Time Series Analysis

Lecture 3

Autoregressive Models and Moving Average Models

datascience@berkeley

Autoregressive Models
Part 3
Expression in Lag Operators

Backshift (or Lag) Operators: An Introduction

Backshift Operator:
A very useful concept is the backshift operator because it and its associated characteristic polynomials can be used to study the properties of $AP(p)$ models (and the $ARIMA(q)$ models in general)

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$

Using the backshift operator, the autoregressive model can be re-written as:

$$\phi(B)x_t = \omega_t$$

The equation $\boxed{\phi(B) = 0}$ is called a *characteristic equation*, and it provides a powerful tool to check if a process is stationary. In particular, if the roots of the characteristic equation *all* exceed unity in absolute value, then the process x_t is *stationary*.

Examples of Using the Backshift Operators

Examples:

1. The random walk model $x_t = x_{t-1} + \omega_t$ has $\phi = 1$ and $\theta = 1 - \mathbf{B}$ with root $\mathbf{B} = 1$. Thus, this is a non-stationary process.

2. The AR(1) model $x_t = \frac{1}{2}x_{t-1} + \omega_t$ has the characteristic equation $1 - \frac{1}{2}B = 0$ and the root of $B = 2 > 1$. Thus, this AR(1) model is stationary.

3. Consider the AR(2) model $x_t = x_{t-1} - \frac{1}{4}x_{t-2} + \omega_t$. Expressed using the backshift operator, $\frac{1}{4}\left(B^2 - 4B + 4\right)x_t = \omega_t$, or $\frac{1}{4}\left(B-2\right)^2 x_t = \omega_t$. The corresponding characteristic equation is $\phi(B) = \frac{1}{4}\left(B-2\right)^2 = 0$, so the root is $B = 2 > 1$. Thus, the AR(2) model is stationary.

4. Consider another AR(2) model $x_t = \frac{1}{2}x_{t-1} + \frac{1}{2}x_{t-2} + \omega_t$, which can be expressed as $-\frac{1}{2}\left(B^2 + B - 2\right)x_t = \omega_t$. The corresponding polynomial $\phi(B) = -\frac{1}{2}(B-1)(B+2)$ has roots $B = 1, -2$. With the unit root $B = 1$, this AR(2) model is *non-stationary*.

Autoregressive Models, Part 4

Intuition of the Properties of the General AR(p) Models

Autoregressive Models in Lag Operators

For a (weakly) stationary $AR(p)$ model, its lag operator presentation is

$$\Phi(L)x_t = \alpha + \epsilon_t$$

$\cancel{x_t} = \alpha x_{t-1} + \varepsilon_t$

$\underbrace{(1 - \alpha L)}x_t = \varepsilon_t$

where

$$\left[\Phi(L) = 1 - \phi_1 L - \cdots - \phi_p L^p\right]$$

$1 - \alpha L = 0$

$\Rightarrow \alpha = \frac{1}{L} > 1$

Recall that the stationarity condition for the $AR(p)$ model is that the roots of the characteristic polynomial $\Phi(z)$ all lie outside of the unit circle (or equivalently, the roots of $z^p - \phi_1 z^{p-1} - \cdots - \phi_p$ all lie inside the unit circle.

Inverse of the

$\frac{1}{\alpha} = L < 1$

The mean of this model is

$$E(x_t) = \frac{\alpha}{1 - \phi_1 - \cdots - \phi_p}$$

The autocovariance function of the general $AR(p)$ model can be derived using the *Yule-Walker* equations. ~~We wi~~

Key Properties of the General AR(p) Model

The general properties of an AR(1) model carries through to an AR(p) model.

1. **Stationarity condition:** An AR(p) process is covariance stationary if and only if the inverse of all roots of the autoregressive lag operator polynomial $\Phi(B)$ are inside the unit circle.

2. **ACF:** The autocorrelation function for the AR(p) process decays gradually with displacement.

3. **PACF:** The partial autocorrelation function has a sharp cut-off at **displacement p**.

Key Properties of the General AR(p) Model (2)

However, there are difference between the general AR(p) models and AR(1) models.

1. Models with higher autoregressive order allows for a richer dynamics, and the autocorrelation function displays a wider variety of patterns.
   - For example, it can display damped, monotonic decays, as in the AR(1) case with a positive coefficient, but it can also have damped oscillation that AR(1) can't have unless its coefficient is negative.

2. The richer patterns of the ACF from the higher-order autoregressive models can mimic a wider range of cyclical patterns.

Autoregressive Models

Simulation of AR(2) Models

Use an AR(2) Model to Illustrate the Properties

Consider an AR(2) model of the specification:

$$y_t = 1.5y_{t-1} - 0.9y_{t-2} + \epsilon_t$$

The corresponding lag operator polynomial is

$$1 - 1.5B + 0.9B^2$$

The roots of this polynomial can be easily found in **R** using the **polyroot** function: `polyroot(c(1, -1.5, 0.9))`.

The result is two complex conjugate roots **0.83 +/- 0.65i** .

```
> polyroot(c(1, -1.5,0.9))
[1] 0.8333333+0.6454972i 0.8333333-0.6454972i
```

abs(polyroot(c(,))) 1.054

Use an AR(2) Model to Illustrate the Properties

The inverse roots are `0.75+/-0.58i`:

```
> 1/polyroot(c(1, -1.5,0.9))
[1] 0.75-0.5809475i 0.75+0.5809475i
```

0.948

Both of these roots are close to 1 but nevertheless inside the unit circle.

Therefore, the process is **covariance stationary**.

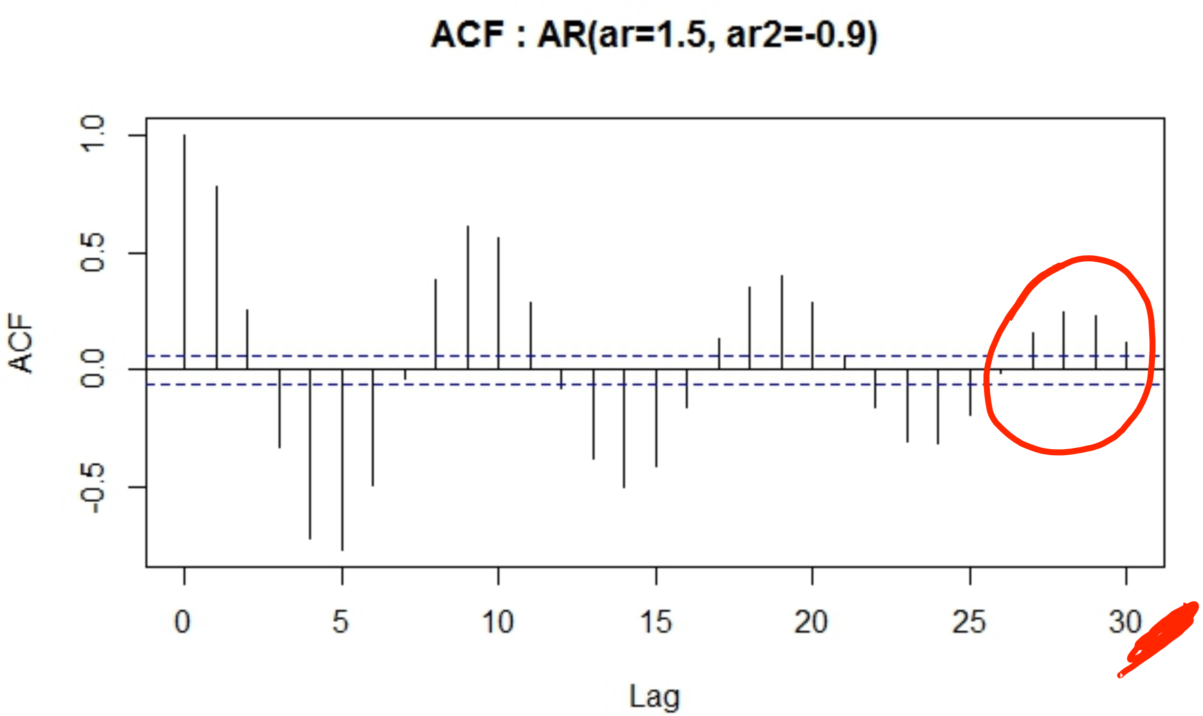
The autocorrelation function of an AR(2) process is

$$
\boxed{
\begin{aligned}
\rho(0) &= 1 \\
\rho(1) &= \frac{\phi_1}{1-\phi_2} \\
\rho(2) &= \phi_1\rho(\tau-1)+\phi_2\rho(\tau-2), \quad \tau = 2,3,\ldots
\end{aligned}
}
$$

Use an AR(2) Model to Illustrate the Properties

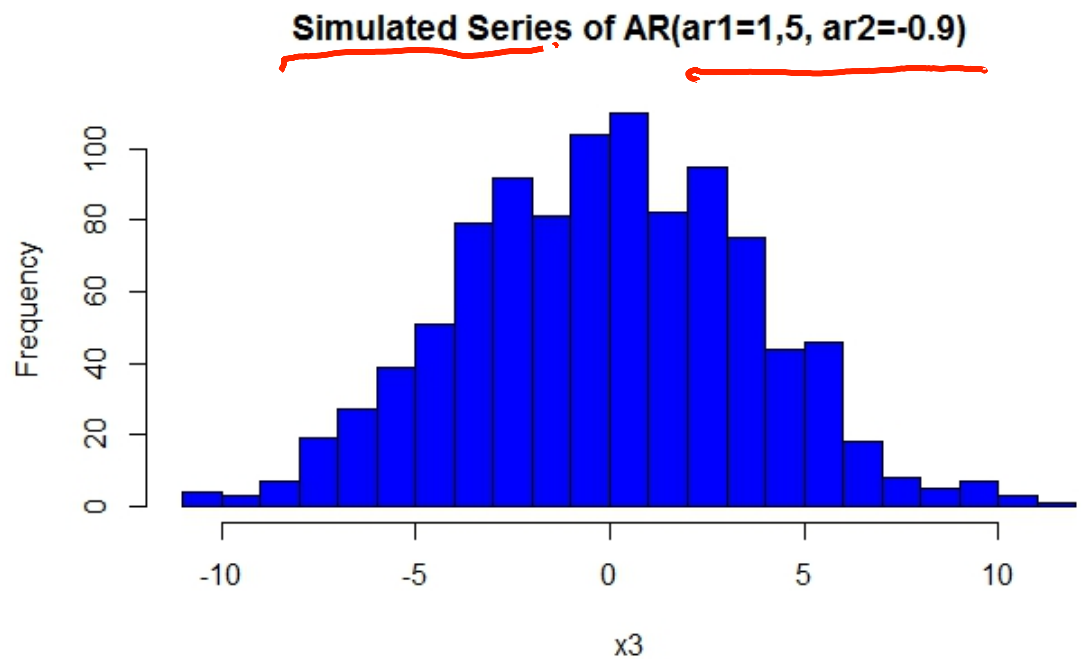
- Because the roots are complex, the autocorrelation function oscillates.

- Because the roots are close to one, the autocorrelation function oscillates slowly.



Use an AR(2) Model to Illustrate the Properties

- The histogram looks fairly symmetric.
- The summary statistics are displayed below the histogram.

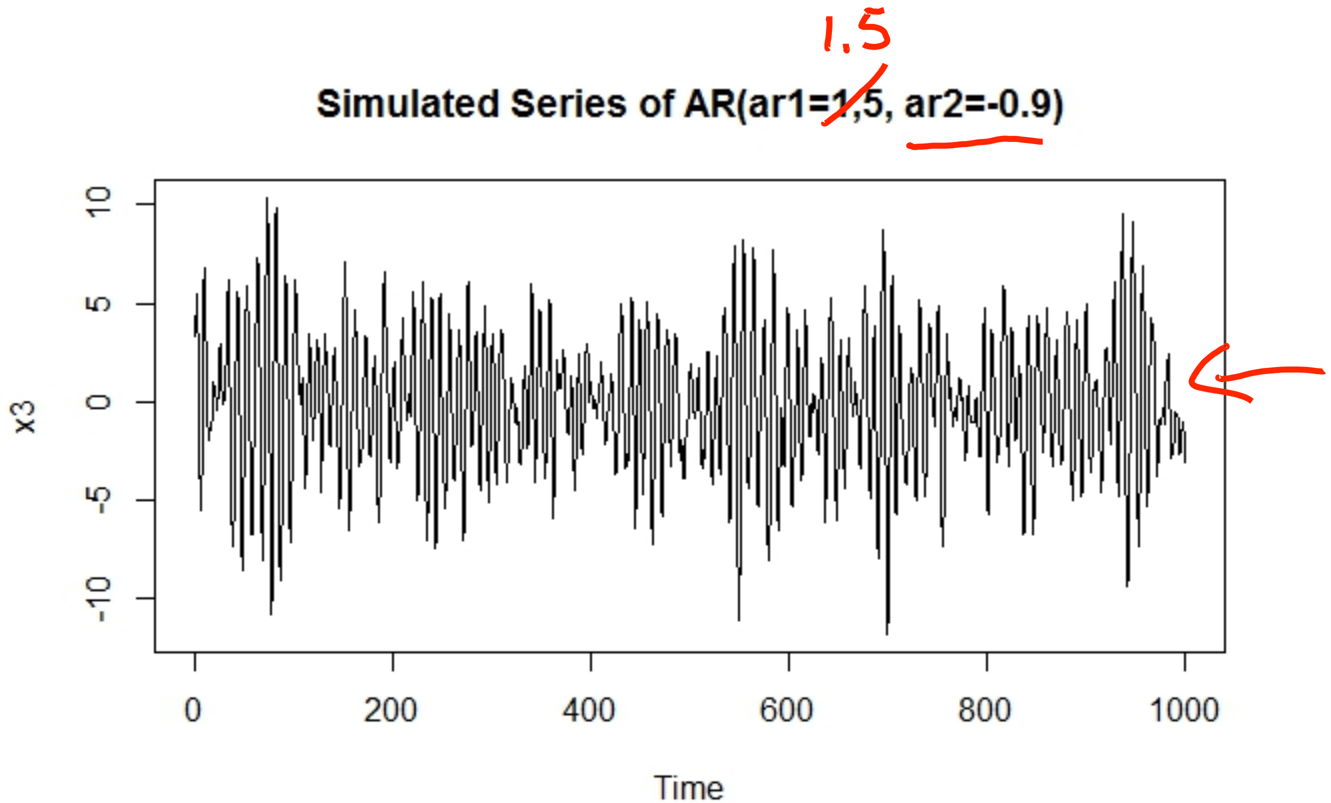



```
> str(x3)
 Time-Series [1:1000] from 1 to 1000: 3.34 5.44 5.03 2 -2.86 ...
> summary(x3)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-11.7900  -2.5550  -0.1784  -0.1619   2.3410  10.3400
```

Use an AR(2) Model to Illustrate the Properties

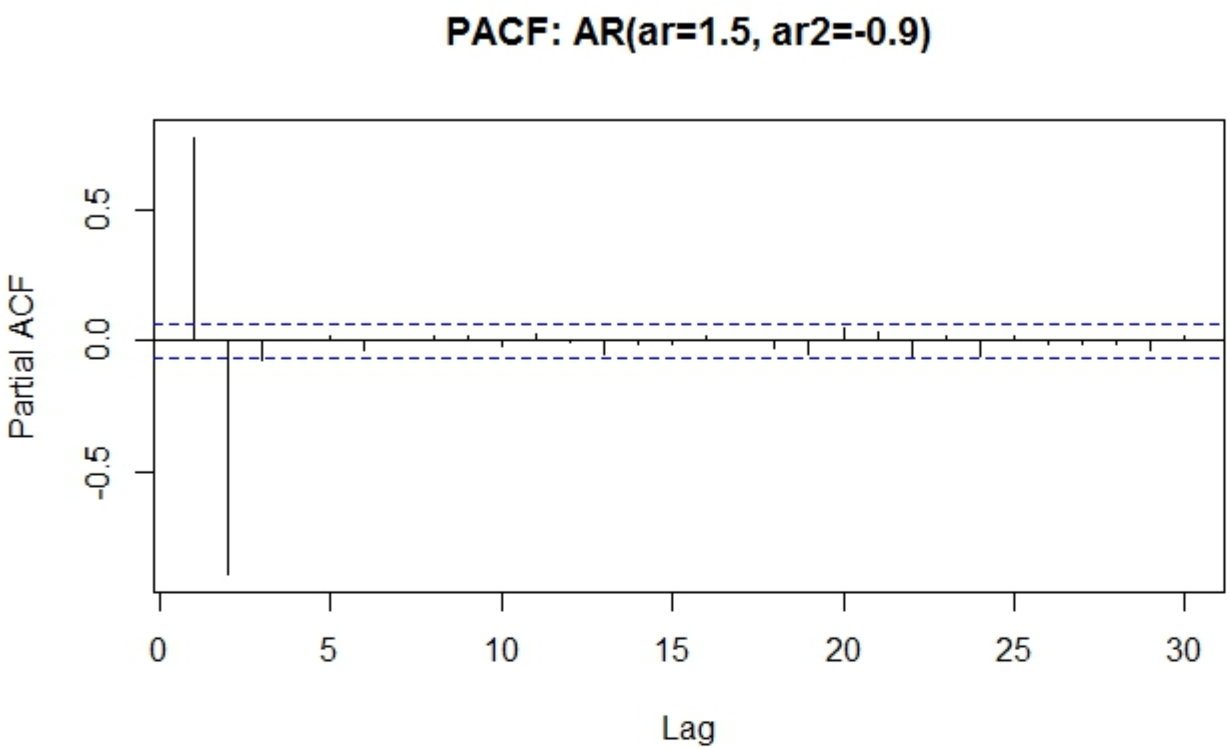
- The time series plot shows that the series display strong fluctuations

- The magnitude of these fluctuations change over time



Use an AR(2) Model to Illustrate the Properties

- As in the AR(1) model in which the PACF has a sharp cut-off at **displacement 1**, the PACF of the simulated AR(2) process has a sharp cut-off at **displacement 2**.



Autoregressive Models,

Model Estimation and Model Selection

Estimation: Example 1–AR(1)

Let's apply an AR model to the series we simulated using an AR(1) process of the following specification.

$$x_t - \mu = 0.7(x_{t-1} - \mu) + \omega_t$$

where μ is the mean of the series

Simulation conducted in R:

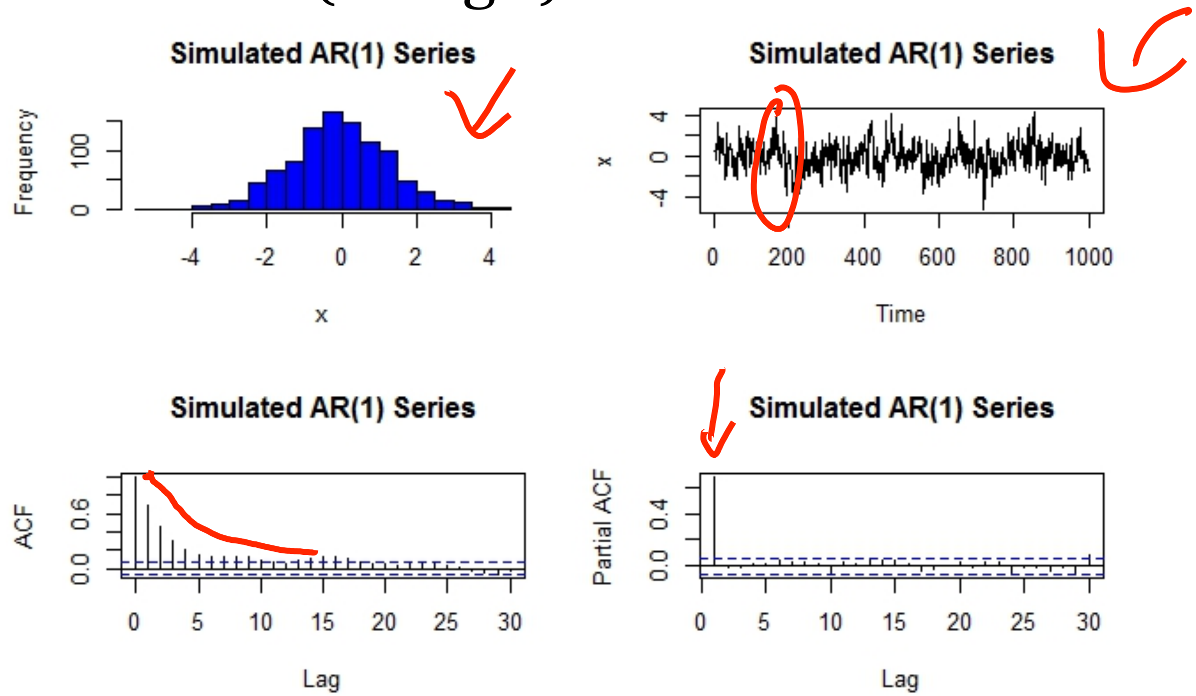
```
x <- w <- rnorm(1000)
x <- arima.sim(n = 1000, list(ar=c(0.7), ma=0))
str(x)
summary(x)
```

Always examine the series after the simulation:

```
> str(x)
 Time-Series [1:1000] from 1 to 1000: 0.5094 0.2424 1.1298 -0.4148 -0.0859 ...
> summary(x)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-5.30700 -0.92890 -0.11660 -0.07005  0.77120  4.32200
```

Estimation: Example 1–AR(1)

- Examine the simulated series by visualizing its distribution (using histogram), its dynamics (using time series plot), and its dependence structure (using ACF and PACF graphs).
- ACF gradually tapers off to zero; PACF falls off sharply after time displacement 1 (or lag 1).



Estimation: Example 1–AR(1)

- The estimation uses R's `ar()` function.
- This function estimates an autoregressive model of the following form:

$$x[t] - m = a[1]*(x[t-1] - m) + \ldots + a[p]*(x[t-p] - m) + e[t]$$

- Below is the specification of the function in **R.**

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

- By default it selects the estimated model with the lowest Akaike Information Criterion (AIC; Akaike, 1974).
- AIC is a goodness-of-fit measure that penalizes the number of parameters used in the model. We will discuss AIC next.

Estimation: Example 1–AR(1)

- The commands used to estimate the AR model are listed below. Note that maximum likelihood estimation (MLE) is chosen for the estimation.

```
x.ar <- ar(x, method = "mle")
x.ar$order # order of the AR model with lowest AIC
x.ar$ar # parameter estimate
sqrt(x.ar$asy.var) # asy. standard error
summary(x.ar)
x.ar$aic
```

1. The first command estimates a series of AR models and stores the estimation "object" in `x.ar`; recall that everything in R is an object.
2. The second command displays the order of the AR model that has the lowest AIC.
3. The third command displays the AR parameter estimates associated with this model.

Estimation: Example 1–AR(1)

```
x.ar <- ar(x, method = "mle")
x.ar$order # order of the AR model with lowest AIC
x.ar$ar # parameter estimate
sqrt(x.ar$asy.var) # asy. standard error
summary(x.ar)
x.ar$aic
```

4. The fourth command takes the square root of the asymptotic variance to arrive at the asymptotic standard error.
5. The fifth command lists different objects that come with the estimated AR object.
6. The last command displays the AIC.

Estimation: Example 1–AR(1)

The outputs of these commands follow:

```
> x.ar$order # order of the AR model with lowest AIC
[1] 1
> x.ar$ar # parameter estimate
[1] 0.6848115
> sqrt(x.ar$asy.var) # asy. standard error
           [,1]
[1,] 0.02304673
> summary(x.ar)
             Length Class  Mode
order           1   -none- numeric
ar              1   -none- numeric
var.pred        1   -none- numeric
x.mean          1   -none- numeric
aic            13   -none- numeric
n.used          1   -none- numeric
order.max       1   -none- numeric
partialacf      0   -none- NULL
resid        1000   ts     numeric
method          1   -none- character
series          1   -none- character
frequency       1   -none- numeric
call            3   -none- call
asy.var.coef    1   -none- numeric
> x.ar$aic
         0          1          2          3          4          5          6
630.074431   0.000000   1.945940   3.785605   4.626213   4.440357   5.231398
         7          8          9         10         11         12
  5.601880   7.464930   9.249846  11.194681  12.716578  14.715200
```

Estimation: Example 1–AR(1)

Note that the AICs associated with AR(0) to AR(12) models are all displayed. However, it is shown as the difference between the AIC of a model with the lowest AIC.

To get the AIC, we can calculate it "manually":

Akaike Information Criterion:

$$AIC = e^{\left(\frac{2k}{T}\right)} \frac{\sum_{t=1}^{T} e_t^2}{T}$$

- k is the number parameters
- n is the length of the sample used in the estimation
- MSE is mean squared error, which is a goodness-of-fit measure: the smaller the MSE, the better the fit.

Estimation: Example 1–AR(1)

Taking natural log transformation of the AIC formula above gives the following:

$$
\begin{aligned}
ln(AIC) &= ln\left(\frac{\sum_{t=1}^{T} e_t^2}{T}\right) + \left(\frac{2k}{T}\right) \\
&= ln(\text{MSE}) + \left(\frac{2k}{T}\right)
\end{aligned}
$$

Some authors/statisticians simply call this "AIC" (instead of ln(AIC). Some other authors use other variants of AIC and called them AIC. For example, our textbook defines AIC as follows:

```
AIC = -2xlog-likelihood + 2k
```

Estimation: Example 1–AR(1)

Since the `ar()` function reports only the difference between each of the AICs and the lowest AIC, we will have to obtain AIC manually using what is provided by the `ar()` object.

The `ar()` object provides the value of the estimated maximum likelihood associated with the AR models with the lowest AIC, so we can use the following formula:

```
AIC = -2xlog-likelihood + 2k
```

```
> -2*(-1446.94) + 2*(3)
[1] 2899.88
```

Estimation: Example 1–AR(1)

Recall the several commands that we used to obtain the estimates from the model:

1. Order of the "best" AR model
2. Parameter estimates associated with it
3. Estimated (asymptotic) standard error
4. Confidence interval

The results are:

```
> x.ar$order # order of the AR model with lowest AIC
[1] 1
> x.ar$ar # parameter estimate
[1] 0.6848115
> sqrt(x.ar$asy.var) # asy. standard error
           [,1]
[1,] 0.02304673
> x.ar$ar + c(-2,2)*sqrt(x.ar$asy.var)
[1] 0.6387180 0.7309049
```

.T

Estimation: Example 1–AR(1)

- Based on this particular realized sample path, the parameter estimates is `0.6748` and the **95% confidence interval** is `(0.6387, 0.7309)`, which includes the “true” parameter of the underlying data-generating process.

- In reality, however, the “true” parameter is unknown, no matter how big the sample is. In this example, we define the underlying data-generating process, so we know the “true” parameter. It just so happens that given this particular realized sample path, the estimated 95% confidence interval includes the true parameter.

Estimation: Example 1–AR(1)

- It is possible, however, that there are other sample paths whose associated confidence intervals do not include the true parameters.

- The interpretation of the 95% confidence interval is that given a large number of sample paths, 95% of those will include the true parameter.

- Try this as an exercise using the R scripts I've already provided. See the assignment for more detailed instructions.

Autoregressive Models

Model Diagnostics and Assumption Testing

Estimation

Model assumption diagnosis and testing:

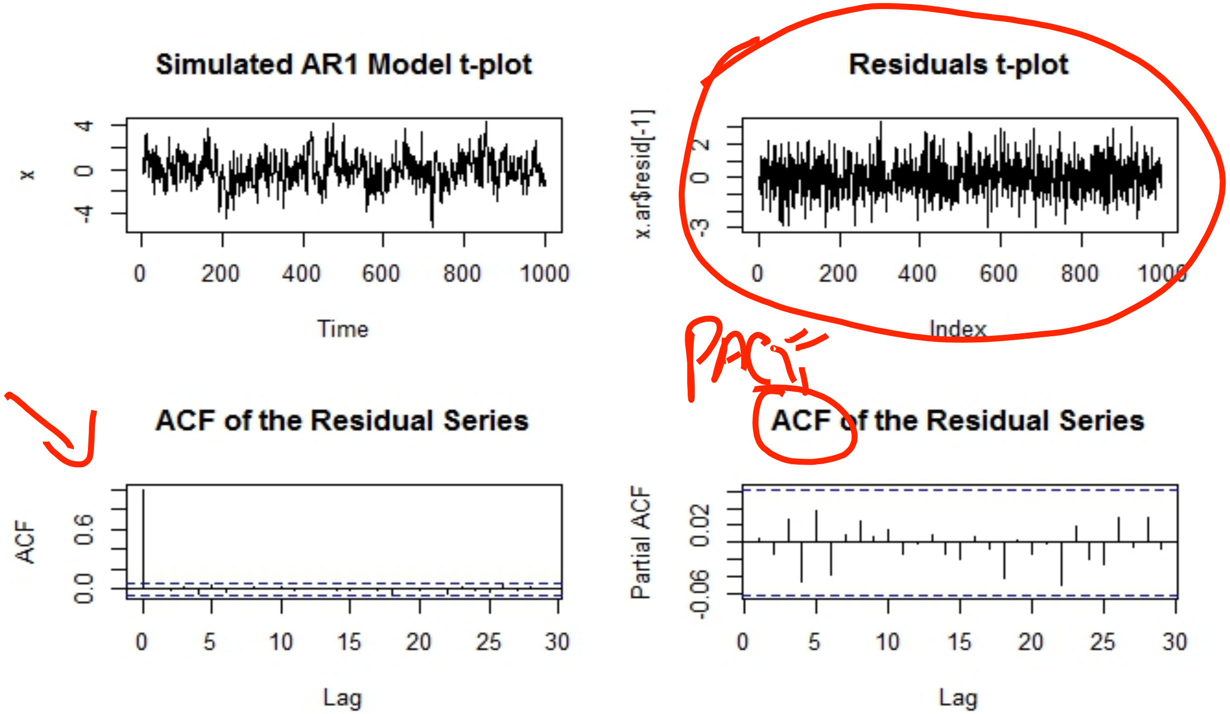
1. AR models have random components resembling that of white noise. **Question: Do the estimated residuals look like the realizations generated by a white noise process?**

2. We are interested in stationary AR models. **Question: Is our estimated AR model stationary (at least statistically)?**

Model Assumption Diagnosis and Testing:

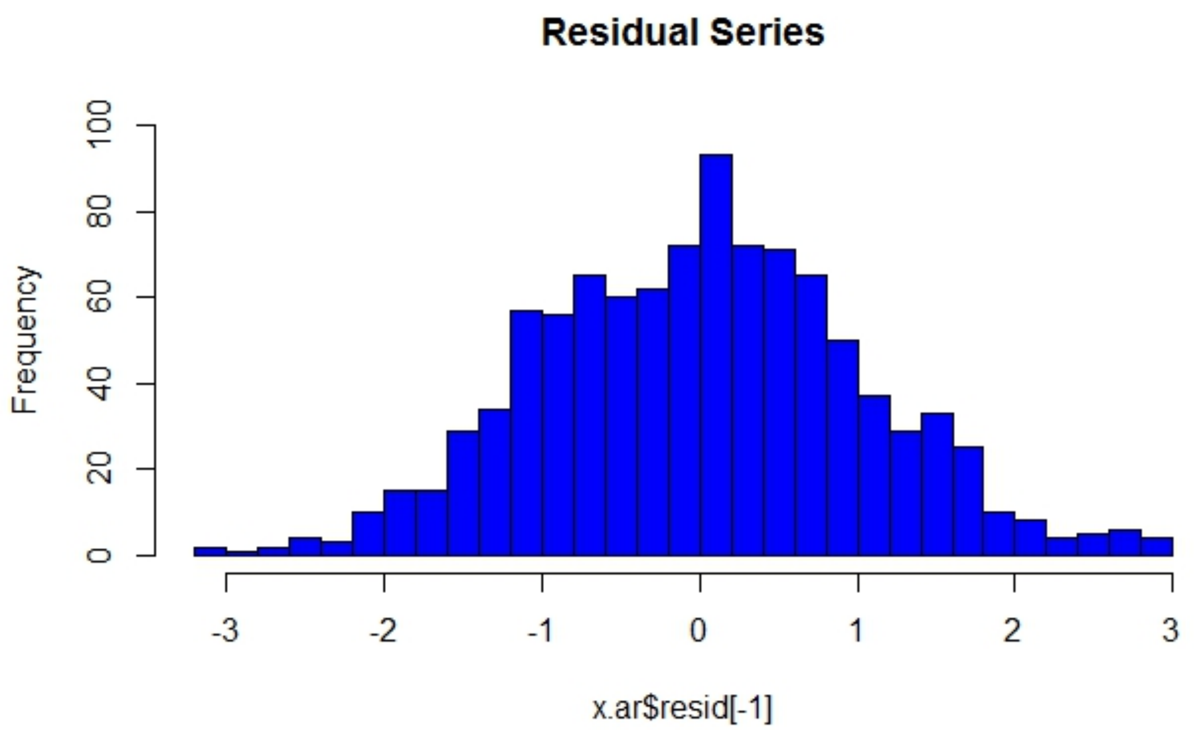
1. Does the residual series look like the realizations generated by a white noise process? The t-plot, correlogram, and PACF plots of the residuals are similar to those of a white noise process.

```
> head(x.ar$resid)
[1]          NA  0.04078616 -0.73661043 -0.26731725  0.24702927 -0.27750065
```



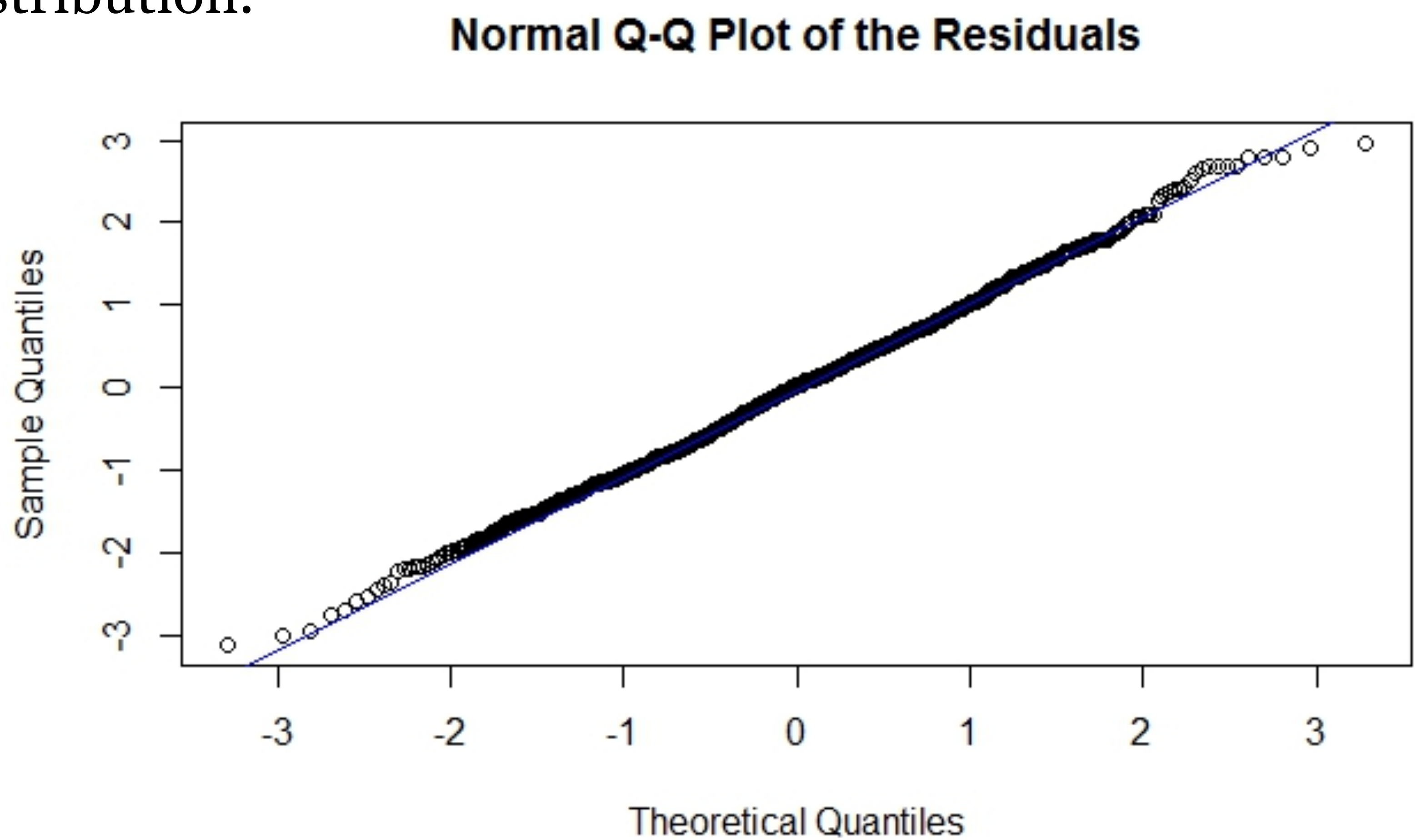
Model Assumption Diagnosis and Testing:

We can also examine the distribution of the residuals. The histogram shows a fairly symmetric distribution.



Model Assumption Diagnosis and Testing:

And, the qqplot against the theoretical normal also provides preliminary evidence that the residuals follow a normal distribution.



Autoregressive Models

Example 2: Estimation, Model Selection, Model Diagnostics, and Assumption Testing

Estimation and the True Data-Generating Process

Let's apply an AR model to the series we just simulated.

Recall that the "true" model (or true underlying data-generating process (DGP)) has the following functional form:

$$y_t = 1.5y_{t-1} - 0.9y_{t-2} + \epsilon_t$$

As we have already examine the series, we will proceed directly to estimation.

Pretend that we do not have knowledge about the underlying DGP, but based on the histogram, t-plot, ACF, and PACF, we will try using a autoregressive model.

Estimation: R Commands

We will first apply the `AR()` function, which selects the model with the lowest AIC.

Similar to example 1, the following commands are used to (1) estimate a series of AR model, (2) list the objects within the estimated AR objects, (3) determine the order of the AR model with the lowest AIC, (4) determine the estimated AR parameters, and (5) determine the difference between AICs and the best AIC.

```
x3.arfit <- ar(x3, method = "mle")
summary(x3.arfit)

x3.arfit$order # order of the AR model with lowest AIC
x3.arfit$ar # parameter estimate
x3.arfit$aic # AICs of the fit models
sqrt(x3.arfit$asy.var) # asy. standard error
x3.arfit$mean
```

Estimation Results

The results:

```
> x3.arfit$order # order of the AR model with lowest AIC
[1] 5
> x3.arfit$ar # parameter estimate
[1]  1.4787350 -0.9085119 -0.0658531  0.1367330 -0.1021317
> x3.arfit$aic # AICs of the fit models
          0           1           2           3           4           5
2516.786599 1579.744987    4.590775    6.550943    8.410185    0.000000
          6           7           8           9          10          11
   1.778044    2.165644    2.169896    4.129782    5.383153    7.338368
         12
   9.025748
> sqrt(x3.arfit$asy.var) # asy. standard error
            [,1]       [,2]       [,3]       [,4]        [,5]
[1,] 0.031273080        NaN 0.02990534        NaN 0.004045091
[2,]         NaN 0.05587106        NaN 0.03040702         NaN
[3,] 0.029905338        NaN 0.06275694        NaN 0.029905338
[4,]         NaN 0.03040702        NaN 0.05587106         NaN
[5,] 0.004045091        NaN 0.02990534        NaN 0.031273080
Warning message:
In sqrt(x3.arfit$asy.var) : NaNs produced
```

Estimation Results Explained

Several points worth noticing:

1. A series of 12 AR models were estimated. This is the default set in the `AR()` function. We can change the maximum order using the option `order.max`. See the R documentation for more details.

2. The “best” model (in terms of AIC) is an AR(5) while the underlying DGP is an AR(2) process, although the first two parameters (1.48 and -0.91 with standard errors being 0.0312 and 0.0559) are very similar to those of the true DGP.

3. The AICs of the AR(0) and AR(1) models are huge, but those of AR(2), AR(5)–AR(10) are within the same ballpark.

4. Some of the off-diagonal elements in the (asymptotic) variance-covariance matrix cannot be computed.

Model Diagnostics

Examine the residuals:

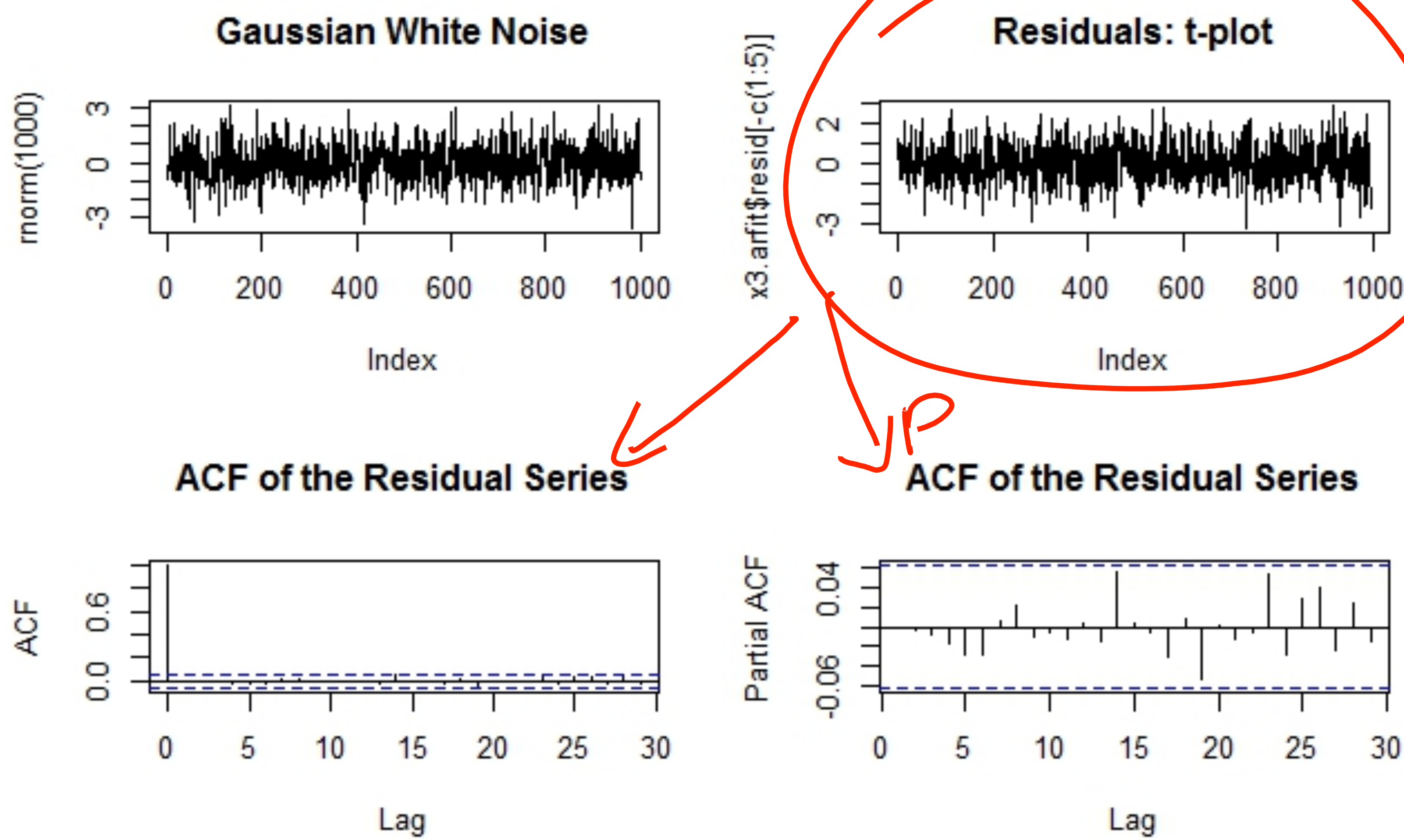
Note that the first five observations in the estimated residual series are missing. It is because the first five observations from the sample path are excluded in the estimation of an AR(5) model.

For this reason, to plot ACF and PACF, these missing observations from the series need to be excluded.

```
> head(x3.arfit$resid, 15)
 [1]          NA          NA          NA          NA          NA  1.19943972
 [7] -0.74451763  0.04727915  0.79347539  0.94224939  0.55571479 -0.81977882
[13]  0.34283689  0.26066242  0.86765075
> head(x3.arfit$resid[-c(1:5)], 15)
 [1]  1.19943972 -0.74451763  0.04727915  0.79347539  0.94224939  0.55571479
 [7] -0.81977882  0.34283689  0.26066242  0.86765075  0.02029650 -0.43598030
[13] -0.19129438  1.37263278  0.06466725
```

Model Diagnostics

The residuals look like a white noise series.



Model Selection

These results call into the question of which model (AR(5) or AR(2)) should we use.

- The AR(5) model gives a lower AIC.
- Or, the AR(2) is the "true" model.
  - But, of course, in reality we do not know what the true model really is.

Before we make other considerations, remember:

1. AIC is but one model performance measure.
2. In this example, we have only used it for in-sample measure (i.e., goodness-of-fit).

Model Selection (2)

Other considerations:

1. Are the differences among the AICs from the several AR models really that big?

2. Even if the focus is "in-sample" fit, we should consider other model performance measure such as Bayesian information criteria, which penalize more on the number parameters used in the model: $\boxed{\texttt{-2ln(likelihood) + ln(N)k}}$
   - It is possible that AIC and BIC give different results. We will discuss AIC and BIC more in the next lecture.

3. What about out-of-sample test?
   - We have not talked about out-of-sample tests.

4. Among all of these considerations, the most important question we need to ask is what is the objective of building this model? What question are we trying to answer?

Model Selection (3)

Suppose the objective is for forecasting. Then, the performance measure should be focusing on using forecast errors to compare among models.

- We will have to ask how long the forecast horizon is.
- Is the forecast short-term or long-term?
- How long is the sample we want to use to estimate the model? Do we always want to use the entire sample, as we did in the previous two simulated examples?
- Should we leave a subset of the observed samples for out-of-sample test?
  - In other courses, you may have come across this concept as dividing the sample into a **training set** and a **testing (or validation) set**. In time series analysis, the sample is often very limited. How much sample to leave out requires a thorough discussion between data scientists/modelers and the business stakeholders.

Autoregressive Models

Model Estimation, Model Identification, Model Selection
Example 3

Example 3: Global Surface Temperature

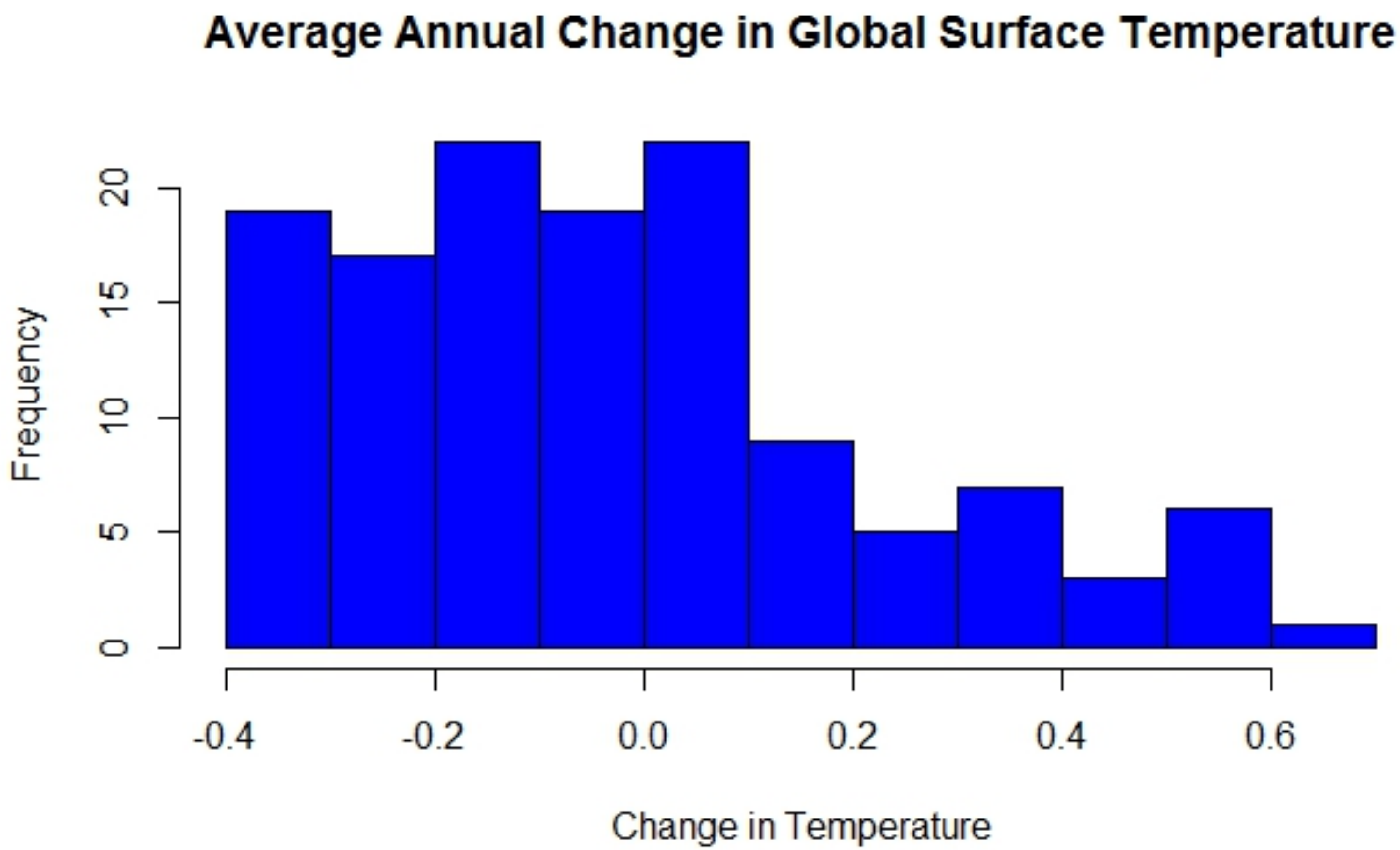
- Global surface temperature measured in annual average.
- Annual series from 1880–2009.
- Let's try to fit an AR model to this series.

- After loading and cleaning the dataset, examine the series:
  - The series has 130 (annual) observations.
  - It measures the annual change.

```
> str(gtemp)
 Time-Series [1:130] from 1880 to 2009: -0.28 -0.21 -0.26 -0.27 -0.32 -0.32 -0
.29 -0.36 -0.27 -0.17 ...
> head(gtemp)
[1] -0.28 -0.21 -0.26 -0.27 -0.32 -0.32
```

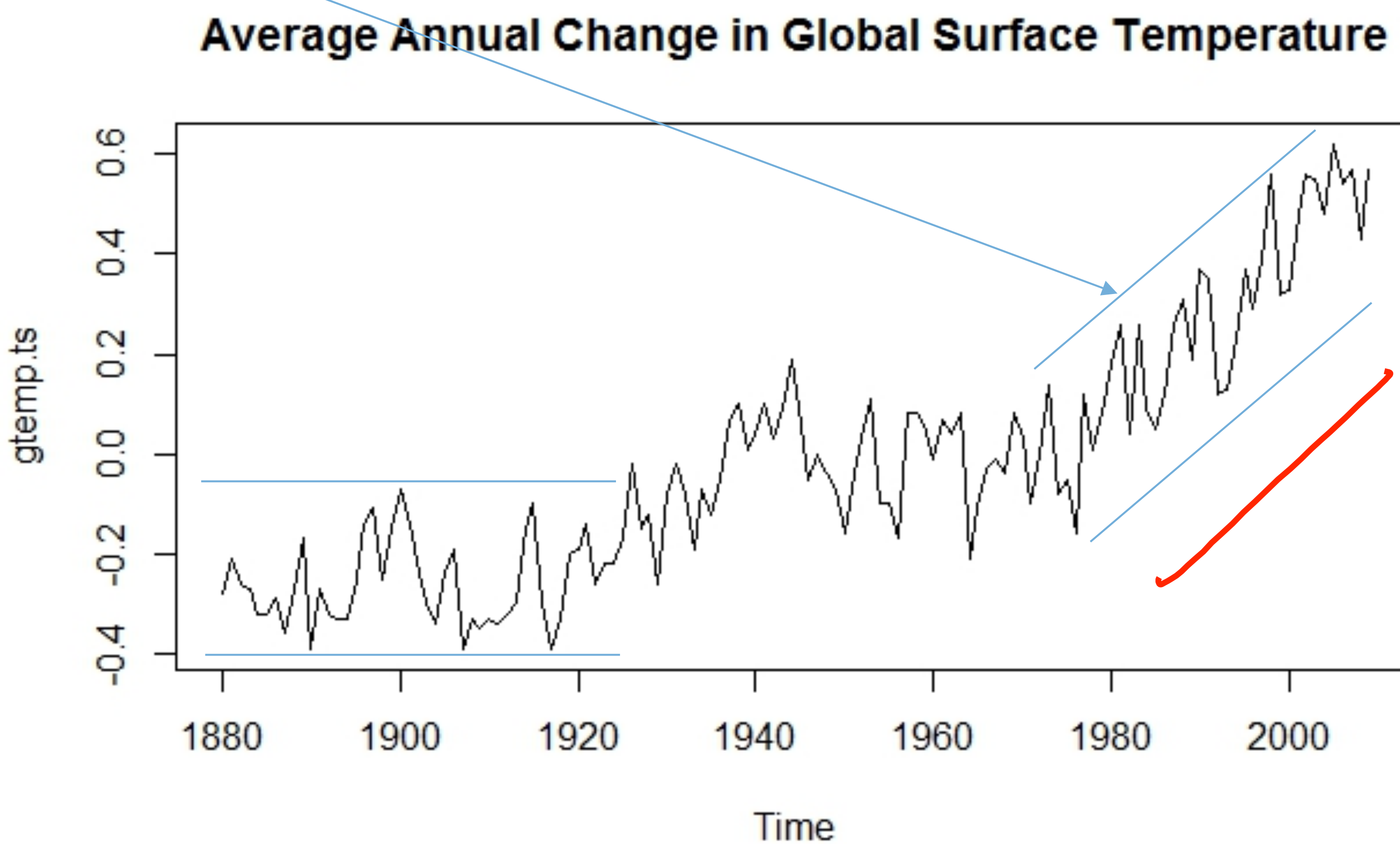
Example 3: Global Surface Temperature

The histogram shows that the density is skewed.



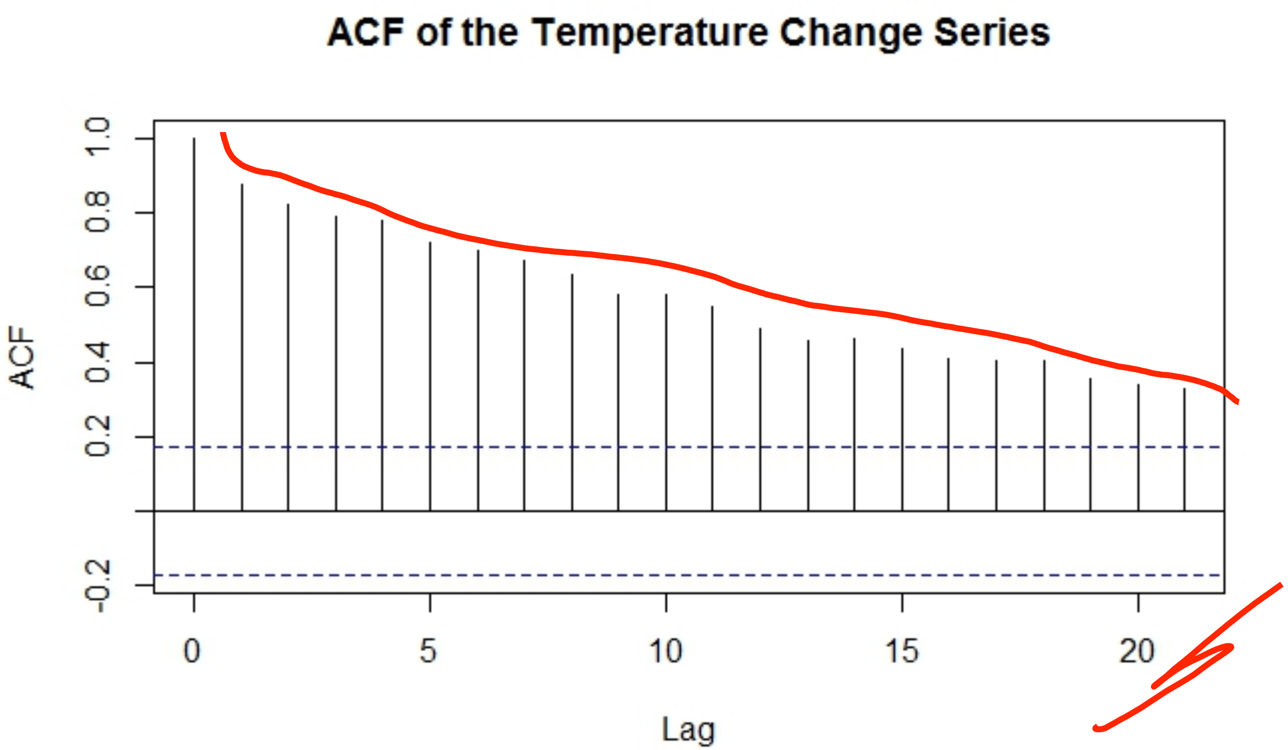
Example 3: Global Surface Temperature

An upward trend is apparent in the time series plot.



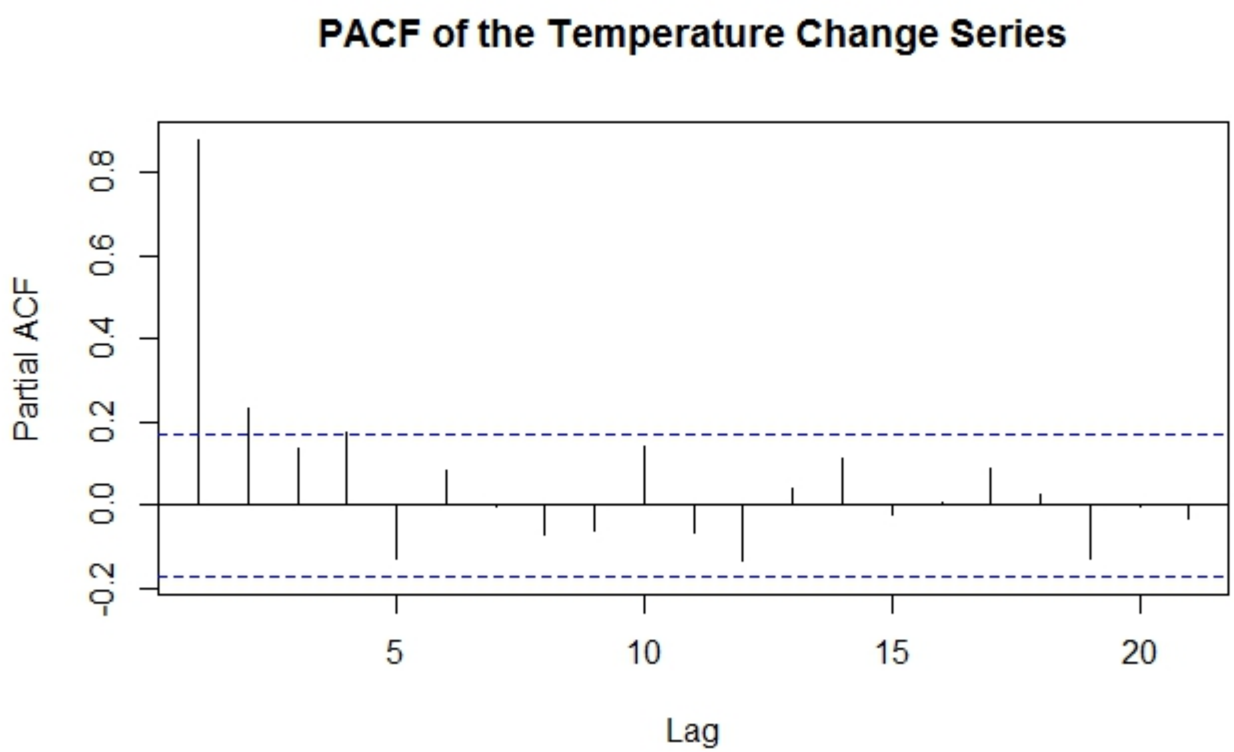
Example 3: Global Surface Temperature

The ACF of the series confirms the strong persistence due to the trend. The autocorrelations decline to zero very slowly.



Example 3: Global Surface Temperature

The PACF shows only a low but slightly statistically significant correlation at lag 1.



Although there is a trend that needs to be account for before fitting an AR model, we have not studied the techniques to handle trends yet.

Let's continue with the example and fit an AR model using the `AR()` function

```
> gtemp.ar <- ar(gtemp.ts, method="mle")
Warning messages:
1: In arima0(x, order = c(i, 0L, 0L), include.mean = demean) :
  possible convergence problem: optim gave code = 1
2: In arima0(x, order = c(i, 0L, 0L), include.mean = demean) :
  possible convergence problem: optim gave code = 1
3: In arima0(x, order = c(i, 0L, 0L), include.mean = demean) :
  possible convergence problem: optim gave code = 1
```

So, the model fails to converge and the estimation cannot be completed!

This is a good news: Even if we "forget" to remove (or ignore) the trend before fitting an AR model, R would generate an error.

Moving Average Models

Mathematical Formulation and Derivation of Properties

Moving Average Models: An Introduction

A moving average model of order q is a linear combination of the current and past q white noises:

$$x_t = \omega_t + \theta_1 \omega_1 + \cdots + \theta_{t-q} \omega_{t-q} \quad (3.0.2)$$

t-1

where $\{\omega_t\}$ is a white noise sequence, each of which has zero mean and variance σ_ω^2. Also, consider x_t as a demeaned series. That is, $x_t - \mu$.

Expressed in backshift operators,

$$\tilde{x}_t = \left(1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q\right) = \boxed{\theta_q (B) \omega_t} \quad (3.0.3)$$

$\theta_q()$ is a polynomial of order q.

- Since the MA(q) process is a finite linear combination of white noises, the moving average processes are **stationary with constant mean, variance, and autocovariance**.

- In other words, the stationarity condition for a moving average process is met regardless of the values of its parameters.

Moving Average Models: An Introduction (2)

- A moving average process is a function of both current and past shocks. These shocks are theoretical, and more importantly, they are unobservable!

- So, we really cannot use moving average models for forecasting, as forecasting generally requires an established statistical relationship between current and past values.

- The question becomes, **"Can we express a moving average model as a function of current and past observable values?"**

- This leads to the concept of **invertibility**.

The Invertibility Condition

- In particular, if a moving average process is invertible, then it can be "inverted" and expressed as a function of current shock and lagged values of the series. We call this form of expression of MA models as an **autoregressive representation**.

In general, an MA(q) process is *invertible* when the roots of $\theta_q(B)$ all exceed 1 in absolute value.

- Invertibility condition is needed because of its practical importance.

- If a MA model is invertible, it can be expressed as autoregressive representation.

The Invertibility Condition (2)

- With the autoregressive representation, one can use it for real-world forecasting because forecasting requires the linkage between the present observations to the past observations.

- With this linkage (or a model), we can extrapolate to form a forecast based on present and past observations.

- For this reason, we will restrict our focus on invertible processes.

- Let's look at the mathematical formulation and the key properties of MA models.

Moving Average Models

Mathematical Formulation of the Model and Derivation of Properties

The Invertibility Condition

To fix the idea, consider an MA(1) model:

$$y_t = \epsilon_t + \theta\epsilon_{t-1}$$
$$\epsilon_t \sim WN(0, \sigma_w^2)$$

Solve for the shocks (or innovations):

$$\epsilon_t = y_t - \theta\epsilon_{t-1}$$

Using recursive substitution,

$$\epsilon_{t-1} = y_{t-1} - \theta\epsilon_{t-2}$$
$$\epsilon_{t-2} = y_{t-2} - \theta\epsilon_{t-3}$$
$$\epsilon_{t-3} = y_{t-3} - \theta\epsilon_{t-4}$$
$$\vdots$$

The Invertibility Condition

We will obtain

$$y_t = \epsilon_t + \theta y_{t-1} - \theta^2 y_{t-2} + \theta^3 y_{t-3} - \cdots$$

Rearrange and use the backward shift operator:

$$y_t - \theta y_{t-1} - \theta^2 y_{t-2} - \theta^3 y_{t-3} - \cdots = \epsilon_t$$
$$(1 - \theta B - \theta^2 B^2 - \theta^3 B^3 - \cdots) y_t = \epsilon_t$$

Therefore, the infinite autoregressive representation of a MA(1) model can be expressed as

$$\frac{1}{1 + \theta B} = \epsilon_t$$

Expressed in backward shift operator, the infinite sequence of $\{\theta^i\}_{i=1}^{\infty}$ converges if $|\theta| < 1$, which is the invertible condition for the MA(1) model.

Mean and Variance of

- The mean of the MA(q) model has a constant mean equal to

$$E(x_t) = \sum_{j=0}^{q} \theta_j E\left(w_{t-j}\right) = 0 \tag{3.0.4}$$

where $\theta_0 = 1$,

It also has a constant variance with the following form:

$$Var(x_t) = \sum_{j=0}^{q} \theta_j Var\left(w_{t-j}\right) = \left(1 + \beta_1 + \cdots + \beta_q\right) \sigma_\omega^2 \tag{3.0.5}$$

Covariance

Using the definition of covariance

$$\gamma(h) = cov(x_{t+h}, x_t) = cov\left(\sum_{j=0}^{q} \theta_j w_{t+h-j}, \sum_{k=0}^{q} \theta_k w_{t-k}\right) \tag{3.0.6}$$

and the variance derived above, we can come up with the autocorrelation function, for $k \geq 0$:

$$\rho(k) = \begin{cases} 1 & \text{for } k = 0 \\ \frac{\sum_{i=0}^{q-k} \beta_i \beta_{i+k}}{\sum_{i=0}^{q} \beta_i^2} & \text{for } k = 1, \ldots, q \\ 0 & \text{for } k > q \end{cases} \tag{3.0.7}$$

where $\beta_0 = 1$.

A MA(2) Model

To get a feel of the above formulas, consider a simple **MA(q)** model: MA(2)

This process has mean μ and variance $\sigma_\omega^2\,(1+\beta_1^2+\beta_2^2)$, and the autocorrelation

$$\rho(k) = \begin{cases} 1 & \text{for } k = 0 \\ \frac{\beta_1(1+\beta_2)}{1+\beta_1^2+\beta_2^2} & \text{for } k = 1 \\ \frac{\beta_2}{1+\beta_1^2+\beta_2^2} & \text{for } k = 2 \\ 0 & \text{for } k > 2 \end{cases} \tag{3.0.9}$$

Moving Average Models

Key Properties Recap

The "Memory" of Moving Average Process

Let's recap the functional form of the variance and autocorrelation function:

The Autocovariance function of an MA(q) process $(k = 0, 1, \ldots, q)$ is

$$\gamma_k = E\left[(\omega_t + \theta_1\omega_{t-1} + \cdots + \theta_q\omega_{t-q})(\omega_{t-k} + \theta_1\omega_{t-k-1} + \cdots + \theta_q\omega_{t-k-q})\right]$$
$$= \theta_k E[\omega_{t-k}^2] + \theta_1\theta_{k+1}E[\omega_{t-k-1}^2] + \cdots + \theta_{q-k}\theta_q E[\omega_{t-q}^2]$$

since the ω_t are uncorrelated, and $\gamma_l = 0$ for $k > q$.

Therefore, the variance of the process is

$$\gamma_0 = \left(1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2\right)\sigma_\omega^2$$

The autocovariance of the process is

$$\gamma_k = \begin{cases} (\theta_k + \theta_1\theta_{k+1} + \theta_2\theta_{k+2} + \cdots + \theta_{q-k}\theta_q)\,\sigma_\omega^2 & k = 1, 2, \ldots, q \\ 0 & k > q \end{cases}$$

The autocorrelation function is

$$\rho_k = \begin{cases} \frac{\theta_k + \theta_1\theta_{k+1} + \theta_2\theta_{k+2} + \cdots + \theta_{q-k}\theta_q}{1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2} & k = 1, 2, \ldots, q \\ 0 & k > q \end{cases}$$

The "Memory" of Moving Average Process

- The autocorrelation function shows that a MA(q) has a "memory" of only **q** periods. If q=1, then the model has memory of only one period.

- It means that the current value is not affected by the values older than q periods.

- This has an important implications on model identification (if the underlying data generating process is indeed a MA(q) process):

> The ACF of a MA(q) model drops off completely after **q** periods.

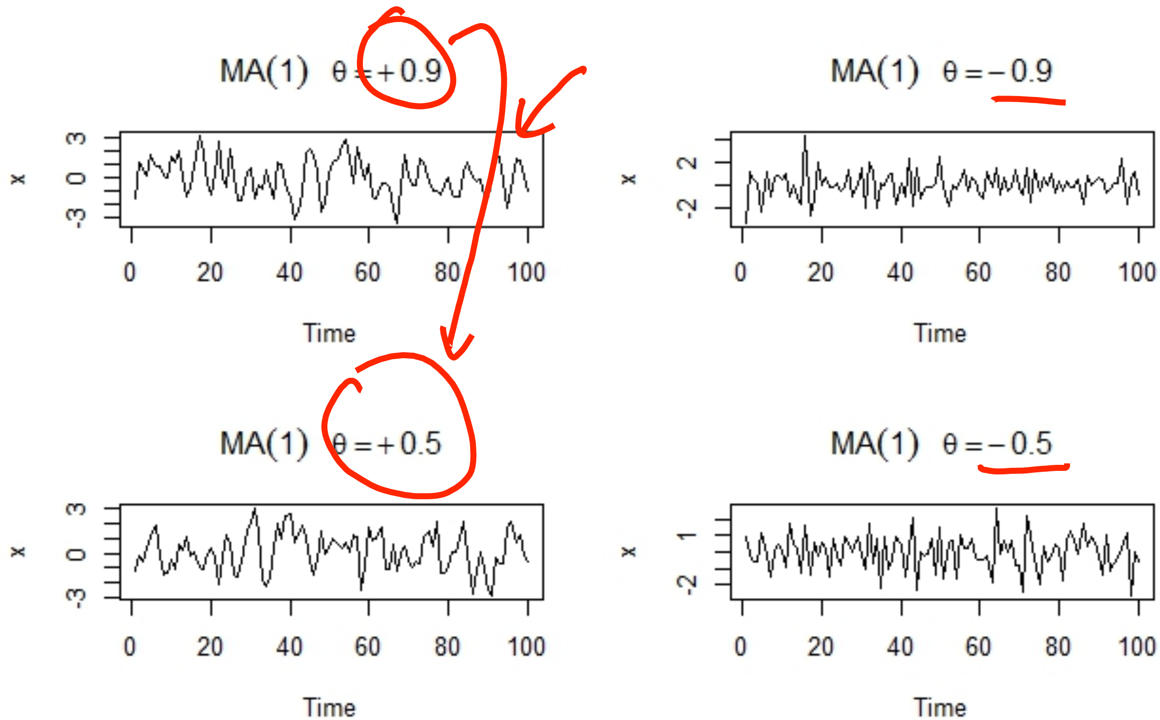
- This means that the ACF provides a considerable amount of information about the order of dependence if the process is a moving average process.

Moving Average Models,

Simulation of MA(q) Models

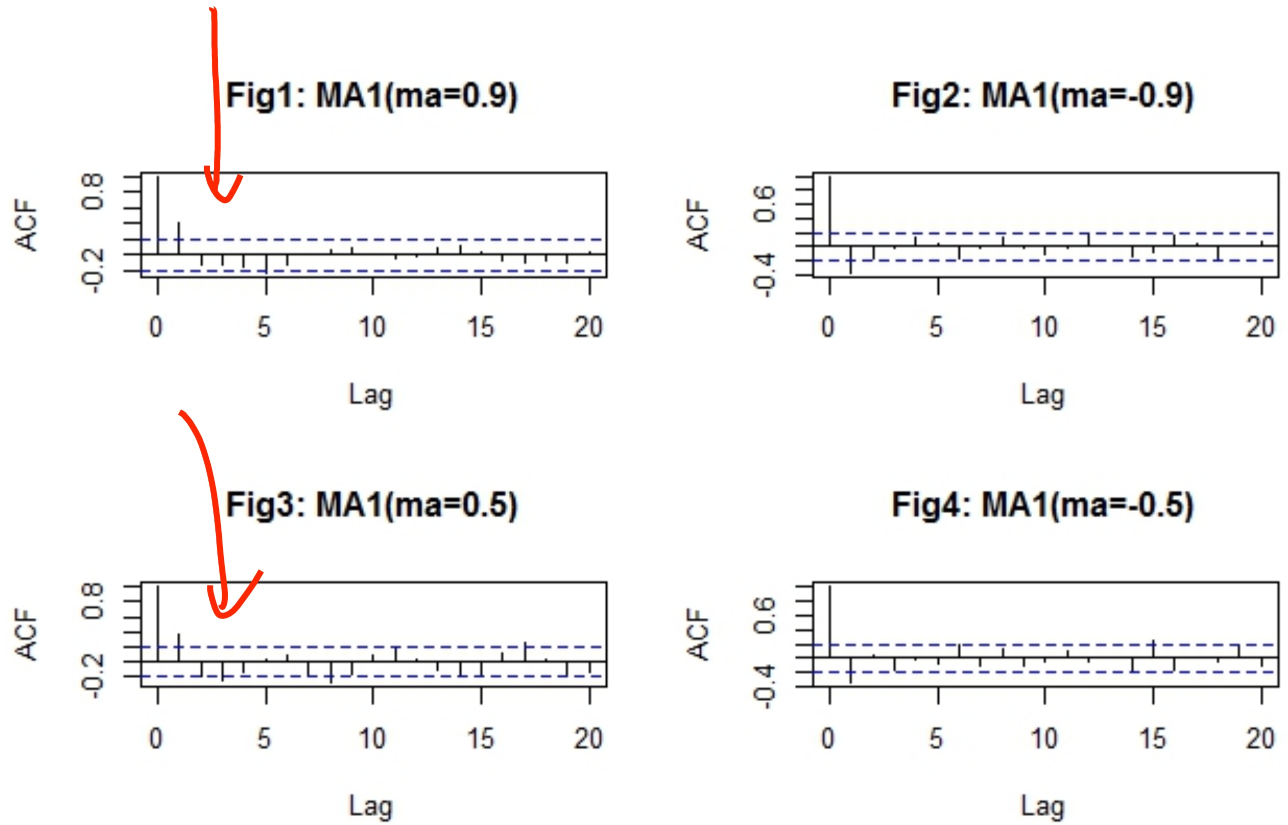
MA(1) Models: Time Series Plots

- MA models in general have “limited” memory, which can be seen from the functional form of its autocorrelation.
- For MA(1) models, its memory only lasts for one period.
- As a result, the difference in dynamic is not as big as those in AR(1) models when the AR parameter values are very different.
- The different dynamic of the MA models with positive and negative coefficients come from the positive and negative autocorrelation.



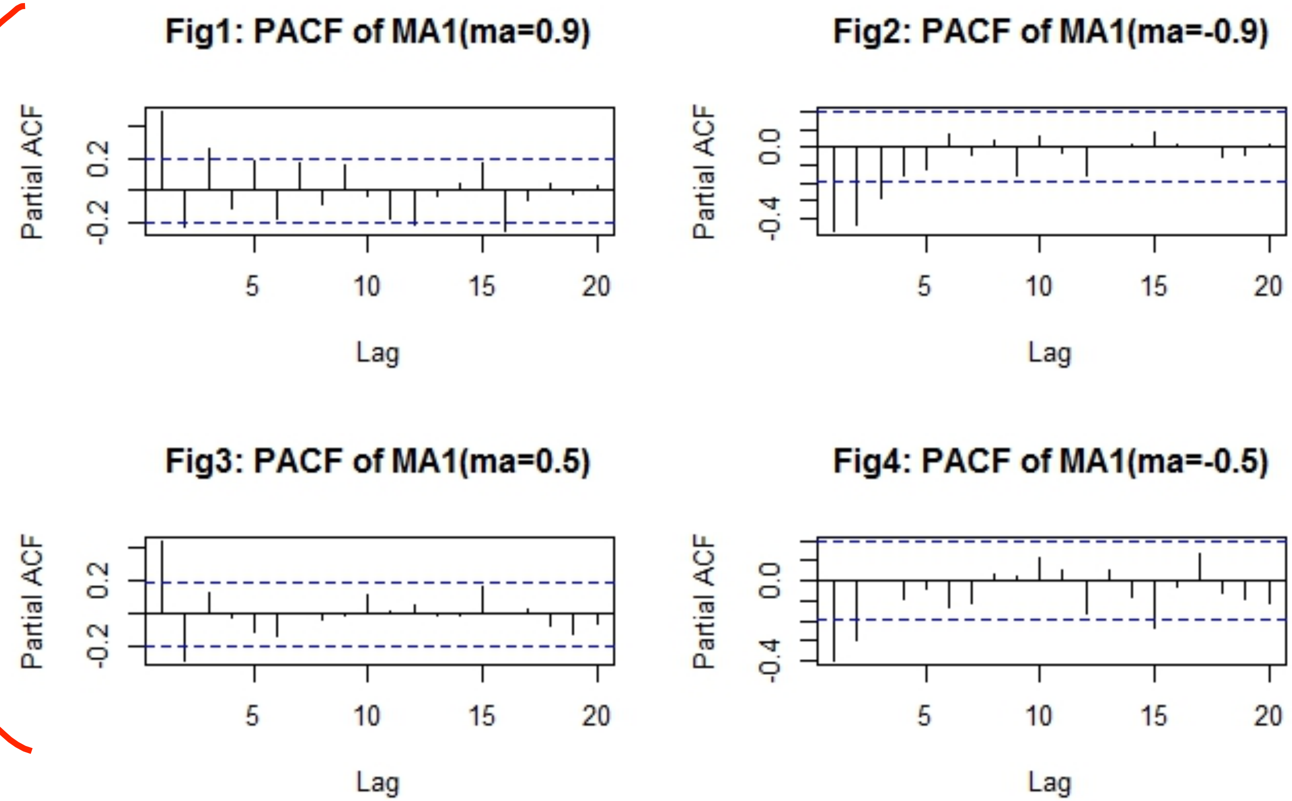
MA(1) Models: ACF Plots

1. The ACF sharply drops off after the first lag.
2. This pattern holds regardless of the value of MA parameter.
3. For MA(1) models, positive MA parameter value leads to a positive correlation and negative parameter value leads to a negative correlation.



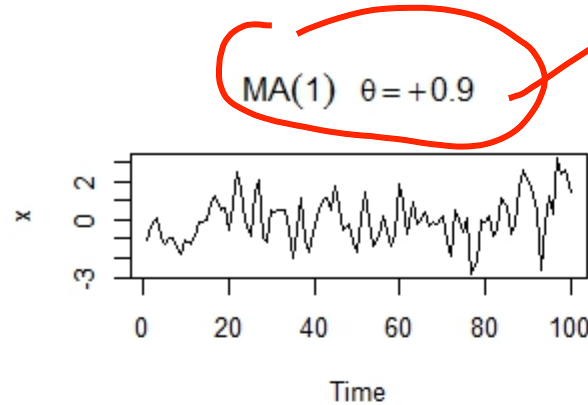
MA(1) Models: PACF Plots

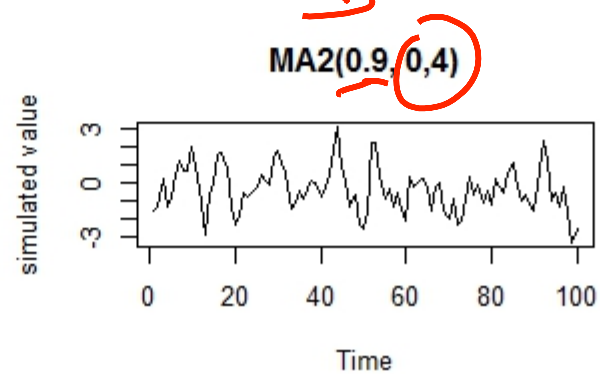
1. Unlike its ACF, the PACF of MA models decay gradually to zero.
2. This can be seen from the infinite autoregressive representation of the MA(1) process.
3. For MA(1) model, if the MA parameter is positive, the decay oscillates to zero. If the parameter is negative, the decay is "mostly" one-sided.

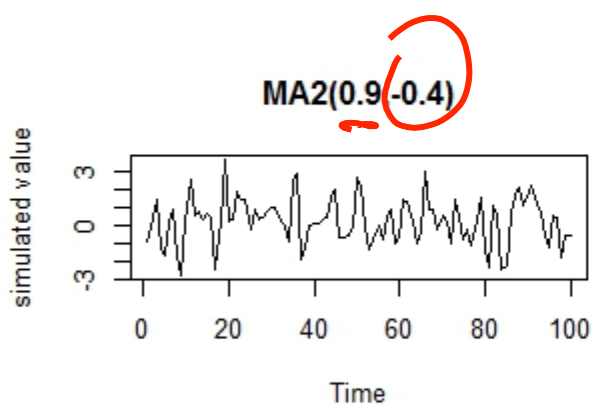


MA(2) Models: Time Series Plots 1

- MA(2) models can allow for richer dynamics than those of MA(1) models, which can be exploited to improve forecast.
- Based solely on the time series plots, it is, however, not easy to distinguish between a MA(1) and MA(2) models if they have the same value on the first MA parameter.

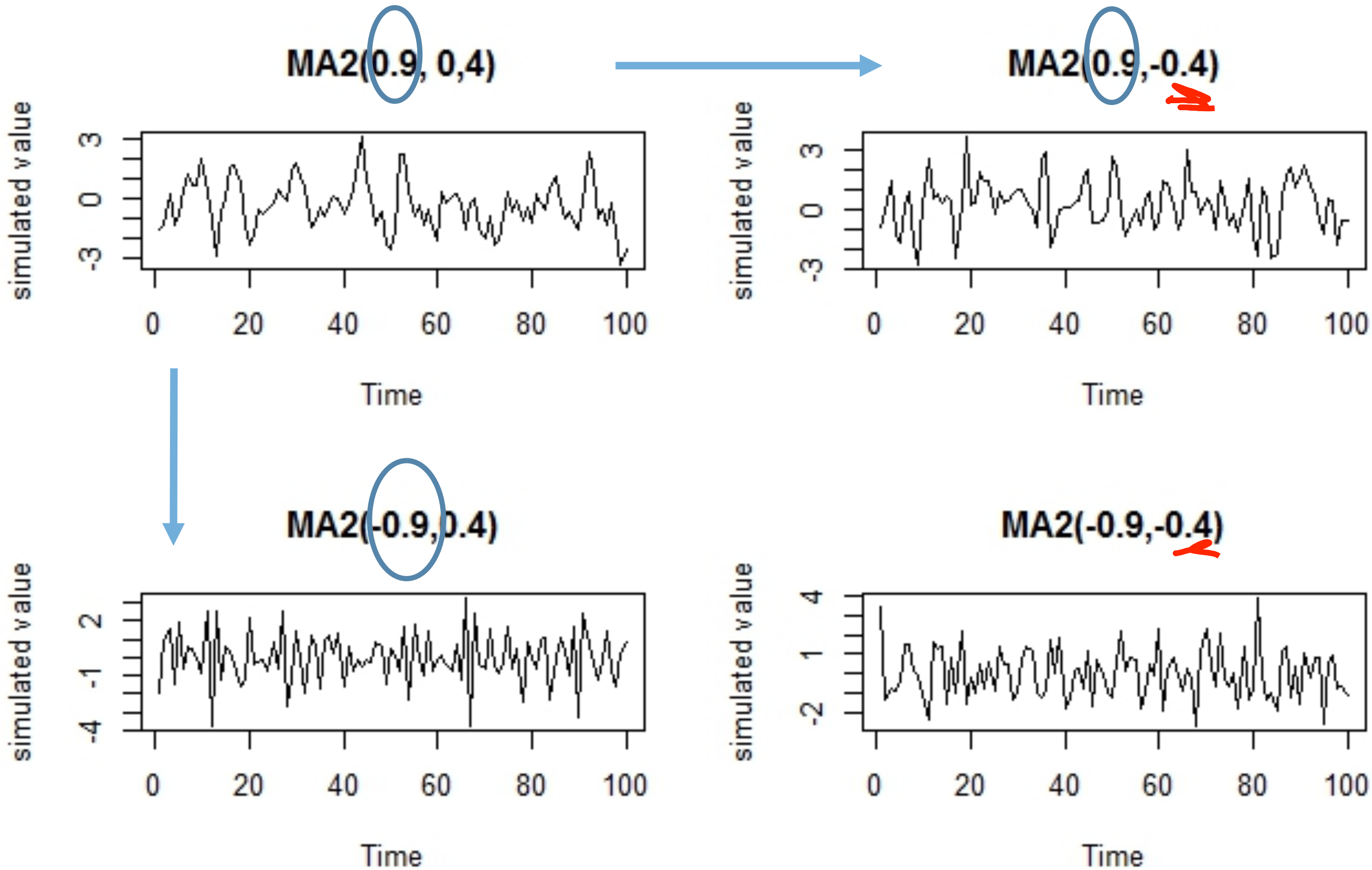






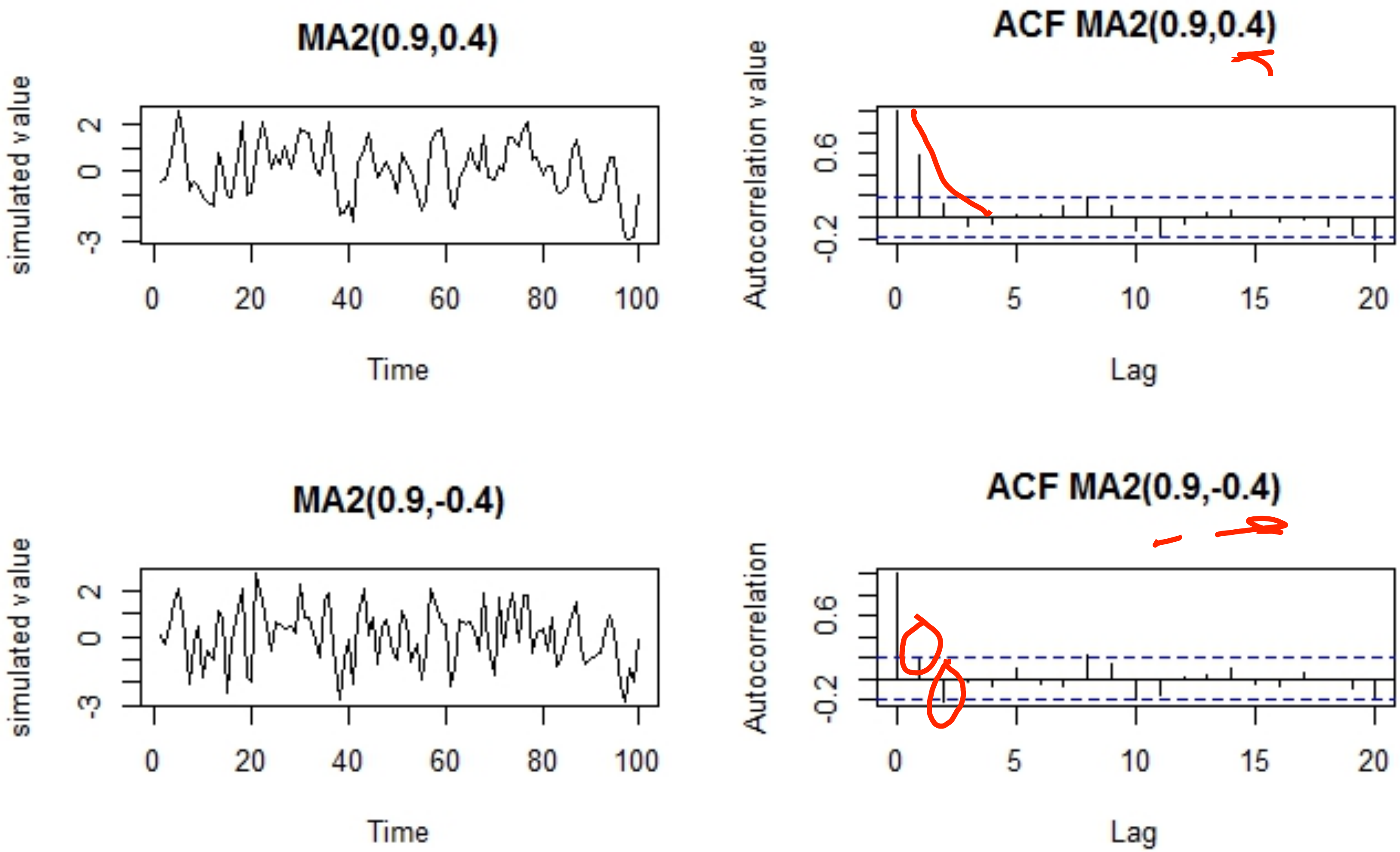
MA(2) Models: Time Series Plots 2

1. Negative MA parameters produce more volatile series.
2. The first MA parameter has a larger impact on the volatility than the second parameter.



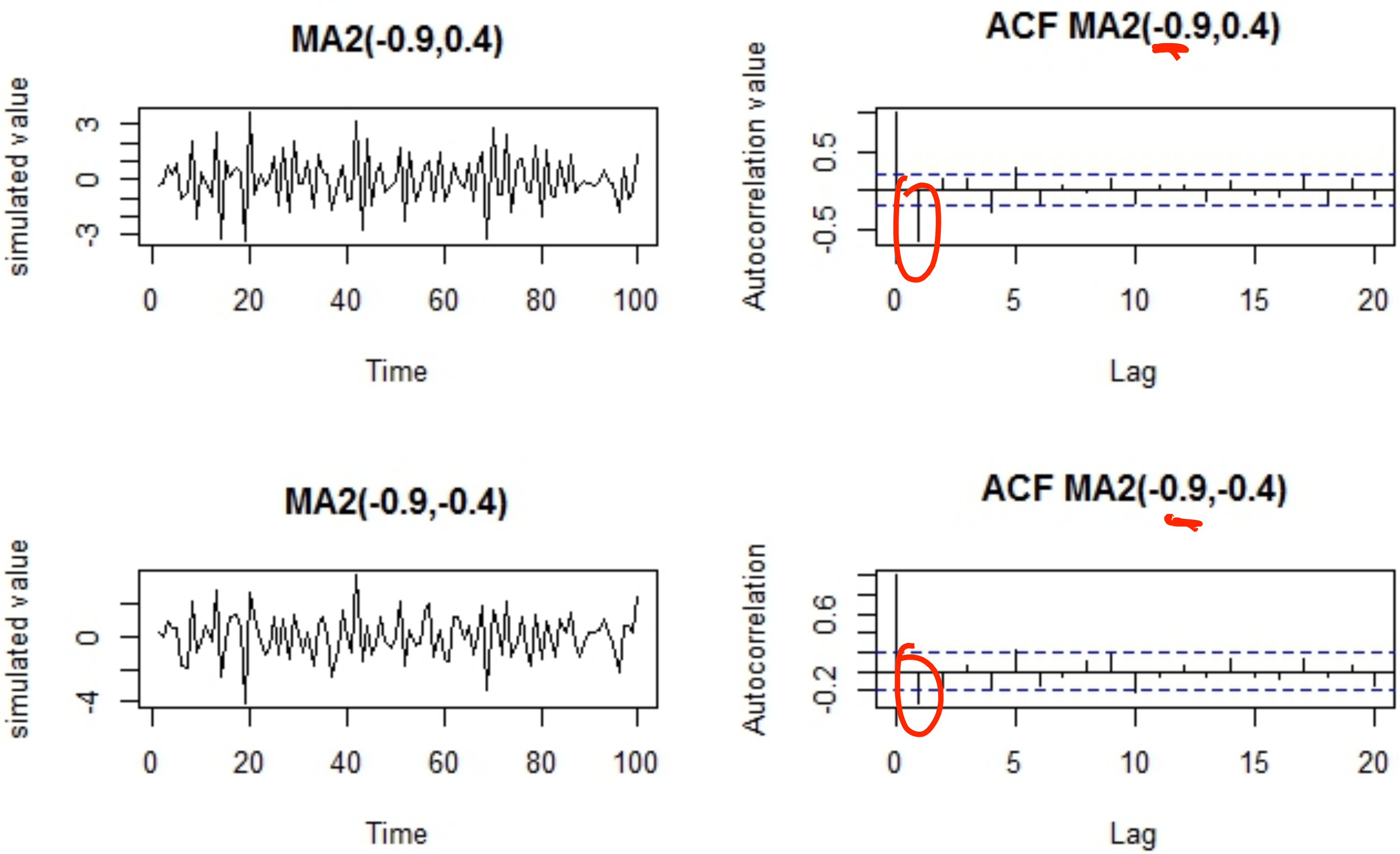
MA(2) Models: Time Series and ACF Plots 1

Observe the impact the second MA parameter has on autocorrelation.



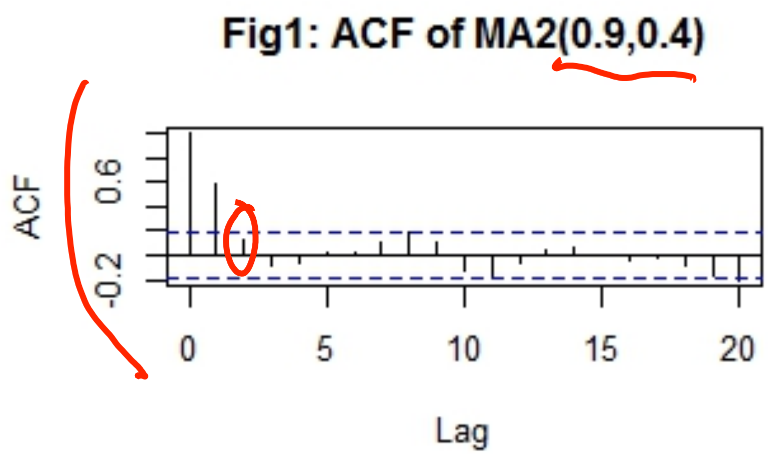
MA(2) Models: Time Series Plots 3

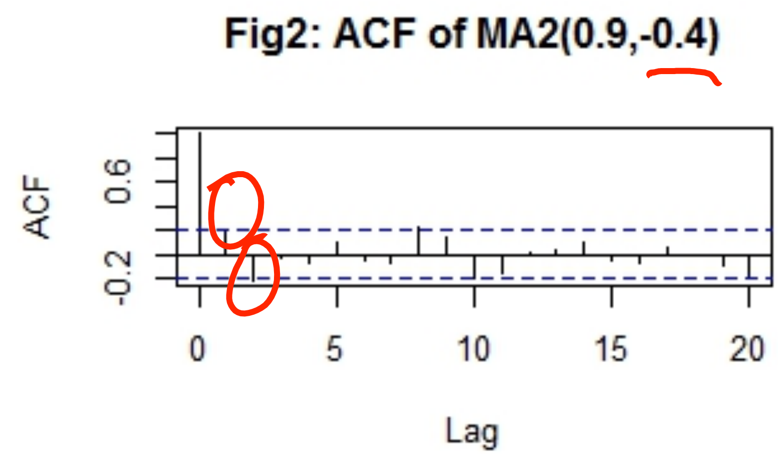
- The first autocorrelation is negative, as influenced by the MA parameters of the models.

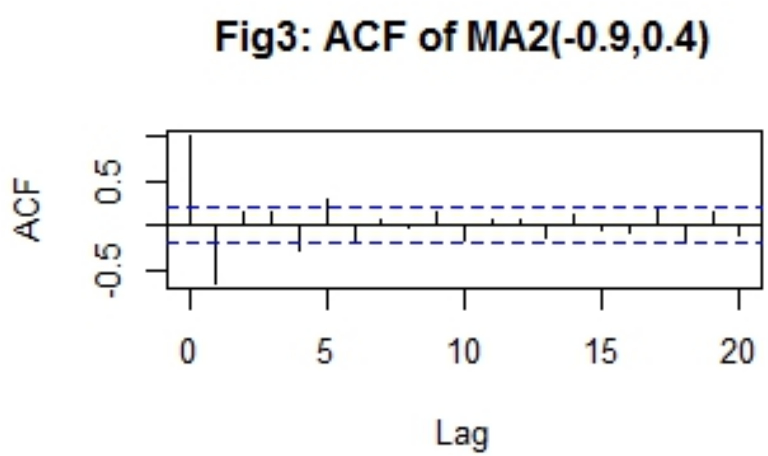


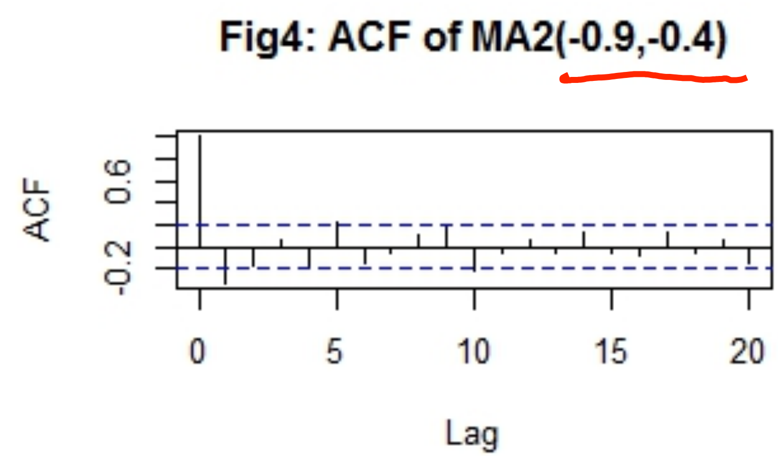
MA(2) Models – Time Series Plots 3

Comparing the ACF of four different MA(2) models:









Moving Average Models

Modeling Using Real-World Data: The Data and Descriptive Statistics

The Data Used in This Example

- Instead of using the data (British pounds vs. New Zealand dollar (NZD)) series provided by the textbook, I downloaded the monthly US dollar–NSZ dollar exchange rate series from the Federal Reserve website:

http://research.stlouisfed.org/fred2/series/EXUSNZ/downloaddata

The Data Used in This Example (2)

- The advantage of obtaining the data series from the official source is that you have complete control over modification and transformation made to the series.

- I will still use the data series provided by the book later in this lecture because I want to extend the example used in the book.

- Alternatively, one could use function in the R library "**quantmod**" to stream the data directly, which I will do in the next lecture.

- **Quantmod** is especially designed for financial time series analysis, and I will not go into the details of this library.

The Data Used in This Example (3)

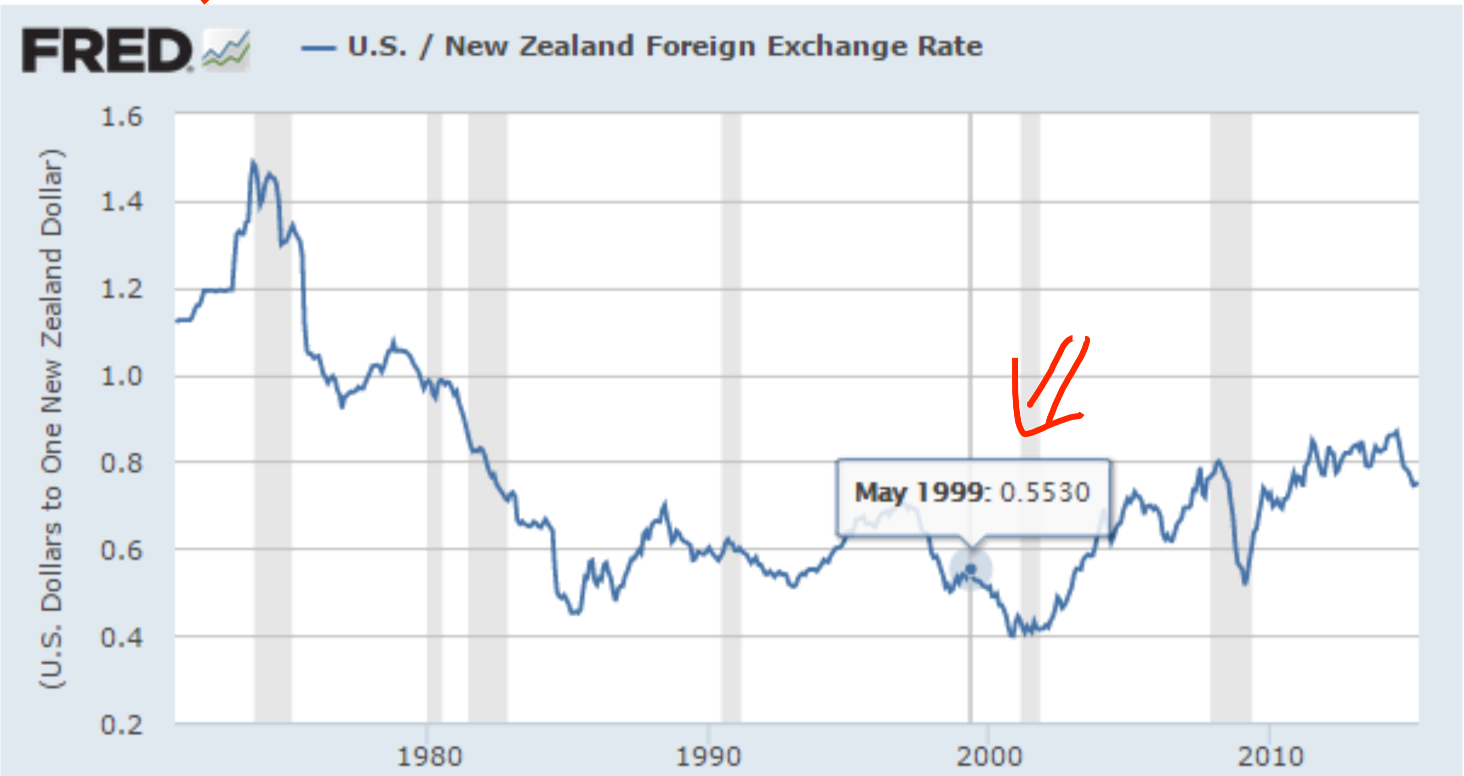
U.S. / New Zealand Foreign Exchange Rate

2015-04: **0.7540** U.S. Dollars to One New Zealand Dollar (+ see more)
Monthly, Not Seasonally Adjusted, EXUSNZ, Updated: 2015-04-13 3:41 PM CDT

Click and drag in the plot area or select dates: 1yr | 5yr | 10yr | Max 1971-01-01 to 2015-04-01

The data series is not seasonally adjusted.

This graph is provided in the FRED website: http://research.stlouisfed.org/fred2/series/EXUSNZ



Examine the Basic Structure of the Data

Import the data from a csv file, and name the data.frame exusnz.

```
exusnz <- read.csv("C:/Users/K/z_Teach/MIDS_AdvStat/data/EXUSNZ.csv",
                   header=TRUE, stringsAsFactors=FALSE)
```

Then, we examine the basic structure of the data:

- 532 observations and 2 variables: date and the exchange rate called "value."

```
> str(exusnz)
'data.frame':	532 obs. of  2 variables:
 $ DATE : chr  "1971-01-01" "1971-02-01" "1971-03-01" "1971-04-01"
 $ VALUE: chr  "1.1194" "1.1250" "1.1254" "1.1250" ...
```

- List the first and last few observations of the data.
- Always a good idea to list some observations to make sure it is what you would expect.

```
> cbind(head(exusnz), tail(exusnz))
        DATE  VALUE       DATE  VALUE
1 1971-01-01 1.1194 2014-11-01 0.7834
2 1971-02-01 1.1250 2014-12-01 0.7766
3 1971-03-01 1.1254 2015-01-01 0.7628
4 1971-04-01 1.1250 2015-02-01 0.7454
5 1971-05-01 1.1254 2015-03-01 0.7460
6 1971-06-01 1.1255 2015-04-01      .
```

This is unexpected!

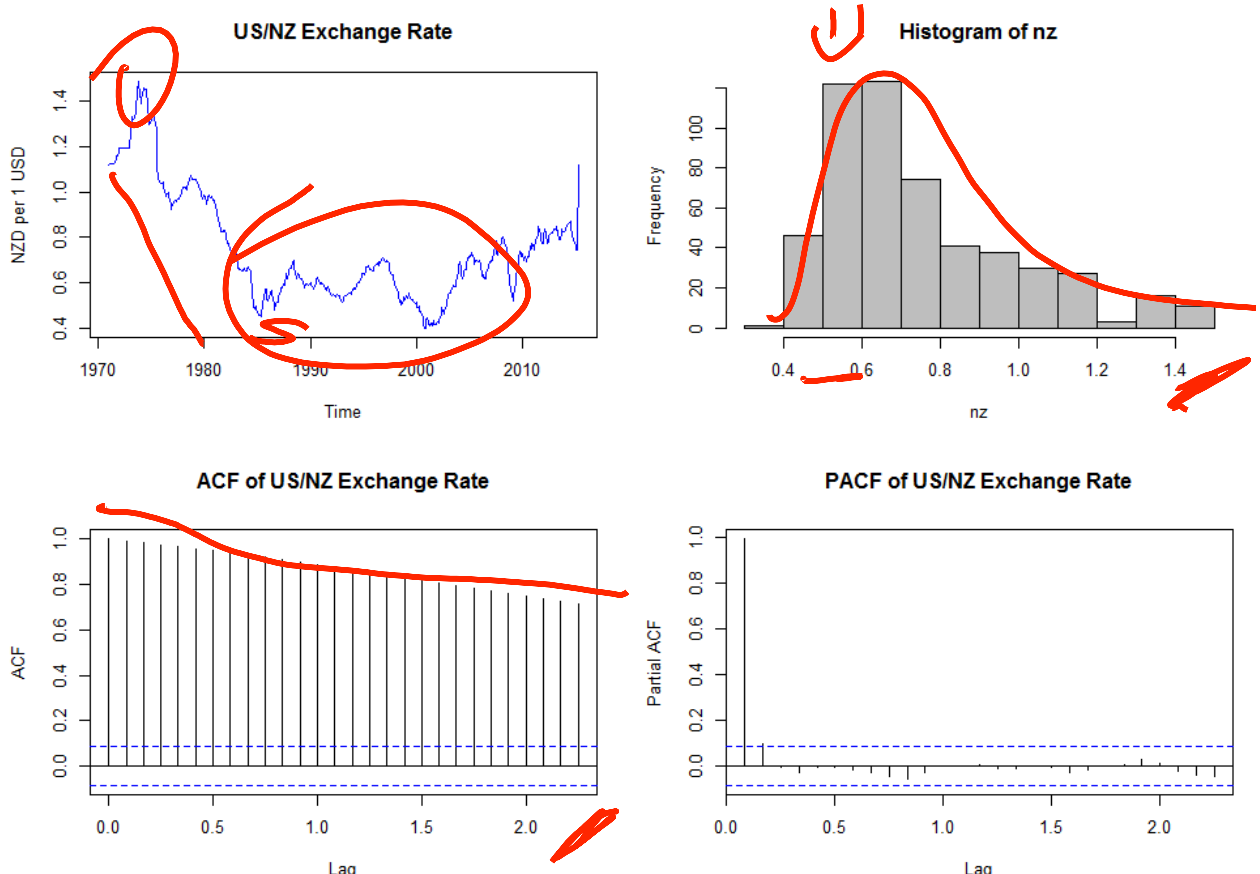
Clean Up the Data—Handling Missing Values

- Delete the last observation. (Note that I could have gone back to the website, downloaded the series again, and examined whether or not the last observation is really unavailable. However, this exercise is for pedagogical purpose, so I simply use the series as is, which already has over 500 observations.)
- I delete the last observation because it contains a missing value, providing no information.
- In general, the root cause of missing values should be thoroughly examined, and one should never simply impute it with mean (or any other values) without a good reason of doing so. One should also not simply delete the information without understanding the reason.

```
> exusnz2 <- exusnz[-532,] # omit the last observation because it is
> str(exusnz2)
'data.frame':	531 obs. of  2 variables:
 $ DATE : chr  "1971-01-01" "1971-02-01" "1971-03-01" "1971-04-01" ...
 $ VALUE: chr  "1.1194" "1.1250" "1.1254" "1.1250" ...
```

Data Visualization and Descriptive Statistics

- The visuals suggests that the series is not a sample path of a MA process.



Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.40	0.57	0.68	0.75	0.85	1.49

1%	5%	10%	25%	50%	75%	90%	95%	99%
0.41	0.46	0.51	0.57	0.68	0.85	1.12	1.30	1.45

Moving Average Models,

Modeling (i.e., Estimation, Model Diagnosis, Model Performance Evaluation, and Statistical Inference)
Using
Real-World Data

Estimation

- We estimate a moving average model of order 4 (MA(4)).
- Each of the estimate parameters is highly significant and so is the intercept.

ARIMA(p,d,q)

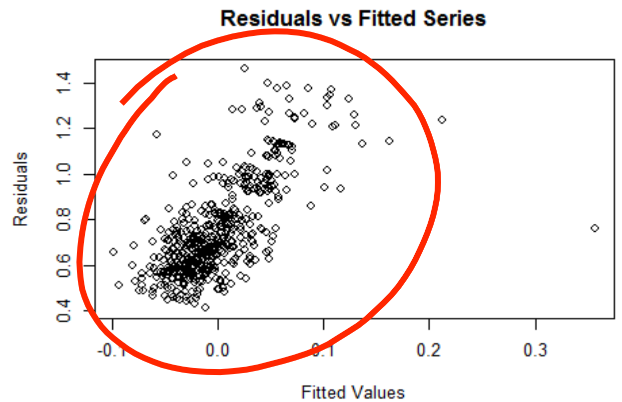
```
> summary(ma4.nzfit)
Series: nz
ARIMA(0,0,4) with non-zero mean

Coefficients:
        ma1    ma2    ma3    ma4   intercept
      1.928   2.08  1.662  0.699       0.755
s.e.  0.045   0.06  0.046  0.037       0.014

sigma^2 estimated as 0.00187:  log likelihood=913
AIC=-1814   AICc=-1814   BIC=-1788

Training set error measures:
                    ME  RMSE   MAE   MPE MAPE MASE ACF1
Training set -0.00024 0.043 0.032 -1.3  4.4  2.3  0.4
```

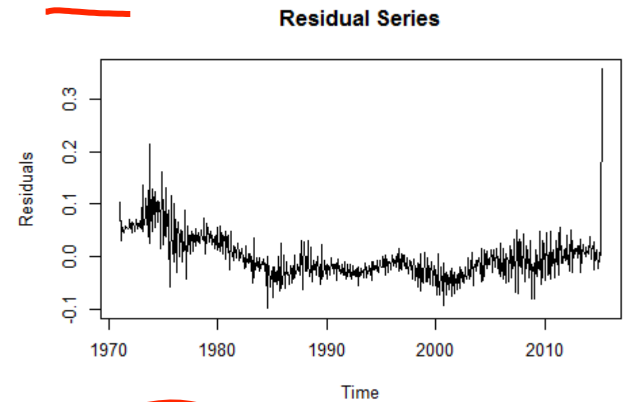
Model Diagnostic Using Residuals

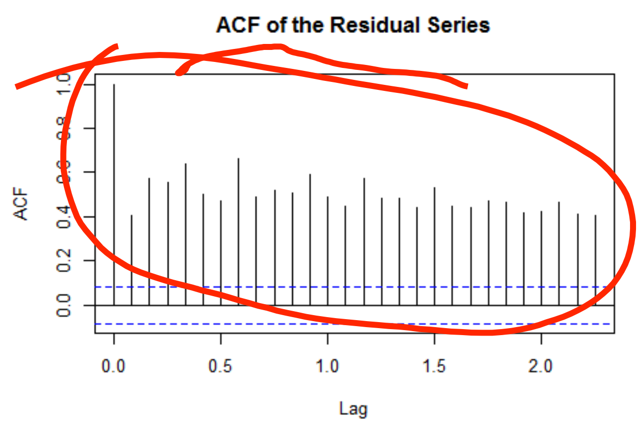
- The residual series confirms that the MA4 model does not capture the NZD series' dynamic well.
- Both the ACF and PACF show evidence of autocorrelation in the residuals.
- Ljung-Box statistic rejects the null hypothesis that the series is uncorrelated.

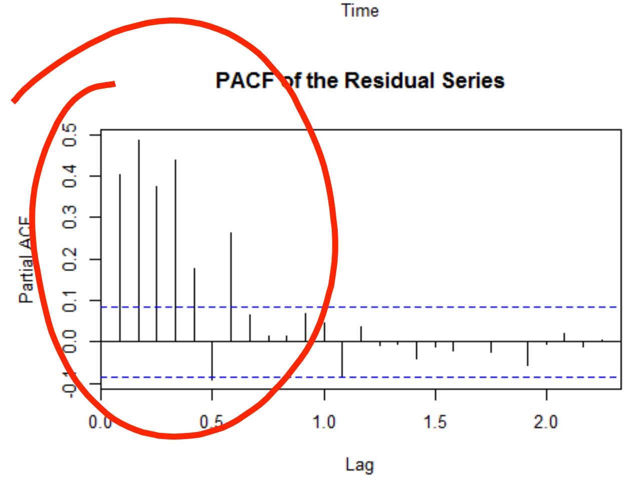
```
> head(ma4.nzfit$resid, 10)
 [1] 0.104 0.032 0.069 0.047 0.049 0.046 0.059 0.055 0.056 0.053
> summary(ma4.nzfit$resid)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -0.10   -0.03   -0.01    0.00    0.02    0.36
```

```
 Box-Ljung test

data:  ma4.nzfit$resid
X-squared = 87, df = 1, p-value < 2.2e-16
```









Moving Average Models

Modeling (i.e., Estimation, Model Diagnosis, Model Performance Evaluation, and Forecasting) Using Simulated Data

Estimation Results of a MA(2) Model

- Using the simulated data from the MA2 models with parameters (0.5, -0.4), estimate a MA(2) model.
- Both of the estimated MA parameters are not statistically different from 0.5 and -0.4.
- The estimated intercept is not statistically significant.

```
Series: ma2c2
ARIMA(0,0,2) with non-zero mean

Coefficients:
        ma1      ma2  intercept
      0.475   -0.525      0.034
s.e.  0.088    0.084      0.085

sigma^2 estimated as 0.796:  log likelihood=-133
AIC=273   AICc=274   BIC=284
```

MA(2) Model: Original, Fitted, and Residual Values

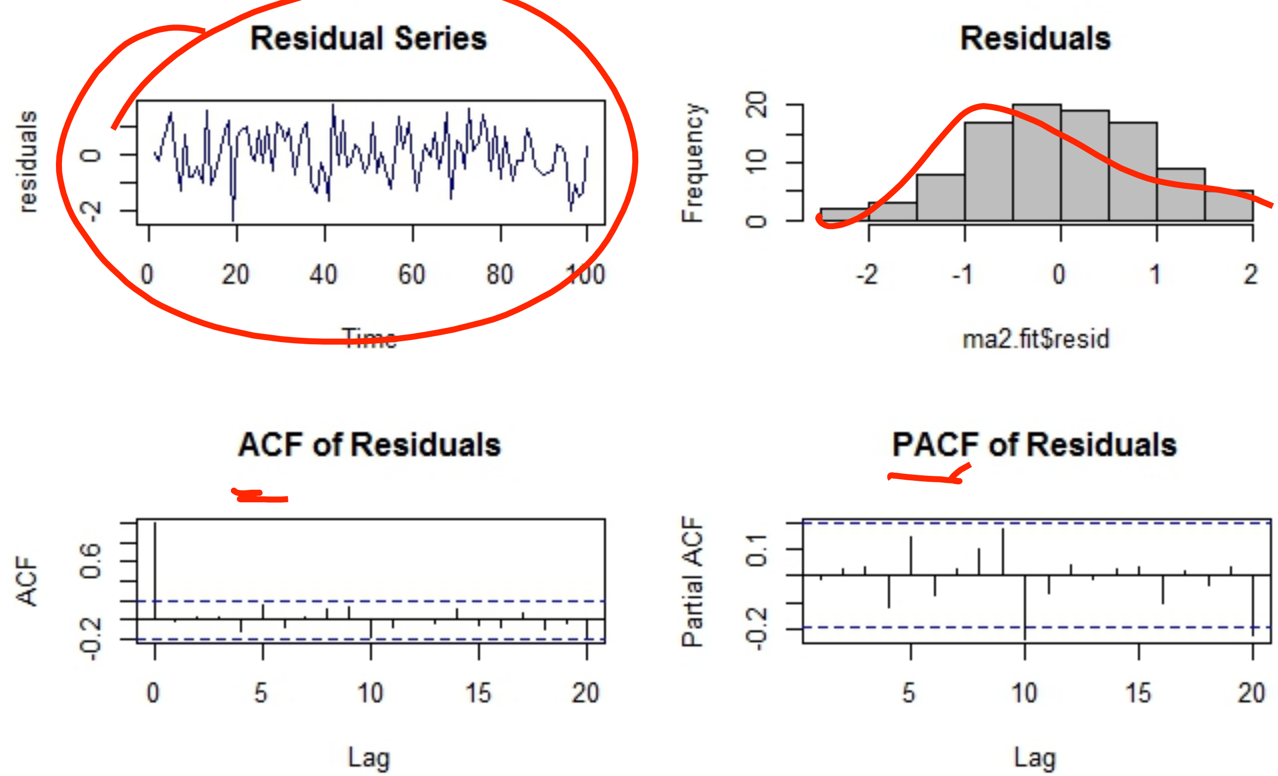
- It is always a good idea to examine the descriptive statistics and list a few values of the original series, the estimated values, and the residual values.

```
> df<-data.frame(cbind(ma2c2, fitted(ma2.fit), ma2.fit$resid))
> stargazer(df, type="text")

=========================================================
Statistic           N   Mean  St. Dev.   Min     Max
---------------------------------------------------------
ma2c2              100 0.027  1.100    -2.400 2.400
fitted.ma2.fit.    100 0.027  0.610    -1.600 1.700
ma2.fit.resid      100 0.001  0.900    -2.400 1.800
---------------------------------------------------------
```

```
> head(cbind(ma2c2, fitted(ma2.fit), ma2.fit$resid),10)
        ma2c2 fitted(ma2.fit) ma2.fit$resid
 [1,]   0.075          0.0413         0.033
 [2,]  -0.242         -0.0093        -0.233
 [3,]   0.529          0.0219         0.507
 [4,]   1.319          0.3856         0.933
 [5,]   1.745          0.2299         1.515
 [6,]   0.293          0.1371         0.156
 [7,]  -2.018         -0.7243        -1.294
 [8,]   0.174         -0.4995         0.673
 [9,]   0.133          0.8989        -0.766
[10,]  -1.435         -0.6438        -0.791
```

MA(2) Model: Examining the Residual Values

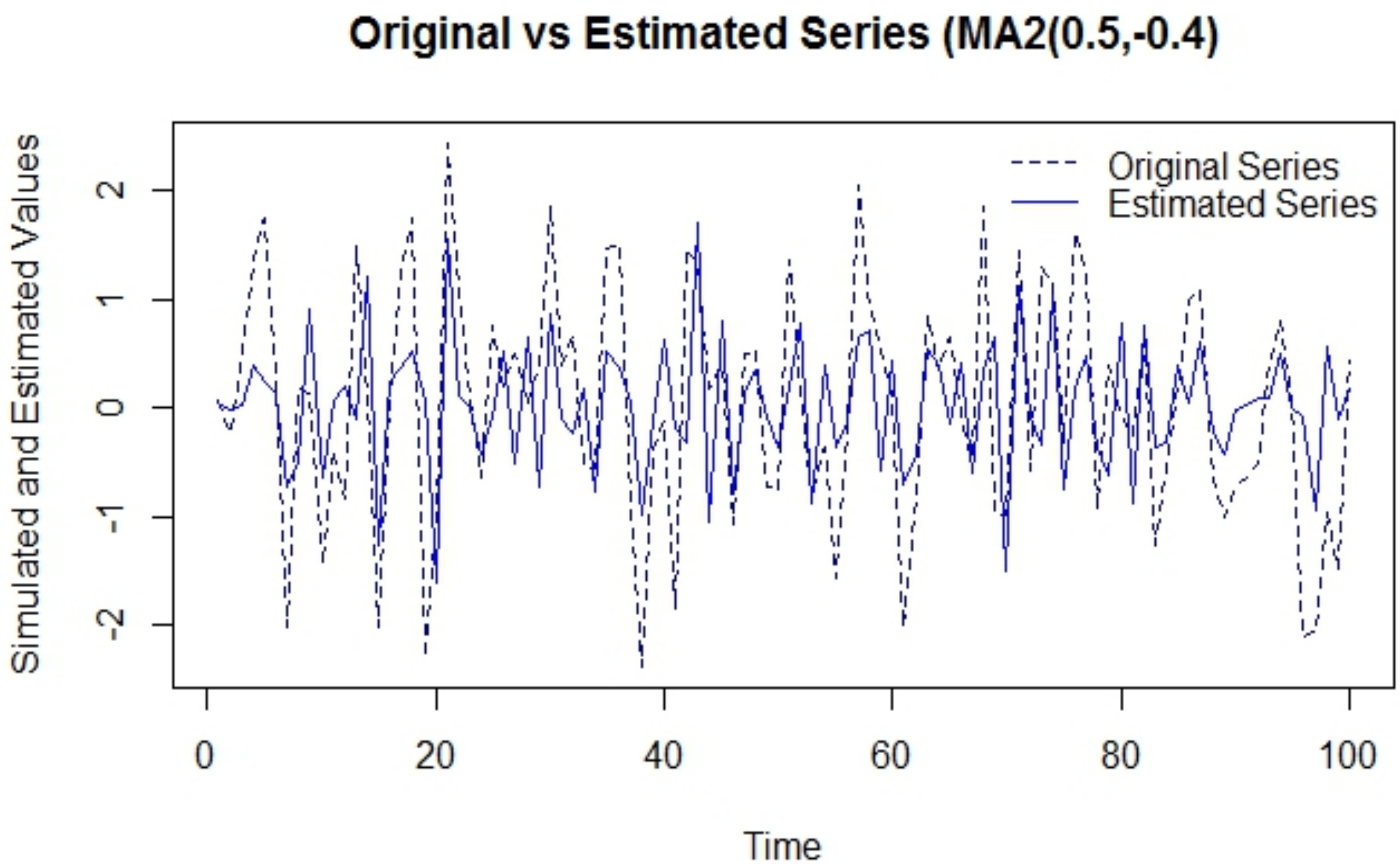
- All of the graphical evidence point to the residuals mimicking white noise
- The Ljung-Box test of residual dynamics cannot reject the null hypothesis that the independence of the residual series.




```
> Box.test(ma2.fit$resid, type="Ljung-Box")

 Box-Ljung test

data:  ma2.fit$resid
X-squared = 0.03, df = 1, p-value = 0.8623
```

Model Performance Evaluation: In-Sample Fit

- The pointwise fit is not perfect.
- However, remember that these are point estimates.



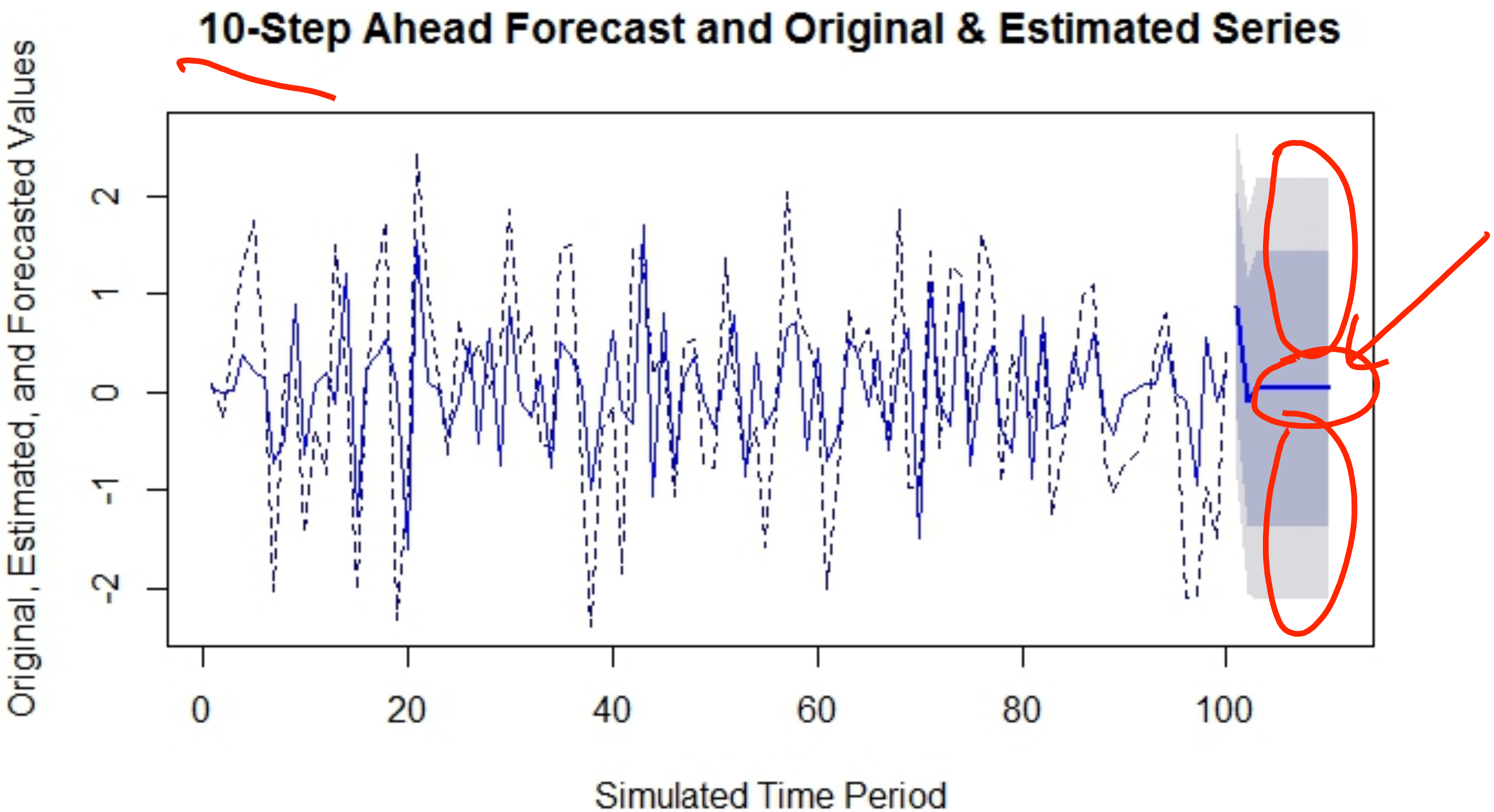
Forecasting

- Note that the forecast after the third step ahead is a constant; this is the feature of MA(2) models.
- This holds true for the high and low forecasts.

```
Model Information:
Series: ma2c2
ARIMA(0,0,2) with non-zero mean

Coefficients:
         ma1      ma2  intercept
       0.475   -0.525      0.034
s.e.   0.088    0.084      0.085

sigma^2 estimated as 0.796:  log likelihood=-133
AIC=273   AICc=274   BIC=284

Error measures:
                  ME RMSE  MAE MPE MAPE MASE   ACF1
Training set 0.00071 0.89 0.74  56  151 0.65 -0.017

Forecasts:
    Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
101          0.880 -0.27   2.0 -0.88   2.6
102         -0.102 -1.37   1.2 -2.04   1.8
103          0.034 -1.37   1.4 -2.11   2.2
104          0.034 -1.37   1.4 -2.11   2.2
105          0.034 -1.37   1.4 -2.11   2.2
106          0.034 -1.37   1.4 -2.11   2.2
107          0.034 -1.37   1.4 -2.11   2.2
108          0.034 -1.37   1.4 -2.11   2.2
109          0.034 -1.37   1.4 -2.11   2.2
110          0.034 -1.37   1.4 -2.11   2.2
```

Forecasting (2)

- The following graph plots the original series, overlaid with the estimated series, a 10-step-ahead forecast series, and the confidence interval of the forecast.



Berkeley SCHOOL OF INFORMATION