ARIMA | Model ID

Model ID for ARIMA(p, d, q) Models

$$(B)(1 \quad B)^{d}(X_t \qquad) = \quad (B)a_t$$

where

- $\varphi(B)$ and $\theta(B)$ are pth and qth order operators, respectively
- All roots of $\varphi(z) = 0$ and $\theta(z) = 0$ are outside the unit circle

Properties of ARIMA Models

Recall:

- The $(1-B)^d$ factor dominates the stationary components
  - In the realizations
  - Autocorrelations
  - Spectral densities (not shown) all have peaks at $f = 0$
- For $d > 1$ this domination is even more striking
- The true autocorrelations are equal to one for all lags
- The sample autocorrelations will always damp (in part because of the way they are calculated)
- **Slowly damping sample autocorrelations is an indication of an ARIMA data**

Recall the Issues with Parameter Estimation in Stationary Models with Roots *Close* to the Unit Circle

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

Real root very close to unit circle

Factor	**Abs recip**	**System freq**
$1\text{-}0.995B$	0.9950	0.0
$1\text{-}1.2B+0.8B^2$	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

- Burg and ML estimates are better than YW in near "nonstationary" case

Model ID in Nonstationary Case: General Comments

- YW and Burg estimates always produce stationary AR models
  - Even if data are from an ARIMA model
- ML estimates may produce nonstationary models
  - However, the nonstationarity is often due to roots inside the unit circle rather than on the unit circle like *(1-B)*.
    - i.e. unacceptable models

Key Point

- A decision whether to fit a stationary or nonstationary model must be made by the time series analyst
  - Simply noting that a fitted model has roots outside the unit circle does not in itself indicate stationarity
- The decision whether to fit a nonstationary model may depend on the physical problem being considered
  - For example, data from stationary models have an attraction to a mean value.
    - Is this a reasonable assumption for your data?

DataScience@SMU

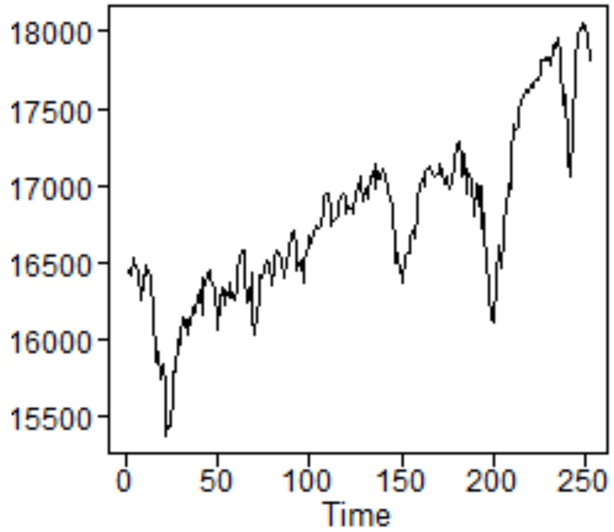
ARIMA | Dow Jones Example Part 1

Dow Jones Data

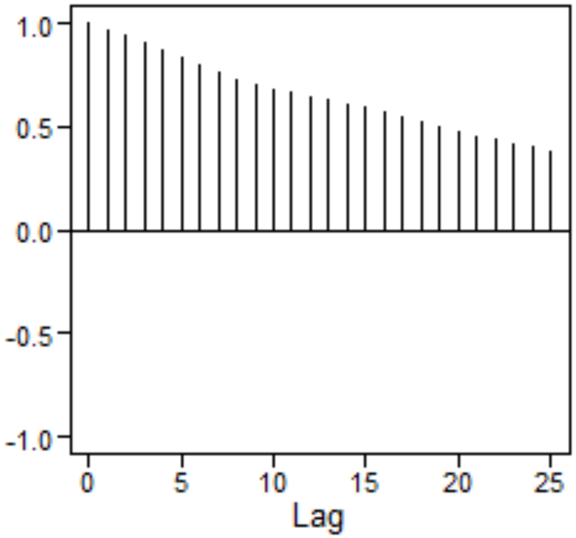
The graphs below are

- A plot of daily closing values of the DOW Jones for 2014
- The associated (damping) sample autocorrelations

DOW Jones 2014



Sample autocorrelations



Dow Jones Data

The tswge commands:

```
data(dowjones2014)
aic.wge(dowjones2014,p=0:5,q=0:2)
```

select the AR(1) model:

$$(1 - .9816B)(X_t - 16778) = a_t$$

Question: Should we use this (stationary) model or a model that includes a $(1-B)$ factor?

- $\hat{\varphi}_1$ is very close to 1
- Sample autocorrelations damp, but they do so slowly (the autocorrelation at lag 10 is nearly .7)
- Do you expect the DOW Jones average to eventually pull back to a mean value like 16,778?
  - Historically the answer to this last question is ***No.***

A nonstationary model is suggested

DataScience@SMU

ARIMA Box-Jenkins Model ID Procedure

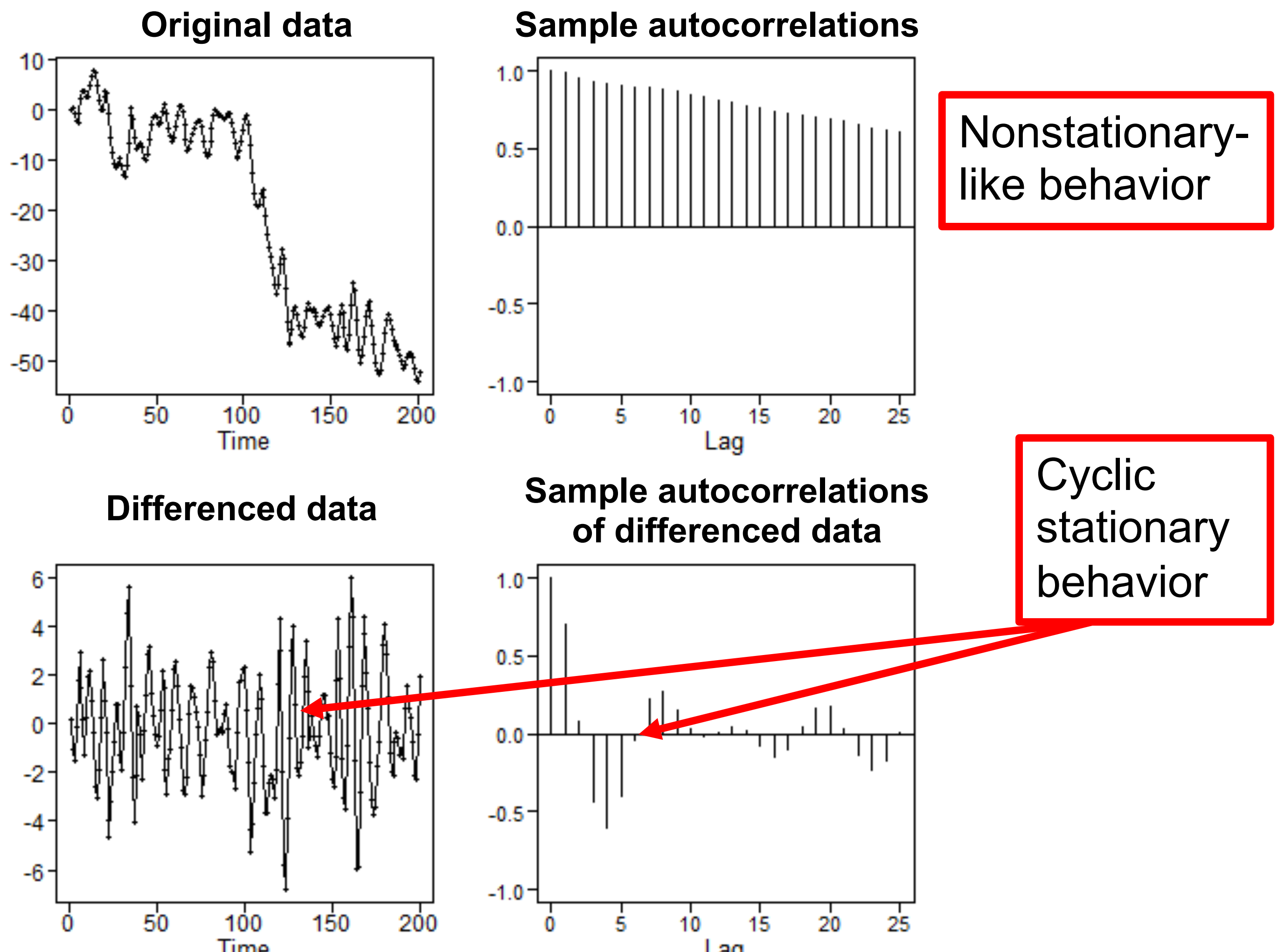
Classical Box-Jenkins Procedure* for Including a Unit Root in the Model

- If data are wandering and sample autocorrelations damp slowly
  - Difference the data
- If differenced data still show evidence of a unit root
  - Difference the data again, etc.

* Reference to "Box-Jenkins" procedures apply to techniques proposed in the Classic Box-Jenkins (1970) text on time series analysis. These include
  - The forecasting procedure we used
  - The stationary model ID procedure we do not recommend
  - The nonstationary differencing techniques listed above

A Model with a Unit Root

$$(1 - B)(1 - 1.2B + .8B^2)X_t = a_t$$



Steps for Obtaining an ARIMA(p,d,q) Model

- Difference the data (possibly multiple times) until the data appear stationary
- Estimate the parameters of the "stationarized data"

DataScience@SMU

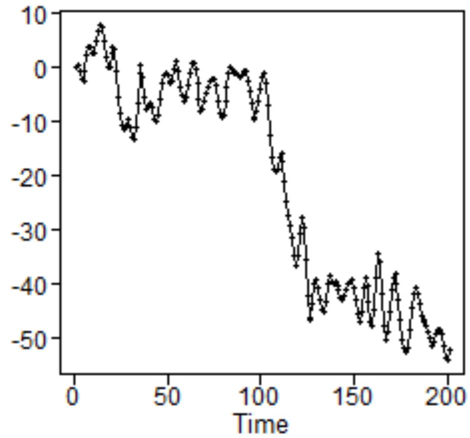
Example 1

1 Unit Root

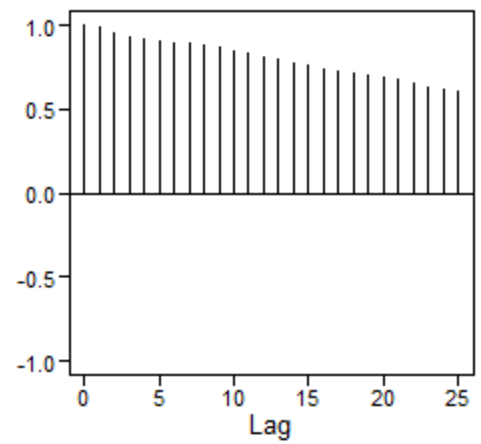
Model with a Unit Root

$$(1 \quad B)(1 \quad 1.2B + .8B^2)X_t = a_t$$

Original data



Sample autocorrelations



Notes:

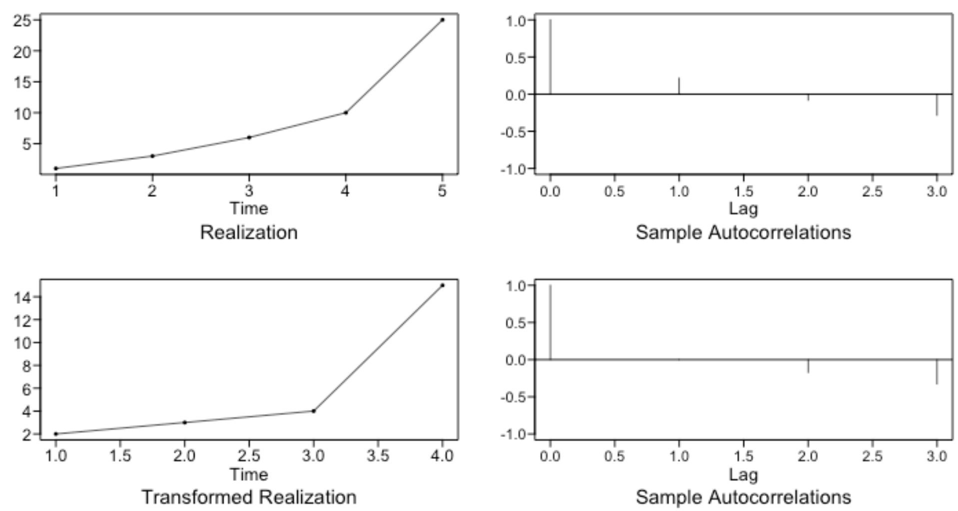
- We decided to difference the data because
  - Dramatic wandering
  - Slowly damping autocorrelations
- Differencing the data adds a factor of $(1–B)$ to the final model of the original data (which will be ARIMA)
- A feasible decision might also have been to go ahead and model the data using a stationary model
  - i.e. fit a model (using AIC) to the original data

Review: “Difference” the Data using tswge

tswge transformation function

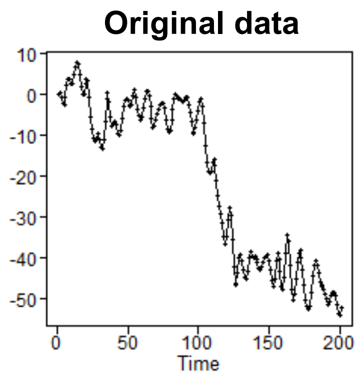
```
# the following command differences the data in x
y=artrans.wge(x,phi.tr=1)
# This simply means that y(i) = x(i) - x(i-1)
# y has length n-1 because x(1) has no x(0) before it.
# Example
x = c(1,3,6,10,25)
y = artrans.wge(x,phi.tr = 1)
y # shows the 4 differences
```

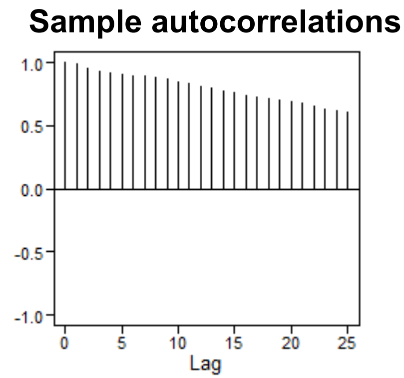
```
> y # shows the 4 differences
[1]  2  3  4 15
```



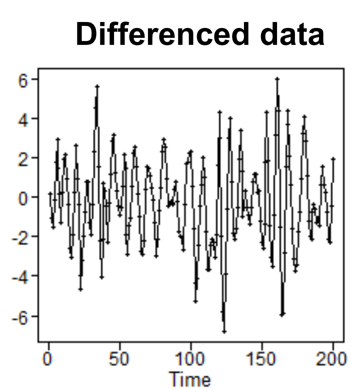
Model with a Unit Root

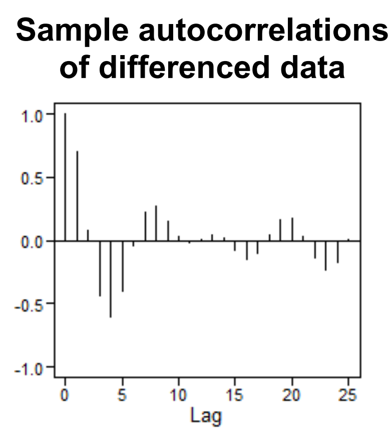
$$(1\quad B)(1\quad 1.2B+.8B^2)X_t = a_t$$






```
# generate ARIMA(2,1,0) data
xd1=gen.arima.wge(n=200,phi=c(1.2,-.8),d=1,sn=56)
# difference the data
xd1.dif=artrans.wge(xd1,phi.tr=1)
# xd1.dif is the differenced data
```

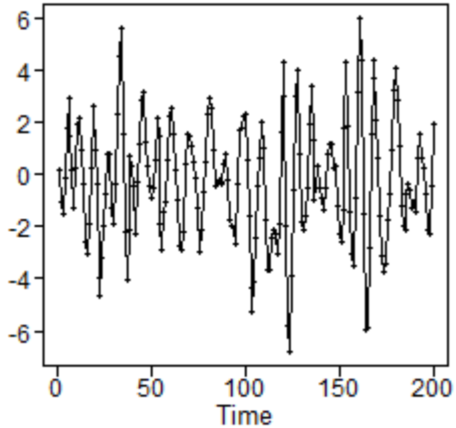




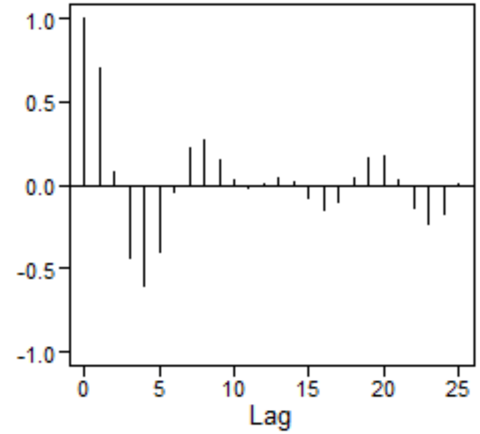
Model with a Unit Root

$$(1 \quad B)(1 \quad 1.2B + .8B^2)X_t = a_t \qquad {}_a^2 = 1$$

Differenced data



Sample autocorrelations of differenced data



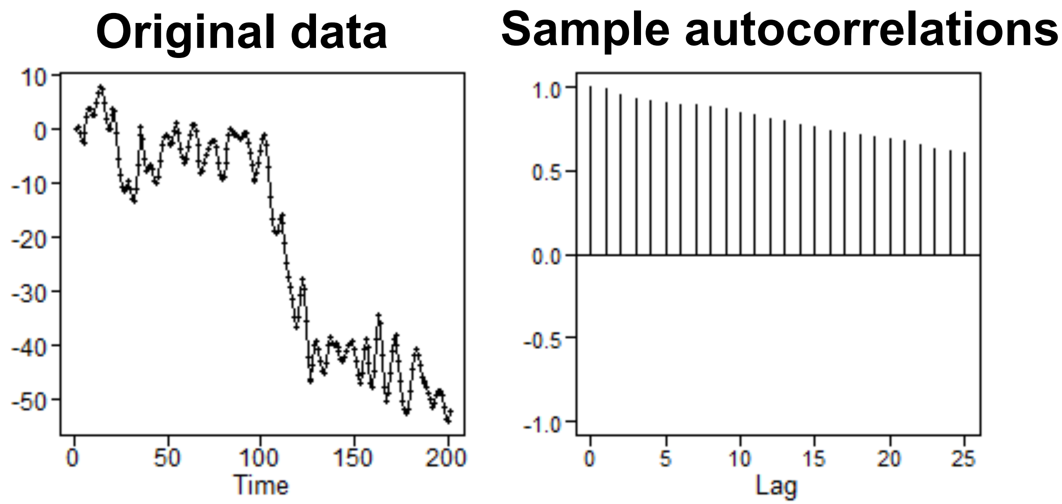
Differenced data (xd1.dif) appear stationary—use AIC

```
aic5.wge(xd1.dif,p=0:5,q=0:2)
#  aic picks AR(2)
est.ar.wge(xd1.dif,p=2)  # $phi [1] 1.265005 -0.802240
#  $avar [1] 0.9293244
mean(xd1)  # -21.22
```

Final model

$$(1-B)(1-1.27B+.80B^2)(X_t+21.22) = a_t \quad \hat{\sigma}_a^2 = .93$$

Or … We Could Have Modelled it with a Stationary Model




```
# generate ARIMA(2,1,0) data
x=gen.arima.wge(n=200,phi=c(1.2,-.8),d=1,sn=56)
# Assume it will eventually make it back to a fixed
# process mean. That is, it is attracted to a constant mean.
aic5.wge(x) # selects a p = 4, q = 0
est.arma.wge(x, p = 4, q = 0)
#phis:  2.3546329 -2.2920782  1.0456011 -0.1101713
#avar: 0.9295115
mean(x) #-21.21951
```

$$(1 - 2.35B + 2.29B^2 - 1.05B^3 + 0.11B^4)(X_t + 21.22) = a_t$$

$$\sigma_a^2 = .9295$$

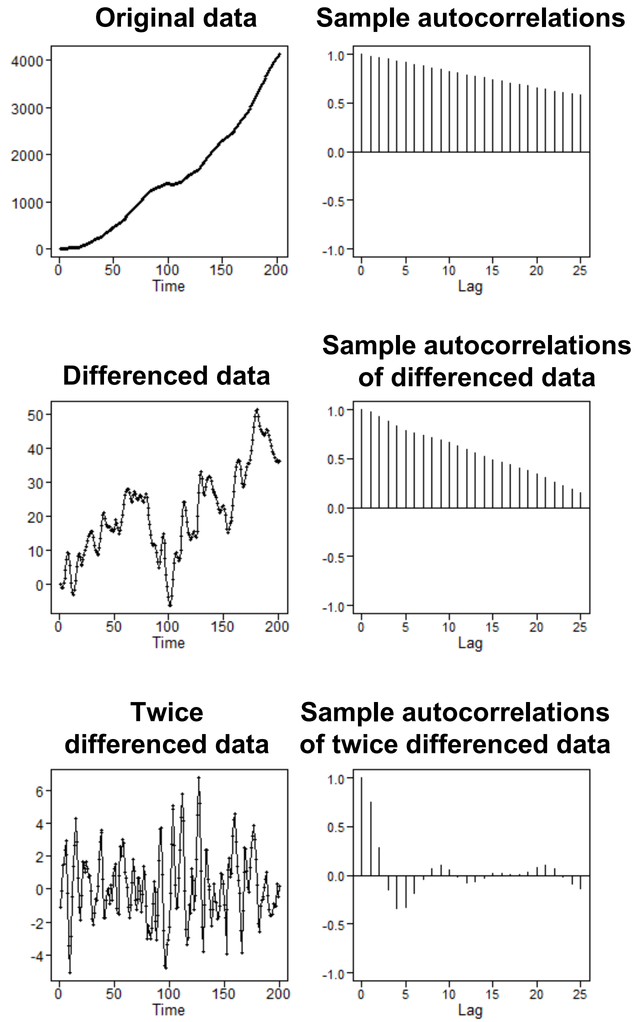
DataScience@SMU

Example 2

2 Unit Roots

A Model with 2 Unit Roots

$$(1-B)^2(1-1.2B+.6B^2)X_t = a_t$$



Nonstationary: no stationary AR(2) behavior visible

Nonstationary: stationary AR(2) behavior not obvious

Stationary AR(2) behavior can now be seen

tswge demo

```
# generate data from the ARIMA(2,2,0) model
x=gen.arima.wge(n=200,d=2,phi=c(1.2,-.6),sn=132,vara=1)
# difference the data
x.d1=artrans.wge(x,phi.tr=1)
# difference the data again since the differenced data
#    still has nonstationary ARIMA characteristics
#     (see previous slide)
 x.d2=artrans.wge(x.d1,phi.tr=1)
#  x.d2 appears to be stationary (see previous slide)
aic5.wge(x.d2,p=0:5,q=0:2)
# AIC picks an AR(2)  # which seems reasonable from
est.ar.wge(x.d2,p=2)
#  $phi[1]  1.2724446 -0.6827008
#   $avar [1] 1.026015
mean(x)  # 1512
```

Final model

$$(1-B)^2(1-1.27B+.68B^2)(X_t-1512)=a_t \quad \hat{\sigma}_a^2=1.03$$

DataScience@SMU

Comments about ARIMA Models

Comments about ARIMA Models

$$(1 \quad B)(1 \quad 1.2B + .8B^2)X_t = a_t$$

$$(1 - B)(1 - 1.27B + .80B^2)(X_t + 21.22) = a_t \quad \hat{\sigma}_a^2 = .93$$

$$(1 - B)^2(1 - 1.2B + .6B^2)X_t = a_t$$

$$(1 - B)^2(1 - 1.27B + .68B^2)(X_t - 1512) = a_t \quad \hat{\sigma}_a^2 = 1.03$$

- Notice that both of the previous ARIMA models had $\mu = 0$.
- However, the two sample means were $\bar{X} = -21.22$ and 1512, respectively.
- This is because there is no attraction to a mean in ARIMA models.
- For ARIMA models, you can always write the model in zero mean form.

DataScience@SMU

Review of Forecasts for ARIMA Models

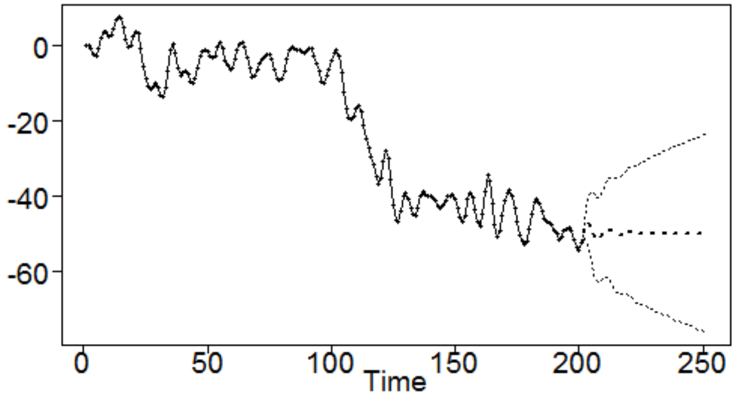
Forecasting with the ARIMA(2,1,0) Model

Fitted model

$$(1-B)(1-1.27B+.80B^2)(X_t+21.22)=a_t \quad \hat{\sigma}_a^2=.93$$

tswge forecasting code

```
# let xd1 denote the realization that produced
#   this model fit
# code for forecasting the next 50 values
fore.aruma.wge(xd1,d=1,phi=c(1.27,-.8),n.ahead=50)
```



Forecasts essentially forecast the last data value (because of a singe $(1-B)$ factor). The AR(2) factor adds a little cyclic behavior for early steps ahead.

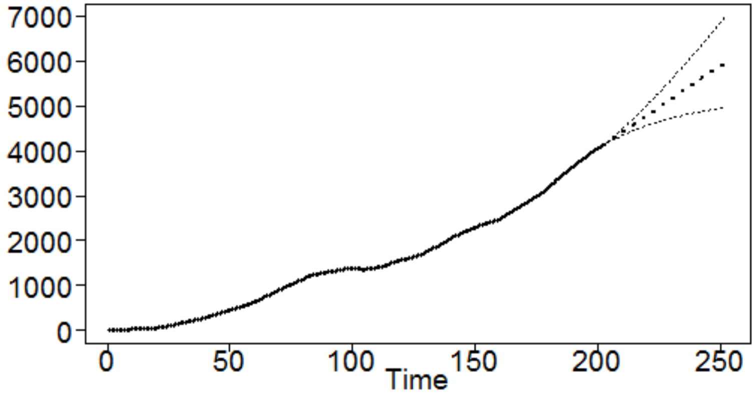
Forecasting with the ARIMA(2,2,0) Model

Fitted model

$$(1-B)^2(1-1.27B+.68B^2)(X_t-1512)=a_t \quad \hat{\sigma}_a^2=1.03$$

tswge forecasting code

```
# let x denote the realization that produced
#   this model fit
# code for forecasting the next 50 values
fore.aruma.wge(x,d=2,phi=c(1.27,-.68),n.ahead=50)
```



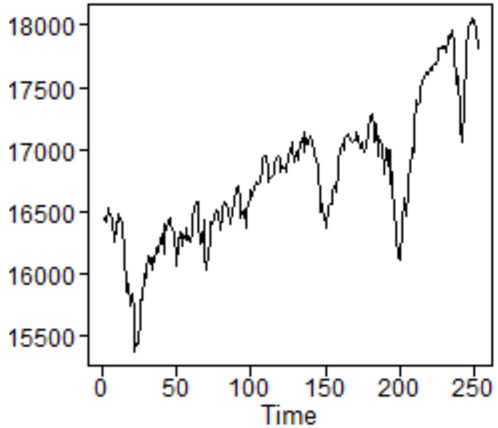
Forecasts essentially forecast a linear trend determined by the last two data values. The effect of the AR(2) is not noticeable.

DataScience@SMU

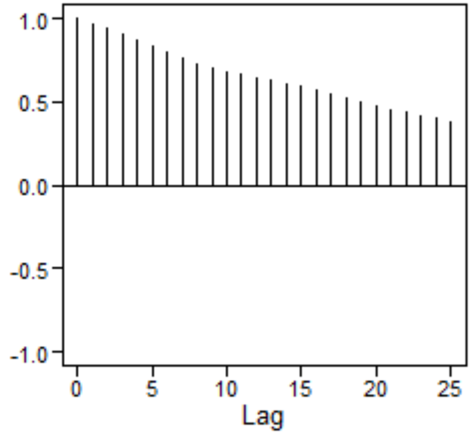
ARIMA | Dow Jones Example Part 2

2014 DOW Jones Data

DOW Jones 2014

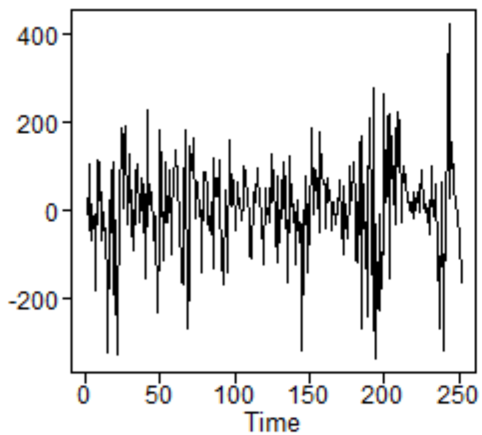


Sample autocorrelations

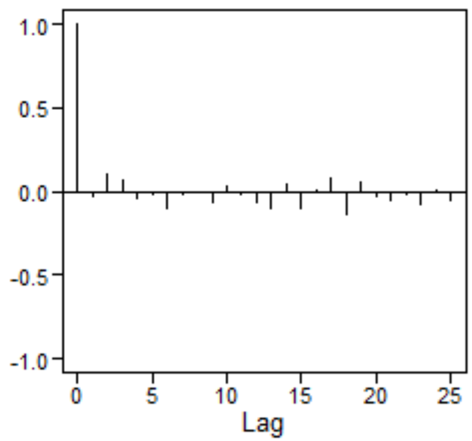


Nonstationary-like behavior

Differenced DOW data



Sample autocorrelations of differenced data



Looks stationary

2014 DOW Jones Data

tswge code for differencing and analyzing the differenced data

```
data(dowjones2014)
# redefine for simplicity
dow=dowjones2014
# difference the data
dow.1=artrans.wge(dow,phi.tr=1)
aic5.wge(dow.1,p=0:5,q=0:2)
#  aic picks ARMA(4,1)
est.arma.wge(dow.1,p=4,q=1)
# $phi [1]  0.9265955  0.1355600 -0.0240474 -
0.1242563
# $theta [1] 0.9999996
#  $avar [1] 12112
mean(dow)  # 16778
```

2014 DOW Jones Data

Let's look at the estimated ARMA(4,1) model

$$(1 \quad .93B \quad .14B^2 + .02B^3 + .12B^4)(X_t \quad 16778) = (1 \quad B)a_t \qquad \hat{\ }_a^2 = 12112$$

which factors into the following factors found using

```
factor.wge(phi=c(.93,.14,-.02,-.12))
```

$$(1 \quad .85B)(1 \quad .68B)(1 + .61B + .21B^2)(X_t \quad 16778) = (1 \quad B)a_t$$

Notes:

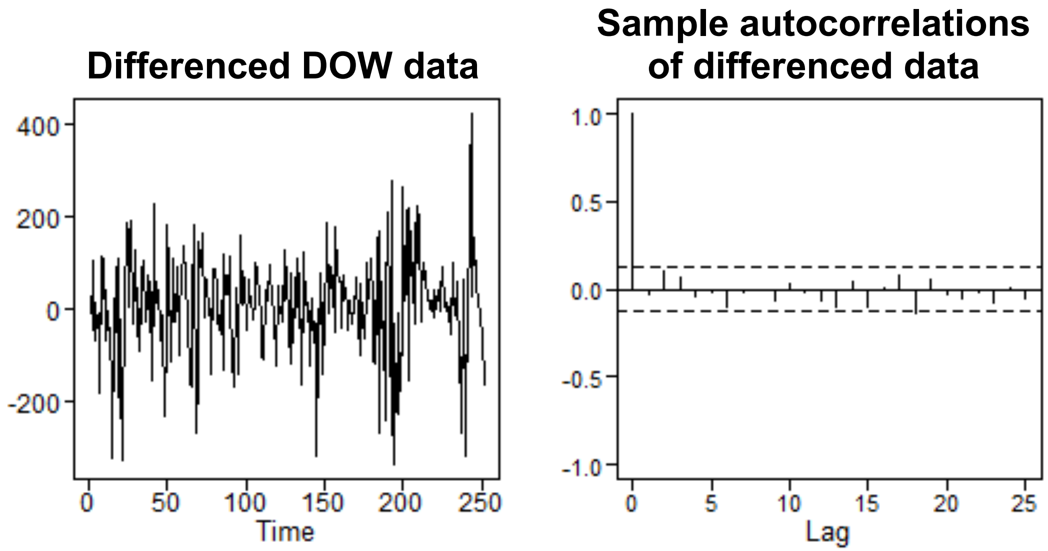
- The above is a noninvertible model, but actually $\hat{\theta}_1$=.9999996 which is "barely" invertible.
- The two factors $(1-.85B)$ and $(1-.68B)$ work to cancel-out the $(1-.9999996B)$ MA factor.
- It appears that a smaller order model is called for.

2014 DOW Jones Data

We try using BIC

```
aic5.wge(dow.1,p=0:5,q=0:2,type='bic')
# BIC selects p=0,q=0, i.e. white noise
```

Looking again at the differenced data we see that white noise is plausible.



We should have modeled it as white noise before applying AIC!

2014 DOW Jones Data

Final model

$$(1 \quad B)(X_t \quad 16778) = a_t$$

or

$$(1 \quad B)X_t = (1 \quad B)16778 + a_t$$

But, $(1 \quad B)16778 = 16778 \quad 16778 = 0$

So, we can write the model ***without the mean***.

Note:

This is true for all ARIMA models with $d \geq 1$

We could have written the final models for the ARIMA(2,1,0) and ARIMA(2,2,0) examples ***without a mean.***

2014 DOW Jones Data

DOW Jones model

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

Note that the differenced data $Y_t = (1 - B)X_t$ is white noise,

- So we can estimate the white noise variance as the sample variance of the differenced DOW data
- We found $\hat{\sigma}_a^2$ using the R command `sd`

```
sd(dow.1)
```

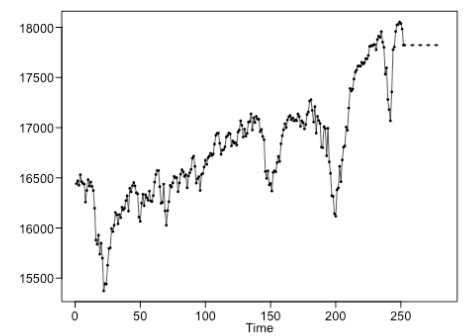
and squaring the result.

Final model $(1 \quad B)X_t = a_t \qquad \hat{\ }_a^2 = 13036$

Note:

- Forecasts from this model will simply repeat the last data value for all steps ahead
- It's difficult to predict the DOW!

DOW Jones



DataScience@SMU

A More General Approach to ARIMA Model ID

A More General Approach for Detecting Nonstationarities

To this point, our approach (Box-Jenkins) for detecting and removing stationarities is as follows:

- If data are wandering and sample autocorrelations damp slowly
  - *Difference the data*
- If differenced data still show evidence of a unit root
  - *Difference the data again, etc.*

A More General Approach for Detecting Nonstationarities

The ARIMA is an extension of the ARMA model to allow one or more factors of $(1-B)$ in the model

- That is, ARIMA models include roots **on** the unit circle
  - For ARIMA models, these roots are +1

ARIMA models are not the only extension of ARMA models that have roots **on** the unit circle

- These roots may be complex. These models include
  - Seasonal models
  - Cyclic ARUMA models (see Woodward, et al., 2017; but not covered here)

DataScience@SMU

Tiao/Tsay and Overfit Tables

1 Unit Root

Result (Tiao/Tsay) (loosely stated):

If a high order AR(p) model is fit to a realization from a nonstationary model with roots on the unit circle, then factors associated with these roots will "show up" in the factor table.

- The nonstationary behavior will "show up" in factor tables for a wide range of orders p.
- Fitting several high order models to a set of data to identify nonstationarities is called ***overfitting.***

Overfitting using the `tswge` package:

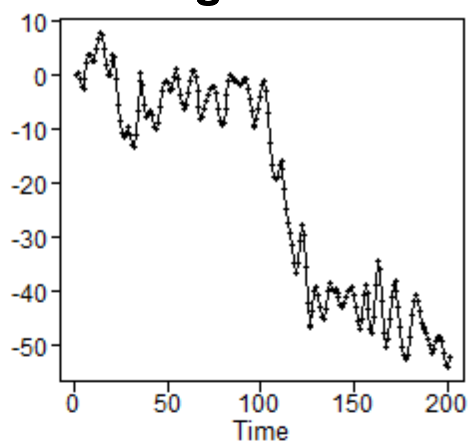
`est.ar.wge` function

- We use Burg estimates (always stationary)
- MLE would be ok

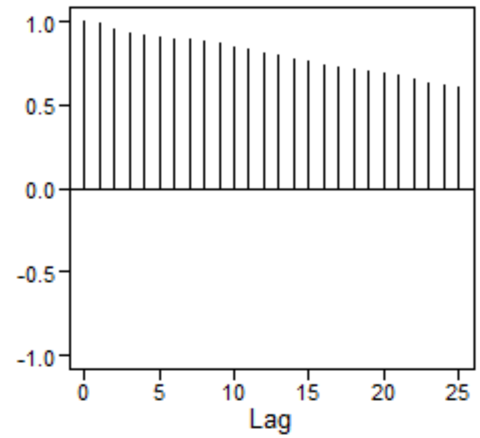
Recall: ARIMA(2,1,0) Model

$$(1 - B)(1 - 1.2B + .8B^2)X_t = a_t$$

Original data



Sample autocorrelations




```
# generate ARIMA(2,1,0) data as before
xd1=gen.arima.wge(n=200,phi=c(1.2,-.8),d=1,sn=56)
# fit an AR(6) and AR(8) to this realization
est.ar.wge(xd1,p=6,type='burg')
est.ar.wge(xd1,p=8,type='burg')
```

Overfit Tables for ARIMA(2,1,0) Data

AR(6) fit

AR factor	Absolute reciprocal of root	System frequency (f_0)
$1-0.9944B$	0.9944	0.0000
$1-1.1978B+0.7584B^2$	0.8708	0.1293
$1-0.4472B+0.0881B^2$	0.2967	0.1142
$1+0.2879B$	0.2879	0.5000

AR(8) fit

AR factor	Absolute reciprocal of root	System frequency (f_0)
$1-0.9951B$	0.9951	0.0000
$1-1.1324B+0.7314B^2$	0.8552	0.1348
$1-1.0807B+0.4241B^2$	0.6513	0.0943
$1+0.4968B$	0.4968	0.5000
$1+0.3615B+0.2464B^2$	0.4964	0.3093

Key point:

Even though the data are not from an AR(6) or AR(8), the factor close to 1-*B* shows up in both factor tables

Recall for ARIMA(2,1,0) Data

We previously analyzed these data and we decided to difference the data because:

- The data are wandering and sample autocorrelations damp slowly.
- We only differenced once because the differenced data appeared to be stationary.

Using overfitting:

- We decide to difference the data because the near 1-*B* factor continued to show up in high order Burg overfits.
- The differenced data will likely appear to be stationary after differencing because only one factor near 1-*B* appears in the overfit tables.

Modeling the ARIMA(2,1,0) Data using Overfitting

Since the factor tables suggest a factor of $(1-B)$, we difference the data and come up with the same model as before:

$$(1-B)(1-1.27B+.80B^2)(X_t+21.22)=a_t \quad \hat{\sigma}_a^2=.93$$

Note:

As mentioned with the DOW data, we can write the model without a mean, i.e.

$$(1-B)(1-1.27B+.80B^2)X_t=a_t \quad \hat{\sigma}_a^2=.93$$

Question:
How do you use overfitting to detect ***more than one unit root?***

DataScience@SMU

Overfit Factor Tables

2 Unit Roots

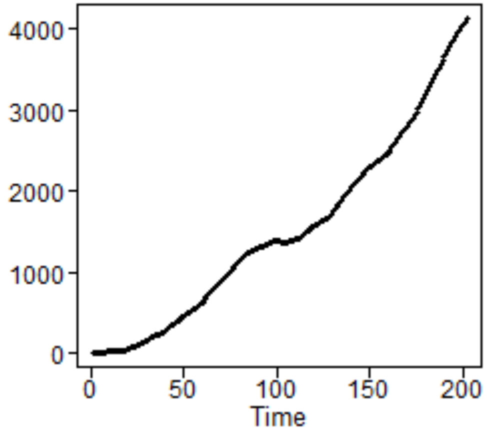
Question:

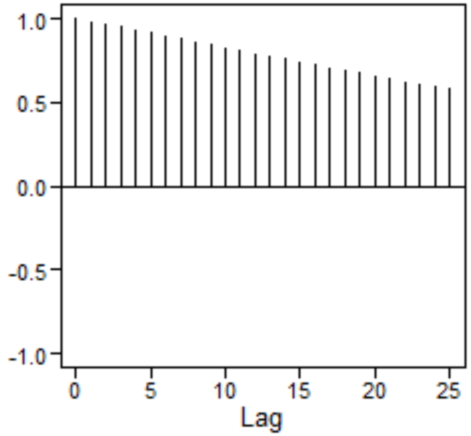
How do you use overfitting to detect ***more than one unit root?***

A Model with 2 Unit Roots

$$(1-B)^2(1-1.2B+.6B^2)X_t = a_t$$

Original data **Sample autocorrelations**






```
# generate ARIMA(2,2,0) data as before
x=gen.arima.wge(n=200,d=2,phi=c(1.2,-.6),sn=132)
# fit an AR(8) and AR(10) to this realization
est.ar.wge(x,p=8,type='burg')
est.ar.wge(x,p=10,type='burg')
```

Overfit Tables for ARIMA(2,2,0) Data

AR(8) fit

Factor	Abs recip	System freq
$1-1.9919B+0.9924B^2$	0.9962	0.0036
$1-1.2575B+0.6987B^2$	0.8359	0.1145
$1-0.8229B$	0.8229	0.0000
$1+0.1891B+0.2798B^2$	0.5290	0.2786
$1+0.4792B$	0.4792	0.5000

AR(10) fit

$1-1.9903B+0.9909B^2$	0.9954	0.0036
$1-0.8684B$	0.8684	0.0000
$1-1.3414B+0.7528B^2$	0.8676	0.1094
$1-0.1407B+0.4330B^2$	0.6581	0.2330
$1+1.0699B+0.4040B^2$	0.6356	0.4092
$1-0.1267B$	0.1267	0.0000

No Factors near 1-*B* ?

Let's examine AR(8) fit further

Factor	Abs recip	System freq
$1-1.9919B+0.9924B^2$	0.9962	0.0036

This factor

- Has roots very close to the unit circle
- Has System Frequency $f_0 = .0036$
  - Associated with period $1/.0036 = 278$
  - This period is longer than the data length (n = 200)
  - Suggests aperiodic data or data with a very long period
- Is very close to $1-2B+B^2 = (1-B)^2$

Suggests two factors of $(1-B)$

Note: AR(10) fit has a similar factor

Recall for ARIMA(2,2,0) Data

We previously analyzed these data and we decided to difference the data twice.

Using overfitting:

- We decide to difference the data twice because of the comments on the previous slide.
- We come up with the same final model as in our previous analysis of this realization.

Note: We did this analysis by looking at factor tables.

- It is ***always*** a good idea to look at the data
  - Before and after differencing, etc.

DataScience@SMU

ARIMA |
Comments about Unit Roots

Comments about Unit Roots (ARIMA Models)

- We mentioned that the decision whether to include a unit root or other nonstationary component in your model is a difficult one.
- We argued that it is a conscious decision that must be made by you.
- We next discuss the “Dicky-Fuller tests” which are popular in Economics.
- These tests are appealing because they seem to “make this decision for you.”

DataScience@SMU

Dicky-Fuller Test

Testing for Unit Roots

- Popular in Economics
- These are formal tests designed to help you decide whether to include one or more factors of $(1 - B)$ in your model, i.e. a root of +1
- They involve a test of the hypothesis

  H_0: *model has a root of* $+1$

  H_A: *the model does not have a root of* $+1$
- These are often called Dickey-Fuller tests

A Problem with the Dicky-Fuller Tests

- A problem with these tests is that a decision to include a unit root in the model is based on a decision to **not reject** the null hypothesis!
    - As data scientists you know that failure to reject a null hypothesis does not mean that it is true!
- The following mini-simulation will illustrate this point
- Using function **`adf.test`** in CRAN package **`tseries`**

Mini-Simulation

- 10 realizations were generated from the ***stationary model*** $(1-\varphi_1 B)X_t = a_t$
  - For $\varphi_1 = .9$, $.95$, and $.975$
  - Realization lengths n=100 and 200
- In each case we record the number of times out of 10 that the Dicky-Fuller test ***failed to reject*** the null hypothesis
  - i.e., ***incorrectly decided there was a unit root***

	$\varphi_1 = .9$	$\varphi_1 = .95$	$\varphi_1 = .975$
n=100	9/10	9/10	10/10
n=200	5/10	6/10	8/10

Comments on Simulation Results

The simulations show that unit root tests do not give you a definitive decision concerning whether a root is:

- On the unit circle (nonstationary)

or

- Close to the unit circle (stationary)

Bottom line:

The decision to include a nonstationary factor in your model must be made by ***you***

- And will often involve physical considerations/domain knowledge

DataScience@SMU

Model ID for Strictly Seasonal Models

Model Identification for Seasonal Models

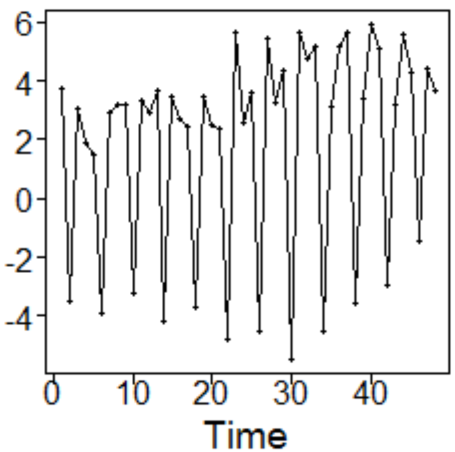
- In the previous slides we have discussed the issues related to model identification of ***ARIMA* models**
- Another type of nonstationary model is the ***seasonal model***, i.e. models that include a factor such as $(1 - B^s)$ for some integer s
  - Fitting seasonal models is the topic of the following material

Recall: Seasonal Models

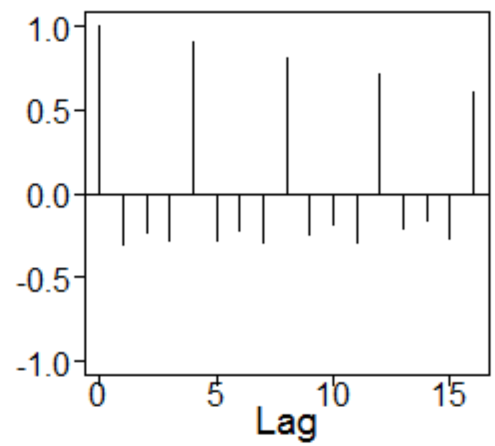
Pure seasonal models $\boxed{(1 - B^s)X_t = a_t}$

Example: $(1 \quad B^4)X_t = a_t$

Realization



Sample autocorrelations



The data go through 20 “irregular periods of 4” in 80 time points. The pattern seems to be 3 values “high” followed by a very low value.

Sample autocorrelations at lags 4, 8, … are ”large” which is consistent with the model.

- that is, X_t, X_{t+4}, X_{t+8}, … would be expected to be “similar”

DataScience@SMU

Model Identification for Seasonal Models

Procedure for including $1 - B^s$ in the model

As in the case of ARIMA models, ***the decision to include a seasonal factor*** in the model ***is one you will have to make*** after considering the evidence.

1. Look at the data
   - Does the behavior in realization tend to repeat every s time periods?
   - Are the sample autocorrelations large at s, 2s, 3s, etc.?
   - Does a seasonal model with season s make sense?

We recommend using factor tables

2. Compare factors in *overfit factor table* with those for $1 - B^s$ under consideration.
3. If "nonstationary" factors match with factors of $1 - B^s$ fairly well, transform the data by $1 - B^s$.

Recall: Seasonal Factor Table, s = 4

Factor	Abs recip	System freq
1−*B*	1	0.0
1+*B*	1	0.5
1+*B*^2	1	0.25

Factor table for seasonal data on previous slide

```
x=gen.aruma.wge(n=48,s=4,sn=23)
est.ar.wge(x,p=8,type='burg')
```

Factor	Abs recip	System freq
1+0.0148*B*+0.9826*B*^2	0.9912	0.2512
1+0.9905*B*	0.9905	0.5000
1−0.9556*B*	0.9556	0.0000
1−0.5126*B*+0.2985*B*^2	0.5464	0.1723
1+0.4753*B*	0.4753	0.5000
1+0.0593*B*	0.0593	0.5000

Factor tables with *p*=10 and *p*=12 are similar

Based on the Factor Tables

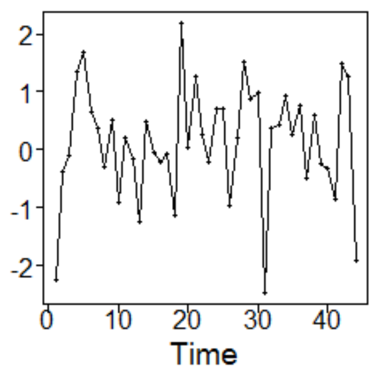
- It appears that the model contains a seasonal factor of the type $(1 - B^4)$.
- The next step is to transform the data to obtain

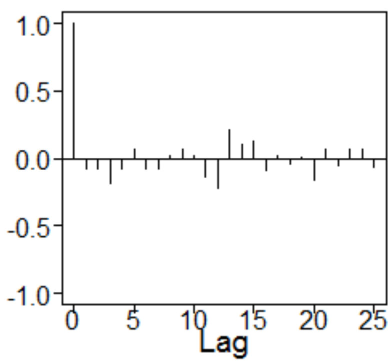
$$Y_t = (1 - B^4)X_t = X_t - X_{t-4}$$

- We can use the tswge command

```
y=artrans.wge(x,phi.tr=c(0,0,0,1))
```

- The transformed data and sample autocorrelations are shown below

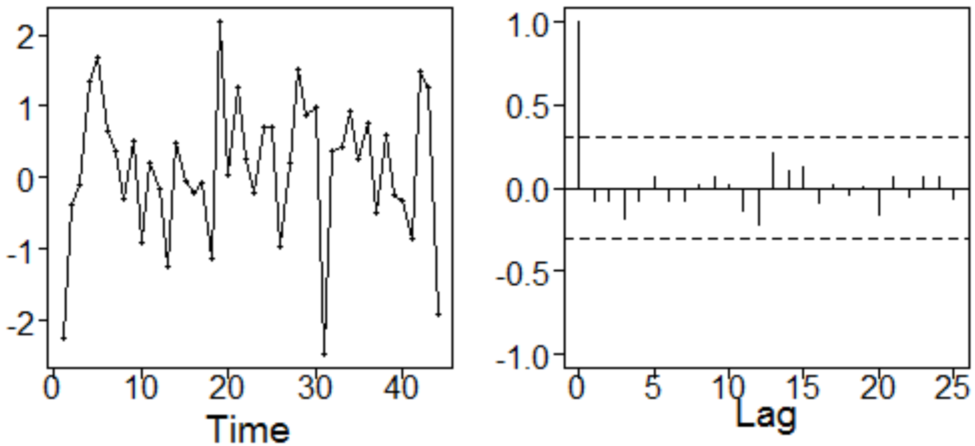




- The next step is to model the transformed data

Finding a Final Model

- Transformed data $Y_t = (1 - B^4)X_t = X_t - X_{t-4}$



- The transformed data appear white
  - Which is consistent with the true model
  - So the **`y`** variable obtained using **`artrans.wge`** is white noise. To find the white noise variance we use

    `wnv=sd(y)^2`

Final model $(1 - B^4)X_t = a_t \qquad \hat{\sigma}_a^2 = 1.006$

DataScience@SMU

Model ID for More General Seasonal Models

Fitting More General Seasonal Models

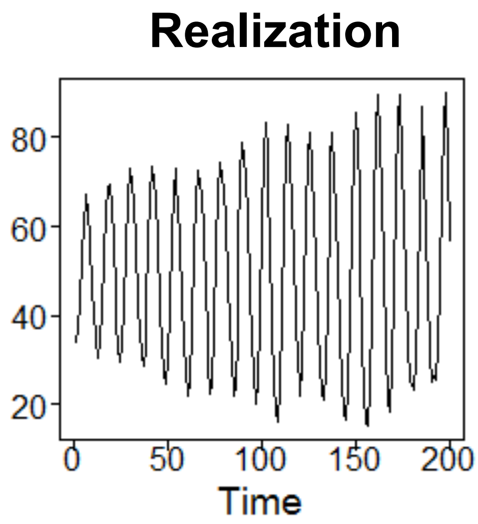
$$(1-B^s)\varphi(B)X_t = \theta(B)a_t$$

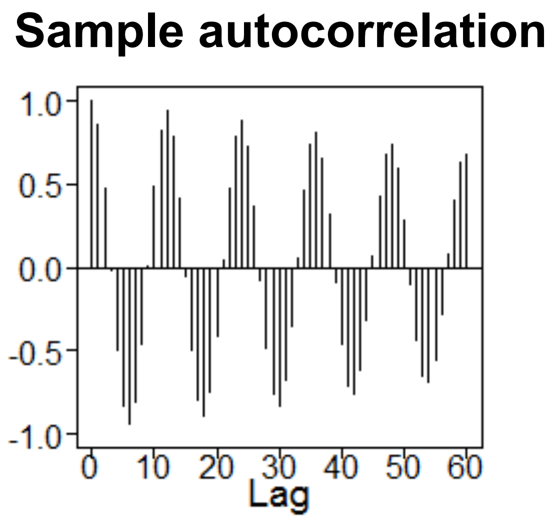
Seasonal factors $(1 - B^s)$ are a type of nonstationary factor. As such they tend to be dominant in factor tables of high order AR models fit to the data (overfitting).

- We have seen this in the previous "purely seasonal" case with *s*=4
- This dominance also tends to occur when stationary components are also in the data
  - As in the case of ARIMA models

Simulated Seasonal Data

$$(1 - B^{12})(1 - 1.5B + .8B^2)(X_t - 50) = a_t$$





Use tswge to simulate and plot the data above

```
x=gen.aruma.wge(n=200,s=12,phi=c(1.5,-.8),sn=87)
x=x+50
plotts.sample.wge(x,lag.max=60)
d15=est.ar.wge(x,p=15, type='burg')
```

Factor table for $(1 - B^{12})$

Factor	Abs recip	System freq
1+*B*+*B*^2	1	0.3333
1+1.732*B*+*B*^2	1	0.4167
1-*B*+*B*^2	1	0.1667
1+*B*	1	0.5000
1-1.732*B*+*B*^2	1	0.0833
1-*B*	1	0.0000
1+*B*^2	1	0.2500

Factor table for AR(15) fit to data on previous slide

```
d15=est.ar.wge(x,p=15, type='burg')
```

Factor	Abs recip	System freq
1-1.7269*B*+0.9978*B*^2	0.9989	0.0839
1+1.0027*B*+0.9948*B*^2	0.9974	0.3338
1-1.0036*B*+0.9944*B*^2	0.9972	0.1661
1-0.0142*B*+0.9907*B*^2	0.9953	0.2489
1+1.7320*B*+0.9904*B*^2	0.9952	0.4180
1+0.9788*B*	0.9788	0.5000
1-0.9522*B*	0.9522	0.0000
1-1.5043*B*+0.7732*B*^2	0.8793	0.0867
1+0.1523*B*	0.1523	0.5000

Factor Table for AR(17) Fit to Data on Previous Slide

```
d15=est.ar.wge(x,p=17, type='burg')
```

Factor	Abs recip	System freq
1-1.7276B+0.9985B^2	0.9992	0.0838
1-1.0028B+0.9955B^2	0.9977	0.1662
1+1.0010B+0.9936B^2	0.9968	0.3337
1-0.0157B+0.9930B^2	0.9965	0.2487
1+1.7329B+0.9899B^2	0.9949	0.4182
1+0.9836B	0.9836	0.5000
1-0.9136B	0.9136	0.0000
1-1.4286B+0.7747B^2	0.8802	0.0993
1-0.7137B	0.7137	0.0000
1+0.7516B+0.3366B^2	0.5801	0.3621

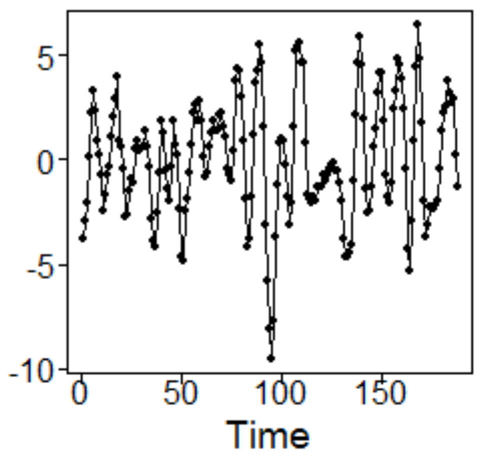
Note: The factor $1 - 0.9136B$ is not very close to the unit circle, but since all other factors are consistent with $(1 - B^{12})$ we conclude that this overfit table suggests $(1 - B^{12})$ also.

Based on Overfit Tables

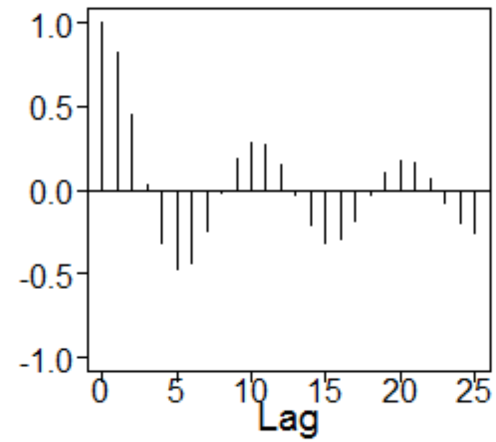
Transform data to create $Y_t = (1 \quad B^{12}) X_t$

```
y=artrans.wge(x,phi.tr=c(0,0,0,0,0,0,0,0,0,0,0,1))
```

Transformed data



Sample autocorrelations



The transformed data appear stationary, so we use AIC to identify a model.

Using tswge to Model the Transformed Data

We use `aic5.wge` to model the transformed data (y) on the previous slide

- When using AIC to model data that has been "stationarized" using the seasonal transform $(1 - B^s)$, it is good practice to allow range of *p* values to include *s*
  - To uncover any seasonal stationary information that might be in the data
  - In R code to follow, we consider the range *p*=0:13 and *q*=0:3

Using tswge to Model the Transformed Data

```
# y is the transformed data
aic5.wge(y,p=0:13,q=0:3)
```

AIC selects an ARMA(4,2) model. Factoring the ARMA(4,2) model we obtain:

AR factors

Factor	Abs recip	System freq
1-0.4098*B*+0.8135*B*^2	0.9019	0.2135
1-1.4733*B*+0.7771*B*^2	0.8815	0.0925

MA factor

Factor	Abs recip	System freq
1-0.4957*B*+0.9262*B*^2	0.9624	0.2085

Note: The dominant behavior of the transformed data is a pseudo cyclic behavior of length about 10 (f_0=.10). The factors associated with frequency about f_0=.2 essentially cancel.

Using tswge to Model the Transformed Data

Based on the comments on the previous slide, we look for a simpler model using BIC.

```
aic5.wge(y,p=0:13,q=0:3,type='bic')
```

BIC picks a simpler AR(2) model. This is consistent with

- The pseudo cyclic data
- Damping cyclical sample autocorrelations

Decision:

We choose to model the transformed data as an AR(2).

Final Model

Based on the decision to fit an AR(2) model, we use the **est.ar.wge** command to obtain ML estimates.

```
est.ar.wge(y,p=2)
#  $phi:  [1] 1.4652837 -0.7596714
#  $avar: [1] 1.036377
mean(x)   #[1] 49.77867
```

So, the final model is

$$(1 - B^{12})(1 - 1.47B + .76B^2)(X_t - 49.78) = a_t \qquad \hat{\sigma}_a^2 = 1.04$$

which is very similar to the model from which the data were simulated:

$$(1 - B^{12})(1 - 1.5B + .8B^2)(X_t - 50) = a_t \qquad \sigma_a^2 = 1$$

DataScience@SMU

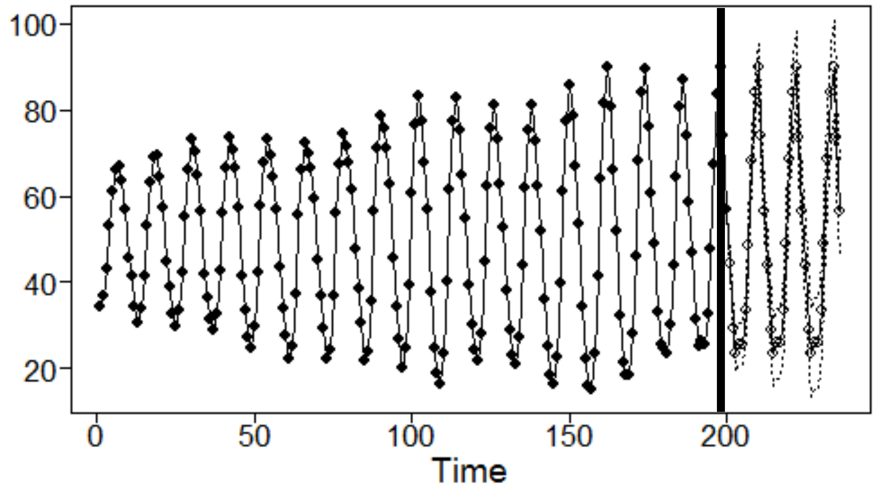
Review of Forecasting with Seasonal Models

Forecasting with the Seasonal Model

$$(1 - B^{12})(1 - 1.47B + .76B^2)(X_t - 49.78) = a_t \qquad \hat{\sigma}_a^2 = 1.04$$

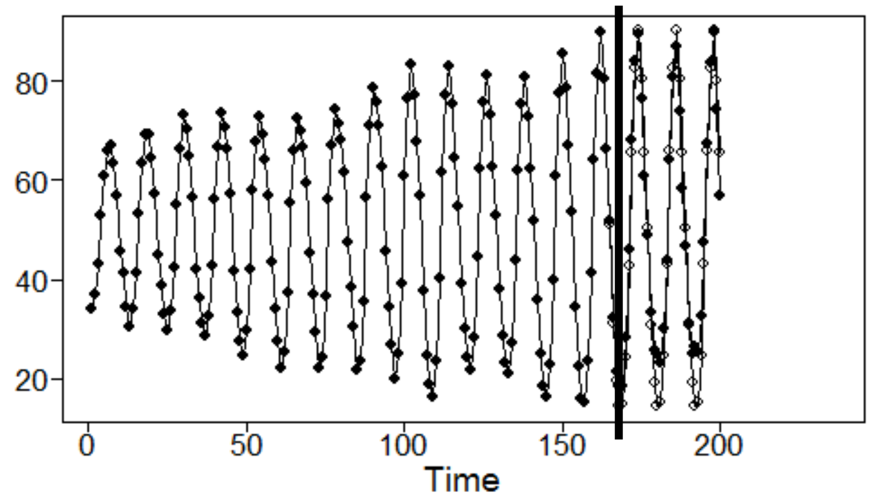
Forecasts: 36 steps ahead

```
x=gen.aruma.wge(n=200,s=12,phi=c(1.5,-.8),sn=87)
x=x+50
fore.aruma.wge(x,s=12,phi=c(1.47,-.76),n.ahead=36,
lastn=FALSE)
```



Forecasts: Last 36 Values

```
fore.aruma.wge(x,s=12,phi=c(1.47,-.76),n.ahead=36,
lastn=TRUE,limits=FALSE)
```



ASE = 26.27

Summary:

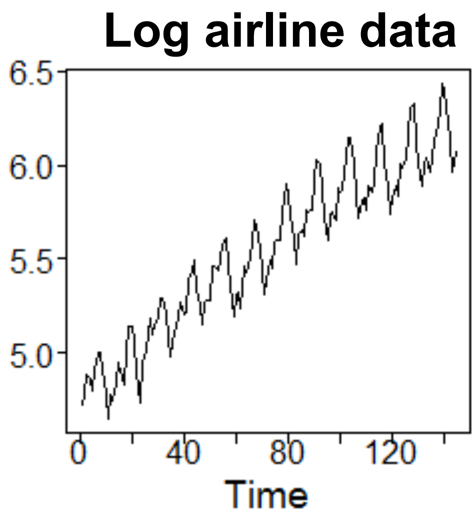
- The forecasts (open circles) are very close to the true values.
- We were able to fit a model very close to the true model.
- Forecasts (ahead and last 36) are quite good.

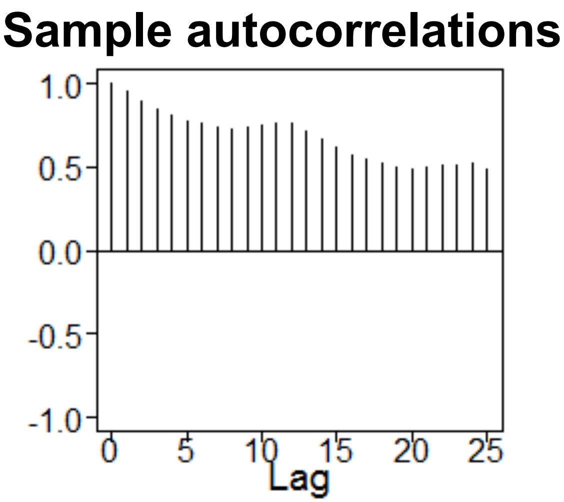
DataScience@SMU

Seasonal Models | Example

Airline Data

Example: Airline Data





Again

- There is evidence of nonstationarity
- We apply the overfit procedure using **`est.ar.wge`**

Factor table for $(1 - B^{12})$

Factor	Abs recip	System freq
1+B+B^2	1	0.3333
1+1.732B+B^2	1	0.4167
1-B+B^2	1	0.1667
1+B	1	0.5000
1-1.732B+B^2	1	0.0833
1-B	1	0.0000 ?
1+B^2	1	0.2500

Factor table for AR(15) fit to log airline data

```
est.ar.wge(lair,p=15,type='burg')
```

Factor	Abs recip	System freq
1-1.7253B+0.9953B^2	0.9976	0.0838
1-0.9861B+0.9950B^2	0.9975	0.1677
1+0.9869B+0.9850B^2	0.9925	0.3328
1-0.0375B+0.9788B^2	0.9893	0.2470
1-1.9697B+0.9704B^2	0.9851	0.0034 ?
1+1.6891B+0.9504B^2	0.9749	0.4168
1+0.8540B	0.8540	0.5000
1+0.6000B	0.6000	0.5000
1-0.1110B	0.1110	0.0000

Notes

- Factor tables for other high orders are similar
- The factor $1-1.9697B+0.9704B^2$ is associated with a frequency of $f_0 = 0.0034$ or a period of $1/.0034 = 294$ (longer than the data record). This suggests aperiodic data or a very long period.
- Also this factor is very close to $1-2B+B^2 = (1-B)^2$ which is associated with frequency $f_0 = 0$
    - We encountered a similar situation with an ARIMA model with $d=2$.
    - $(1-B)^2$ provides the last "piece" needed for a factor of $(1-B^{12})$ plus an extra $(1-B)$ factor
    - That is, the factor table suggests the presence of nonstationary factors $(1-B)(1-B^{12})$
- Although $(1+.85B)$ is not as close to the unit circle "as we would expect for $s=12$ data, the "total picture" suggests a factor of $(1-B^{12})$

In the log airline factor table

$(1-B)^2$ provides the last "piece" needed for a factor of $(1 - B^{12})$ plus an extra $(1-B)$ factor

Recall:

We earlier discussed the following model that is useful for modeling ***seasonal data with a trend***

$$(1 \quad B)(1 \quad B^{s}) \quad (B)(X_t \qquad) = \quad (B)a_t$$

We referred to it as the ***airline model***

- For obvious reasons!

Note:

The airline model allows for ***stationary components***

- To find these, we transform to find

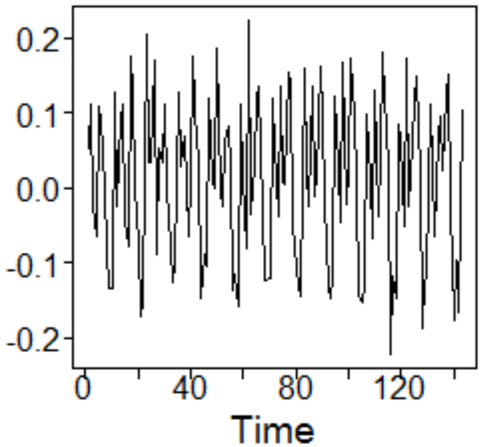
$$Y_t = (1 - B)(1 - B^{12})X_t$$

Transform to Stationarity

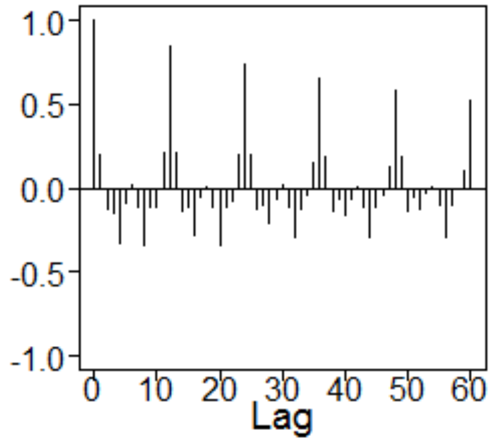
Using **tswge** to transform log airline data (lair) to remove seasonal and ARIMA components

```
#Difference the data
d1=artrans.wge(lair,phi.tr=1)
```

Differenced data (d1)



Sample autocorrelations

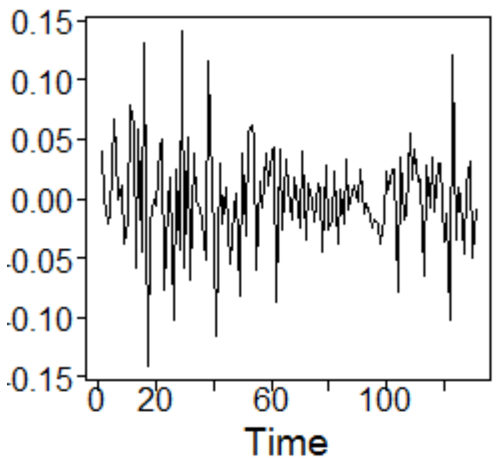


- Trend has been removed
- 12 month seasonal behavior remains

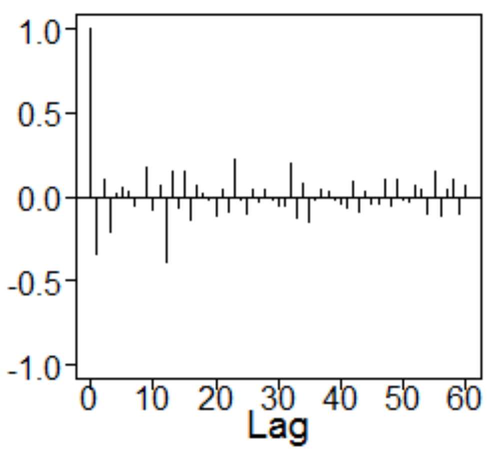
Differenced Data Transformed by (1-B^{12})

```
# Transform differenced data with "seasonal difference"
d1.12=artrans.wge(d1,phi.tr=c(0,0,0,0,0,0,0,0,0,0,0,1))
```

(1-*B*)(1-*B*^{12})X_t (d1.12)



Sample autocorrelations



- Appears to be stationary
- No trending or seasonality present
- We will find a stationary model for this realization

Finding a Final Model for the (log) Airline Data

Use `tswge` to model the transformed data, `d1.12`

```
aic5.wge(d1.12,p=0:13,q=0:3)
```

AIC picks an ***ARMA(12,1).***

- In order to see if a lower order model could satisfactorily model the data, we use BIC

```
aic5.wge(d1.12,p=0:13,q=0:3,type='bic')
```

BIC picks an ***MA(1)*** as the first choice and an ***AR(1)*** as second choice.

- Examining the data and sample autocorrelations, neither model seems appropriate
- We decide to use the ARMA(12,1) model chosen by AIC

Final Model for (log) Airline Data

We estimate the parameters of the ARMA(12,1) model using est.arma.wge

```
est.arma.wge(d1.12,p=12,q=1)
# $phi  [1]  0.00846435  0.07967002 -0.10710183 -0.02069279
0.08039524  0.04050674 -0.05516681  0.03629598  0.13277817
[10] -0.05297698 -0.12340975 -0.40343184
# $theta  [1] 0.4536856
# $avar  [1] 0.001310444
mean(lair)   # [1] 5.542176
```

$(1-B)(1-B^{12})\varphi_{12}(B)(X_t - 5.54) = (1-.45B)a_t \;\; \hat{\sigma}_a^2 = .0013$

$\varphi_{12}(B)$=$1 - .008B - .080B^2 + .107B^3 + .021B^4 - .080B^5 - .041B^6$
$+.055B^7 - .036B^8 - .133B^9 + .053B^{10} + .0123B^{11} + .403B^{12}$

Comments

- Recall from an earlier unit that forecasts using the airline model are quite good
  - We previously used the AR(13) (Woodward / Gray Model) instead of the ARMA(12,1) (Box Model) but forecasts are very similar
- Recall also that $(1-B)(1-B^{12})$ contains two factors of $(1-B)$ which accounts for the fact that the trend in the airline data is predicted to continue
- It is clear that seasonal models are useful for data that occur monthly or quarterly or some other sampling interval where similar patterns are likely to be repeated

DataScience@SMU

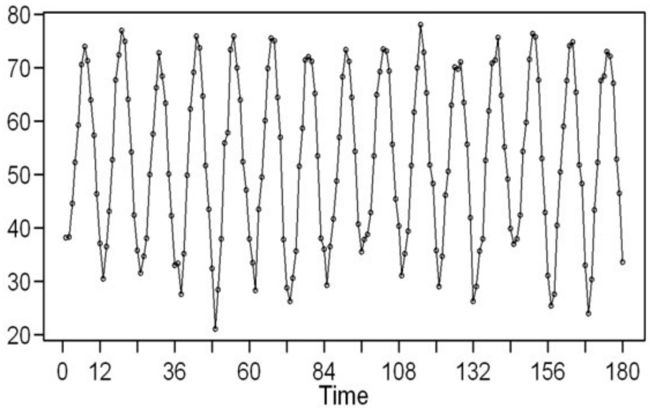
Seasonal Models | Example

Pennsylvania Temperature Data

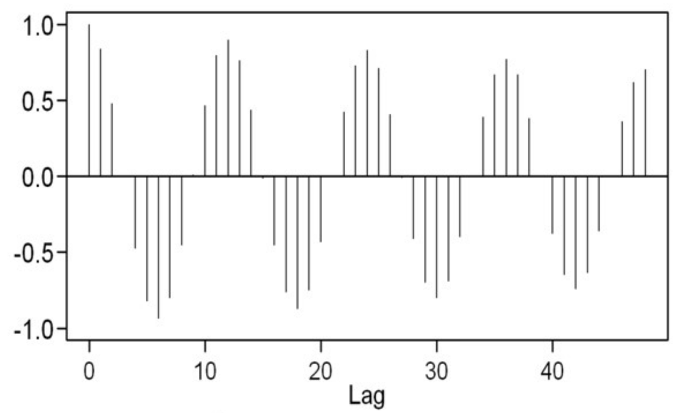
Pennsylvania Monthly Temperature Data

We conclude this section by analyzing another monthly time series

- Monthly average temperatures for Pennsylvania over a 15 year period



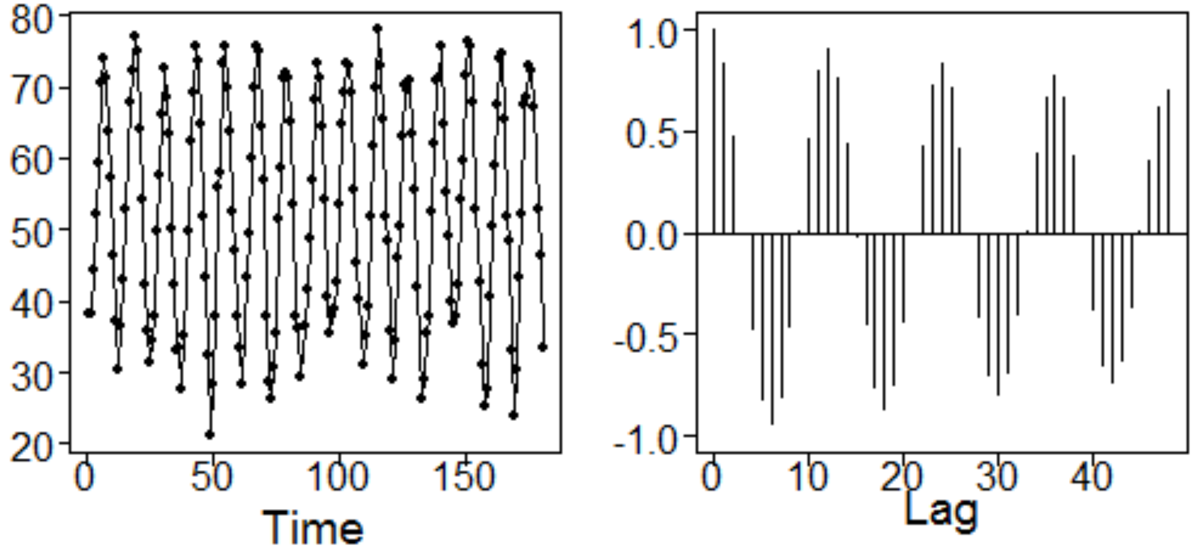
(a) Monthly Pennsylvania temperature data



(b) Sample autocorrelations for (a)

Again! Use overfit procedure

Pennsylvania Monthly Temperature Data



Note:

- There certainly seems to be a “seasonal” pattern
- Autocorrelations damp slowly
- ***Again! Use overfit procedure***

Factor Tables for p=14 and 15
Overfit to Pennsylvania Temperature Data (patemp)

```
data(patemp)  # the data set is available in tswge
est.ar.wge(patemp,p=14,type='burg')
```

Factor	Abs recip	System freq
1-1.7267B+0.9971B^2	0.9985	0.0838
1-0.9007B	0.9007	0.0000
1+0.1854B+0.8005B^2	0.8947	0.2665
1-0.8397B+0.7441B^2	0.8626	0.1691
1+0.9827B+0.6948B^2	0.8335	0.3503
1+1.5999B+0.6922B^2	0.8320	0.4557
1+0.4007B	0.4007	0.5000
1-0.3283B	0.3283	0.0000
1+0.1672B	0.1672	0.5000

```
est.ar.wge(patemp,p=15,type='burg')
```

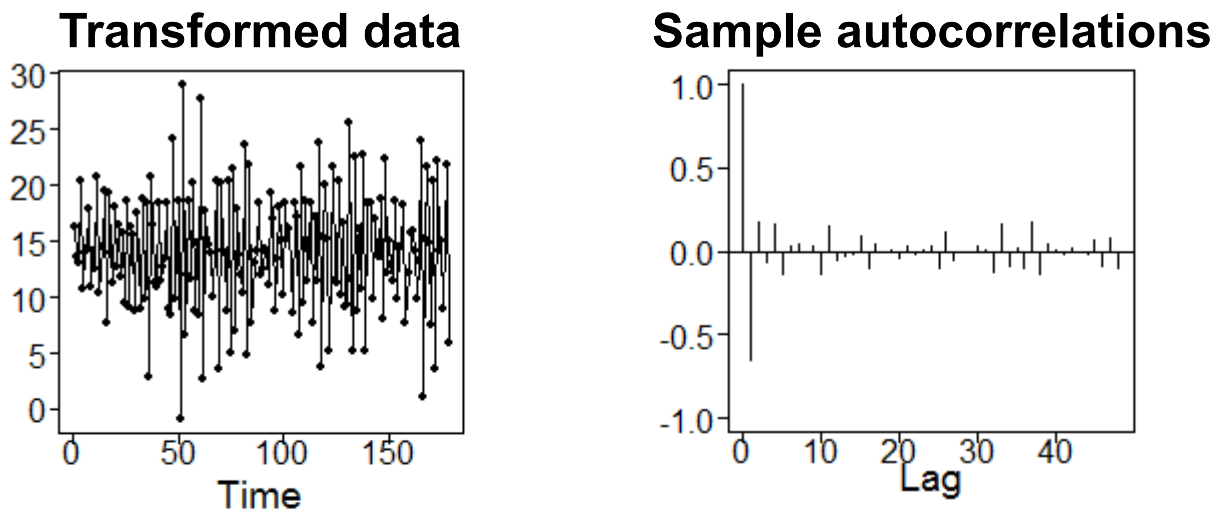
Factor	Abs recip	System freq
1-1.7268B+0.9973B^2	0.9987	0.0838
1-0.8743B	0.8743	0.0000
1-0.8987B+0.7623B^2	0.8731	0.1640
1+0.2200B+0.7559B^2	0.8694	0.2702
1+1.0663B+0.7499B^2	0.8660	0.3556
1+1.6060B+0.7278B^2	0.8531	0.4452
1+0.7830B	0.7830	0.5000
1+0.0141B+0.4368B^2	0.6609	0.2517
1-0.6487B	0.6487	0.0000

Factor Tables for Pennsylvania Temperature Data

- These factor tables are ***not what we expected***
  - We expected to see the factors of $(1 - B^{12})$ approximated by factors with roots very close to the unit circle
  - This is what we saw in the previous two seasonal examples
- The only factor of $(1 - B^{12})$ that has the behavior we expected is $(1 - 1.732B + B^2)$
  - Each factor table had a factor very close to this one with roots close to the unit circle
- This suggests that the stationarizing transformation is $Y_t = (1 - 1.732B + B^2)X_t$

We Transform the Data by $Y_t = (1 - 1.732B + B^2)X_t$

```
y.tr=artrans.wge(patemp,phi.tr=c(1.732,-1))
```



- The transformed data appear stationary
- Using aic5.wge with p=0:13, q=0:3, an AR(9) is selected
- Using aic5.wge with the same range for p and q and using BIC, an AR(3) is selected
- Let's go with the simpler model

Final Model

Estimating the parameters of the AR(3) model we get

```
est.ar.wge(y.tr,p=3)
#  $phi  [1] -1.1425939 -0.8431575 -0.4144801
# $avar  [1] 10.76795
mean(patemp)   #  [1] 52.62667
```

$$(1 - 1.732B + B^2)(1 + 1.14B + .84B^2 + .41B^3)(X_t - 52.63) = a_t,$$
$$\hat{\sigma}_a^2 = 10.77$$

Note:

The above model is an example of what Woodward, et al., 2017, refer to as an ARUMA model

- It is a more general nonstationary model that can contain roots on the unit circle that are not necessarily +1
- Seasonal models are ARUMA models

Notes

- The issue here is that the seasonal behavior is very nearly sinusoidal
  - It can be explained with a second order factor (with roots on the unit circle and a cycle length of 12)
  - The other seasonal data sets, for example the airline data, had a non-sinusoidal pattern within the 12 month year
    - For the airline data November and December are high and January is low, etc.
    - A pattern, but not sinusoidal
- A signal-plus-noise with a sine or cosine signal might also be a good model for these data

Key Point

Do not include $1-B^{12}$, for example, in the model simply because there is a 12^{th} order nonstationarity or because the data show a period of 12

- Or just because you have monthly data

Check the factor tables!!

DataScience@SMU

Signal + Noise Models | Testing for Trend

OLS Method

Fitting Signal-Plus-Noise Models

In our discussion of fitting nonstationary models to data we have considered

- ARIMA models
- Seasonal models

We next address the topic of fitting deterministic ***signal-plus-noise models*** to data

Deterministic Signal-Plus-Noise Models

$$\boxed{X_t = s_t + Z_t}$$

s_t is a deterministic signal

Z_t is a zero-mean, stationary process

Example signals: $s_t = a + bt$

$s_t = a + bt + ct^2$

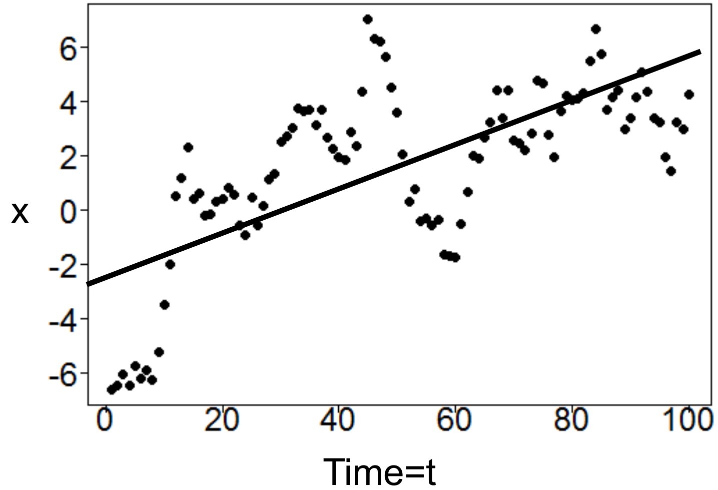
$s_t = A\cos(2\pi ft + C)$, C constant

Testing for Linear Trend

Recall: It is difficult to distinguish between random trends and deterministic trends.

- We will look at this issue in more detail in this section

Question: Is there a deterministic linear trend component in the data below?



Note:

It looks like a linear regression problem.

Linear Regression Strategy to Test for Trend

- Let *t* = Time be the independent variable
- Let *x* = X_t be the dependent variable
- Consider the standard regression model
  $X_t = a + bt + Z_t$
- Conclude there is a trend if $H_0: b = 0$ is rejected.

Results:

Base R program **`lm`** can be used for regression. If

- *x* is the vector of length 100 plotted on vertical axis
- t=1:100

The R code:

```
re=lm(formula=x~t)
summary(re)
```

produces (among other things) the regression model

$$X_t = -2.082 + .073t + Z_t$$

and $H_0: b = 0$ is rejected with $p < .001$.

Conclusion

There is clearly a significant (upward) trend

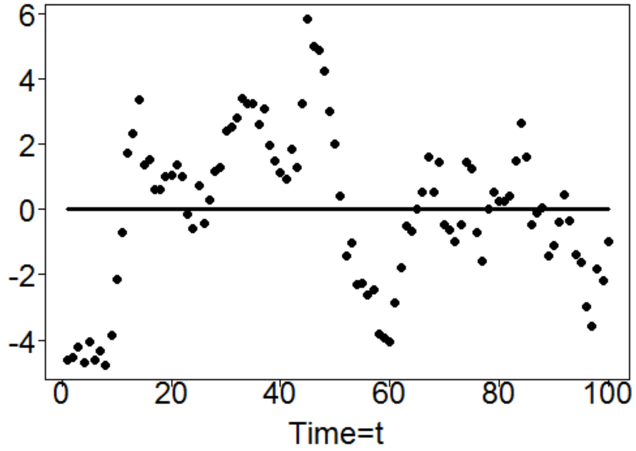
Check residuals

As good data scientists, we know to check the residuals to see if they are ***uncorrelated.***

Using the R code

```
z=x -(2.082+.073*t)
```

we obtain the residual plot below:



Oops!

Obviously the Residuals Are Correlated

Question:

Does that really make a difference? (Or is that just something statisticians worry about?)

Note:

The data set in the previous analysis satisfies the model $X_t = 0 + 0 * t + Z_t$ where Z_t is a realization of length n=100 from the AR(1) model $Z_t - .95Z_{t-1} = a_t$

That is:

$H_0: b = 0$ is ***True*** (and we rejected it dramatically!)

Was This Just a Fluke?

- We expect to reject $H_0: b = 0$ about 5% of the time when $b = 0$.
- Maybe this is just one of those 5%
- In order to check this we have run a Mini-Simulation
  - We simulate several realizations from models $X_t = 0 + 0 * t + Z_t$ where Z_t is from AR(1) models like $Z_t - .95Z_{t-1} = a_t$
    - How unusual is it to reject $H_0: b = 0$ (which is true in these simulated realizations)?

DataScience@SMU

Mini-Simulation

Type 1 Error Rate

Mini-Simulation: Using Regression

- We consider the model
  $X_t = a + bt + Z_t$ where $(1 - \varphi_1 B)Z_t = a_t$ ***where b=0***
  - We test at the .05 level of significance
- 10 realizations were generated from the model above
  - For $\varphi_1 = .9,\ .95.\ .975$
  - Realization lengths n=100 and 200
- In each case we record the number of times out of 10 we ***(incorrectly)*** rejected $H_0{:}\, b = 0$

	$\varphi_1 = .9$	$\varphi_1 = .95$	$\varphi_1 = .975$
n=100	8/10	8/10	9/10
n=200	5/10	8/10	7/10

Realizations from $X_t = 0 + 0 * t + Z_t$ where Z_t satisfies $Z_t - .95Z_{t-1} = a_t$

Obviously, $H_0: b = 0$ is true in the above model.

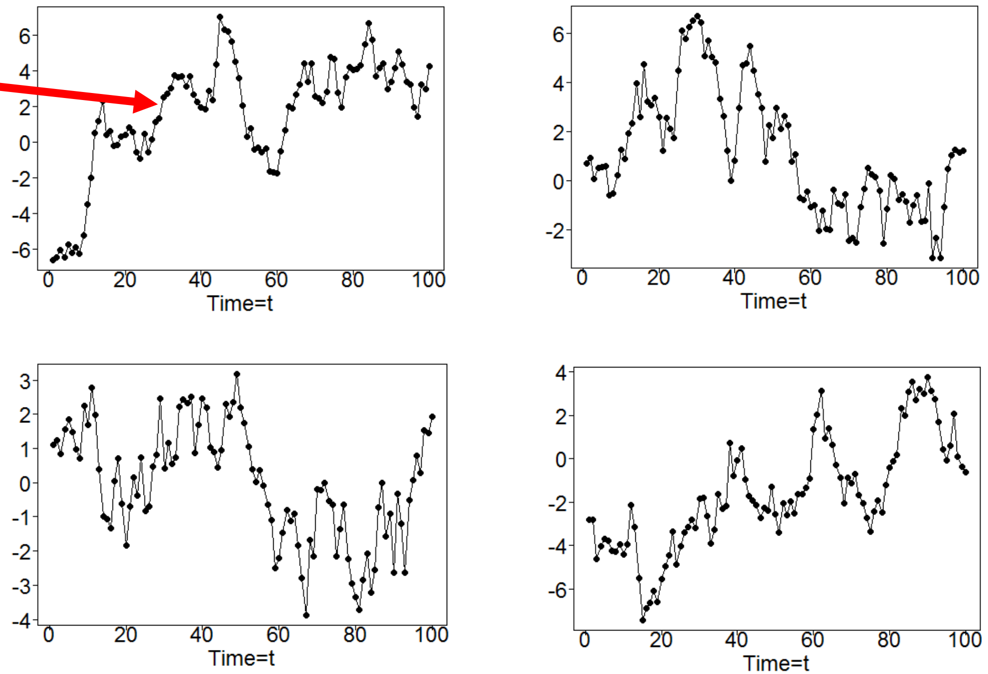
- These realizations show "random trends" that are difficult to distinguish from "deterministic trends"
- And cause $H_0: b = 0$ to be incorrectly rejected

Realization we previously analyzed and rejected

$H_0: b = 0$

with $p < .001$.

We'll come back to this



Comments on Simulation Results

The simulations show that using standard regression techniques to test $H_0: b = 0$ with AR(1) residuals

- Where φ_1 is close to +1
- Ignoring the fact that the residuals are correlated

Results in the null hypothesis are being rejected ***much more than 5% of the time.***

Bottom line

This is a really bad way to test for trend in time series data

- You will detect a (deterministic) trend in many cases in which no such trend exists
- The test is picking up on the random trends in AR(1) data

Caution:

When we have tests with inflated “observed significance rates”

- Then if you find a significant deterministic trend, you don’t have “confidence” that the trend is actually deterministic
- The next example illustrates the problems this can cause

DataScience@SMU

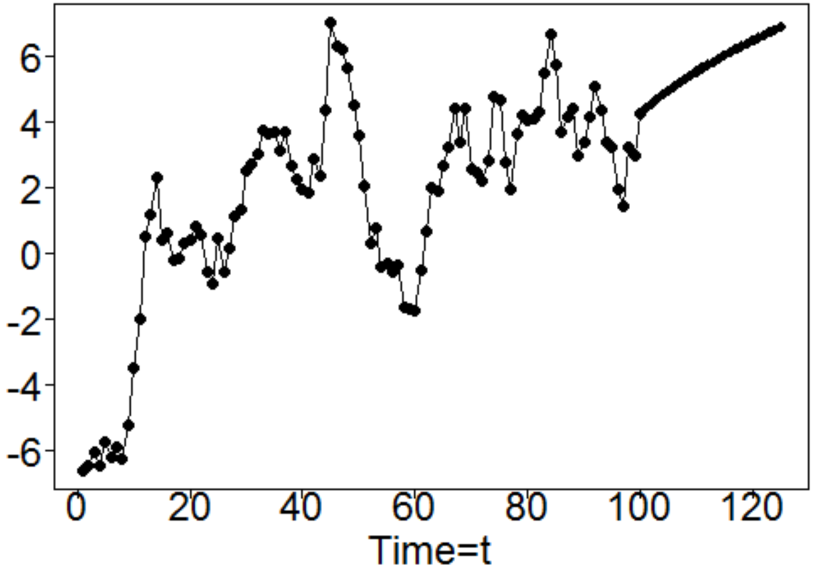
Forecasting and False Trend

True model: $X_t = 0 + 0 * t + Z_t$ **where** Z_t **satisfies** $Z_t - .95Z_{t-1} = a_t$

We fit the model: $X_t = -2.082 + .073t + Z_t$

($H_0: b = 0$ **is rejected with** $p < .001$.)

Forecasts vs. actual next 25 data values from "random trend" model



Note: The trend didn't continue as predicted by deterministic trend model!

oops again!!

Clearly We Need a Better Way to Test for Trend!

Estimation procedures exist for deciding whether a trending behavior in data is due to either a

- Deterministic component

  or

- Random/correlation driven component

  or both

These include:

- Cochrane-Orcutt
- ML techniques
- Bootstrap techniques
- ...

We illustrate the ***Cochrane-Orcutt*** method next

DataScience@SMU

Cochrane-Orcutt Method

Cochrane-Orcutt Method

Consider: $X_t = a + bt + Z_t$ where $(1 - \varphi_1 B)Z_t = a_t$.

1. Fit a linear regression to the original data and find estimates $\hat{a}$ and $\hat{b}$.
2. Find $\hat{Z}_t = X_t - \hat{a} - \hat{b}\mathrm{t}$ which should be approximately AR(1).
3. Estimate $\hat{\varphi}_1$ by fitting an AR(1) model to $\hat{Z}_t$
4. Calculate $\hat{Y}_t = (1 - \hat{\varphi}_1 B)X_t, \quad t = 2, \ldots, n$
5. Note that $\hat{Y}_t = (1 - \hat{\varphi}_1 B)X_t = (1 - \hat{\varphi}_1 B)(a + bt + Z_t)$

$$= a(1 - \hat{\varphi}_1) + bt - b\hat{\varphi}_1(t - 1) + (1 - \hat{\varphi}_1 B)Z_t$$
$$= c + bt_{\hat{\varphi}_1} + g_t, \text{ where } t_{\hat{\varphi}_1} = t - \hat{\varphi}_1(\mathrm{t} - 1).$$

where $\mathrm{c} = a(1 - \hat{\varphi}_1)$, $t_{\hat{\varphi}_1} = t - \hat{\varphi}_1(\mathrm{t} - 1)$, and
$g_t = (1 - \hat{\varphi}_1 B)Z_t$

light board

Why did we do that?

Logic behind the Cochrane-Orcutt Method

We transformed: $X_t = a + bt + Z_t$ where $\boxed{(1 - \varphi_1 B)Z_t = a_t.}$

To the new model: same slope

$\boxed{\Rightarrow Z_t = (1 - \varphi_1 B)^{-1} a_t}$

$\hat{Y}_t = (1 - \hat{\varphi}_1 B)X_t = c + b\, t_{\hat{\varphi}_1} + g_t$, where

$c = a(1 - \hat{\varphi}_1)$, $t_{\hat{\varphi}_1} = t - \hat{\varphi}_1(t - 1)$, and $\boxed{g_t = (1 - \hat{\varphi}_1 B)Z_t}$

?

new dependent variable

Let's look at g_t

new constant

new independent variable

$g_t = (1 - \hat{\varphi}_1 B)Z_t$

$$g_t = (1 - \hat{\varphi}_1 B)Z_t = (1 - \hat{\varphi}_1 B)\boxed{(1 - \varphi_1 B)^{-1} a_t} \leftarrow \boxed{Z_t}$$

The two factors: $(1 - \hat{\varphi}_1 B)$ and $(1 - \hat{\varphi}_1 B)$ nearly cancel each other out

- i.e. $g_t \approx a_t$ so g_t should be "nearly uncorrelated"

Finally: Test $H_0: b = 0$ in this "new" model

- Using standard regression methods

light board

Cochrane-Orcutt Test

Test H_0: $b = 0$ using the test statistic

$$t_{\text{co}} = \frac{\hat{b}_{\text{co}}}{\widehat{SE}(\hat{b}_{\text{co}})}$$

where $\hat{b}_{\text{co}}$ is the least squares (regression) estimate of slope b in the "new" model, and $\widehat{SE}(\hat{b}_{\text{co}})$ is its usual standard error.

Note:

On the next slide we discuss the performance of the Cochrane-Orcutt method on simulated data

light board

DataScience@SMU

Cochrane-Orcutt in R (Orcutt Package)

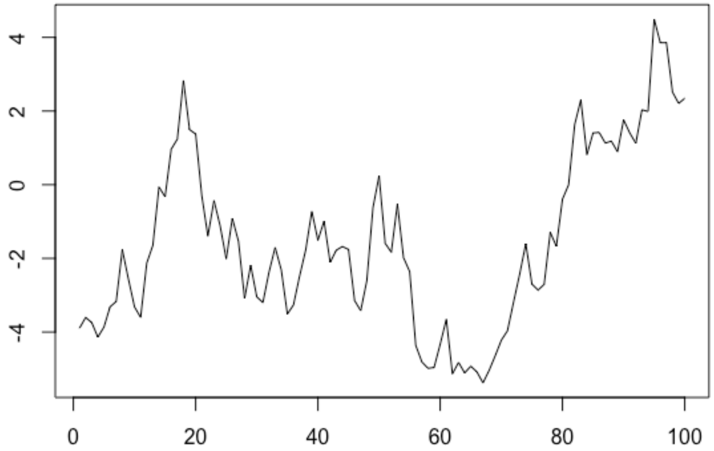
Cochrane-Orcutt Test in R (in Orcutt Package)

```
install.packages("orcutt")
library(orcutt)

x = gen.sigplusnoise.wge(100, b0 = 0, b1= 0, phi= .95, sn = 21)

t = seq(1,100,1)
df = data.frame(x = x, t= t)
fit = lm(x~t, data = df)
summary(fit)

cfit = cochrane.orcutt(fit)
summary(cfit)
```




```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.078211   0.450925  -6.826 7.31e-10 ***
t            0.030428   0.007752   3.925 0.000161 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.360341   2.811634  -1.195   0.2349
t            0.041244   0.040656   1.014   0.3129
```

Mini-Simulation: Using Cochran-Orcutt

- We consider the model
  $X_t = a + bt + Z_t$ where $(1 - \varphi_1 B)Z_t = a_t$ ***where b=0***
  - We test at the $.05$ level of significance
- 10 realizations were generated from the model above
  - For $\varphi_1 = .9,\ .95,\ .975$
  - Realization lengths n=100 and 200
- In each case we record the number of times out of 10 we ***(incorrectly)*** rejected $H_0: b = 0$

	$\varphi_1 = .9$	$\varphi_1 = .95$	$\varphi_1 = .975$
n=100	2/10	4/10	4/10
n=200	1/10	2/10	2/10

Note: Results are much better than those using simple regression

- Still higher than 5%   ??

More Extensive Cochrane-Orcutt Simulation Results

$Y_t = a + bt + Z_t$ where $(1 \quad {}_1B)Z_t = a_t$

- $b = 0$ (i.e. null hypothesis of zero slope is true)
- 1000 replicates generated from each model

Observed significance levels for tests $H_0 : b = 0$

n	${}_1 = .8$	${}_1 = .9$	${}_1 = .95$
50	18.4	27.2	37.2
100	16.0	20.0	28.4
250	8.0	12.4	17.6
1000	4.8	8.4	9.6

Conclusion: Cochrane-Orcutt has significance levels inflated over 5%

Comments about Inflated Rejection Rates

Recall:

Type I error rate: Probability of rejecting the null when it is true.

- We typically run hypothesis tests controlling the Type I error rate to be small (commonly $\alpha = .05$)
- We want there to be a small chance of rejecting the null *when it is true*
  - In our case we want there to be a small chance of declaring there is a deterministic trend ($b \neq 0$) when it is actually zero
  - i.e. when the only trending in the data is correlation-based random trends

Testing for Trend: Summary

$$X_t = a + bt + Z_t$$

- It's a difficult problem when Z_t has a positive correlation structure
  - That produces "random trends"
  - And are difficult to distinguish from "deterministic trends"
- Ignoring the correlation structure in the residuals and running a ***standard regression analysis is a very bad idea***
- The Cochrane-Orcutt technique helps adjust for the correlation structure but still rejects the null hypothesis more often than it should
- Other methods such as ML (used by SAS) have similar problems with inflated significance levels
  - See Woodward, Bottone, and Gray (1993) for more discussion of these issues

DataScience@SMU

Bonus Material

Bootstrap Method

Woodward, Bottone, and Gray (1997). *Journal of Agricultural, Biological, and Environmental Statistics (JABES)*, 403-416.

These authors devised a method for improving the observed significance levels

- Based on a bootstrap = based technique

Bivin: let's talk

Simulation Results

$Y_t = a + bt + Z_t$ where $(1 \quad {}_1B)Z_t = a_t$

- $b = 0$ (i.e. null hypothesis of zero slope is true)
- 1000 replicates generated from each model

Observed significance levels for tests $H_0 : b = 0$

n	${}_1 = .8$		${}_1 = .9$		${}_1 = .95$	
50	18.4	*5.9*	27.2	*8.1*	37.2	*10.0*
100	16.0	*6.4*	20.0	*5.3*	28.4	*7.6*
250	8.0	*3.8*	12.4	*5.1*	17.6	*6.4*
1000	4.8	*4.1*	8.4	*3.8*	9.6	*6.1*

Black: Cochrane-Orcutt

Red italics: Bootstrap

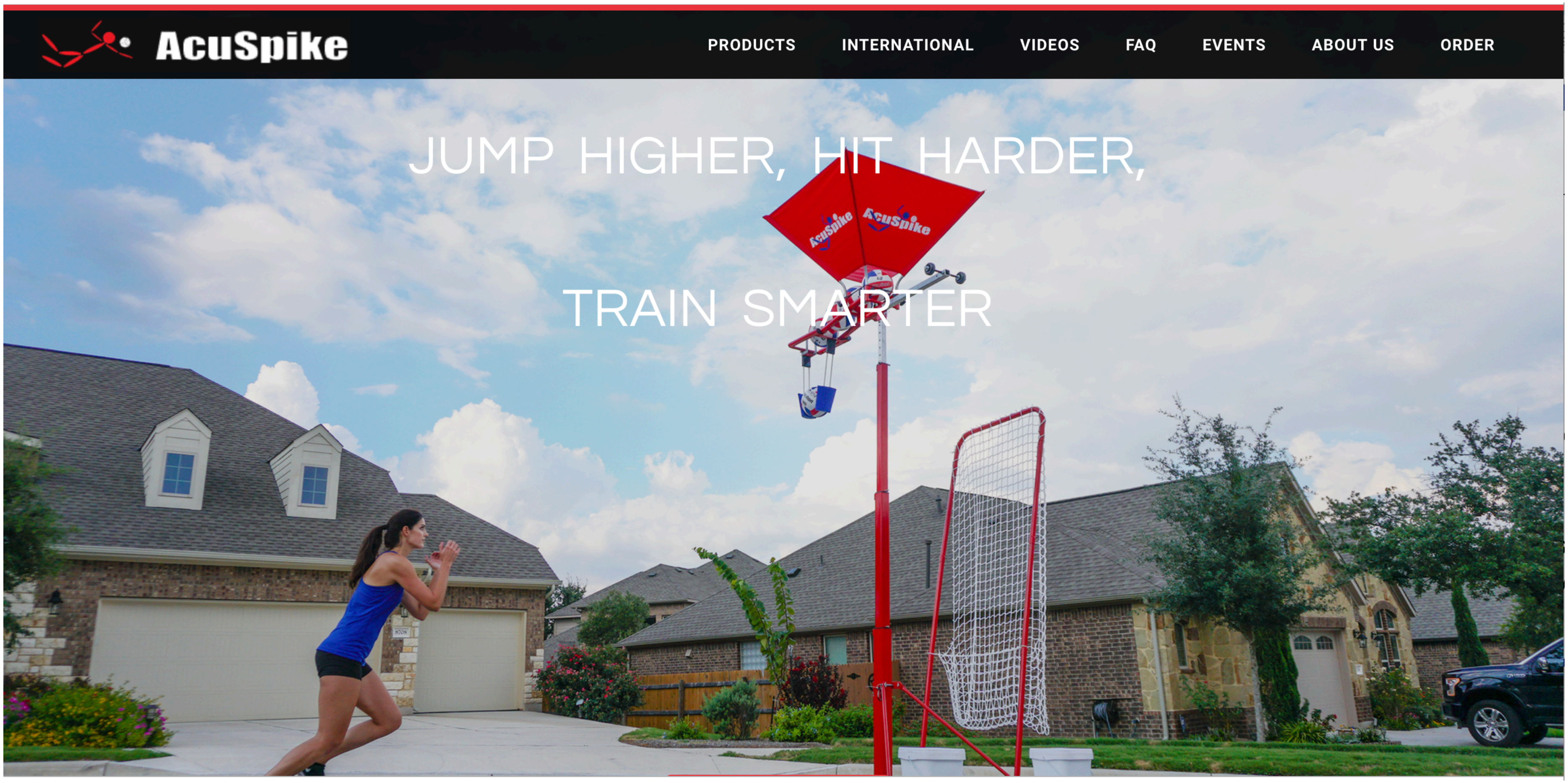
Final Remarks

We have discussed the problem of detecting a deterministic trend in time series data

- Even if a deterministic trend is detected, caution should be exercised for all but short term forecasts
  - Unless there is sufficient evidence that current conditions exist into the future
  - This is essentially the extrapolation problem in the time series setting

We have not discussed the issues involved in testing for deterministic components other than linear trends.

- For example, cosine trends can be difficult to distinguish from cyclic nonstationary and near nonstationary behavior


AcuSpike
PRODUCTS
INTERNATIONAL
VIDEOS
FAQ
EVENTS
ABOUT US
ORDER
JUMP HIGHER, HIT HARDER,
TRAIN SMARTER
AcuSpike
AcuSpike

DataScience@SMU