STAT 436 / 536 - Lecture 10

October 5, 2018

Regression with Seasonality

- We have seen the presence of seasonality in several of the datasets we have considered. This section focuses on regression techniques for seasonality.

Indicator variables for seasonality

- One approach for seasonality

- Assume we are looking at monthly data (e.g. airline passengers),

- Write out a linear model for seasonality with no trend.

$$x_t = s_t + z_t$$

- This model can be fit in R (for the airline passengers) using the following commands

```
data("AirPassengers")
AirPassengers.df <- data.frame(season = as.factor(as.numeric(cycle(AirPassengers))),
                               count = as.numeric(AirPassengers))
lm(count ~ season - 1, data = AirPassengers.df)
```

```
##
## Call:
## lm(formula = count ~ season - 1, data = AirPassengers.df)
##
## Coefficients:
##  season1   season2   season3   season4   season5   season6   season7
##    241.8     235.0     270.2     267.1     271.8     311.7     351.3
##  season8   season9  season10  season11  season12
##    351.1     302.4     266.6     232.8     261.8
```

- Now consider the same framework with a trend too

- *We can formulate this in an additive framework as*

- This model can be fit in R (for the airline passengers) using the following commands

```
data("AirPassengers")
AirPassengers.df <- data.frame(season = as.factor(as.numeric(cycle(AirPassengers))),
                count = as.numeric(AirPassengers), time = 1:length(AirPassengers))
lm(count ~ time + season - 1, data = AirPassengers.df)
```

```
##
## Call:
## lm(formula = count ~ time + season - 1, data = AirPassengers.df)
##
## Coefficients:
##     time   season1   season2   season3   season4   season5   season6
##     2.66     63.51     54.10     86.60     80.86     82.95    120.12
##  season7   season8   season9  season10  season11  season12
##   157.13    154.22    102.89     64.40     27.99     54.33
```

- Or

- We will discuss fitting this model shortly, but note that it can be acheived with a log transform.

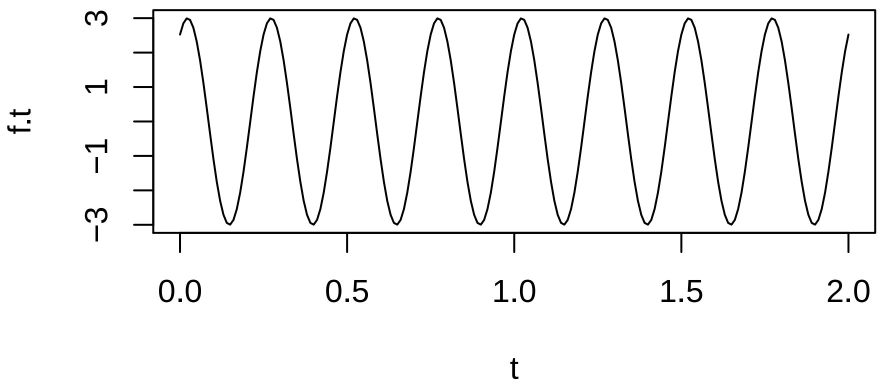
Harmonic seasonal models

- The indicator variables cause a stair-step like seasonal pattern where each season has a step function or a separate intercept. An alternative for a smooth seasonal pattern is to use sine and cosine functions.

- A sine wave with frequency f, phase shift ϕ, and amplitude A can be written as

$$A\sin(2\pi f t + \phi)$$

```
A <- 3
f <- 4
phi <- 1
t <- seq(0,2, by=.01)
f.t <- A * sin(2 * pi * f * t + phi )
plot(t, f.t, type='l')
```



- The representation above is not linear, as ϕ is in the sine function; however, this can be re-expressed as

$$A\sin(2\pi f t + \phi)$$

where $\alpha_s = A\cos(\phi)$ and $\alpha_c = A\sin(\phi)$. This representation is linear in the parameters α_s and α_c and standard techniques (OLS) can be used to estimate these parameters.

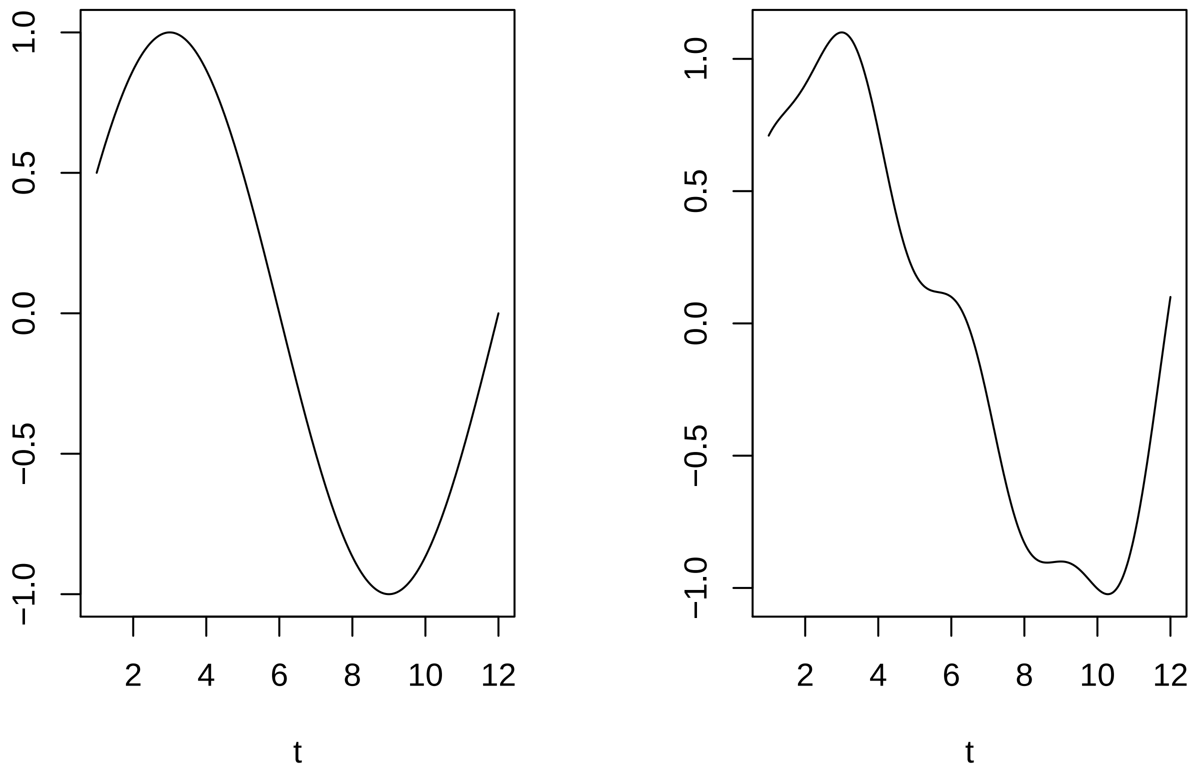
- Using this framework for harmonics with a time series $\{x_t\}$ with s seasons results in $[s/2]$ possible cycles, where [ ] retains the integer.

- This model can be written as

$$x_t = m_t + \sum_{i=1}^{[s/2]} \left(s_i \sin(2\pi i t/s) + c_i \cos(2\pi i t/s)\right) + z_t$$

where

- For instance consider the two curves below



- The first curve is defined by $x_t = \sin(2\pi t/12)$.

- The second curve has harmonic terms with frequencies at $\frac{1}{12}$, $\frac{2}{12}$ and $\frac{4}{12}$ and can be written as:

- Revisiting the equation above, this would equate to:

-$s_1 =$

-$c_1 =$

-$s_2 =$

-$c_2 =$

-$s_3 =$

-$c_3 =$

-$s_4 =$

-$c_4 =$

- Create a figure that contains the four separate harmonic curves (`par(mfcol=c(4,1))` is one way to do this)

- Now we are going to build on the model from above such that

$$
\begin{aligned}
x_t &= .05 * t \\
&+ \sin(2\pi(t)/12) \\
&+ 0.2\sin(2\pi(2t)/12) \\
&+ 0.1\sin(2\pi(4t)/12) \\
&+ 0.1\cos(2\pi(4t)/12) \\
&+ w_t
\end{aligned}
$$

  where $w_t \sim N(\mu = 0, \sigma^2 = 1^2)$. Simulate a time series from this model with a total fo 240 time points.

- The final step is to fit a time series model. Our goal here is learn the values for s_i and c_i. To do so, we need to construct the sine and cosine components. These serve the same purpose as covariates typically denoted as X in a simple regression model.

```
SIN <- COS <- matrix(nrow = length(t), ncol=4) # restricting to 4 components
for (i in 1:4){
  COS[,i] <- cos(2 * pi * i * t / 12)
  SIN[,i] <- sin(2 * pi * i * t / 12)
}

lm.harm <- lm(x ~ t + COS + SIN)
summary(lm.harm)
```

```
##
## Call:
## lm(formula = x ~ t + COS + SIN)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.80990 -0.72876  0.07232  0.71650  2.16092
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0808591  0.1312197  -0.616    0.538
## t            0.1004270  0.0009443 106.349   <2e-16 ***
## COS1         0.1277193  0.0924207   1.382    0.168
## COS2        -0.1225709  0.0924207  -1.326    0.186
## COS3         0.0302556  0.0924207   0.327    0.744
## COS4         0.0092122  0.0924207   0.100    0.921
## SIN1         1.0934354  0.0924830  11.823   <2e-16 ***
## SIN2         0.1371539  0.0924303   1.484    0.139
## SIN3        -0.0011151  0.0924207  -0.012    0.990
## SIN4         0.0915840  0.0924175   0.991    0.323
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.012 on 230 degrees of freedom
## Multiple R-squared:  0.9802, Adjusted R-squared:  0.9794
## F-statistic:  1265 on 9 and 230 DF,  p-value: < 2.2e-16
```