        - The more linearized the coefficients are in the nonlinear model, the easier it is to estimate the model (for the numerical search algorithm)

- If we didn't know *f*, the `nls` version of the model could also attempt to estimate it (but the estimation process is even more dependent on "wise" parameter initializations)

- We can also add other linear terms to nls models in a similar fashion to those in an lm:

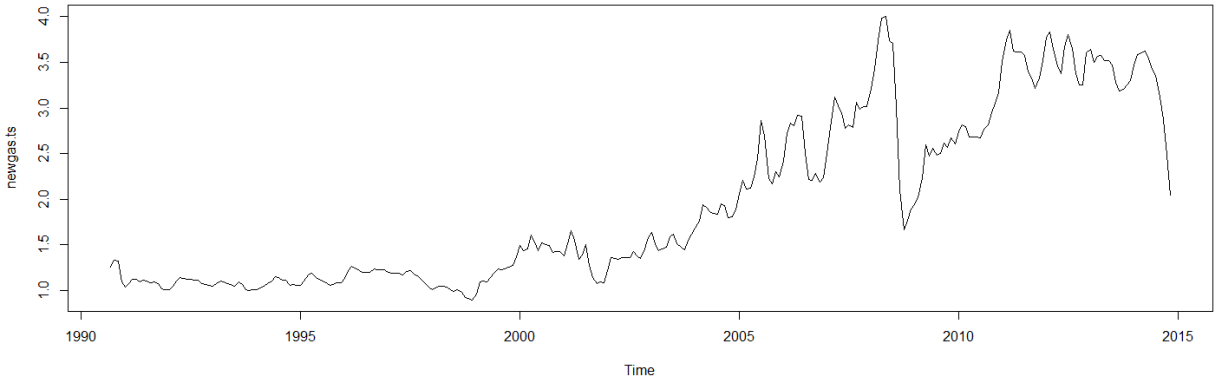
```
> model3_trend<-nls(signalts[,3]~a+b*cos(2*pi*time/12+c)+d*time,start=list(a=
0,b=2,c=0.6*pi,d=0))
> summary(model3_trend)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
a  0.109034   0.980803   0.111  0.91171
b  2.721547   0.682508   3.988  0.00013 ***
c  1.845750   0.255133   7.234 1.14e-10 ***
d -0.005012   0.016861  -0.297  0.76691

Residual standard error: 4.862 on 96 degrees of freedom

Number of iterations to convergence: 3
Achieved convergence tolerance: 1.446e-08
```

CM 5.6 (p101-) adds the details for having additional sine/cosine components at higher harmonic frequencies (mentioned on CC p35-36):

- If the seasonal component has *s* seasons (same as the Period as defined above), there can be [*s/2*] cycles per seasonal
    - [ ] here is the floor function

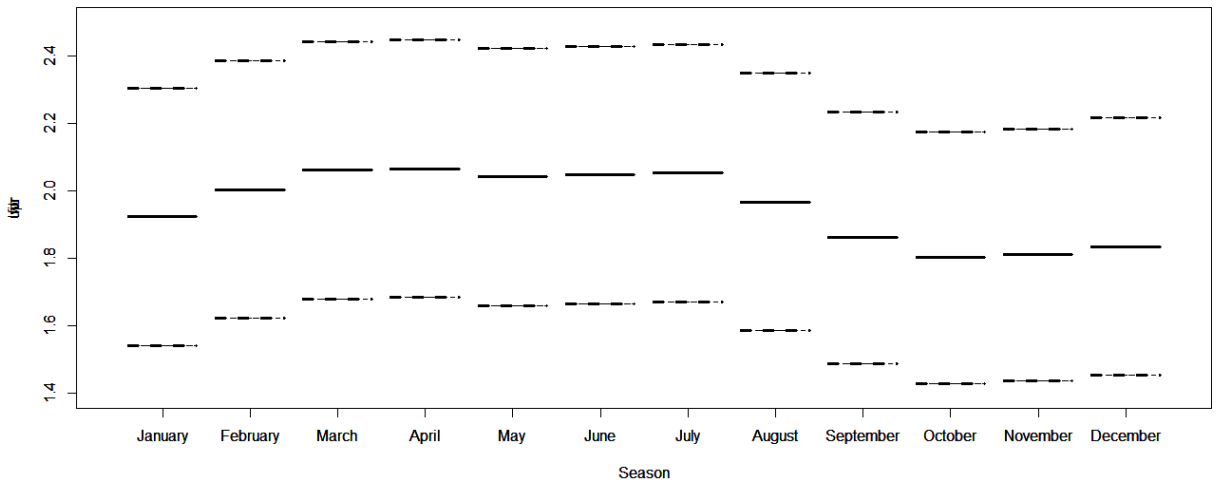
- Add additional linearized sine/cosine functions at higher frequencies of oscillation – perturbs shape of sine function to a possibly more realistic shape
- Basically is using a set of fourier basis functions to do a fourier decomposition of the signal, constrained in frequencies at or above the seasonal cycle
    - Attempts to explain any variation that is occurring seasonally – not just variation that matches a single sine function
    - excludes low frequency – trend type components since the frequency is constrained
    - add the trend in using another model component
- Starts to provide the same flexibility as cyclic cubic splines defined below
- Formula for added seasonal components:

Updated monthly gas price time series:




```
> FRED.GASREGCOVM <- read.csv("https://dl.dropboxusercontent.com/u/77307195/F
RED-GASREGCOVM.csv")
> head(FRED.GASREGCOVM)
       Date Value
1  1/1/2015 2.046
2 12/1/2014 2.488
3 11/1/2014 2.875
4 10/1/2014 3.120
5  9/1/2014 3.354
6  8/1/2014 3.425
> n<-length(FRED.GASREGCOVM[,1])
> newgas<-FRED.GASREGCOVM[n:1,2]
>
> newgas.ts<-ts(newgas,start=c(1990,9),freq=12)
> plot(newgas.ts)
> require(TSA)
> seasonalmeans<-lm(newgas.ts~season(newgas.ts)-1)
> summary(seasonalmeans)
                          Estimate Std. Error t value Pr(>|t|)
season(newgas.ts)January     1.9241     0.1941   9.915   <2e-16 ***
season(newgas.ts)February    2.0045     0.1941  10.329   <2e-16 ***
season(newgas.ts)March       2.0623     0.1941  10.627   <2e-16 ***
season(newgas.ts)April       2.0663     0.1941  10.648   <2e-16 ***
season(newgas.ts)May         2.0426     0.1941  10.526   <2e-16 ***
season(newgas.ts)June        2.0479     0.1941  10.553   <2e-16 ***
season(newgas.ts)July        2.0534     0.1941  10.581   <2e-16 ***
season(newgas.ts)August      1.9673     0.1941  10.138   <2e-16 ***
season(newgas.ts)September   1.8614     0.1901   9.790   <2e-16 ***
season(newgas.ts)October     1.8022     0.1901   9.478   <2e-16 ***
season(newgas.ts)November    1.8108     0.1901   9.523   <2e-16 ***
season(newgas.ts)December    1.8348     0.1941   9.455   <2e-16 ***

Residual standard error: 0.9507 on 279 degrees of freedom
Multiple R-squared:  0.8156,   Adjusted R-squared:  0.8077
F-statistic: 102.8 on 12 and 279 DF,  p-value: < 2.2e-16

> seasonalmeans2<-lm(newgas.ts~season(newgas.ts))
> summary(seasonalmeans2)
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                1.92413    0.19406   9.915   <2e-16 ***
season(newgas.ts)February  0.08037    0.27444   0.293    0.770
season(newgas.ts)March     0.13821    0.27444   0.504    0.615
season(newgas.ts)April     0.14221    0.27444   0.518    0.605
season(newgas.ts)May       0.11850    0.27444   0.432    0.666
season(newgas.ts)June      0.12379    0.27444   0.451    0.652
season(newgas.ts)July      0.12929    0.27444   0.471    0.638
season(newgas.ts)August    0.04321    0.27444   0.157    0.875
season(newgas.ts)September -0.06273    0.27168  -0.231    0.818
```

```
season(newgas.ts)October   -0.12197    0.27168  -0.449    0.654
season(newgas.ts)November  -0.11337    0.27168  -0.417    0.677
season(newgas.ts)December  -0.08938    0.27444  -0.326    0.745

Residual standard error: 0.9507 on 279 degrees of freedom
Multiple R-squared:  0.01158,  Adjusted R-squared:  -0.02739
F-statistic: 0.2971 on 11 and 279 DF,  p-value: 0.9861
> res1<-data.frame(Season=season(newgas.ts),predict(seasonalmeans,interval="c
onfidence"))
> head(res1,20)
      Season      fit      lwr      upr
1  September 1.861400 1.487112 2.235688
2    October 1.802160 1.427872 2.176448
3   November 1.810760 1.436472 2.185048
4   December 1.834750 1.452744 2.216756
5    January 1.924125 1.542119 2.306131
6   February 2.004500 1.622494 2.386506
7      March 2.062333 1.680327 2.444339
8      April 2.066333 1.684327 2.448339
9        May 2.042625 1.660619 2.424631
10      June 2.047917 1.665911 2.429923
11      July 2.053417 1.671411 2.435423
12    August 1.967333 1.585327 2.349339
13 September 1.861400 1.487112 2.235688
14   October 1.802160 1.427872 2.176448
15  November 1.810760 1.436472 2.185048
16  December 1.834750 1.452744 2.216756
17   January 1.924125 1.542119 2.306131
18  February 2.004500 1.622494 2.386506
19     March 2.062333 1.680327 2.444339
20     April 2.066333 1.684327 2.448339
> plot(fit~Season,data=res1[5:16,],ylim=c(1.4,2.5))
> plot(upr~Season,data=res1[5:16,],add=T,lty=2,col="red")
> plot(lwr~Season,data=res1[5:16,],add=T,lty=2,col="red")
```




```
> harmonic1<-lm(newgas.ts~harmonic(newgas.ts))
> summary(harmonic1)
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                  1.95631    0.05488  35.647   <2e-16 ***
harmonic(newgas.ts)cos(2*pi*t) -0.04953    0.07787  -0.636   0.5252
harmonic(newgas.ts)sin(2*pi*t)  0.12766    0.07735   1.650   0.0999 .

Residual standard error: 0.9361 on 288 degrees of freedom
Multiple R-squared:  0.01075,  Adjusted R-squared:  0.003877
F-statistic: 1.564 on 2 and 288 DF,  p-value: 0.211
```

```
> harmonic6<-lm(newgas.ts~harmonic(newgas.ts,m=6))
> AIC(seasonalmeans,harmonic1,harmonic6)
              df      AIC
seasonalmeans 13 810.1389
harmonic1      4 792.3833
harmonic6     13 810.1389
```

- Maybe we need to “detrend” the gas series or decompose it into a long term trend and seasonal trend:

- Polynomial trends are interesting but can be inefficient approximators to complicated trends.
  - Higher order polynomials can become numerically unstable (see our exploration of multi-collinearity in polynomial linear models)
  - Perhaps some lower order components are not needed, but it is not common practice to delete lower order terms and retain higher order terms (think of x^2 as x*x)

- A “better” option: Smoothing spline trend + seasonal component
  - Most easily accomplished in the Generalized Additive Model (GAM) or Semiparametric framework.
  - **Generalized**: Exponential family distributions (normal, binomial, poisson, negative binomial, gamma most common) are possible
  - **Additive** components (s(x)) are nonparametric, smoothness estimated within the model
  - Generalized Additive **Mixed** Models (GAMMs) allow for correlated or random effects in the models.
    - Be careful with AICs extracted from the mgcv packages gamm function
  - For each model component, define a set of basis functions:
    - Design matrix generated when each basis is evaluated for all values in x
    - Cubic spline, thin plate spline common choices
      - truncated power and fourier bases also can used but not as common
    - Circular cubic spline:

  - Then estimate the spline coefficients, subject to some sort of roughness penalty, selected to optimize the model by generalized cross-validation or… (see STAT 448 for connections to mixed models)

  - Estimated effect(s) is based on multiplying the basis functions by their estimated spline coefficients.