *p* and *q*.

Method 1: AIC

Method 2: Box-Jenkins

AIC

Akaike's Information Criterion

- A general criterion for statistical model identification that you have seen before in other settings
- Is applicable to the problem of identifying p and q in an ARMA model
- AIC is actually one of a number of information-based criteria for model selection; others include:
  - AICC
  - BIC
  - Others
- ***The technique we recommend***
  - Here, the term AIC is used generically to represent the information-based techniques

DataScience@SMU

AIC in General

Multiple Regression Example

`Job Score:` measure of job performance

`Test1 and Test2:` skills tests

Question: Can skills tests predict job performance, and which tests are needed?

Prediction just using Test1

`Job Score= -87.136+2.575*Test1`

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	71925.647	1	71925.647	66.004	.000[b]
	Residual	25063.640	23	1089.723		
	Total	96989.288	24			

"Unexplained sum-of-squares"

Will using Test1 and Test2 help explain more?

`Job Score= -93.71+2.534*Test1+.106*Test2`

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	72125.930	2	36062.965	31.910	.000[b]
	Residual	24863.358	22	1130.153		
	Total	96989.288	24			

"Unexplained sum-of-squares" is smaller
⇒ use Test2?

Comments

- Recall from multiple regression that adding a new variable will always decrease the unexplained variability.
  - Even if the new variable is totally unrelated to the dependent variable
- Suppose that you have $\hat{y} = 6 + 3x_1$ as your prediction equation with only x_1.
  - If you add x_2, then you have a multitude of choices for the coefficients b_0, b_1, and b_2 in the prediction equation $\hat{y} = b_0 + b_1 x_1 + b_2 x_2$
  - One of these choices is $b_0 = 6$, $b_1 = 3$, and $b_2 = 0$
    - That is, you can't do worse than you did using only x_1.

NOTE: The multiple regression example is from Elliott and Woodward (2016), and the computer output is from IBM SPSS.

Recall: *in multiple regression*

- Adding variables to an existing equation will always reduce the residual/unexplained variability
- So, the fact that the residual/unexplained variability is reduced by adding a new variable is not sufficient evidence to conclude that the new variable is important to the prediction
- Our goal is to reduce the residual/unexplained variability—***to a point***
  - If the goal were to reduce residual/unexplained variability as much as possible, we would always use **all** the available independent variables.

Key point: *ARMA modeling is similar*

- When fitting an ARMA model to a set of data, a goal is to explain as much of the variability in the data as is "reasonably possible."
  - in ARMA modeling, the unexplained variability is measured by $\widehat{\sigma}_a^2$
- Suppose that we fit an AR(1) model, say $X_t - .8X_{t-1} = a_t$, to a realization and associated with this fit is estimated white-noise variance $\widehat{\sigma}_a^2$.
  - Then, if we fit an AR(2) model to the data, one choice for the AR(2) model is $X_t - .8X_{t-1} + 0X_{t-2} = a_t$, so the new estimated white-noise variance must be at least as small as $\widehat{\sigma}_a^2$ obtained from the AR(1) fit.

DataScience@SMU

AIC in ARMA Modeling

In ARMA modeling, we need a strategy for increasing the order of our model only if the increase in order is sufficiently helpful.

AIC: imposes a penalty for adding terms

- Theoretically involves maximized likelihood considerations
- We will use the following approximate AIC that is easy to interpret and implement:

Select the ARMA(p,q) model for which

$$AIC = \ln(\hat{\sigma}_a^2) + 2(p+q+1)/n$$

is minimized.

AIC Criterion

Select the ARMA(p,q) model for which

$$AIC = \ln(\hat{\sigma}_a^2) + 2(p+q+1)/n$$

is minimized.

Comments:

- The AIC decision rule is to choose the orders p and q that reduce the (ln) white-noise variance subject to a penalty for increasing the number of parameters.
- A variety of other penalty functions have been proposed. Two that are discussed in Woodward et al. (2017) and included as options in tswge are:
  - AICC
  - BIC

AIC Comments

- AIC has a tendency to select a higher-order model as the realization length increases.
- BIC imposes a stronger penalty for increasing p and q .
  - It tends to pick models with fewer parameters.
  - It avoids the problem of “over-modeling” large data sets.
  - AICC has a penalty between that for AIC and BIC.
- AIC and its variations are for selecting a ***stationary ARMA model.***

DataScience@SMU

AIC in tswge

AIC in tswge

AIC is implemented by specifying a range of values for p and q that will be considered as possible model orders.

- Given a p,q pair in the specified range, an ARMA(p,q) model is fit to the data and AIC is calculated.
- This procedure is repeated for each p,q pair in the range.
  - Example: p=0:2, q=0:1 means that the following ARMA(p,q) models will be fit to the data
    - ARMA(0,0) ARMA(0,1)
    - ARMA(1,0) ARMA(1,1)
    - ARMA(2,0) ARMA(2,1)
- If the ARMA(p_1,q_1) model has the smallest AIC value, then p_1 and q_1 are selected as the model orders.

AIC in tswge

tswge has two functions that use AIC (and its variations) for model identification:

1. aic.wge
2. aic5.wge

aic.wge:

Call statement:

```
aic.wge (x,p=0:5,q=0:2,type="aic")
```

Input: **x** is the data set
p=p1:p2, **q**=q1:q2 (default: **p=0:5** and **q=0:2**)
type: aic (default), **bic**, or **aicc.**

Output: **value**: smallest value of selected type (aic, bic ,or aicc)
p and **q** selected
phi and **theta** estimates (MLE)
vara: white-noise variance estimate for "best" model

DataScience@SMU

tswge

AIC Demo

tswge AIC demo

$$(1 \quad 2.55B + 2.42B^2 \quad .855B^3)X_t = a_t \qquad \sigma_a^2 = 1$$

$$(1 \quad .95B)(1 \quad 1.6B + .9B^2)X_t = a_t$$

```
# fig3.16a is a realization from the AR(3) model
data(fig3.16a)
# plotts.sample.wge provides a "look" at the
# data (which we recommend you always do)
plotts.sample.wge(fig3.16a)
aic.wge(fig3.16a,p=0:5,q=0:2, type='aic')
```

Notes:

- Selects AR(3)
- phi estimates: [1] 2.5245680 -2.3447622 0.8104239
- **vara**=[1] 1.030867

Final model: $(1 \quad 2.52B + 2.34B^2 \quad .81B^3)(X_t \quad 1.57) = a_t \qquad \hat{\sigma}_a^2 = 1.03$

(mean(fig3.16a)=1.57)

tswge AIC demo

$(1 \quad 1.6B + .9B^2)(X_t \quad 10) = (1 \quad .8B)a_t \qquad \sigma_a^2 = 1$

```
x=gen.arma.wge(n=100,phi=c(1.6,-.9),theta=.8,sn=67)
x=x+10
plotts.sample.wge(x)
# no type listed below so it will use aic
aic.wge(x,p=0:8,q=0:4)
# picks ARMA(2,1)
est.arma.wge(x,p=2,q=1)
```

```
$phi    [1]  1.6194830 -0.9131788
$theta  [1] 0.868127
$vara   [1] 1.076196
```

Final model: $(1 \quad 1.62B + .91B^2)(X_t \quad 10.08) = (1 \quad .87B)a_t \qquad \hat{\sigma}_a^2 = 1.08$

(mean(x)=10.08)

DataScience@SMU

A Comment about Model ID in Practice

A Comment about Model Identification

The previous two examples may have left the impression that, given data from an ARMA(p,q) model, AIC will correctly identify the model orders.

In Practice

- AIC or other criteria will not always identify the order (even when data are from an ARMA model).
- As mentioned before, real data will not be ARMA to start with.
- The goal is to find the "best" model in some sense.
  - AIC, BIC, and AICC try to explain as much of the variability as possible.
    - They attempt to minimize the white-noise variance
    - With a constraint on too many parameters

Comments

- Sometimes AIC (or BIC, AICC) will select a model that "you don't like."
  - It has more parameters than needed.
    - Some AR and MA factors nearly cancel.
    - Certain factors of the model seem so weak (roots far away from the unit circle) that they seem pointless.
  - The model is not stationary (or not invertible).
- In this case, it would be helpful to view not just the model with the smallest AIC, but maybe the smallest five: ***aic5.wge().***
- Or, you may want to try another information criterion.
  - BIC might be an option if the model seems to have too many parameters.

DataScience@SMU

tswge and AIC5/Top 5 Models

tswge AIC demo

$$(1 \quad 2.30B + 1.92B^2 \quad .56B^3)(X_t \quad 30) = (1 + .8B)a_t \qquad {}^{2}_{a} = 1$$

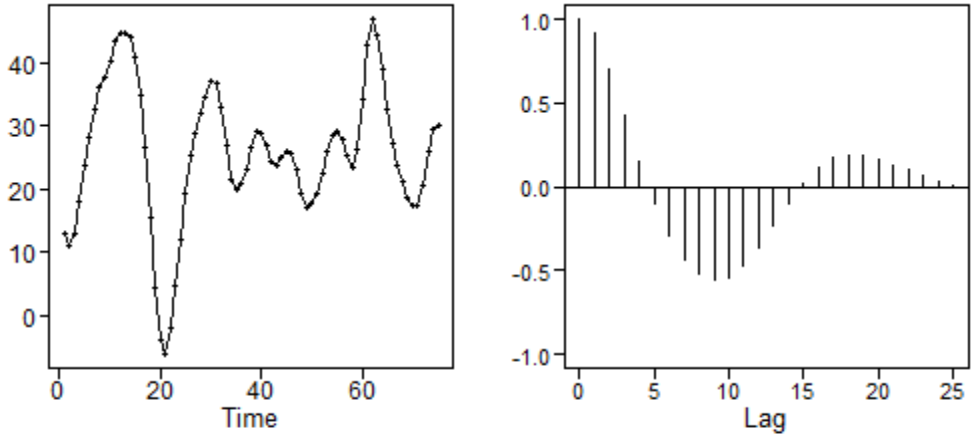
```
x31=gen.arma.wge(n=75,phi=c(2.3,-1.92,+.56),theta=-
.8,sn=61)
x31=x31+30
plotts.sample.wge(x31)
aic5.wge(x31,p=0:8,q=0:2)
# picks (6,1)   try BIC
aic5.wge(x31,p=0:8,q=0:2,type="bic")
#  BIC picks (3,1) - decide to use it
#  showing est.arma.wge results for ARMA(3,1)
est.arma.wge(x31,p=3,q=1)
mean(x31)
#  mean is 25.74
```

Final model: $(1 \quad 2.06B + 1.53B^2 \quad .38B^3)(X_t \quad 25.74) = (1 + .94B)a_t \quad \hat{\sigma}_a^2 = 0.94$

$$(1 \quad 1.55B + .74B^2)(1 \quad .50B)(X_t \quad 25.74) = (1 + .94B)a_t$$

tswge output and discussion

```
plotts.sample.wge(x31)
```



Sample autocorrelations look like a damped sinusoid and the data are pseudo-cyclic.

```
aic5.wge(x31,p=0:8,q=0:2)
```

Five Smallest Values of aic

p	q	aic
6	1	0.04601791
7	1	0.05788330
3	1	0.06646122
8	1	0.07820027
4	1	0.08249288

AIC picks an ARMA(6,1) with an ARMA(3,1) as a third choice.

- ARMA(6,1) and ARMA(7,1) are fairly high-order.
- Is there a lower-order model that is reasonable?

tswge output and discussion

Factor Table for AR(6) Part

Factor	Abs Recip	System Freq
$1 - 1.7790B + 0.8890B^2$	0.9429	0.0538
$1 - 1.1361B + 0.7250B^2$	0.8515	0.1338
$1 + 0.7792B + 0.4583B^2$	0.6770	0.3476

We try aic5 using BIC.

Five Smallest Values of bic

p	q	bic
3	1	0.2209604
2	2	0.2613963
4	1	0.2678919
6	1	0.2932166
5	1	0.2990913

The 6th-order model is probably okay.

- First factor is fairly similar to the 2nd-order factor in model
- No effect of the other two factors is apparent in data, autocorrelations, or Parzen spectrum (not shown)

BIC picks ARMA(3,1). We decided to go with it for simplicity and it showed up 3rd on AIC list.

AIC Model Identification Comments

There will often be more than one model that could be used to model a given set of data.

- Beware of using a model with too few parameters just for simplicity.
  - Key features of the data may be lost.
- If you allow AIC to select a model from a range of p (or q) values, say $p = 0{:}6$ and AIC pick a 6th order, it is good practice to expand the range to determine whether more parameters were needed.

DataScience@SMU

Box-Jenkins and MA(q)

Suppose that we have decided that the data are not white noise, and we wish to fit an appropriate AR/ARMA model to the data.

Second step: Identify/estimate *p* and *q*.

Method 1: AIC

Method 2: Box-Jenkins

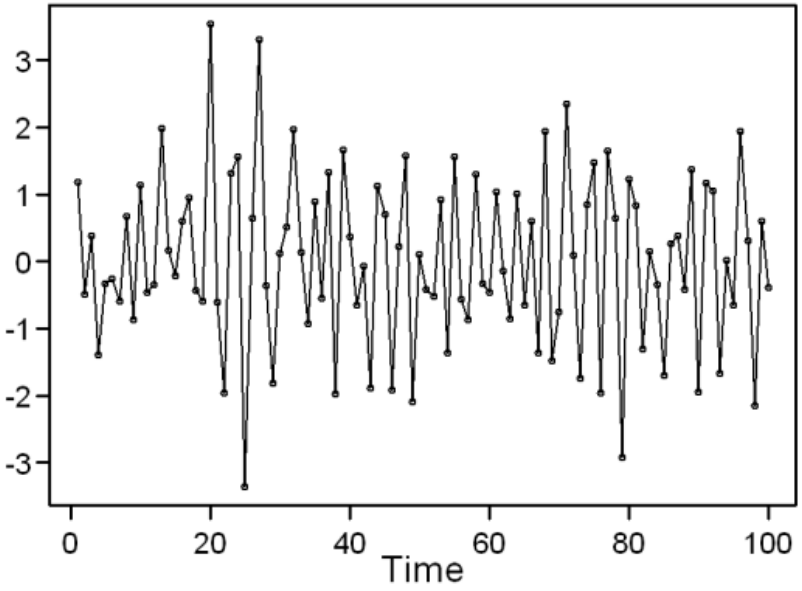
Box-Jenkins Model Identification

This was the earliest or at least one of the first popular ways to fit ARMA models to data.

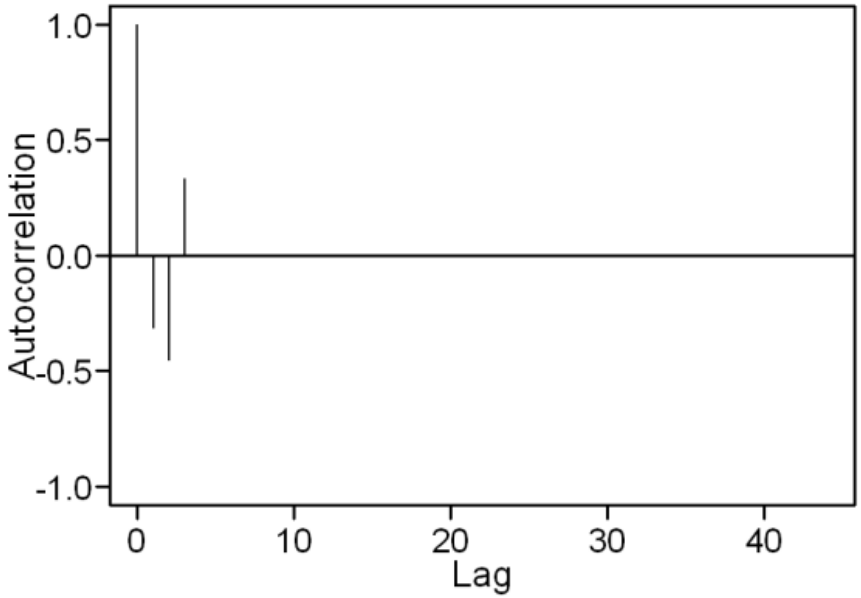
- It is still widely used (although we prefer AIC-type methods).
- It is a pattern-recognition technique.
- It involves examining the autocorrelations and “partial autocorrelations.”

Model Identification for MA(q) Models

Identifying characteristic is that $\rho_k = 0,$ for $k > q$



(a) Realization



(b) True Autocorrelations

Model Identification for MA(q) Models

Identifying characteristic is that $\rho_k = 0,$ for $k > q$

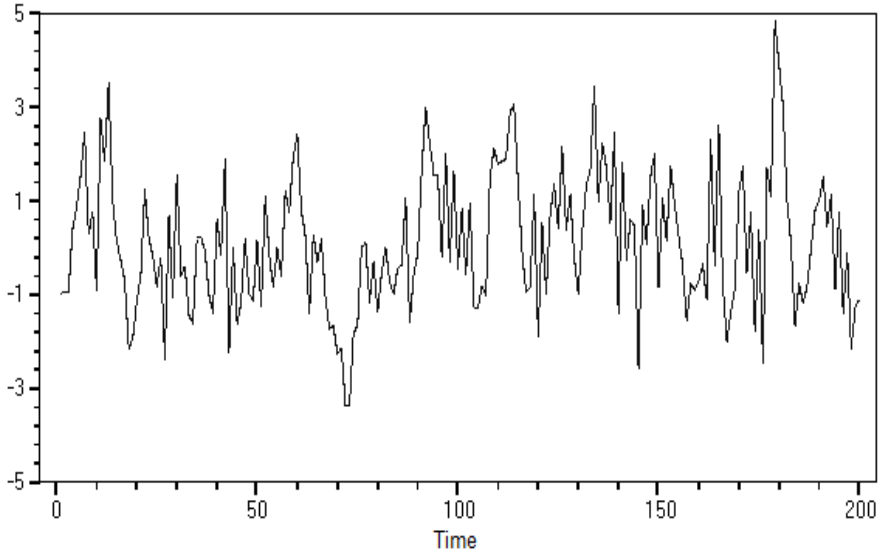
So, for an MA(q), it should follow that $\hat{}_k \quad 0,\ k > q$

A rough guide for testing for an MA(*q*)

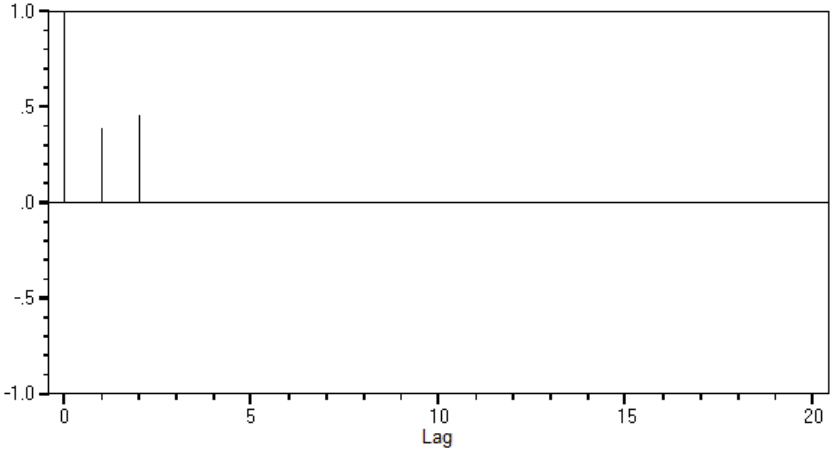
$$| \hat{}_k | > 2\sqrt{(1 + 2 \sum_{v=1}^{q} \hat{}_k^2) / n} \quad \text{for some } k > q$$

is evidence against an MA(q) at $= .05$ level

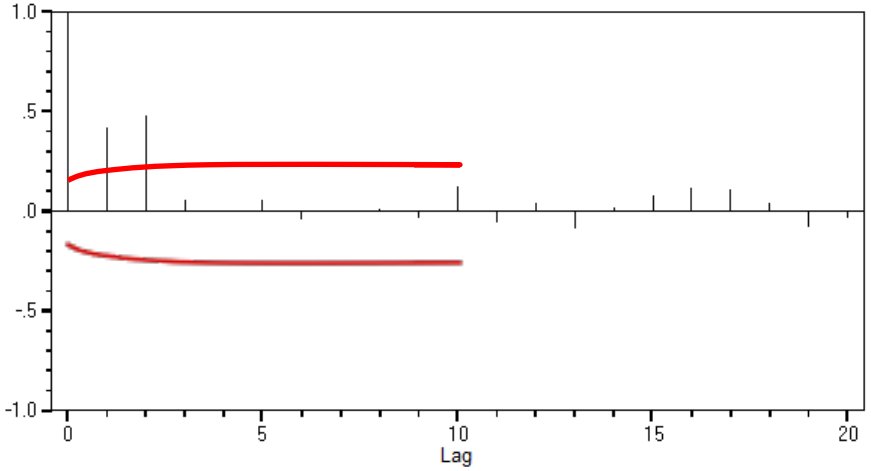
Realization from $X_t = a_t + .4a_{t-1} + .9a_{t-2}$



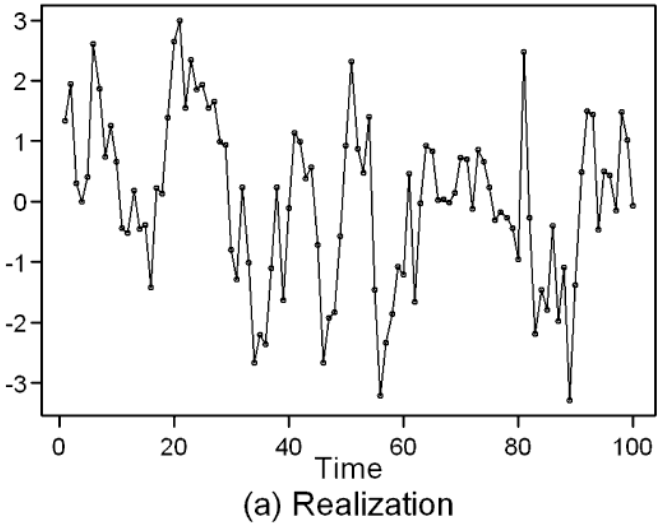
True Autocorrelations



Sample Autocorrelations

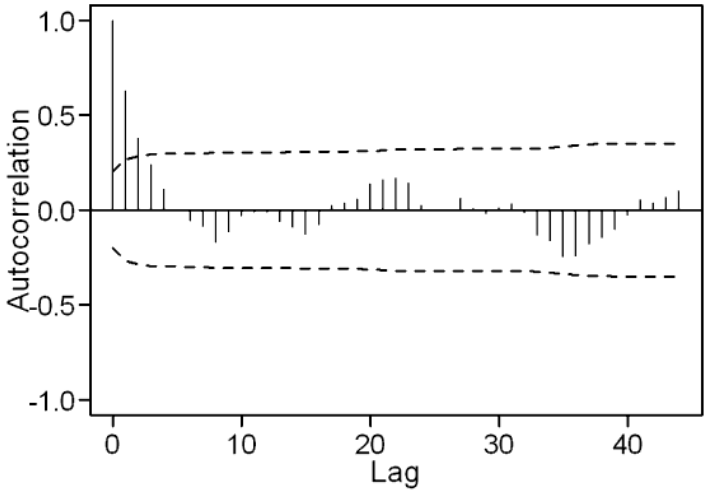


Realization



(a) Realization

Sample autocorrelations



(c) Sample Autocorrelations

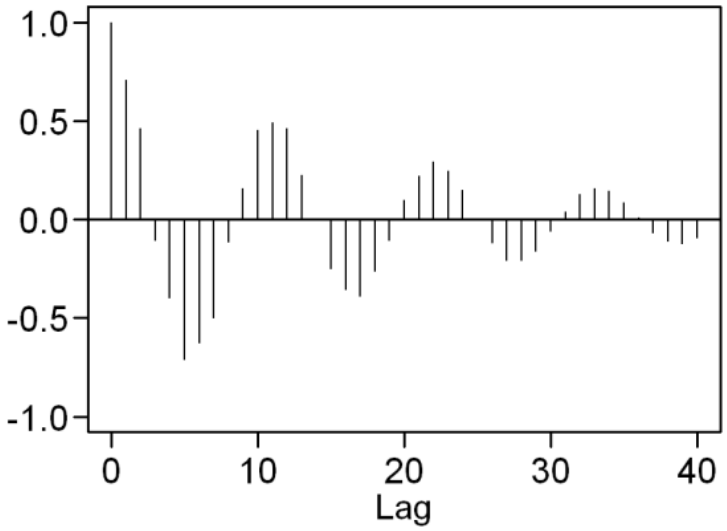
Only the first 2 sample autocorrelations are outside the limit lines.

- But there is not a distinct drop to “nearly zero” after lag 2
- Looks sort of AR-like but not sure

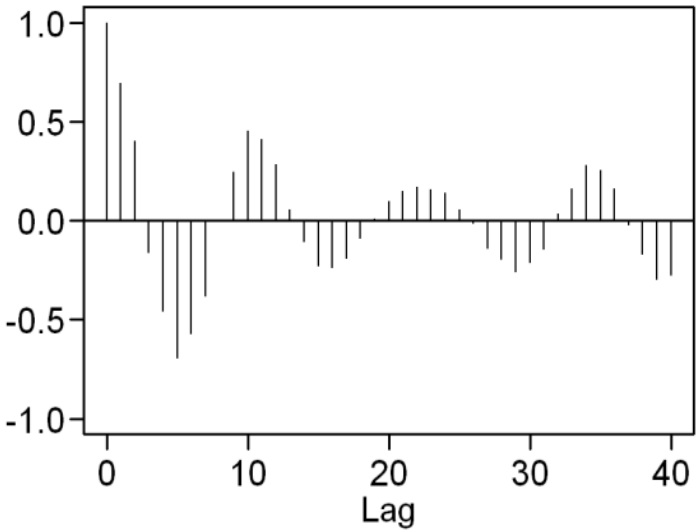
MA(2)? AR(2)?

What Model?

True autocorrelations



Sample autocorrelations



Autocorrelation alone (even true autocorrelation) alone does not tell us.

MA? **AR?**

ARMA?

DataScience@SMU

Box-Jenkins and AR(p)

Partial Autocorrelation Function

Partial Autocorrelation Function: φ_{kk}

Two definitions

φ_{kk} - correlation between X_t and X_{t-k} after removing the effect of the intervening variables $X_{t-1}, X_{t-2}, ..., X_{t-k+1}$

φ_{kk} - kth autoregressive coefficient obtained by "assuming" X_t is AR(k) and solving the YW equations for φ_k

Note: If X_t is an AR(2) $X_t = {}_1X_{t\ 1} + {}_2X_{t\ 2} + a_t$

- We know $\varphi_2 \neq 0$ (or it wouldn't be an AR(2))
- φ_1 could be either zero or non-zero
- If we wrote the AR(2) model as

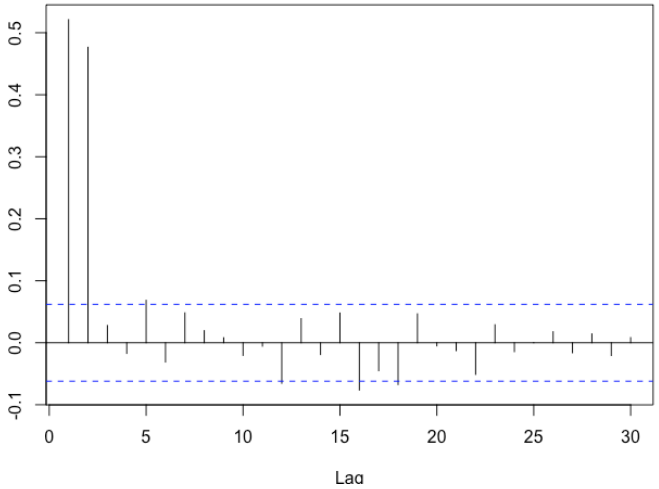
$$X_t = {}_1X_{t\ 1} + {}_2X_{t\ 2} + {}_3X_{t\ 3} + a_t$$

then $\varphi_3 = 0$

- If we write this as an AR(p) for $p > 2$, then $\varphi_3 = \varphi_4 = \cdots = \varphi_p = 0$

So: If $X_t \sim$ AR(2)

- $\varphi_{22} \neq 0$ (or it wouldn't be an AR(2))
- $\varphi_{kk} = 0,\ k > 2$




```
t = gen.arma.wge(1000,phi = c(.3,.5), sn = 1)
pacf(t)
```

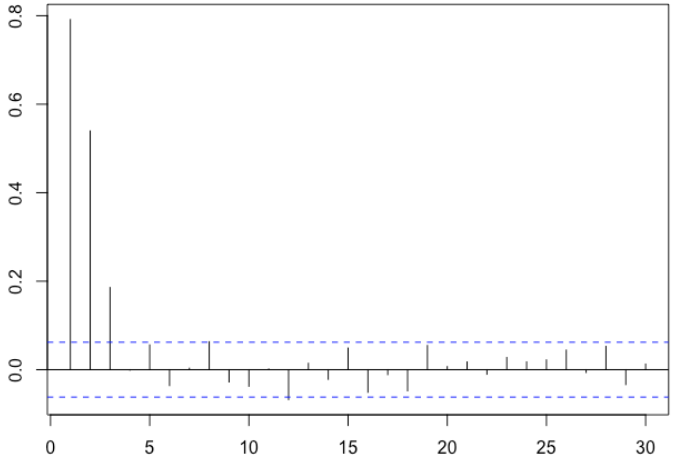
In general:

If $X_t \sim \text{AR}(p)$

- $\varphi_{pp} \neq 0$ (or it wouldn't be an AR(p))
- $\varphi_{kk} = 0,$ for $k > p$

$$1 - .3X_{t-1} - .5X_{t-2} - .15X_{t-3} = at$$

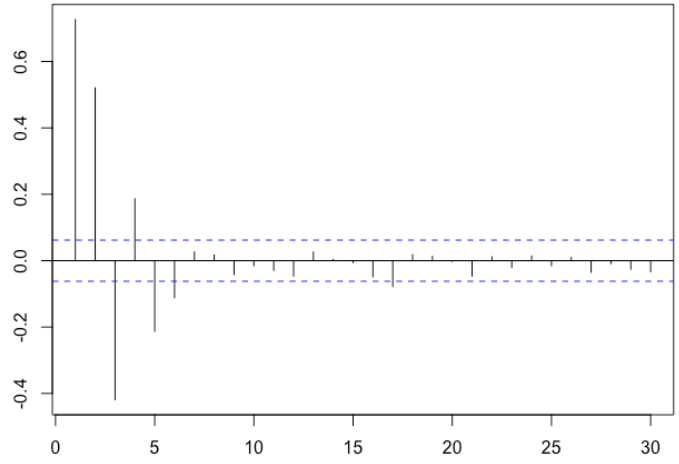
Sample *partial* autocorrelations



t = gen.arma.wge(1000,phi = c(.3,.5,.15), sn = 1)
pacf(t)

$$1 - .7X_{t-1} - .5X_{t-2} + .5X_{t-3}$$
$$-.4X_{t-4} + .2X_{t-5} + .1X_{t-6} = at$$

Sample *partial* autocorrelations



t = gen.arma.wge(1000,phi = c(.7,.5,-.5,.4,-.2,-.1), sn = 1)
pacf(t)

DataScience@SMU

Estimating Partial Autocorrelation from Data

Estimating Partial Autocorrelations from Data

To estimate φ_{kk}, $k = 1, 2, \ldots$

- Use any estimation procedure (Burg, YW, MLE, etc.)

Procedure

To the data set in question, successively fit:

- An AR(1) (and $\hat{\varphi}_{11} = \hat{\varphi}_1$)
- An AR(2) (and $\hat{\varphi}_{22} = \hat{\varphi}_2$)

…

- An AR(k) (and $\hat{\varphi}_{kk} = \hat{\varphi}_k$)

(These are called sample partial autocorrelations.)

Decision rule

Determine whether or not it is reasonable to conclude $\varphi_{kk} = 0$ beyond some point

- by comparing sample partial autocorrelations, $\hat{}_{kk}$, against limits $\pm 2/\sqrt{n}$

DataScience@SMU

Box-Jenkins

Summary for MA(q) and AR(p)

Summarizing

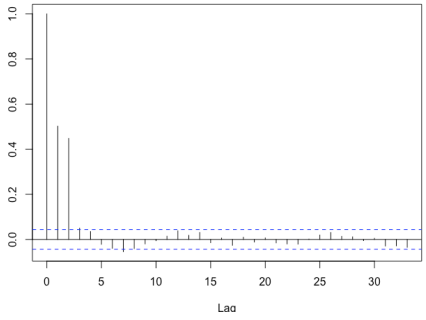
Box-Jenkins model identification procedure for AR(*p*) and MA(*q*) models

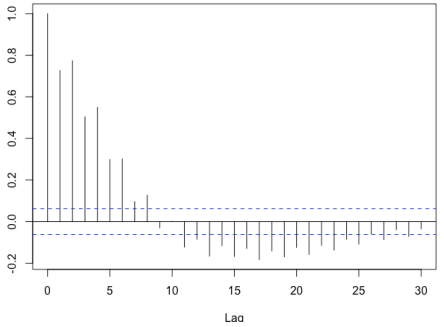
Examine plots of $\hat{\rho}_k$ and $\hat{\varphi}_{kk}$ looking for patterns:
that approximate the following theoretical patterns

Sample autocorrelations **Sample *partial* autocorrelations**

MA(*q*)

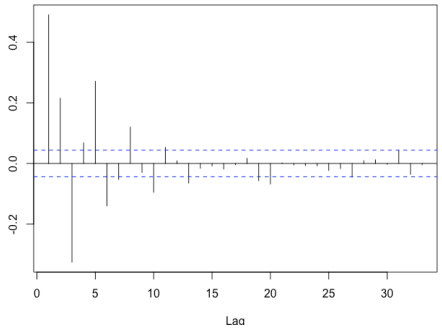
$\rho_k = 0$ for $k > q$

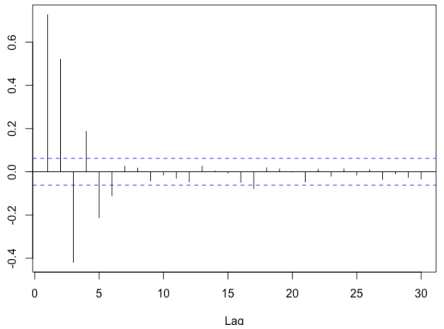




AR(*p*)

$\varphi_{kk} = 0$ for $k > p$





DataScience@SMU

Box-Jenkins: The Problem

ARMA(p,q)

The Problem Comes When Identifying ARMA(p,q) Models with $p > 0$ and $q > 0$

Box-Jenkins procedure is based on the fact that, for an ARMA(p,q):

ρ_k is a mixture of damped exponentials

and/or sinusoids for $k > q - p$

φ_{kk} is dominated by a mixture of damped exponentials

and/or sinusoids for $k > p - q$

These patterns can be difficult to identify, even with true autocorrelations and partial autocorrelations.

- Woodward and Gray (1981) developed the idea of generalized partial autocorrelations to address this issue.
- **However, modern model ID techniques involve AIC-type procedures.**

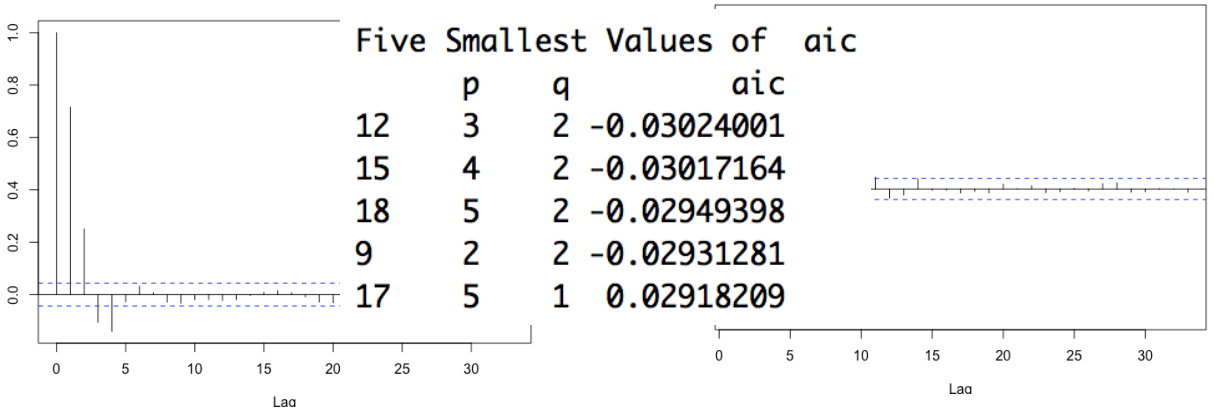
See Woodward et al. (2017).

Summarizing

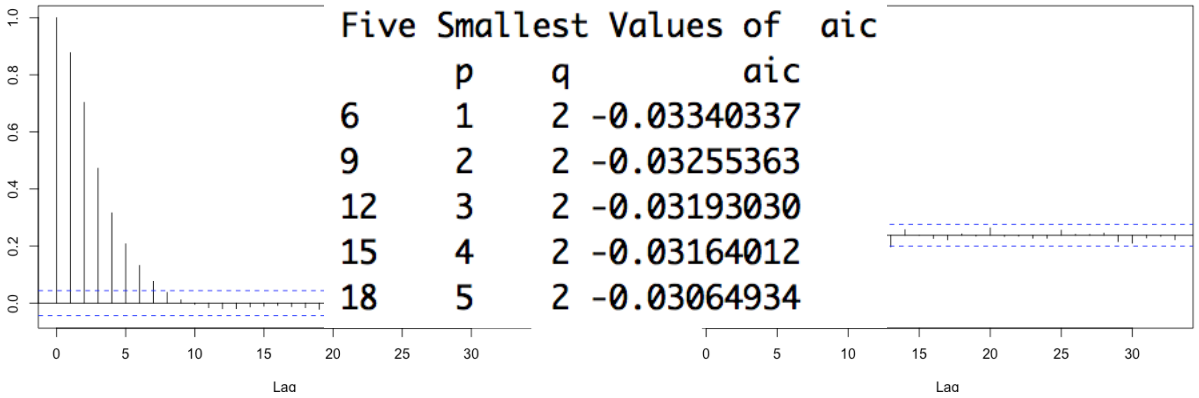
Box-Jenkins model identification procedure for AR(*p*) and MA(*q*) models

Sample autocorrelations **Sample *partial* autocorrelations**

ARMA (*3,2*)



ARMA(*1,2*)



DataScience@SMU

Putting It All Together

Let's Put It All Together!

In practice, we will (assuming we believe the data are from a stationary process):

1. Visualize/plot the data/check for white noise
2. Use aic.wge() or aic5.wge() to identify estimates of *p* and *q*
3. Use the estimate of *p* and *q* to get estimates of the phis and thetas using est.ar.wge() or est.arma.wge()
4. Use the estimated model to forecast and so on

Consider the data in the file PutItAllTogether.csv. Use these data to fit a the model that the AIC favors the most, and use this model to forecast the next 20 values. We will review the answers in the next element!

Let's Put It All Together!

Consider the data in PutItTogether1.csv.

1. Visualize/plot the data.

   plotts.wge(PutItTogether) #data read into array PutItTogether

2. Use aic.wge() or aic5.wge() to identify estimates of *p* and *q*.

   aic5.wge(PutItTogether)

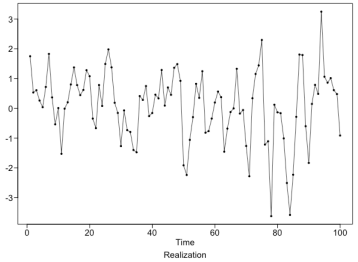
3. Use the estimate of *p* and *q* to get estimates of the phis and thetas.

   m = est.arma.wge(PutItTogether, p = 2, q = 2)

4. Use the estimated model to forecast and so on.

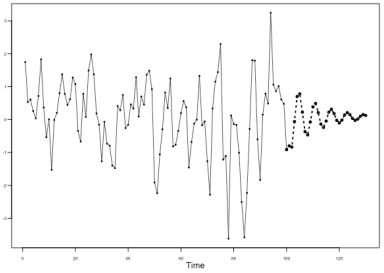
   fore.arma.wge(PutItTogether, phi = m$phi, theta = m$theta, n.ahead = 20)

Note: We are assuming a stationary process and have not investigated the spectral density, acfs, and so on to make sure that key properties of the data are preserved. We will address all of these topics together in the model-building unit to come!




```
Five Smallest Values of  aic
      p    q         aic
9     2    2 -0.07985840
12    3    2 -0.06708593
15    4    2 -0.05356675
18    5    2 -0.03918021
16    5    0 -0.01122233
```

```
> m$phi
[1]  0.9464566 -0.8277521
> m$theta
[1]  0.5355606 -0.8072125
```



DataScience@SMU

Example

Jet Fuel A

Examples: Jet Fuel

In a previous units, we looked at jet fuel prices and found that the aic identified a model with $p = 1$ and $q = 1$ and estimated the model to be

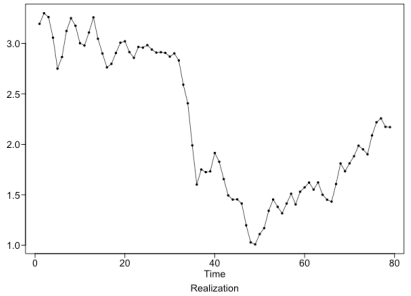
$$(1 - .967B)(X_t - 2.20) = (1+.477B)a_t$$

We will verify the estimates of *p, q,* and the model parameters and use that model to forecast the next 8 months of fuel prices. *Note: This assumes that we have already made the assumption that the data are stationary. We know that this assumption may be wrong and will definitely affect the forecasts.*

Examples: Jet Fuel

1. Visualize/plot the data.

   plotts.wge(jet) #data read into array jet



2. Use aic.wge() or aic5.wge() to identify estimates of *p* and *q*.

   aic5.wge(jet)

```
Five Smallest Values of  aic
      p    q        aic
5     1    1  -4.195998
10    3    0  -4.190446
6     1    2  -4.172064
8     2    1  -4.171285
13    4    0  -4.165614
```

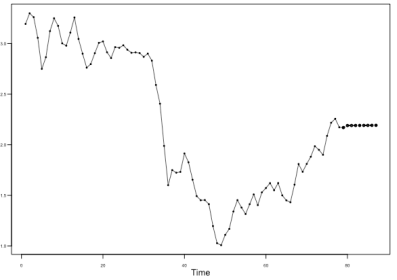
3. Use the estimate of *p* and *q* to get estimates of the phis and thetas.

   est = est.arma.wge(jet, p = 1, q = 1)

```
> est$phi
[1] 0.9673663
> est$theta
[1] -0.476637
```

4. Use the estimated model to forecast and so on.

   fore.arma.wge(jet, phi = est$phi, theta = est$theta, n.ahead = 8)



DataScience@SMU

Example

Canadian Lynx

Examples: Canadian Lynx Data

In a previous units, we looked at the log Canadian Lynx and found that the aic identified a model with $p = 4$ and $q = 1$ and estimated the model to be:

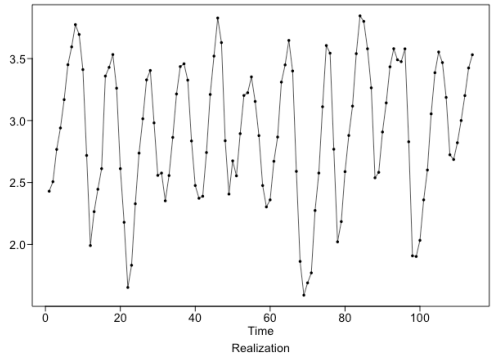
$$(1 - 0.7B - 0.1B^2 + 0.2B^3 + 0.3B^4)(X_t - 2.9) = (1 + .6B)a_t$$

We will verify the estimates of *p, q,* and the model parameters and use that model to forecast the next 24 months of fuel prices. *Note: This assumes that we have already made the assumption that the data are stationary. We know that this assumption may be wrong and will definitely affect the forecasts.*

Examples: Canadian Lynx Data

1. Visualize/plot the data.

   data(llynx)
   plotts.wge(llynx)



2. Use aic.wge() or aic5.wge() to identify estimates of *p* and *q*.

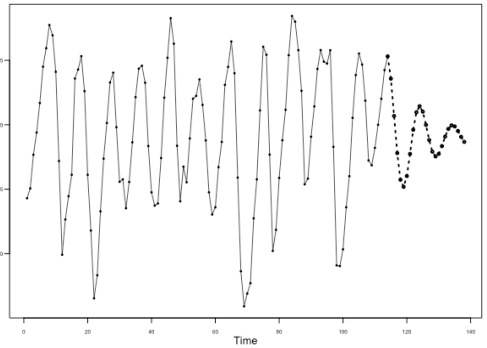
   aic5.wge(llynx)

```
Five Smallest Values of  aic
      p    q        aic
14    4    1  -2.951518
16    5    0  -2.951301
13    4    0  -2.949515
12    3    2  -2.942965
17    5    1  -2.936873
```

3. Use the estimate of *p* and *q* to get estimates of the phis and thetas.

   est = est.arma.wge(llynx, p = 4, q = 1)

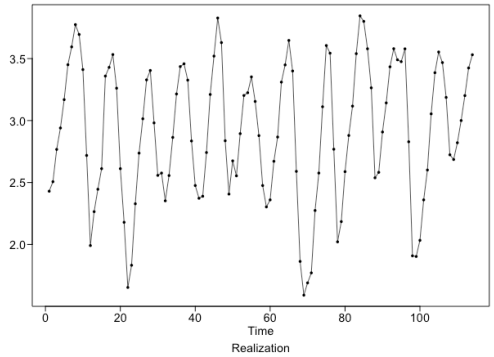
```
> est$phi
[1]  0.68232931  0.06296222 -0.20555174 -0.26022793
> est$theta
[1] -0.6211485
```



4. Use the estimated model to forecast and so on.

   fore.arma.wge(llynx, phi = est$phi, theta = est$theta, n.ahead = 24)

Examples: Canadian Lynx Data



1. Visualize/plot the data.

   data(llynx)
   plotts.wge(llynx)

2. Use aic.wge() or aic5.wge() to identify estimates of *p* and *q*.

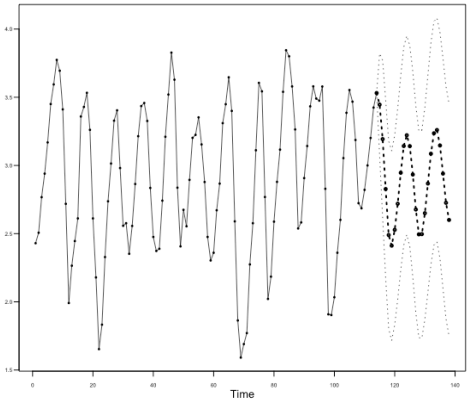
   aic5.wge(llynx, p = 0:15, q = 0:2)

```
Error in aic calculation at 14 2
Five Smallest Values of  aic
      p    q        aic
37   12    0  -3.128705
35   11    1  -3.121965
34   11    0  -3.121655
36   11    2  -3.115068
40   13    0  -3.113269
```

3. Use the estimate of *p* and *q* to get estimates of the phis and thetas.

   est = est.arma.wge(llynx, p = 11, q = 0)

```
Coefficients of Original polynomial:
1.1676 -0.5446 0.2662 -0.3094 0.1540 -0.1463 0.0569 -0.0294 0.1346 0.2021 -0.3394
```



4. Use the estimated model to forecast and so on.

   fore.arma.wge(llynx, phi = est$phi, theta = est$theta, n.ahead = 24)

DataScience@SMU

Closing Remarks

Model Identification and Parameter Estimation

Closing Remarks Parameter Estimation and Model Identification Summary

- The basic parameter estimation method is maximum likelihood.
  - Burg estimates have desirable properties for AR models.
  - We don't recommend using Yule-Walker estimates.
- We recommend AIC-type model identification over the Box-Jenkins pattern recognition approaches.

Notes:

- In this unit, we have discussed parameter estimation and model identification based on ***stationary models.***
- In the next unit, we will discuss fitting non-stationary ARIMA, seasonal, and signal-plus-noise models.

DataScience@SMU