Seasonality

This Lecture

Advantages and disadvantages of three approaches to modeling seasonality:

- Seasonal dummy/indicator/binary variables
- Seasonal ARIMA
- Harmonic regression/Fourier terms

Seasonality

- Regular swings with fixed period (unlike stochastic cycles with random periods)
- Natural gas inventories, electricity consumption, etc.
- Many macroeconomic series are seasonally adjusted by federal agencies. What if we need to work with the raw data?
- Applications: Forecasting, controlling for seasonal variation in X or Y to "partial-out" the seasonal effect, removing seasonal autocorrelation from residuals.

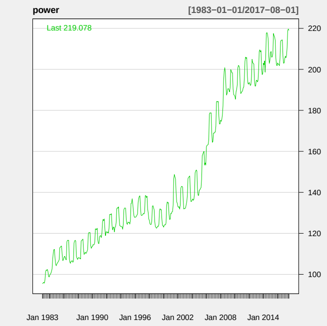
Ignore this warning message

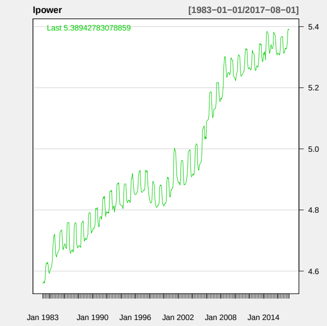
```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0.  You will still be able to use
## 'loadSymbols' to automatically load data.  getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## [1] "CUUR0000SEHF01"
```

Example: Electricity Prices

- Monthly consumer price index of electricity for all urban consumers
- Log the series if there is exponential growth and/or increasing variance.

```
power <- CUUR0000SEHF01[paste("1983-01-01","2017-08-01",sep="/")]
lpower <- log(power)
chartSeries(power,theme="white")
chartSeries(lpower,theme="white")
```





Seasonal Dummy/Indicator/Binary

- Use a dummy variable for each season. Here, month (or quarter).
- $ln(P)_t = p_t$

$$p_t = \beta_0 + \beta_1 x_t + \delta_1 D_{Jan} + \cdots + \delta_{11} D_{Nov} + e_t$$

item $D_{Jan} = 1$ of month is January, $= 0$ otherwise.

- Why only 11 dummies?
- Advantages: simple for low-frequency data/seasonality
- Disadvantages: Cumbersome for high-frequency or multiple seasonality

Seasonal Dummies in R

- Using xts to create dummies:

```
yrmo = factor(month(index(lpower)),
              labels = c("Jan","Feb","Mar","Apr","May","Jun",
                         "Jul","Aug","Sep","Oct","Nov","Dec"))
yrmo <- createDummyFeatures(yrmo,cols="var") # function in "mlr" package
yrmo.xts <- xts(yrmo[,c(1:12)],order.by = as.yearmon(index(lpower)))
lpower.mdums <- merge.xts(lpower,yrmo.xts)
head(lpower.mdums)

##            CUUR0000SEHF01 Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1983-01-01       4.559126   1   0   0   0   0   0   0   0   0   0   0   0
## 1983-02-01       4.562263   0   1   0   0   0   0   0   0   0   0   0   0
## 1983-03-01       4.565389   0   0   1   0   0   0   0   0   0   0   0   0
## 1983-04-01       4.561218   0   0   0   1   0   0   0   0   0   0   0   0
## 1983-05-01       4.575741   0   0   0   0   1   0   0   0   0   0   0   0
## 1983-06-01       4.614130   0   0   0   0   0   1   0   0   0   0   0   0
```

Seasonal Dummies in R

- Using as.factor with xts in a regression:

```
texreg(lm(lpower~as.factor(month(index(lpower)))+index(lpower)),
       fontsize="tiny",include.loglik=FALSE,single.row=TRUE,
       caption="Pretty Table Using TeXReG")
```

	Model 1
(Intercept)	4.26 (0.01)***
as.factor(month(index(lpower)))2	−0.00 (0.01)
as.factor(month(index(lpower)))3	−0.00 (0.01)
as.factor(month(index(lpower)))4	−0.00 (0.01)
as.factor(month(index(lpower)))5	0.01 (0.01)
as.factor(month(index(lpower)))6	0.06 (0.01)***
as.factor(month(index(lpower)))7	0.07 (0.01)***
as.factor(month(index(lpower)))8	0.07 (0.01)***
as.factor(month(index(lpower)))9	0.06 (0.01)***
as.factor(month(index(lpower)))10	0.02 (0.01)
as.factor(month(index(lpower)))11	−0.00 (0.01)
as.factor(month(index(lpower)))12	−0.01 (0.01)
index(lpower)	0.00 (0.00)***
R^2	0.94
Adj. R^2	0.94
Num. obs.	416

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table: Pretty Table Using TeXReG

Seasonal Dummies in R

- Using ts:

```
lpower.ts <-ts(lpower[,1],start=start(index(power)),freq=12)
# tslm(lpower.ts ~ season + trend) # alternative to dynlm()
texreg(dynlm(lpower.ts ~ season(lpower.ts)+ trend(lpower.ts)),
       fontsize="tiny",include.loglik=FALSE,single.row=TRUE,
       caption="Pretty Table Using TeXReG")
```

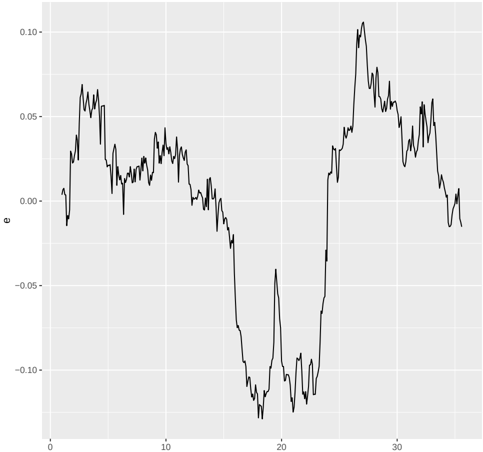
	Model 1
(Intercept)	4.55 (0.01)***
season(lpower.ts)Feb	−0.00 (0.01)
season(lpower.ts)Mar	−0.00 (0.01)
season(lpower.ts)Apr	−0.00 (0.01)
season(lpower.ts)May	0.01 (0.01)
season(lpower.ts)Jun	0.06 (0.01)***
season(lpower.ts)Jul	0.07 (0.01)***
season(lpower.ts)Aug	0.07 (0.01)***
season(lpower.ts)Sep	0.06 (0.01)***
season(lpower.ts)Oct	0.02 (0.01)
season(lpower.ts)Nov	−0.00 (0.01)
season(lpower.ts)Dec	−0.01 (0.01)
trend(lpower.ts)	0.02 (0.00)***
R^2	0.94
Adj. R^2	0.94
Num. obs.	416

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Seasonal Dummies in R

- After removing trend & seasons, may still be random walk:

```
e = residuals(dynlm(lpower.ts ~ season(lpower.ts)+ trend(lpower.ts)))
autoplot(e)
```



Seasonal ARIMA

- Autocorrelation may exist at *seasonal frequency s* as well as immediate lags.
- Can have unit root at seasonal frequency and/or at immediate lag.
- Seasonal unit root:

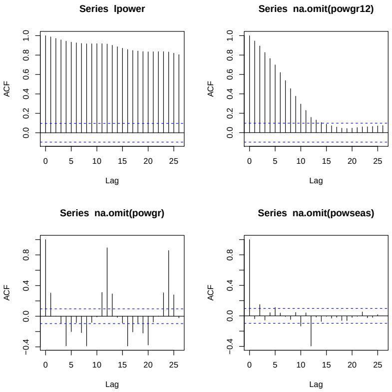
$$p_t = p_{t-s} + a_t \Longrightarrow \Delta_s p_t = (1 - L^s)p_t = a_t$$

- Regular and seasonal unit root:

$$\Delta p_t - \Delta p_{t-s} = \Delta_s \Delta p_t = (1-L^s)(1-L)p_t = p_t - p_{t-1} - p_s + p_{s+1} + a_t$$

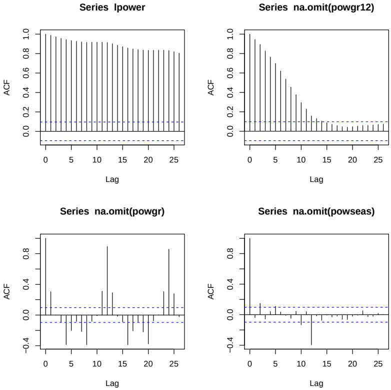
Seasonal ARIMA

```
powgr <- diff(lpower) # price growth (returns)
powgr12 <- diff(lpower,lag=12,differences=1) # seasonal difference of prices
powseas <- diff(powgr,lag=12,differences=1) # seasonal difference of returns
```



Seasonal ARIMA

- Δp_t (bottom left) has large ACF every 12 steps.
- $\Delta_{12}\Delta p_t$ (bottom right) still has one at lag 12 but not further. We can deal with that using seasonal MA.
- $\Delta_{12} p_t$ (top right) might be stationary, unclear.



More General SARIMA

- Sometimes called Multiplicative Seasonal Models or the Airline Model
- (arrival rate of airline passengers at the terminal)

$$(1 - \phi_s L^s)(1 - \phi_1 L)p_t = (1 - \theta_1 L)(1 - \theta_s L^s)a_t$$

- $\phi_s \neq 1$ and $\phi_1 \neq 1$ allow stationary autoregressive behavior at regular and seasonal lags.
- Advantages: flexibly model seasonal variation that may change from year to year
- Disadvantages: too constraining in high frequency data

Modeling seasonal electricity prices

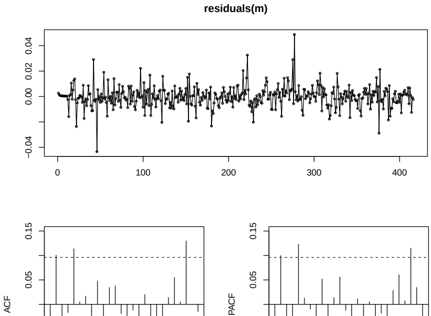
- "order=c(p,d,q)" for regular ARIMA
- "seasonal=list(order=c(P,D,Q))" for seasonal ARIMA.
- based on PACF of $\Delta_{12}\Delta p_t$, try SMA(2)

$$\Delta_s \Delta p_t = (1 - \theta_s B^s - \theta_{2s} B^{2s}) a_t$$

```
m1 = Arima(lpower,order=c(0,1,0),seasonal=list(order=c(0,1,2),period=12))
```

Modeling seasonal electricity prices

```
## Series: lpower
## ARIMA(0,1,0)(0,1,2)[12]
##
## Coefficients:
##          sma1     sma2
##       -0.7087  -0.2003
## s.e.   0.0506   0.0491
##
## sigma^2 estimated as 7.374e-05:  log likelihood=1338.59
## AIC=-2671.18   AICc=-2671.12   BIC=-2659.19
```



Modeling seasonal electricity prices

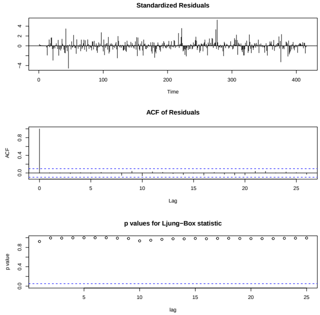
- I experimented a little, auto.arima() was not helpful, but I found this: AIC lower, BIC higher, less residual autocorrelation

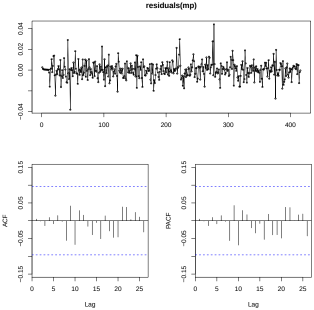
$$(1-\phi_s B^s - \phi_{2s} B^{2s})\Delta_s \Delta p_t = (1-\theta_1 B - \ldots - \theta_5 B^5)(1-\theta_s B^s) a_t$$

```
mp = Arima(lpower,order=c(0,1,5),seasonal=list(order=c(2,1,1),period=12))
mp

## Series: lpower
## ARIMA(0,1,5)(2,1,1)[12]
##
## Coefficients:
##           ma1     ma2      ma3      ma4     ma5    sar1    sar2     sma1
##       -0.0324  0.1114  -0.0456  -0.0204  0.1411  0.2799  0.0932  -0.9880
## s.e.   0.0496  0.0495   0.0490   0.0514  0.0535  0.0597  0.0580   0.1583
##
## sigma^2 estimated as 6.965e-05:  log likelihood=1348.53
## AIC=-2679.06   AICc=-2678.6   BIC=-2643.07
```

Modeling seasonal electricity prices

```
tsdiag(mp,gof=25)
tsdisplay(residuals(mp))
```



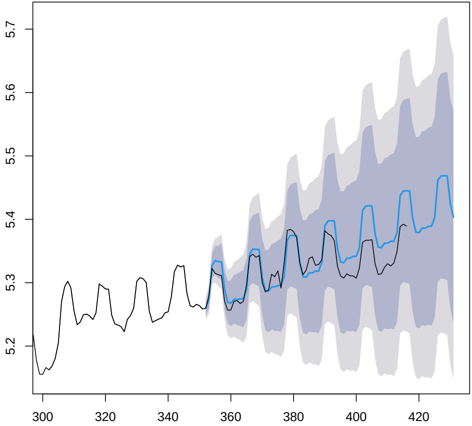


SARIMA Forecast

- Fit it on a holdout sample up until the last 65 observations.
- Use Arima() instead of arima() to ensure that intercept is estimated.

```
fit_no_holds <- Arima(lpower[-c(352:416)],order=c(0,1,5),
                      seasonal=list(order=c(2,1,1),period=12),
                      include.constant = T)
fcast_no_holds <- forecast(fit_no_holds,h=80)
plot(fcast_no_holds,main=" ",include=50)
lines(ts(lpower))
```

SARIMA Forecast



Harmonic Regression

- Sometimes more convient to use a combination of sine and cosine functions of time.
- These are called Fourier terms.
- If s is season length (e.g., 12 for monthly, 52 for weekly data), then Fourier terms that can be included consist of

$$
\begin{aligned}
f_{1t} = &\quad sin\left(\frac{2\pi t}{s}\right) \\
f_{2t} = &\quad cos\left(\frac{2\pi t}{s}\right) \\
f_{3t} = &\quad sin\left(\frac{4\pi t}{s}\right) \\
f_{4t} = &\quad cos\left(\frac{4\pi t}{s}\right) \\
etc.
\end{aligned}
$$

- Include up to $K = s/2$ pairs, e.g., 6 for monthly data.
- Pick the one that minimizes the AIC/BIC, etc.

Harmonic Regression: Advantages and disadvantages

Advantages:

- Handles “wiggly” seasonal patterns and multiple “seasons”, e.g., natural gas consumption with a summer peak.
- Useful for high frequency data (weekly, daily, hourly, etc.)
- Often requires fewer parameters than seasonal dummies, especially with high frequency (large season length s) or multiple seasonality.
  - e.g., hourly data: hour 12 on the 42nd day of this year may not respond exactly to hour 12 of the 42nd day of last year the way a seasonal ARIMA would impose.
  - e.g., hourly data: there is a daily seasonality, as well as possibly weekly and monthly. Fourier terms can be included at different frequencies.

Harmonic Regression: Advantages and disadvantages

Disadvantages:

- Seasonal pattern forced to be identical throughout time (same as seasonal dummies), whereas seasonal ARIMA adapts to last season's value.

Fitting harmonic terms in practice

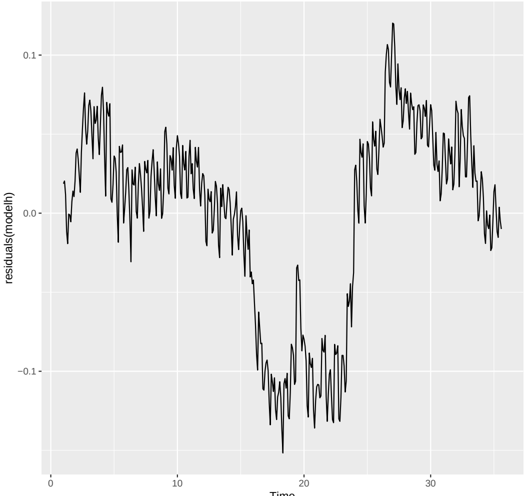
Use dynlm() with harmon() option:

```
modelh = dynlm(lpower.ts ~ harmon(lpower.ts,order=1)+ trend(lpower.ts) )
summary(modelh)

##
## Time series regression with "ts" data:
## Start = 1(1), End = 35(8)
##
## Call:
## dynlm(formula = lpower.ts ~ harmon(lpower.ts, order = 1) + trend(lpower.ts))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15168 -0.02426  0.01527  0.04189  0.12023
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.5762275  0.0059679 766.803  < 2e-16 ***
## harmon(lpower.ts, order = 1)cos -0.0376661  0.0042048  -8.958  < 2e-16 ***
## harmon(lpower.ts, order = 1)sin -0.0109697  0.0042200  -2.599  0.00967 **
## trend(lpower.ts)                0.0226471  0.0002976  76.089  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Fitting harmonic terms in practice

Still appears to be a random walk after controlling for seasons:

```r
autoplot(residuals(modelh))
```



Fitting harmonic terms in practice

Can also use arima(), but what to use for p, d, q and harmonic order?

```
arima(lpower.ts,order=c(1,0,1),xreg = fourier(lpower.ts,K=1))

##
## Call:
## arima(x = lpower.ts, order = c(1, 0, 1), xreg = fourier(lpower.ts, K = 1))
##
## Coefficients:
##          ar1      ma1  intercept    S1-12    C1-12
##       0.9986  -0.0520     4.9217  -0.0285  -0.0272
## s.e.  0.0018   0.0798     0.3091   0.0024   0.0024
##
## sigma^2 estimated as 0.0003503:  log likelihood = 1061.84,  aic = -2111.67
```

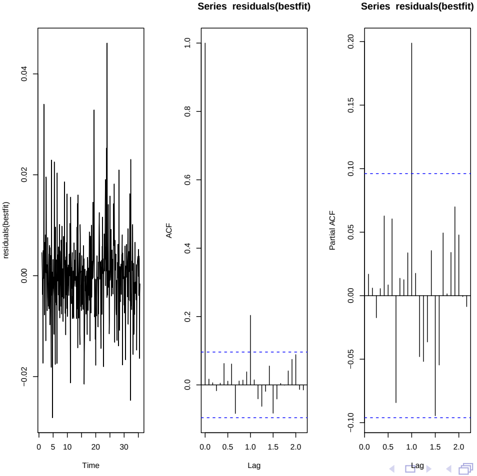
Fitting harmonic terms in practice

Loop over harmonic order to get lowest AIC:

```
bestfit <- list(aicc=Inf)
for(K in seq(6)) {
  fit <- auto.arima(lpower.ts, xreg=fourier(lpower.ts,K=K),
                    seasonal=FALSE)
  if(fit[["aicc"]] < bestfit[["aicc"]]) {
    bestfit <- fit
    bestK <- K
  }
}
bestfit

## Series: lpower.ts
## Regression with ARIMA(2,1,2) errors
##
## Coefficients:
##          ar1      ar2     ma1     ma2   drift    S1-12    C1-12   S2-12
##      -0.7057  -0.8678  0.6814  0.9459  0.0018  -0.0283  -0.0274  0.0163
## s.e.  0.0458   0.0569  0.0284  0.0431  0.0004   0.0011   0.0011  0.0006
##        C2-12   S3-12    C3-12   S4-12   C4-12    S5-12    C5-12
##      -0.0013  0.0017  -0.0041  0.0019  0.0072  -0.0024  -0.0039
## s.e.  0.0006  0.0004   0.0004  0.0004  0.0004   0.0003   0.0003
##
```

Evaluate the fit

Do I get white noise errors from the model I called "bestfit"? Might need additional seasonal AR(1):
arima(lpower.ts,order=c(2,1,2),seasonal=list(order=c(1,0,0)), xreg=fourier(lpower.ts,K=5))



Forecasting

Hold back 65 periods

```
lpower.hold<-ts(lpower.ts[-c(352:416)],freq=12)
fit_no_holds3 <- Arima(lpower.ts[-c(352:416)],order=c(2,1,2),include.drift = TR
                       seasonal = list(order=c(1,0,0)),
                       xreg=fourier(lpower.hold,K=c(5)))
fcast_no_holds3 <- forecast(fit_no_holds3,
                            xreg=fourier(lpower.hold,K=c(5),h=80))
plot(fcast_no_holds3,main="SARIMA(2,1,2)(1,0,0),Fourier",include=50)
lines(ts(lpower.ts))
```