Peru Lima Temperature

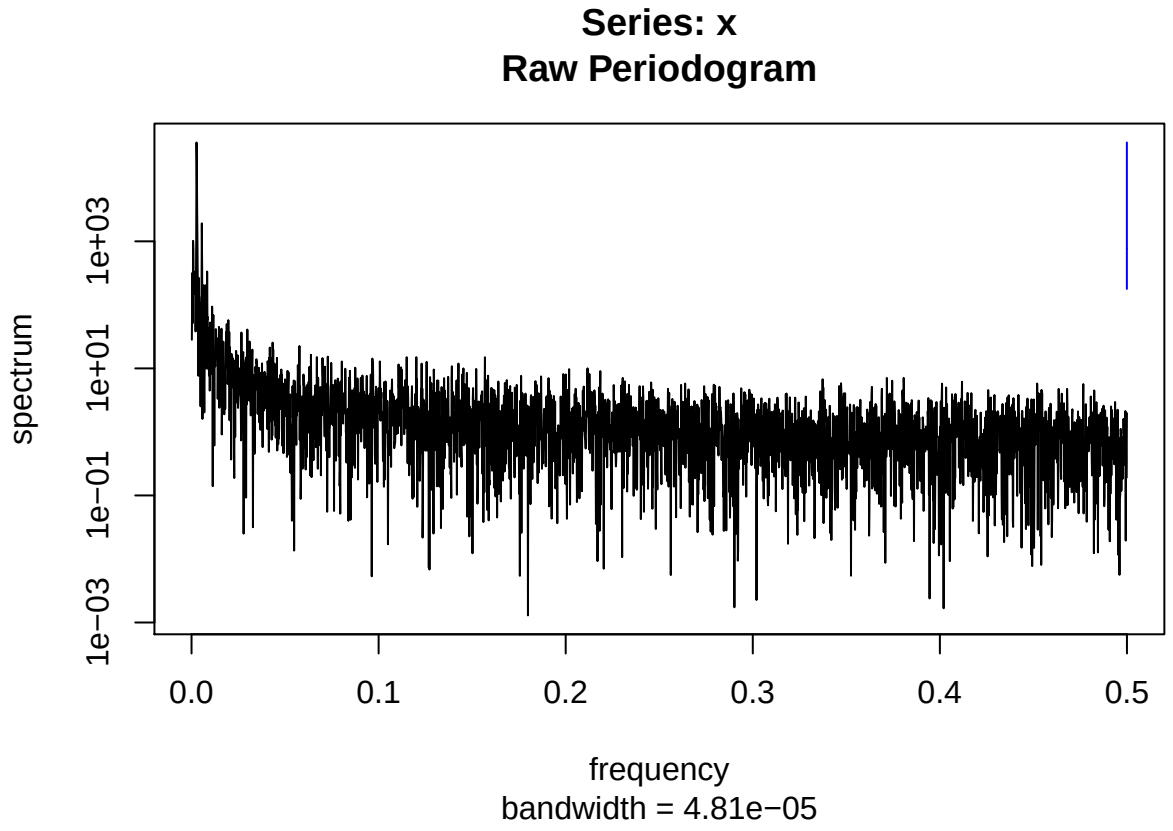
Jing Xu

April 14, 2017

```
setwd('C:/Users/zmxj/Documents/stocasticProject')
data<-read.csv("Peru3.csv")
y<-data[,3]
y<-as.numeric(as.character(y))
t<-1:length(y)
data$date<-as.Date(data$date,"%m/%d/%Y")
date<-data$date
```

Linear fitting for Peru [1]

```
ssp <- spectrum(y)
```



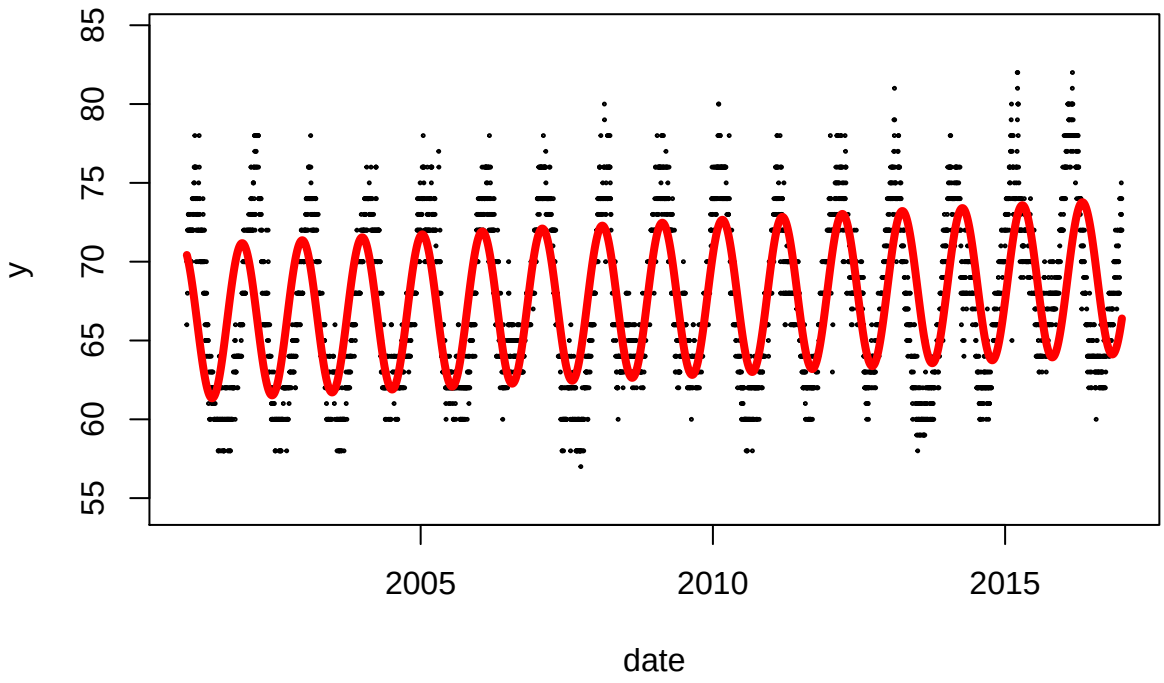

```
per <- 1/ssp$freq[ssp$spec==max(ssp$spec)]
reslm <- lm(y ~ sin(2*pi/per*t)+cos(2*pi/per*t))
summary(reslm)
```

```
##
## Call:
## lm(formula = y ~ sin(2 * pi/per * t) + cos(2 * pi/per * t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2680  -2.5709  -0.0755   2.3899  12.1618
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          67.57438    0.04907 1377.06   <2e-16 ***
## sin(2 * pi/per * t) -2.23322    0.06955  -32.11   <2e-16 ***
## cos(2 * pi/per * t)  4.31729    0.06922   62.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.75 on 5841 degrees of freedom
## Multiple R-squared:  0.4567, Adjusted R-squared:  0.4565
## F-statistic:  2455 on 2 and 5841 DF,  p-value: < 2.2e-16
```

```
rg <- diff(range(y))
plot(y~date,ylim=c(min(y)-0.1*rg,max(y)+0.1*rg),cex=0.2)
# including 2nd harmonic really improves the fit
reslm2 <- lm(y ~ t+sin(2*pi/per*t)+cos(2*pi/per*t))#+sin(4*pi/per*t)+cos(4*pi/per*t))
summary(reslm2)
```

```
##
## Call:
## lm(formula = y ~ t + sin(2 * pi/per * t) + cos(2 * pi/per * t))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7169  -2.5709   0.0368   2.4029  12.8530
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.613e+01  9.568e-02  691.13   <2e-16 ***
## t                    4.943e-04  2.835e-05   17.43   <2e-16 ***
## sin(2 * pi/per * t) -2.229e+00  6.781e-02  -32.87   <2e-16 ***
## cos(2 * pi/per * t)  4.333e+00  6.750e-02   64.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.656 on 5840 degrees of freedom
## Multiple R-squared:  0.4836, Adjusted R-squared:  0.4833
## F-statistic:  1823 on 3 and 5840 DF,  p-value: < 2.2e-16
```

```
lines(fitted(reslm2)~date,col='red',lwd=4)    # solid red line is periodic with second harmonic
```

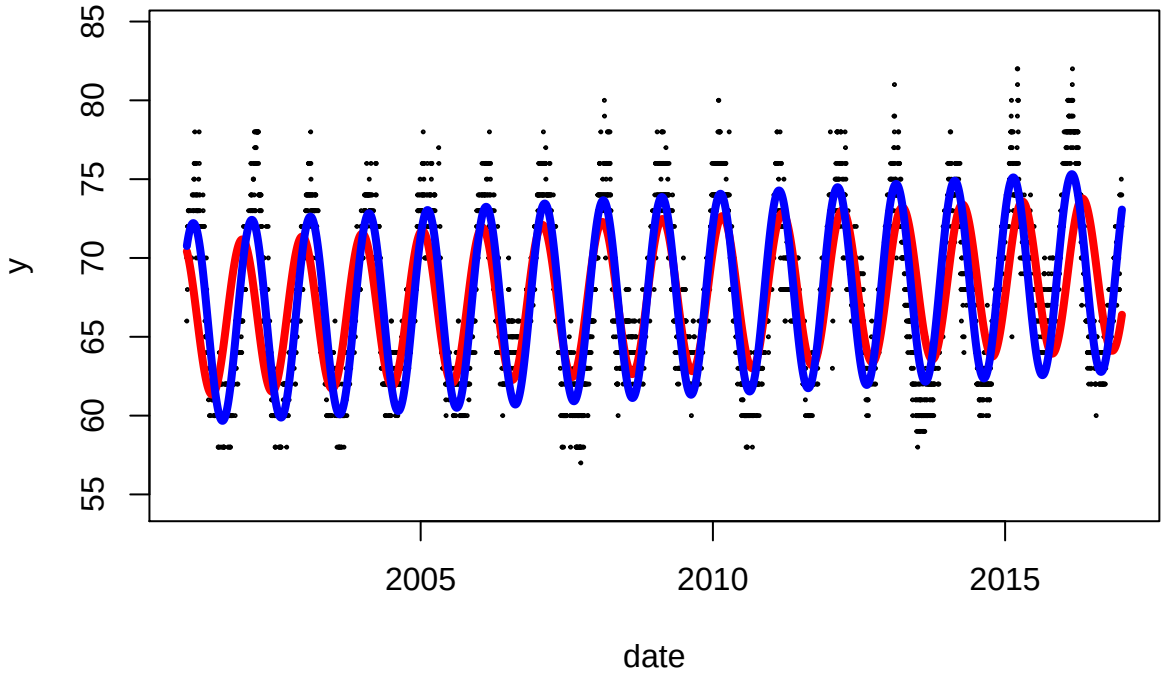


Nonlinear fitting

```
W = 2*pi/per
reslm_1 <- nls(y ~ cons+A*t+B*sin(W*t)+C*cos(W*t), start=list(cons=66.13, A=0.0004943, B=-2.229,W=W,C=4
summary(reslm_1)
```

```
##
## Formula: y ~ cons + A * t + B * sin(W * t) + C * cos(W * t)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## cons 6.587e+01  6.015e-02 1095.05   <2e-16 ***
## A    5.654e-04  1.783e-05   31.72   <2e-16 ***
## B    4.088e+00  7.089e-02   57.67   <2e-16 ***
## W    1.717e-02  3.986e-06 4307.08   <2e-16 ***
## C    4.823e+00  6.419e-02   75.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.296 on 5839 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 3.867e-06
```

```r
plot(y~date,ylim=c(min(y)-0.1*rg,max(y)+0.1*rg),cex=0.2)
lines(fitted(reslm2)~date,col='red',lwd=4)
lines(fitted(reslm_1)~date,col='blue',lwd=4)
```




```r
fitmean2 <- fitted(reslm_1)
```

Sigma

```r
sigma1=sd(diff(y)) #using difference of y, the one on the paper
sigma2=sd(y-fitted(reslm2)) #approximate residual standard error
```

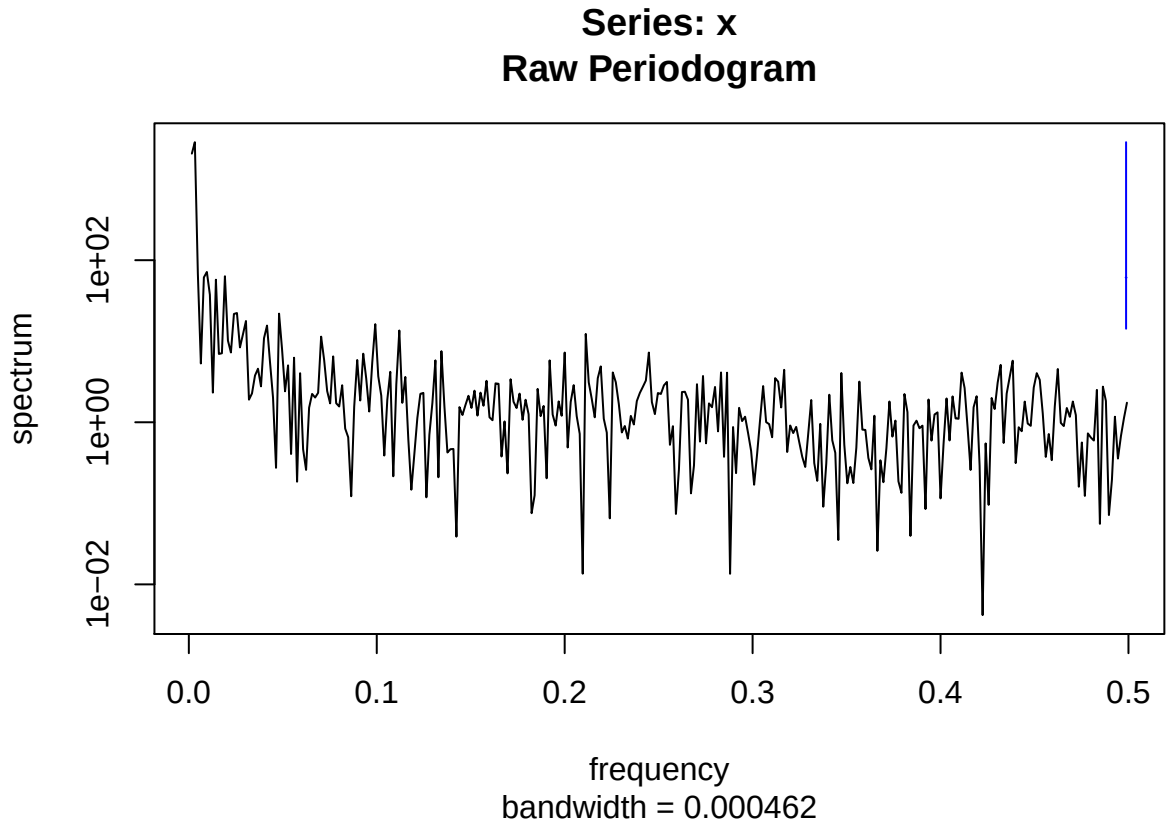
Setting the trigger

```r
#El Nino
for (i in 366:(length(y)-90)){
diff=mean(y[i:(i+30)])-mean(y[(i-365):(i-365+30)])
diff2=mean(y[(i+30):(i+60)])-mean(y[(i+30-365):(i-365+60)])
diff3=mean(y[(i+60):(i+90)])-mean(y[(i+60-365):(i-365+90)])
if (diff>0.1){
```

```
  if (diff2>0.1){
    if (diff3>0.1){
      {
        #print(date[i])
        #print(i)
  }}}}}
#It prints i from 4906 to 5515, which is from 2014/6/8 to 2016/2/6
```

Linear Modeling for El Nino[2]

```
y2<-y[4906:5515]
t2<-4906:5515
date2<-data$date[4906:5515]
ssp <- spectrum(y2)
```

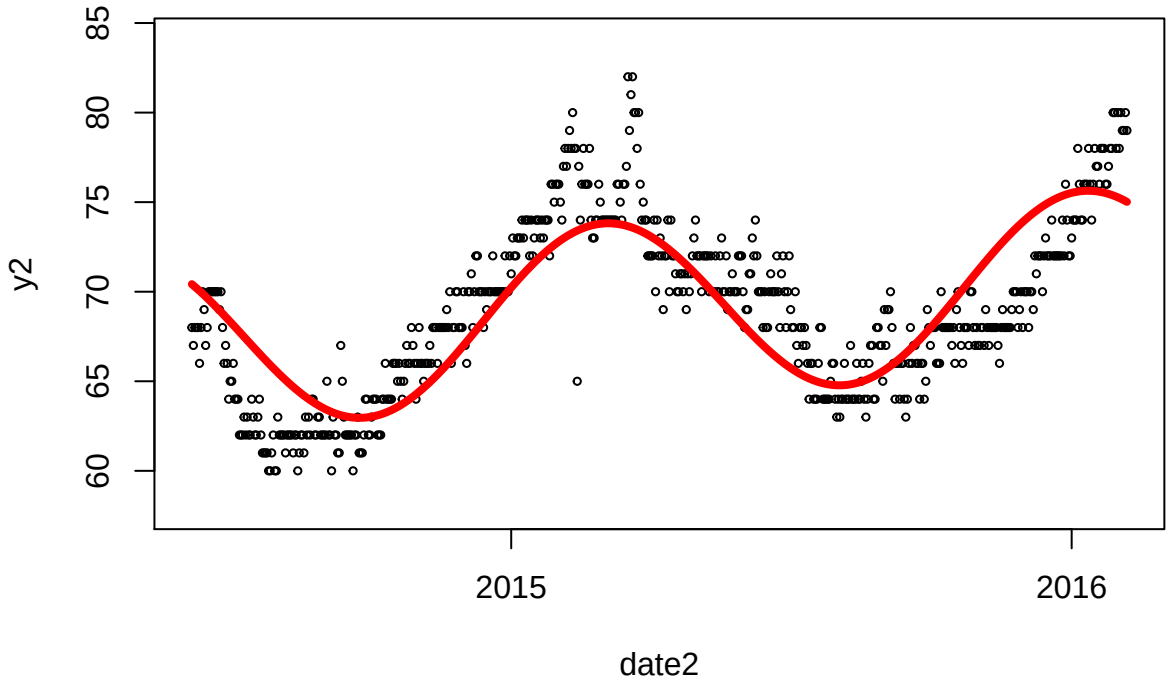



```
per <- 1/ssp$freq[ssp$spec==max(ssp$spec)]

rg <- diff(range(y2))
plot(y2~date2,ylim=c(min(y2)-0.1*rg,max(y2)+0.1*rg),cex=0.5)


# including 2nd harmonic really improves the fit
```

```r
reslm4 <- lm(y2 ~ t2+sin(2*pi/per*t2)+cos(2*pi/per*t2))#+sin(4*pi/per*t)+cos(4*pi/per*t))
summary(reslm4)
```

```
##
## Call:
## lm(formula = y2 ~ t2 + sin(2 * pi/per * t2) + cos(2 * pi/per *
##     t2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4101 -1.9604  0.1918  1.9867  8.4258
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)          38.7837446  3.6248827  10.699  < 2e-16 ***
## t2                    0.0058093  0.0006954   8.354 4.49e-16 ***
## sin(2 * pi/per * t2) -1.7693087  0.1630322 -10.853  < 2e-16 ***
## cos(2 * pi/per * t2) -4.6415275  0.1689548 -27.472  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.772 on 606 degrees of freedom
## Multiple R-squared:  0.6668, Adjusted R-squared:  0.6651
## F-statistic: 404.2 on 3 and 606 DF,  p-value: < 2.2e-16
```

```r
lines(fitted(reslm4)~date2,col="red",lwd=4)
```

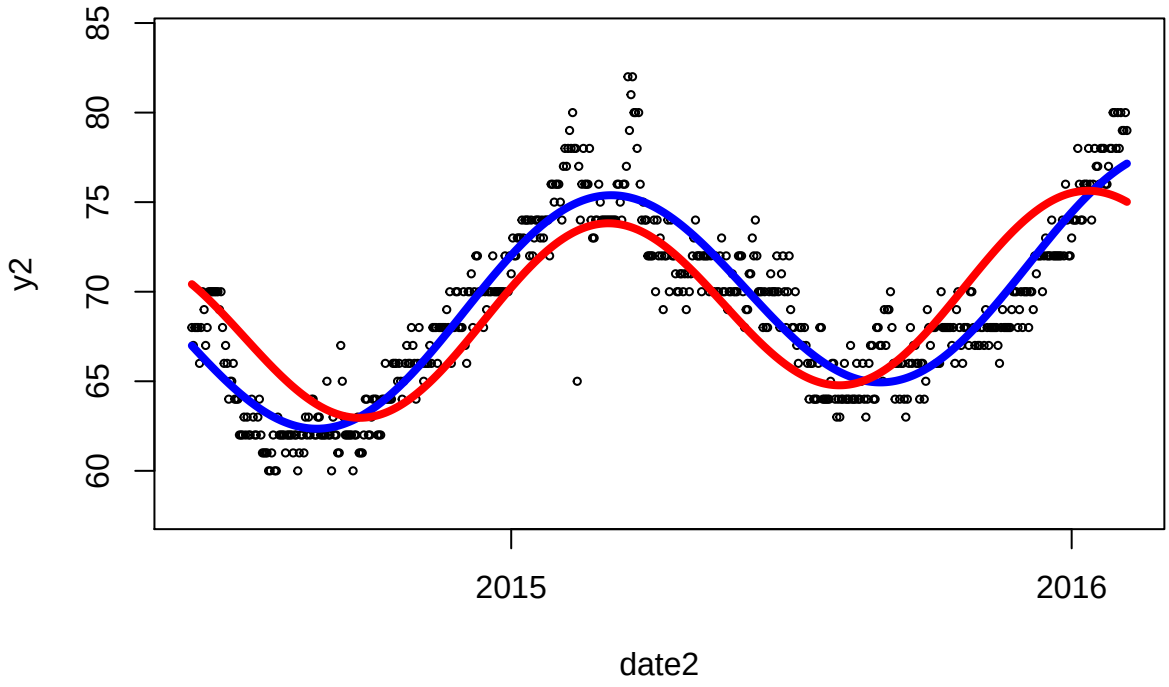


Nonlinear fit for El Nino

```
reslm_2 <- nls(y2 ~ cons+A*t2+B*sin(W*t2)+C*cos(W*t2), start=list(cons=38.7837446, A=0.0058093, B=-1.769
summary(reslm_2)
```

```
##
## Formula: y2 ~ cons + A * t2 + B * sin(W * t2) + C * cos(W * t2)
##
## Parameters:
##       Estimate Std. Error t value Pr(>|t|)
## cons 3.306e+01  2.470e+00  13.386   <2e-16 ***
## A    7.045e-03  4.748e-04  14.838   <2e-16 ***
## B    3.438e+00  2.670e+00   1.288   0.1982
## W    1.712e-02  1.080e-04 158.466   <2e-16 ***
## C    4.741e+00  1.934e+00   2.452   0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.89 on 605 degrees of freedom
##
## Number of iterations to convergence: 18
## Achieved convergence tolerance: 2.764e-06
```

```
plot(y2~date2,ylim=c(min(y2)-0.1*rg,max(y2)+0.1*rg),cex=0.5)
lines(fitted(reslm_2)~date2,col='blue',lwd=4)
lines(fitted(reslm4)~date2,col="red",lwd=4)
```




```
fitmean2 <- fitted(reslm_2)
```

Sigma

```
sigma3=sd(diff(y2)) #using difference of y, the one on the paper
sigma4=sd(y2-fitted(reslm_2)) #approximate residual standard error
```

reference: [1][2] R code, http://stats.stackexchange.com/questions/60994/fit-a-sinusoidal-term-to-data, April 7, 2017