Seasonal ARIMA

Seasonality

Not a good test to determine if seasonality is in the series.

Analyst will need to make this decision.

If seasonality is observed in the model, we need to appropriately model the functional form (and then can also look for seasonal autocorrelation terms too).

We will begin by how to model the “functional form” of seasonality.

Seasonal ARIMA models

If there is a functional form for seasonality (i.e. visible pattern), then we need to model this:

- Seasonal dummy variables
- Trigonometric functions/Fourier

Deterministic pattern

- Seasonal differences (i.e. Seasonal random walk)

After taking care of the seasonal pattern (and trend if there is one) then we can also look for seasonal dependence (correlation) structure

- Seasonal AR
- Seasonal MA terms

Dummy variables

We can fit dummy variables for seasonal levels

- For example
  - 11 dummy variables for monthly data
  - 3 dummy variables for quarterly data
  - 23 dummy variables for hourly data

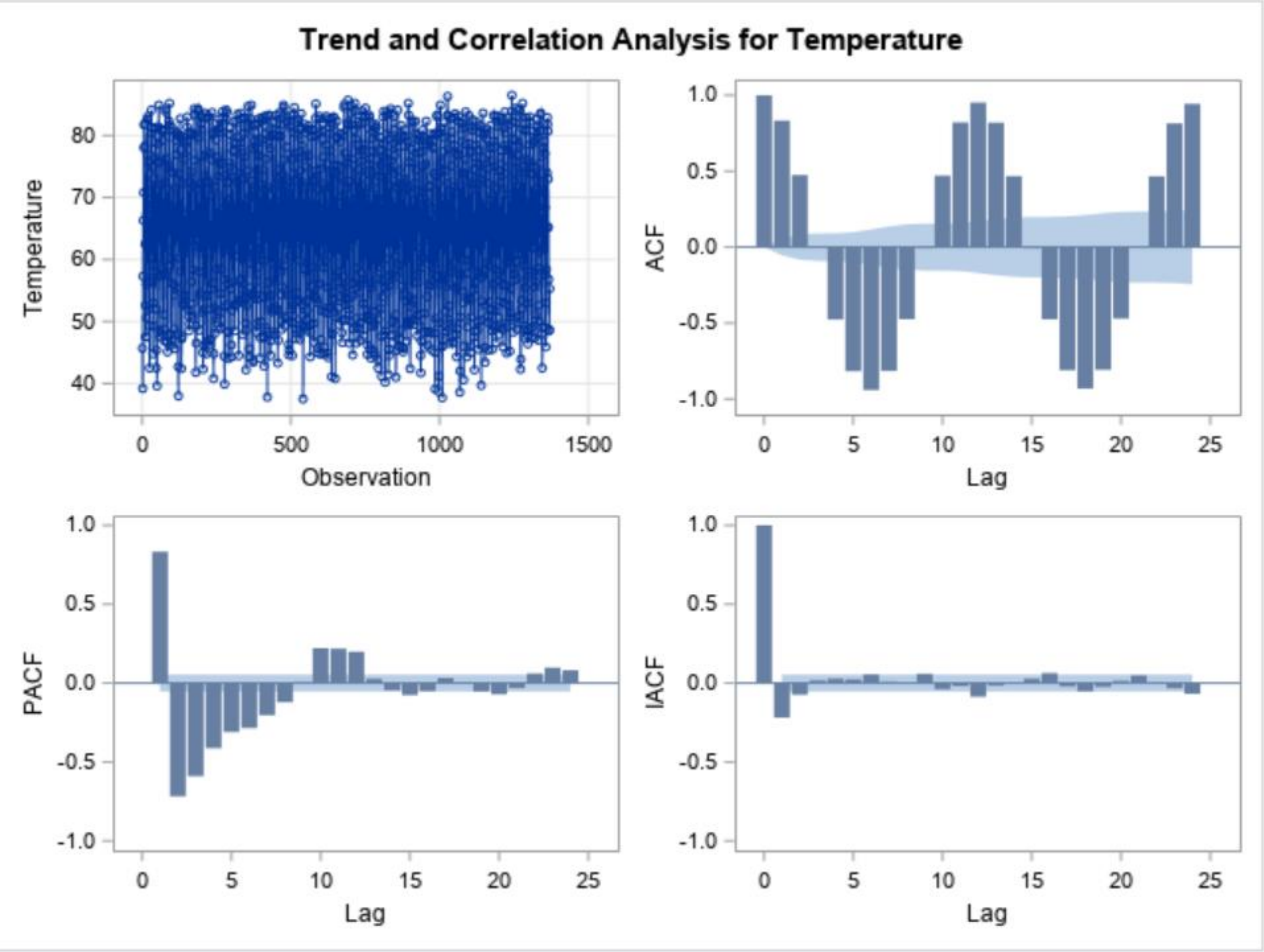
```
proc arima data=season;
identify var=y1 crosscorr=(x1 x2 x3);
estimate input=(x1 x2 x3);
run;
quit;
```

```
season=matrix(rep(c(1,0,0,0,1,0,0,0,1,0,0,0),12),byrow=T,nrow=48)
arima.season=Arima(y2.ts,xreg=season,order=c(0,0,0))
```

However, if there are too many levels might want to move to trig functions (for example, daily data).

```
data txnoaa;
set time.txnoaa;
if month=1 then seas1=1; else seas1=0;
if month=2 then seas2=1; else seas2=0;
if month=3 then seas3=1; else seas3=0;
if month=4 then seas4=1; else seas4=0;
if month=5 then seas5=1; else seas5=0;
if month=6 then seas6=1; else seas6=0;
if month=7 then seas7=1; else seas7=0;
if month=8 then seas8=1; else seas8=0;
if month=9 then seas9=1; else seas9=0;
if month=10 then seas10=1; else seas10=0;
if month=11 then seas11=1; else seas11=0;
run;

proc arima data=txnoaa;
identify var=Temperature crosscorr=(seas1 seas2 seas3 seas4 seas5 seas6 seas7 seas8 seas9 seas10 seas11);
estimate p=1 input=(seas1 seas2 seas3 seas4 seas5 seas6 seas7 seas8 seas9 seas10 seas11);
forecast lead=12;run;
quit;
```



Trend and Correlation Analysis for Temperature
Temperature
Observation
ACF
Lag
PACF
Lag
IACF
Lag

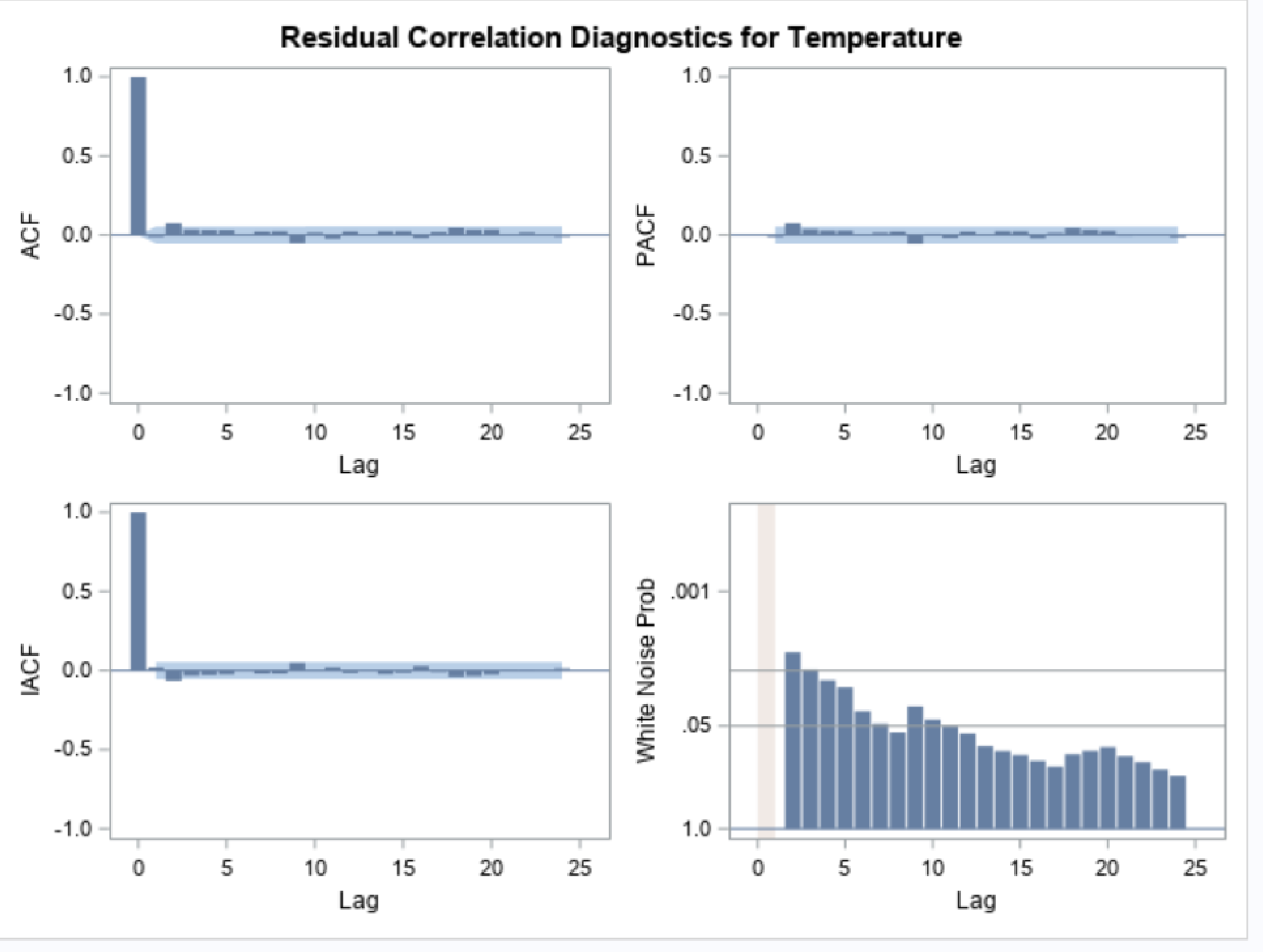
Model for variable Temperature	
Estimated Intercept	47.67982

Input Number 1	
Input Variable	seas1
Overall Regression Factor	-1.27287

Input Number 2	
Input Variable	seas2
Overall Regression Factor	2.275828

Input Number 3	
Input Variable	seas3
Overall Regression Factor	9.664912

Input Number 4


Residual Correlation Diagnostics for Temperature
ACF
1.0
0.5
0.0
-0.5
-1.0
0
5
10
15
20
25
Lag
PACF
IACF
White Noise Prob
.001
.05
1.0

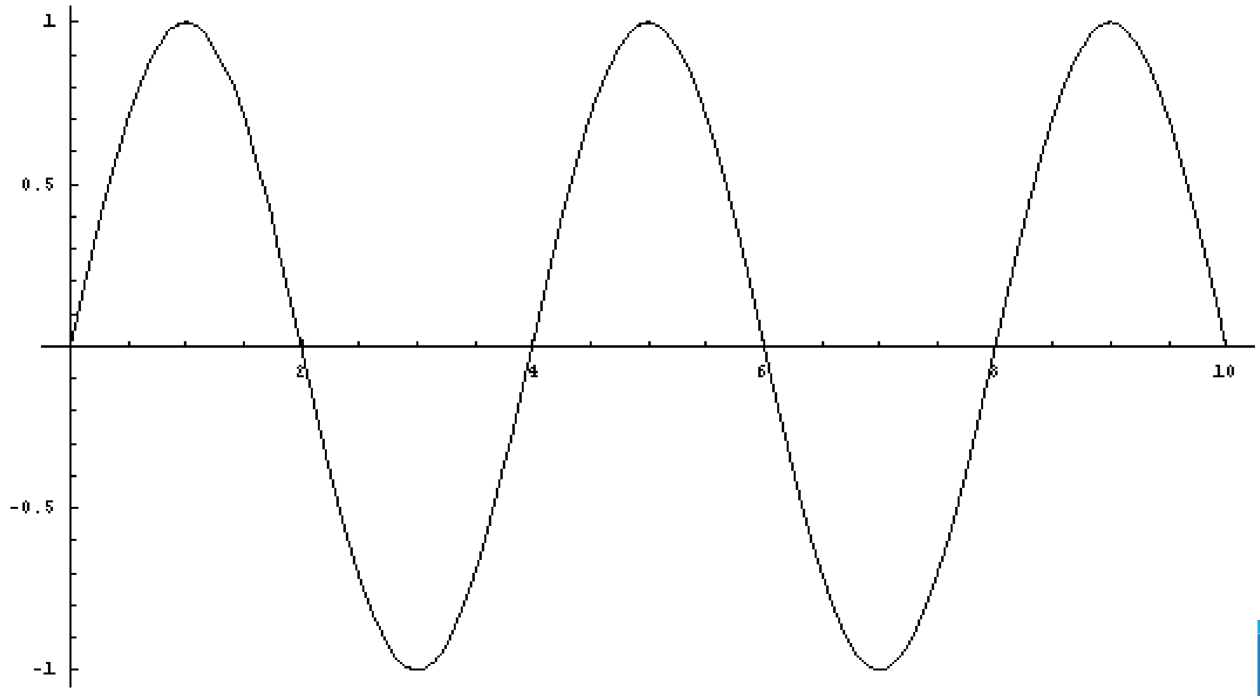
```
tx.ts<-ts(tx$Temperature[1:1358],frequency = 12)
 tx.test<-tx$Temperature[1359:1370]
month.tx=factor(tx$Month[1:1358])
 reg.tx=model.matrix(~month.tx)
 reg.tx=reg.tx[,-1]
 tx.seas=Arima(tx.ts,xreg=reg.tx)
 summary(tx.seas)
```

Fitting Dummy is not a good option with too many levels

If the length of the season is long (for example 365 or potentially even 24), we can fit trigonometric functions instead!

Trigonometric Functions

- Trigonometric functions in mathematics have a cyclical pattern.
- Use trigonometric functions, such as sine and cosine, to model seasonality.



Trigonometric Regression

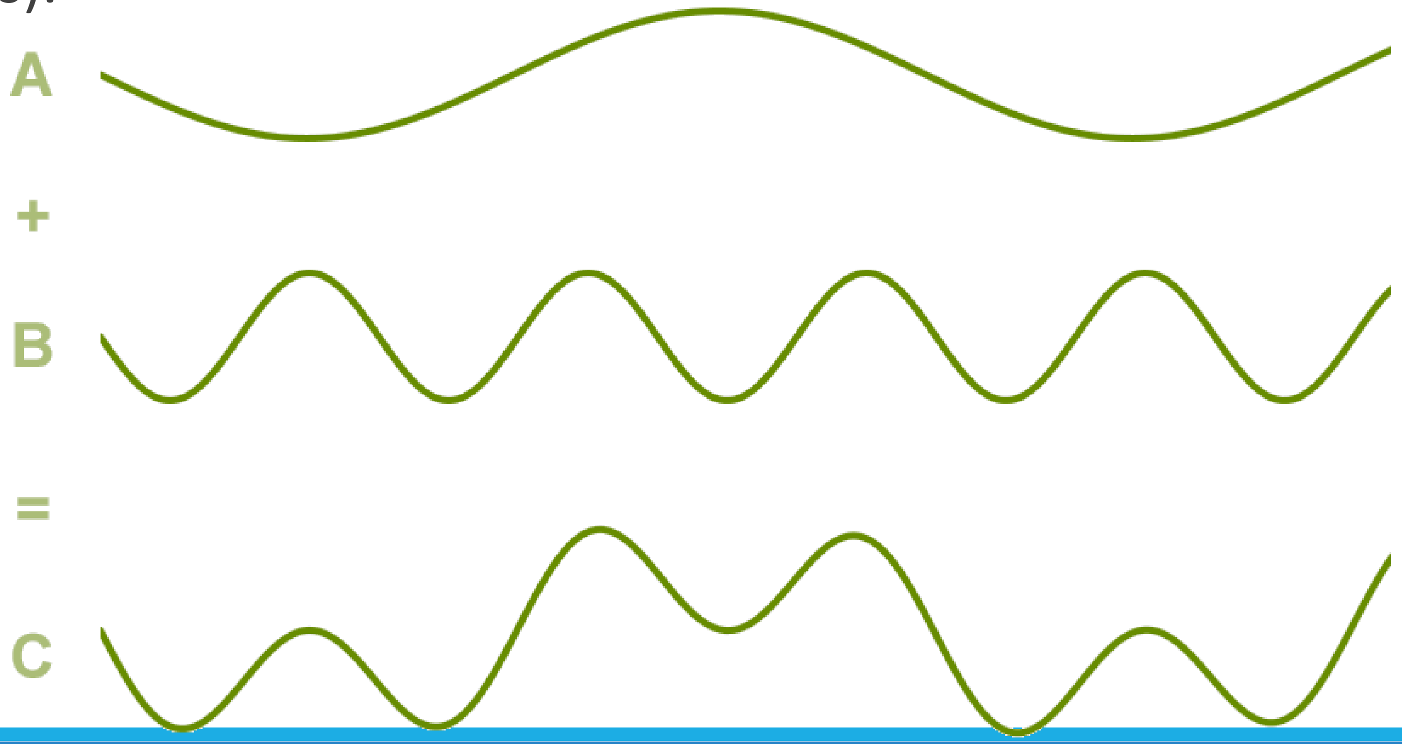
- Define trigonometric variables in the modeling data set and use them for forecasting.

$$X_t = \sin\left(\frac{2\pi t}{S}\right)$$

$$Y_t = \beta_0 + \beta_1 X_t + e_t$$

Trigonometric Regression

- You don’t have to limit yourselves to only one sine or cosine variable.
- Mixing sine and cosine functions might better fit your data (Fourier analysis).



SAS Code

```
data txnoaa;
set time.txnoaa;
pi=constant("pi");
s1=sin(2*pi*1*_n_/12);
c1=cos(2*pi*1*_n_/12);
s2=sin(2*pi*2*_n_/12);
c2=cos(2*pi*2*_n_/12);
s3=sin(2*pi*3*_n_/12);
c3=cos(2*pi*3*_n_/12);
s4=sin(2*pi*4*_n_/12);
c4=cos(2*pi*4*_n_/12);
run;
proc arima data=txnoaa plots=all;
identify var=temperature crosscorr=(s1 c1 s2 c2 s3 c3 s4 c4);
estimate p=1 input=(s1 c1 s2 c2 s3 c3 s4 c4);
forecast lead=12;
run;
quit;
```

Rcode

```
index.ts=seq(1,length(tx.ts))
x1.sin=sin(2*pi*index.ts*1/12)
x1.cos=cos(2*pi*index.ts*1/12)
x2.sin=sin(2*pi*index.ts*2/12)
x2.cos=cos(2*pi*index.ts*2/12)
x3.sin=sin(2*pi*index.ts*3/12)
x3.cos=cos(2*pi*index.ts*3/12)
x4.sin=sin(2*pi*index.ts*4/12)
x4.cos=cos(2*pi*index.ts*4/12)
x.reg=cbind(x1.sin,x1.cos,x2.sin,x2.cos,x3.sin,x3.cos,x4.sin,x4.cos)
arima.1<-Arima(tx.ts,order=c(0,0,0),xreg=x.reg)
summary(arima.1)
arima.2<-Arima(tx.ts,order=c(0,0,0),xreg=fourier(tx.ts,K=4))
summary(arima.2)
```

Long seasonality

Careful when dealing with daily and/or weekly data (seasonality for daily would be 365.25 days or 52.179 weeks [365.25/7])

Fourier (sin/cos) can also be used for complex seasonality

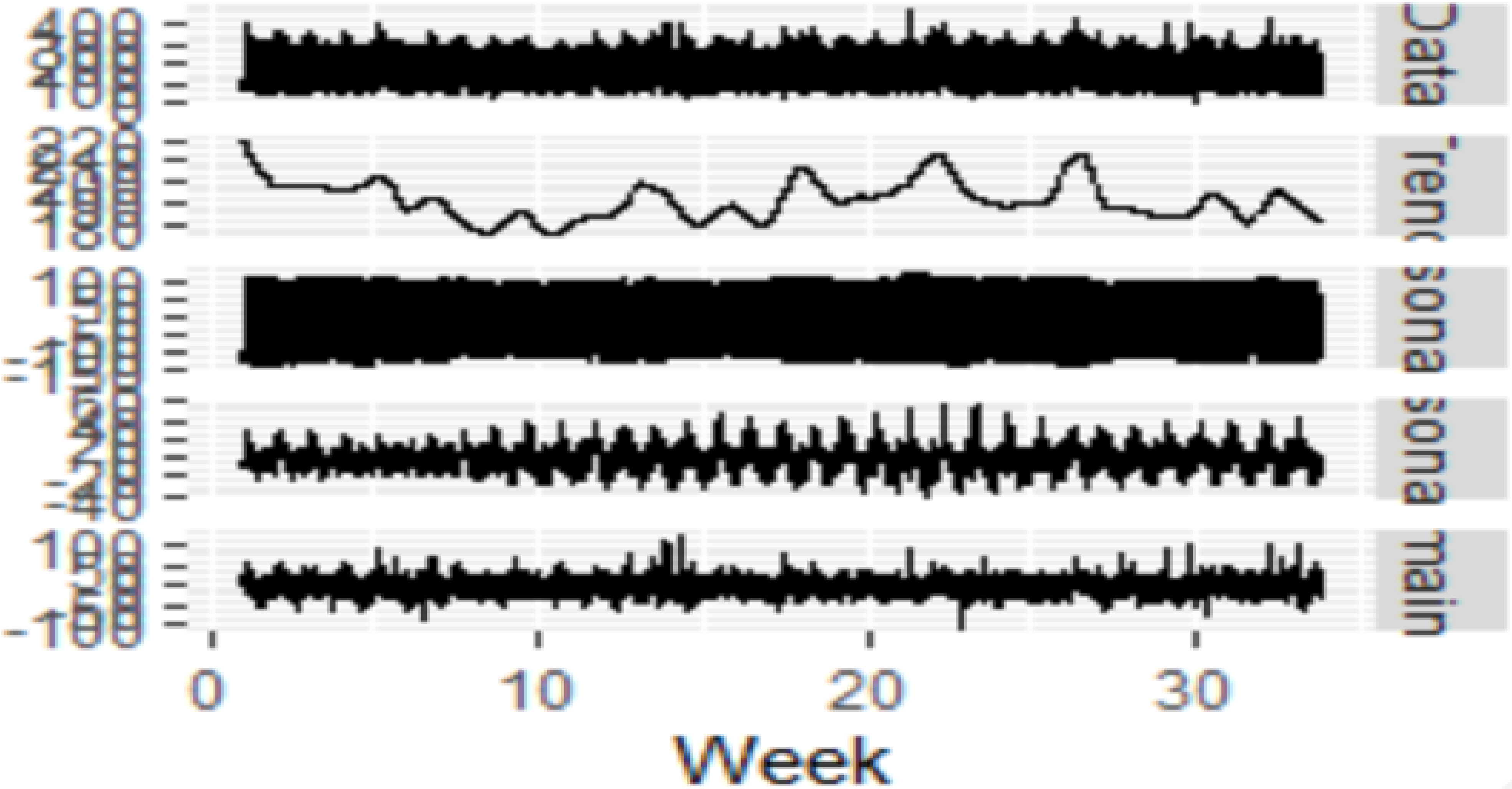
- Need to create Fourier series or sin/cos sequences for each seasonality
- Example of complex seasonality (call data)

R code

```
calls <-
unlist(read.csv("https://robjhyndman.com/data/callcenter.txt",
                    header=TRUE,sep="\t")[,-1])
 calls <- msts(calls, start=1, seasonal.periods = c(169, 169*5))

 calls %>% mstl() %>%
   autoplot() + xlab("Week")

 fit <- auto.arima(calls, seasonal=FALSE,
                 xreg=fourier(calls, K=c(10,10)))
 summary(fit)
 fit %>%
   forecast(xreg=fourier(calls, K=c(10,10), h=2*169)) %>%
   autoplot(include=5*169) +
   ylab("Call volume") + xlab("Weeks")
```


Data
0
10
20
30
Week

Seasonal Random Walks

For seasonal data with period *S*, express the current value as a function that includes the value *S* time units in the past.

$$Y_t = Y_{t-s} + \cdots$$

$$Y_t - Y_{t-s} = \varepsilon_t$$

Examples:

- Monthly → January is a function of last January
- Weekly → Sunday is a function of last Sunday

Seasonal Random Walks

Take a 'seasonal difference'

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

Question: When should we fit a deterministic function (dummy variables or Fourier) and when should we take a seasonal difference?....

Seasonal ADF test

The Augmented Dickey-Fuller test can be extended to check for seasonal lags as well.

The tests will be differenced on specified seasonal lengths instead of single differences.

Tests only able to be checked for seasons up to length 12.

```
proc arima data=txnoaa;
identify var=Temperature stationarity=(adf=2 dlag=12);
run;
quit;
```

Seasonal Augmented Dickey-Fuller Unit Root Tests					
Type	Lags	Rho	Pr < Rho	Tau	Pr < Tau
Zero Mean	0	-1.7834	0.4178	-0.87	0.2348
	1	-2.5931	0.3619	-1.08	0.1757
	2	-3.0963	0.3296	-1.21	0.1428
Single Mean	0	-55.6396	0.0013	-5.44	<.0001
	1	-70.5150	0.0013	-6.10	<.0001
	2	-70.9479	0.0013	-6.06	<.0001

```
nsdiffs(tx.ts)
nsdiffs(tx.ts,test='ch')
```

Seasonal AR and MA terms

In addition to modeling the “functional” form for seasonality (the pattern that you can see!), you can also have correlations in seasons (for example, this January is related to values last January; or this Monday is related to last Monday)

We fit these dependencies by using seasonal AR and MA terms

Remove all signal first (seasonal patterns, trend patterns and/or random walk….have stationary time series) and then we can use ACF, PACF and IACF to help us uncover patterns

Box-Jenkins seasonal arima

- The *p, d,* and *q* are the orders of the nonseasonal terms (typical ARIMA part)
- *P* is the order of the **seasonal** autoregressive structure.
- *Q* is the order of the **seasonal** moving average structure.
- *D* is the order of the **seasonal** difference structure.
- *S* is the length of the seasonal period.

$$\text{ARIMA}(p, d, q)(P, D, Q)_S$$

Using notation

ARIMA$(0, 1, 0)(0, 1, 0)_7$

ARIMA$(3, 0, 2)(1, 1, 0)_{12}$

ARIMA$(4, 1, 0)(2, 0, 3)_4$

ARIMA$(3, 1, 5)(2, 1, 2)_{24}$

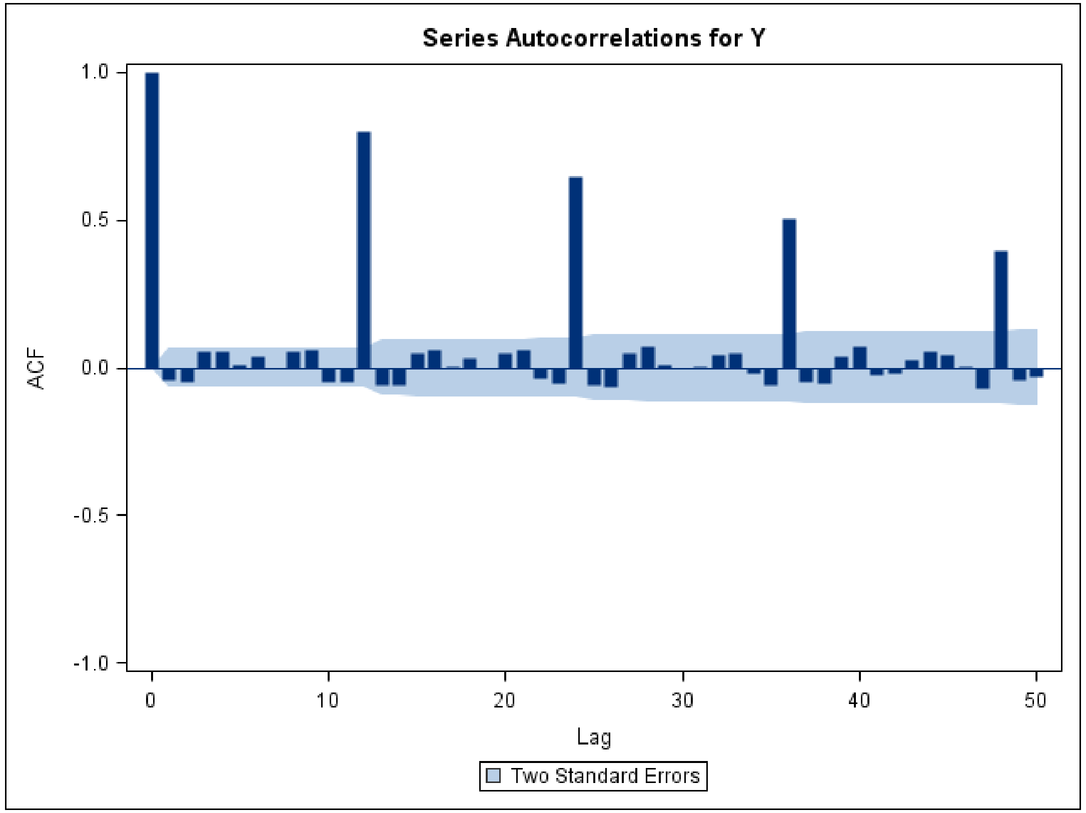
ARIMA(0, 0, 0)(1, 0, 0)$_{12}$

$$Y_t - \mu = \alpha(Y_{t-12} - \mu) + e_t$$

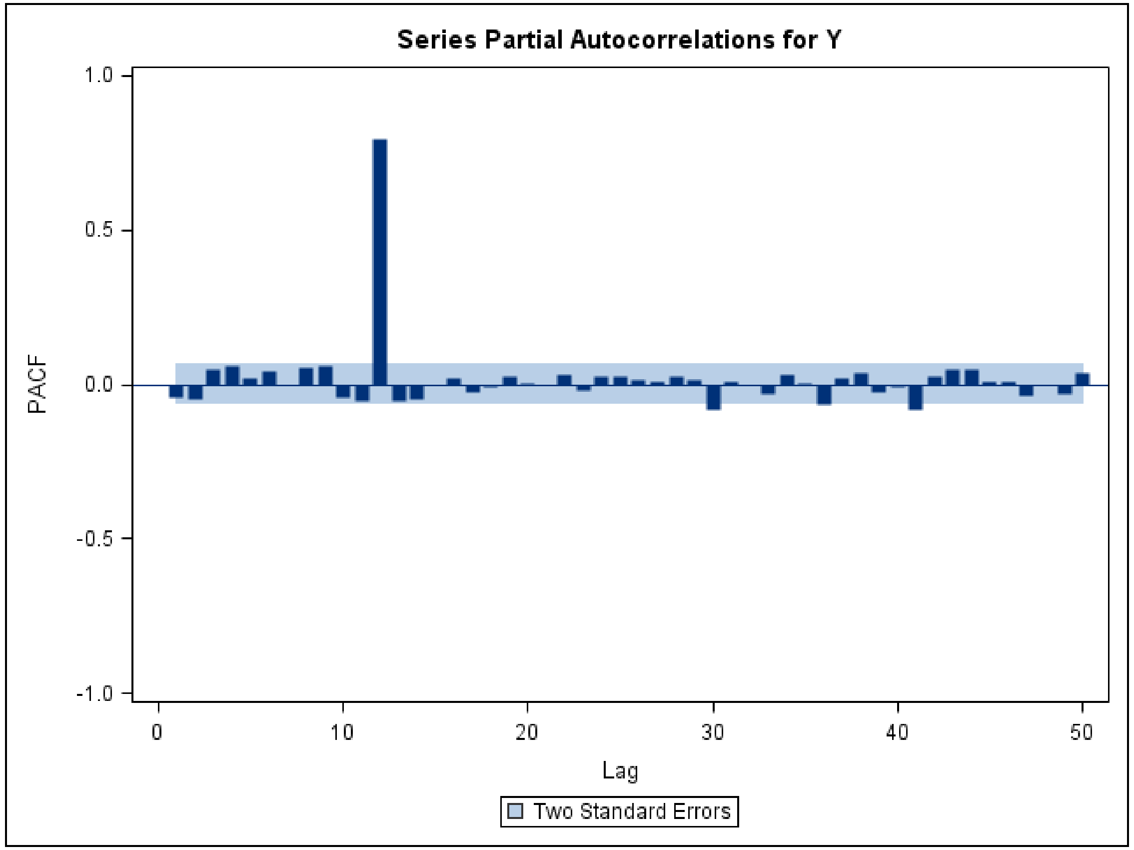
- This is similar to the AR(1) model with the lag value now at *t-12*.
  - The same pattern with the AR(1) model will exist with the ARIMA(0, 0, 0)(1, 0, 0)$_{12}$, just with the seasonal lags.
  - Let's examine the ACF, PACF, and IACF for the following model:

$$Y_t - \mu = 0.8(Y_{t-12} - \mu) + e_t$$

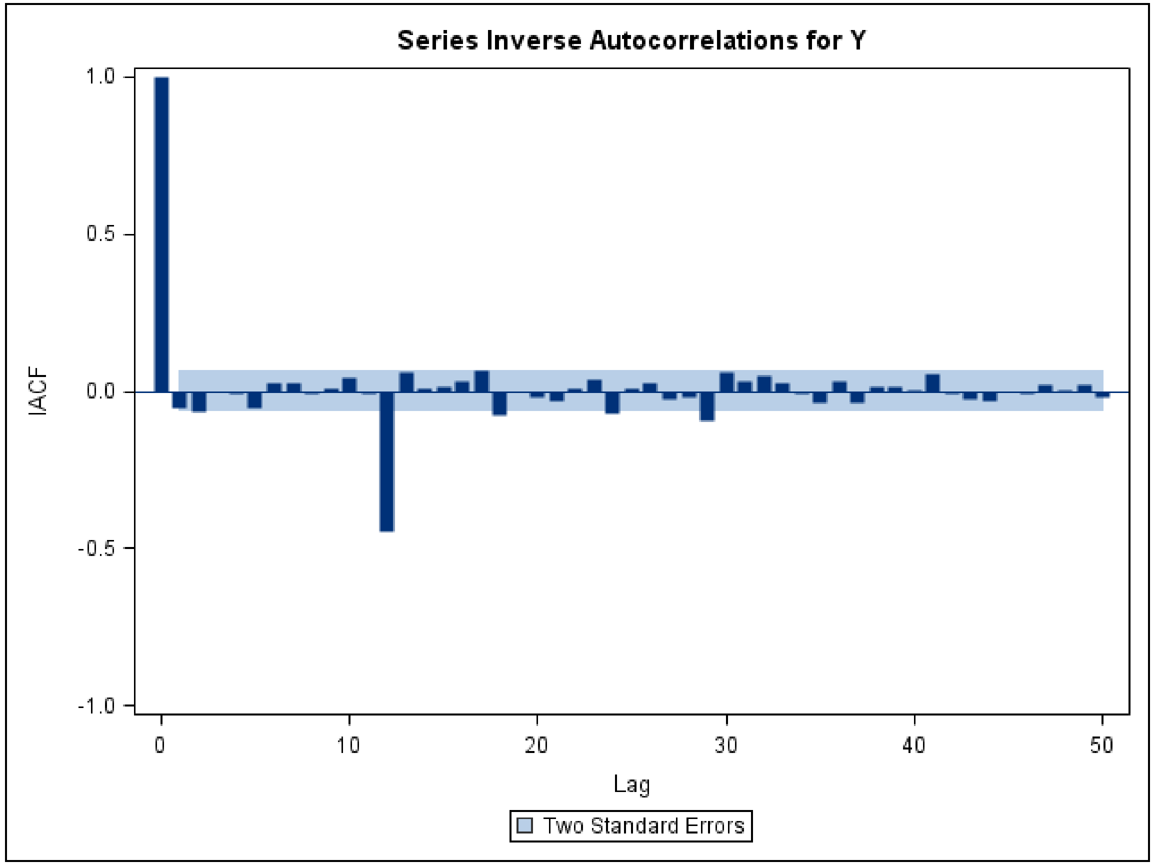
ARIMA(0, 0, 0)(1, 0, 0)$_{12}$ – ACF



ARIMA(0, 0, 0)(1, 0, 0)$_{12}$ – PACF



ARIMA(0, 0, 0)(1, 0, 0)$_{12}$ – IACF



Multiplicative vs Additive seasonal effects

Backshift Operator – B

- The backshift operator is the mathematical operator to convert observations to their lags.
  - $\mathrm{B}(Y_t) = Y_{t-1}$

- This can be extended to any number of lags.
  - $\mathrm{B}^2(Y_t) = \mathrm{B}(Y_{t-1}) = Y_{t-2}$

Problems

- Models become extremely complicated.
- Multiplicative vs. additive structure to the seasonal effects.

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

vs.

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

Problems

- Models become extremely complicated.
- Multiplicative vs. additive structure to the seasonal effects.

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

vs.

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

Additive

Problems

- Models become extremely complicated.
- Multiplicative vs. additive structure to the seasonal effects.

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

Multiplicative

vs.

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

Problems

- Models become extremely complicated.
- Multiplicative vs. additive structure to the seasonal effects.

$$(1-\alpha_1 \mathrm{B})(1-\alpha_2 \mathrm{B}^{12})Y_t = e_t$$

vs.

$$(1-\alpha_1 \mathrm{B}-\alpha_2 \mathrm{B}^{12})Y_t = e_t$$

$$(1-\alpha_1 \mathrm{B}-\alpha_2 \mathrm{B}^{12}+\alpha_1\alpha_2 \mathrm{B}^{13})Y_t = e_t$$

Multiplicative vs. Additive

- Multiplicative models are just a special case of the additive model.

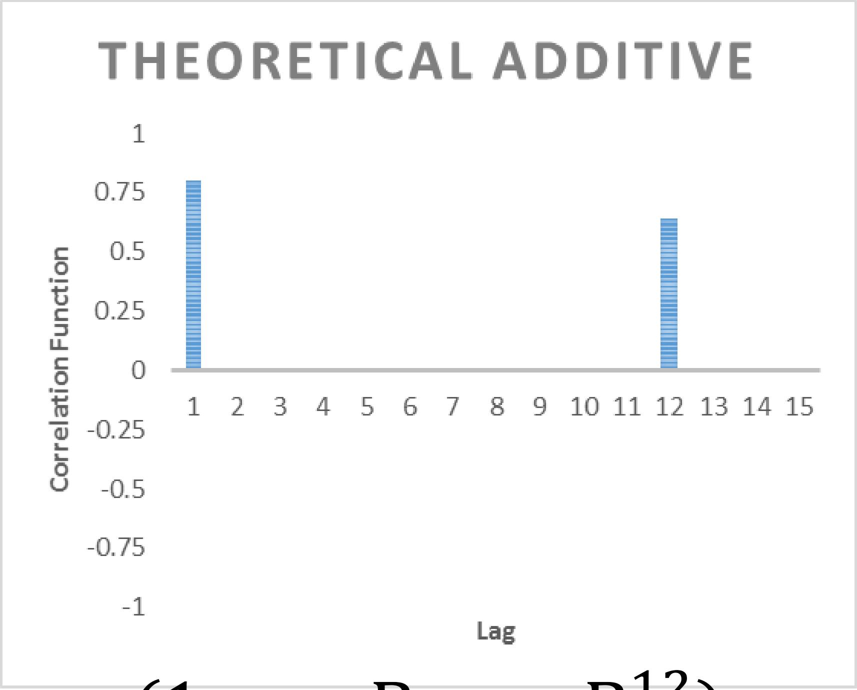
$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12} + \alpha_1 \alpha_2 \mathrm{B}^{13}) Y_t = e_t$$

$\alpha_3 = -\alpha_1 \alpha_2$?

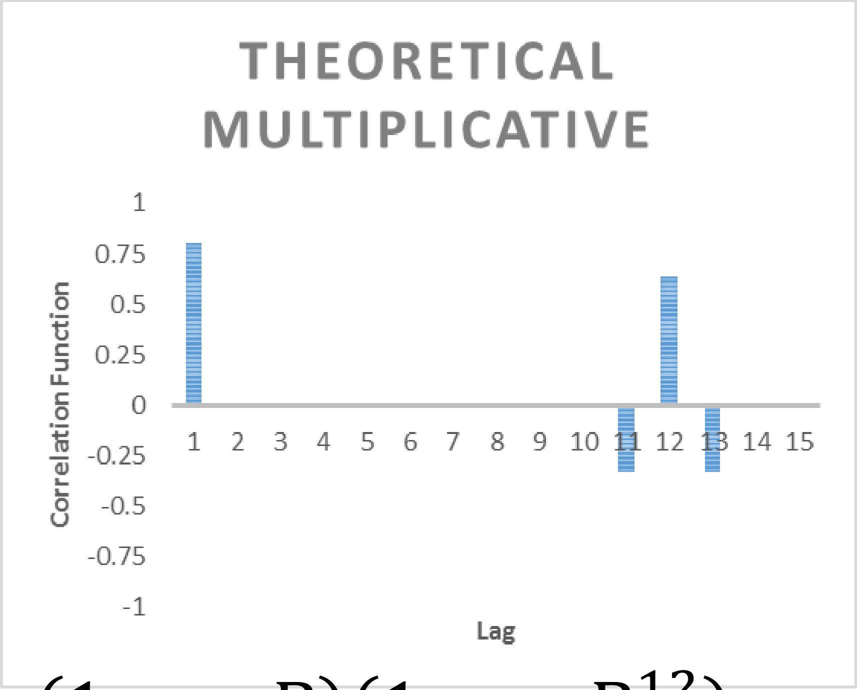
$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12} - \alpha_3 \mathrm{B}^{13}) Y_t = e_t$$

Multiplicative vs. Additive

Correlations have distinct pattern differences if a model is additive compared to multiplicative.



$$(1-\alpha_1 \mathrm{B}-\alpha_2 \mathrm{B}^{12})$$



$$(1-\alpha_1 \mathrm{B})(1-\alpha_2 \mathrm{B}^{12})$$

Multiplicative vs. Additive – SAS

```
proc arima data=Time.Airline plot=all;
    identify var=LogPsngr(1,12) nlag=25;
    estimate q=(1,12,13);
    estimate q=(1)(12);
    forecast lead=24;
run;
quit;
```

Multiplicative vs. Additive? – R

```
S.ARIMA <- Arima(air.ts, order=c(0,1,1), seasonal=c(0,1,1),method="ML")
summary(S.ARIMA)

S.ARIMA <- Arima(air.ts, order=c(0,1,13), seasonal=c(0,1,0),
fixed=c(NA,0,0,0,0,0,0,0,0,0,0,NA,NA), method="ML",)
summary(S.ARIMA)
```