Regression Diagnostics

Yalin Yang

2020-05-03

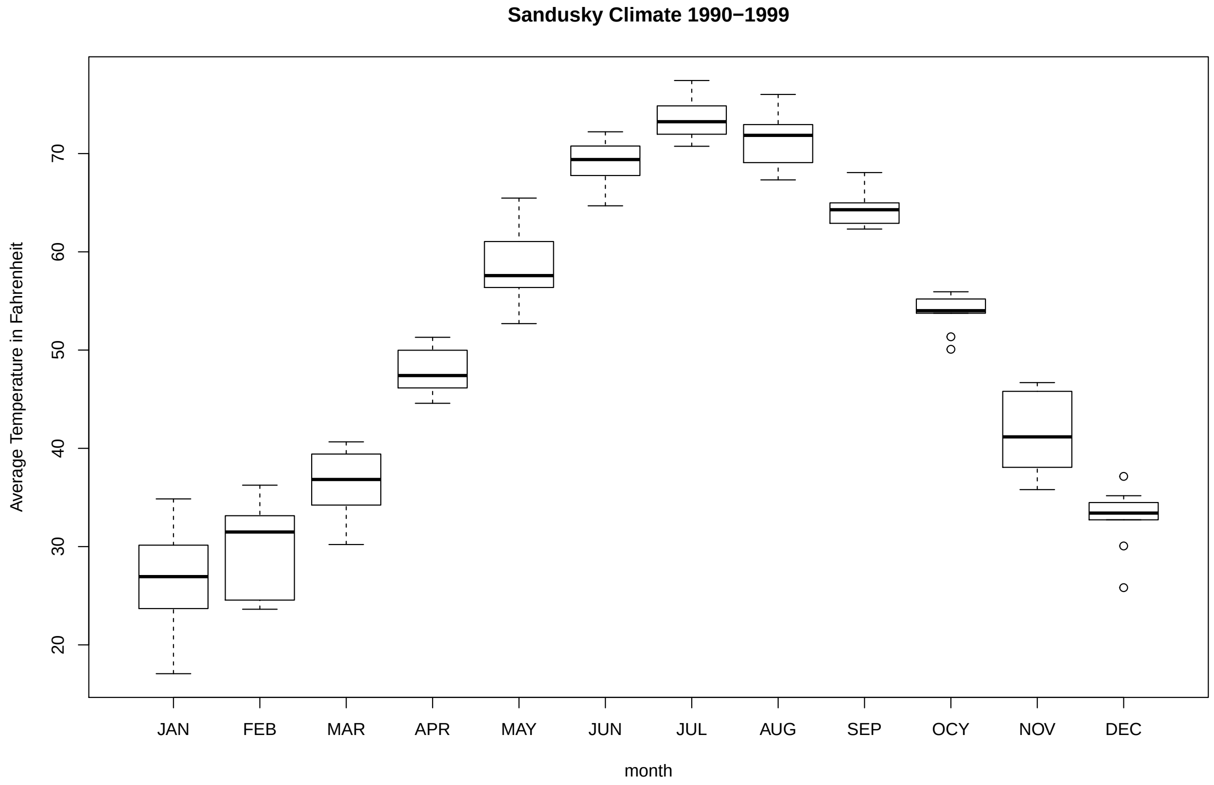
Contents

Standard Regression Diagnositcs

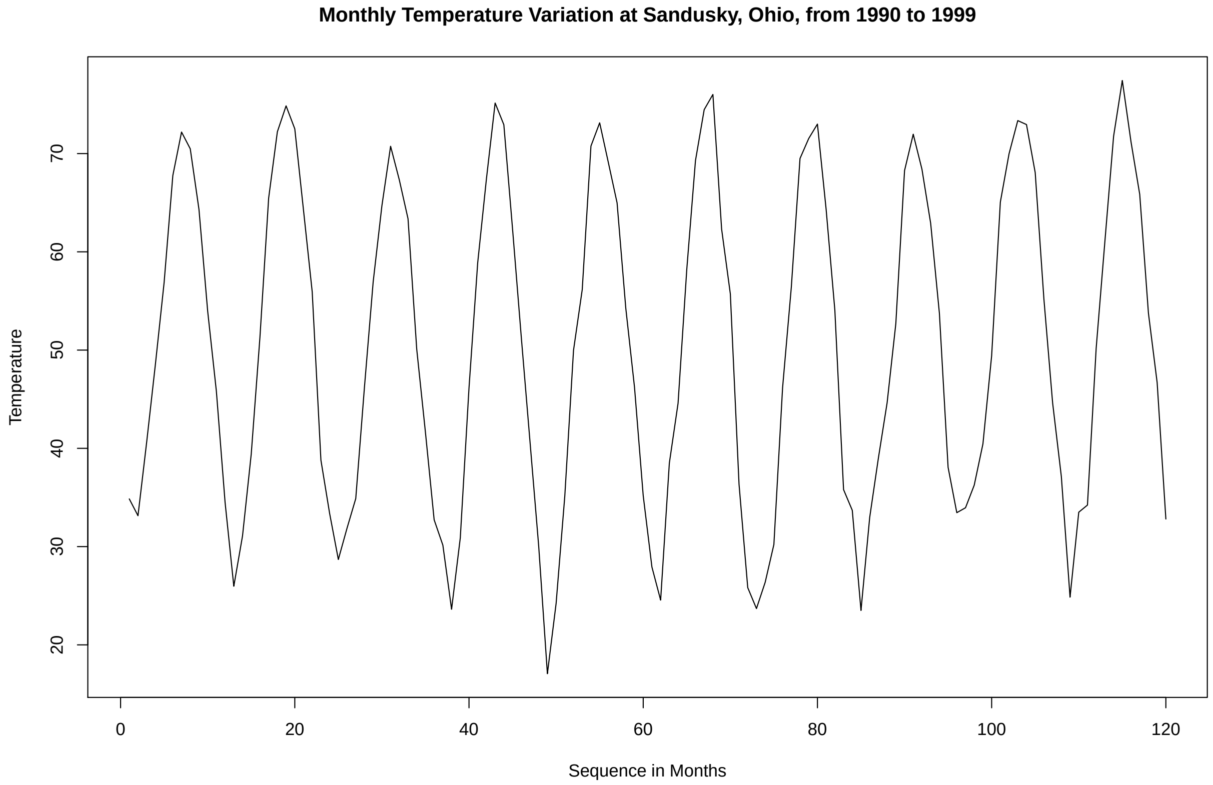
Quick view of dataset

Evaluate monthly cycle and variance heterogeneity

```r
sandusky <- foreign::read.spss("SanduskyTemperature.sav", use.value.labels = TRUE, to.data.frame = TRUE
boxplot(avg7447~month, data=sandusky, ylab="Average Temperature in Fahrenheit", main="Sandusky Climate
```



```
plot(avg7447~time.idx, data=sandusky, main="Monthly Temperature Variation at Sandusky, Ohio, from 1990
     xlab="Sequence in Months", ylab="Temperature", type="l")
```



Generate harmonic variables and add them to the data-frame

Fouier regression with 2 wave parameters

```
sandusky$r.cos <- cos(sandusky$time.idx/12*2*pi)
sandusky$r.sin <- sin(sandusky$time.idx/12*2*pi)
fourier1.lm <- lm(avg7447~time.idx+r.cos+r.sin, data=sandusky)
summary(fourier1.lm,cor=T)
```

```
##
## Call:
## lm(formula = avg7447 ~ time.idx + r.cos + r.sin, data = sandusky)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6296  -2.1067   0.1529   2.2148   7.4496
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  50.145855   0.600117  83.560   <2e-16 ***
## time.idx      0.006018   0.008615   0.699    0.486
## r.cos       -18.214306   0.420795 -43.285   <2e-16 ***
## r.sin       -13.945078   0.421934 -33.050   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.259 on 116 degrees of freedom
## Multiple R-squared:  0.9625, Adjusted R-squared:  0.9615
## F-statistic:   992 on 3 and 116 DF,  p-value: < 2.2e-16
##
## Correlation of Coefficients:
##          (Intercept) time.idx r.cos
## time.idx -0.87
## r.cos     0.02       -0.02
## r.sin    -0.07        0.08     0.00
```

Variance inflation factors (reported in variance NOT std)

```
library(car)
vif(fourier1.lm)
```

```
## time.idx    r.cos    r.sin
## 1.006259 1.000419 1.005840
```

covariance among estimated parameters

```
round(vcov(fourier1.lm),2)
```

```
##             (Intercept) time.idx r.cos r.sin
## (Intercept)        0.36        0  0.00 -0.02
## time.idx           0.00        0  0.00  0.00
## r.cos              0.00        0  0.18  0.00
## r.sin             -0.02        0  0.00  0.18
```

Fixed effect panel model

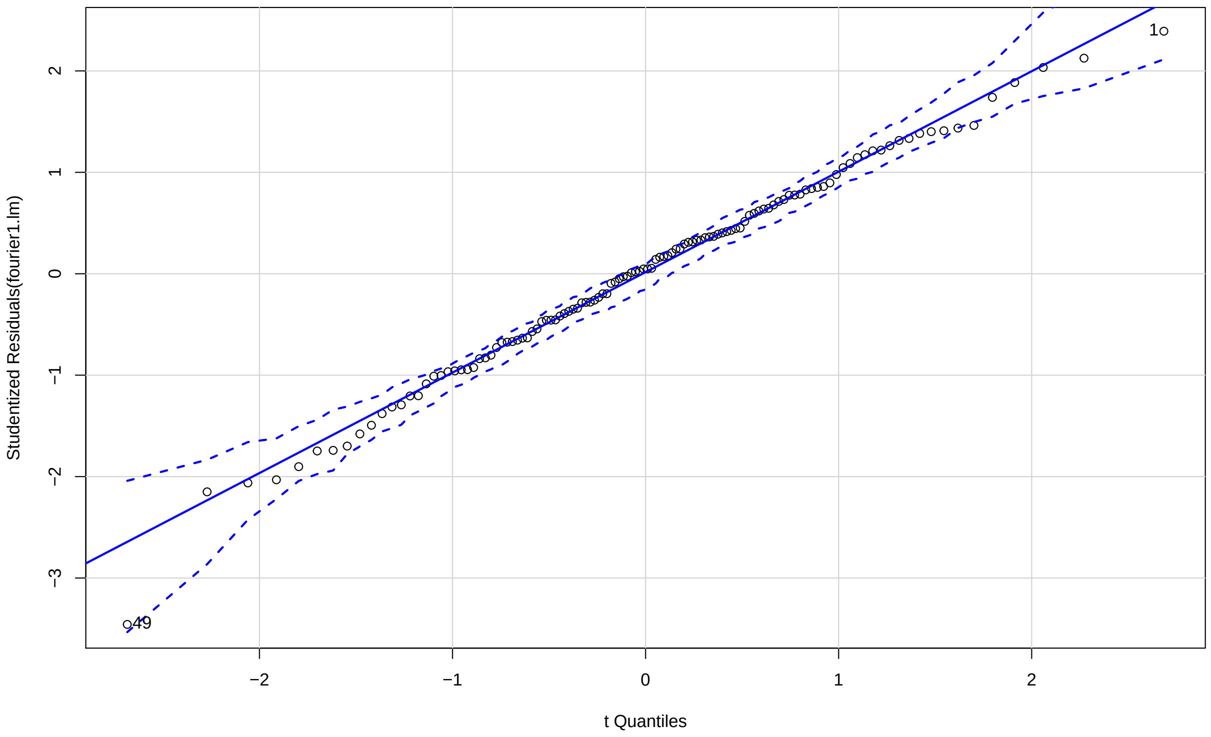
```
month.lm <- lm(avg7447~time.idx + month, data=sandusky)
summary(month.lm)
```

```
##
## Call:
## lm(formula = avg7447 ~ time.idx + month, data = sandusky)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9657 -2.0025  0.1849  2.0056  8.0987
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.750473   1.172385  22.817  < 2e-16 ***
## time.idx     0.005710   0.008918   0.640 0.523387
## monthFEB     2.687840   1.505938   1.785 0.077123 .
## monthMAR     9.266000   1.506018   6.153 1.35e-08 ***
## monthAPR    20.643297   1.506150  13.706  < 2e-16 ***
## monthMAY    31.686498   1.506335  21.035  < 2e-16 ***
## monthJUN    42.071130   1.506572  27.925  < 2e-16 ***
## monthJUL    46.383093   1.506862  30.781  < 2e-16 ***
## monthAUG    44.285837   1.507206  29.383  < 2e-16 ***
## monthSEP    37.158138   1.507601  24.647  < 2e-16 ***
## monthOCY    26.691998   1.508050  17.700  < 2e-16 ***
## monthNOV    14.347351   1.508551   9.511 6.54e-16 ***
## monthDEC     5.746869   1.509104   3.808 0.000234 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.367 on 107 degrees of freedom
## Multiple R-squared:  0.963,  Adjusted R-squared:  0.9589
## F-statistic: 232.4 on 12 and 107 DF,  p-value: < 2.2e-16
```

Diagnostic plots

Test of normality based on t-distribution

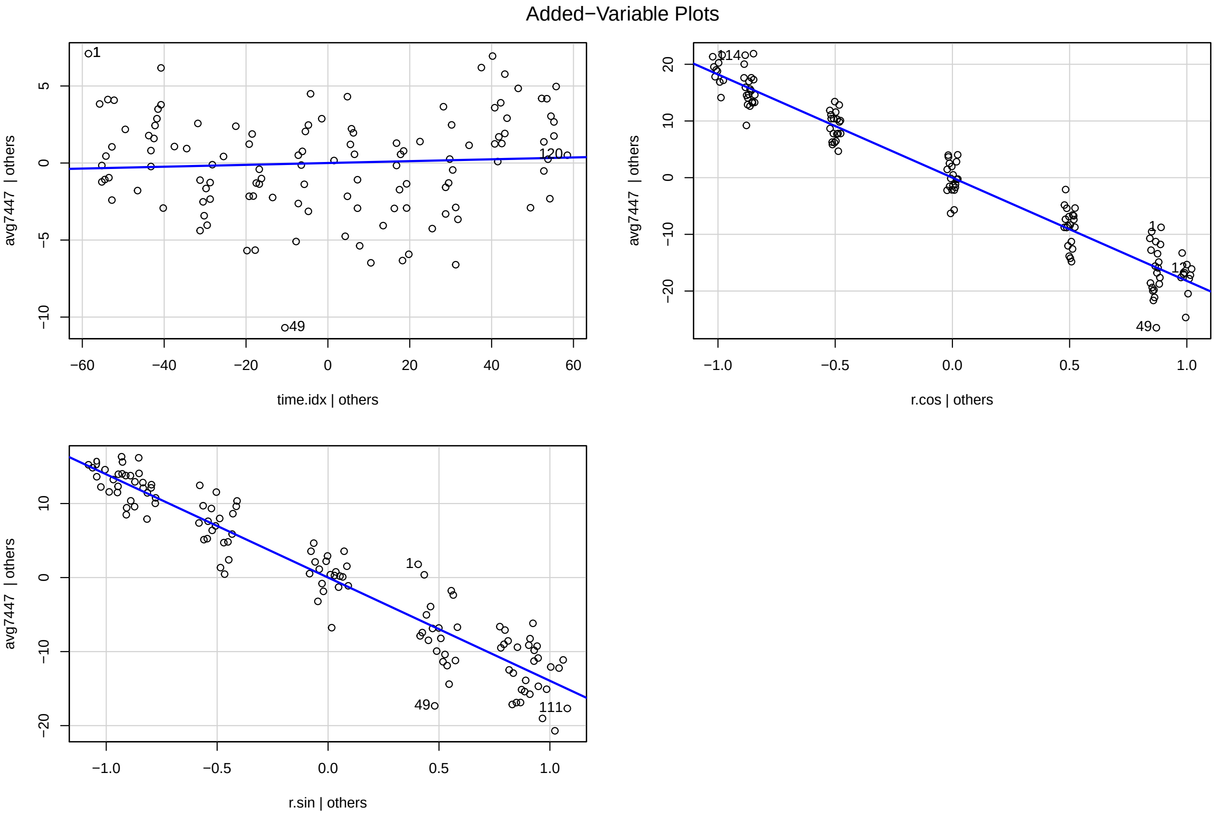
```
qqPlot(fourier1.lm)
```



```
## [1]  1 49
```

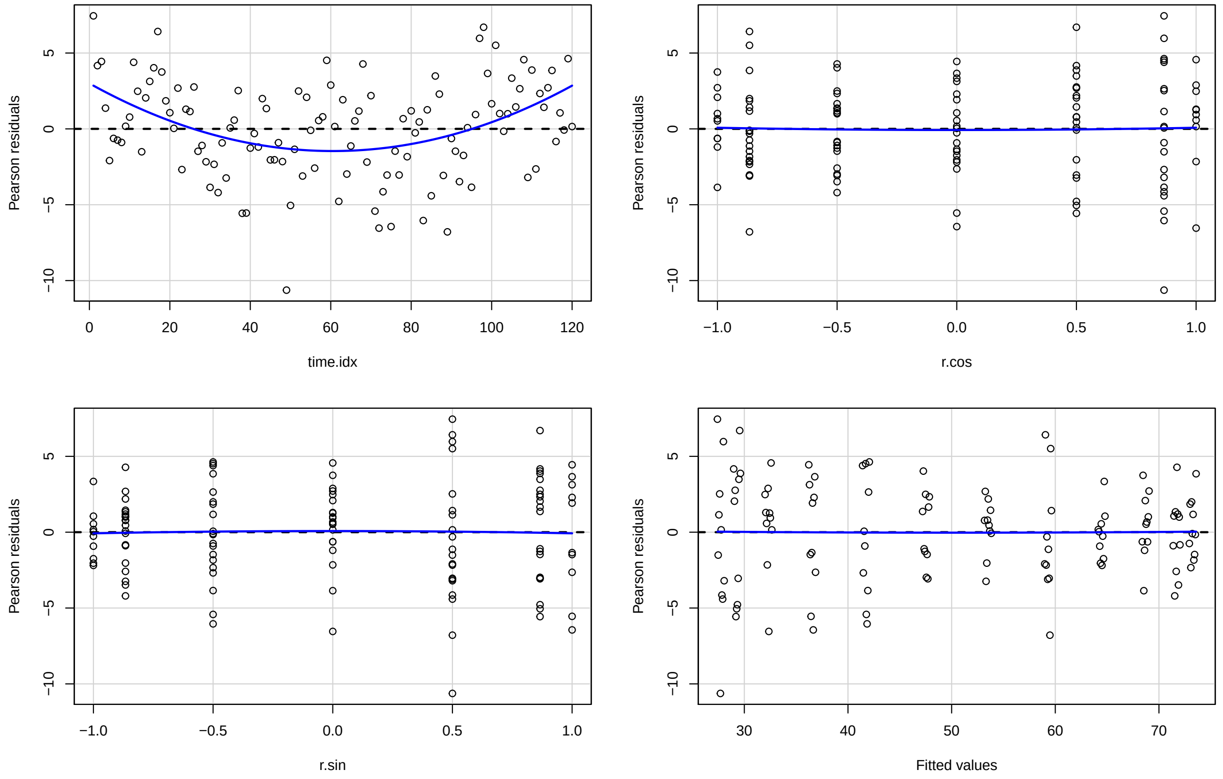
Partial effects plots

```
avPlots(fourier1.lm)
```



Residual plots (Tukey test)

```
residualPlots(fourier1.lm)
```



```
##              Test stat Pr(>|Test stat|)
## time.idx        4.7907        5.001e-06 ***
## r.cos           0.1760           0.8606
## r.sin          -0.1760           0.8606
## Tukey test      0.0769           0.9387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Update the model by adding I(time.idx^2)

```
fourier2.lm <- update(fourier1.lm, .~.+I(time.idx^2))
summary(fourier2.lm)
```

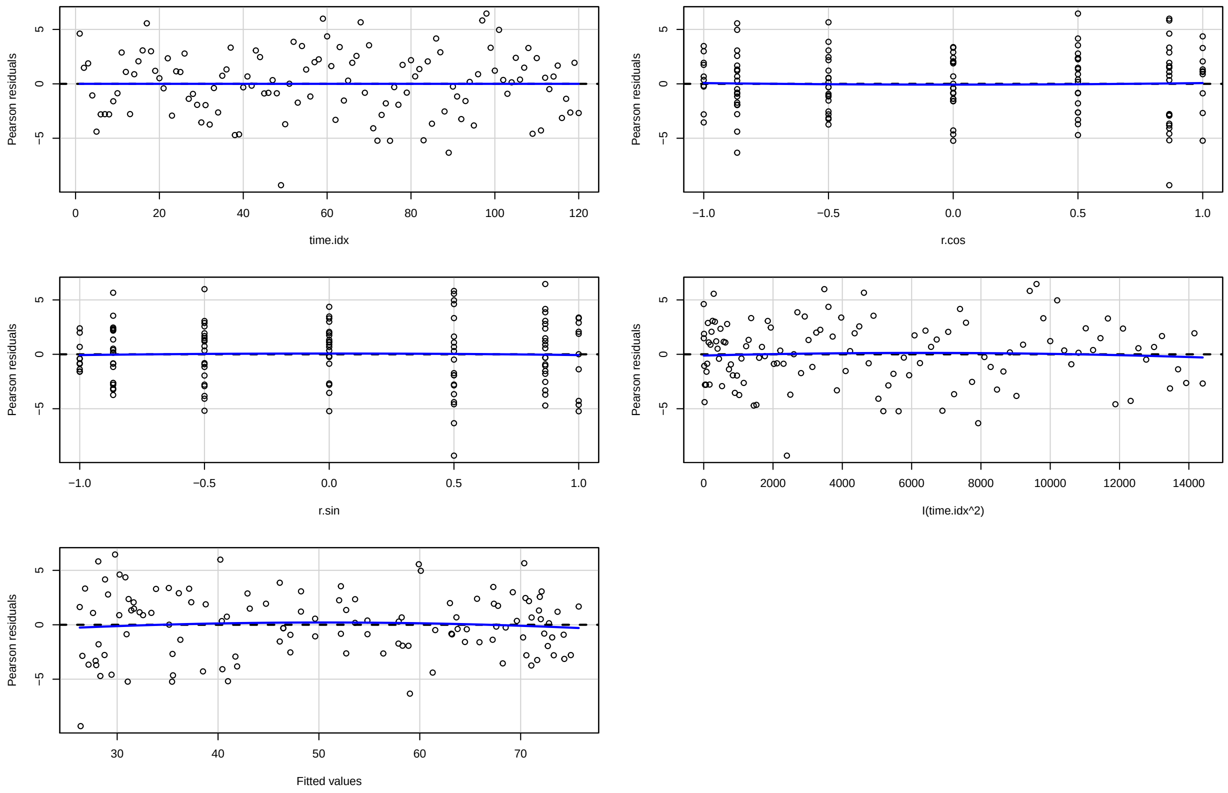
```
##
## Call:
## lm(formula = avg7447 ~ time.idx + r.cos + r.sin + I(time.idx^2),
##     data = sandusky)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3124 -1.9355  0.2357  2.0649  6.4595
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.314e+01  8.331e-01  63.790  < 2e-16 ***
## time.idx      -1.413e-01  3.176e-02  -4.451 1.99e-05 ***
## r.cos         -1.823e+01  3.859e-01 -47.246  < 2e-16 ***
## r.sin         -1.395e+01  3.869e-01 -36.054  < 2e-16 ***
## I(time.idx^2)  1.218e-03  2.542e-04   4.791 5.00e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.988 on 115 degrees of freedom
## Multiple R-squared:  0.9687, Adjusted R-squared:  0.9676
## F-statistic: 890.5 on 4 and 115 DF,  p-value: < 2.2e-16
```

recheck for non-linearity

```
anova(fourier1.lm, fourier2.lm)
```

```
## Analysis of Variance Table
##
## Model 1: avg7447 ~ time.idx + r.cos + r.sin
## Model 2: avg7447 ~ time.idx + r.cos + r.sin + I(time.idx^2)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    116 1231.9
## 2    115 1026.9  1    204.95 22.951 5.001e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
residualPlots(fourier2.lm)
```



```
##                Test stat Pr(>|Test stat|)
## time.idx          1.5857           0.1156
## r.cos             0.1825           0.8555
## r.sin            -0.1825           0.8555
## I(time.idx^2)    -0.7468           0.4567
## Tukey test       -0.5560           0.5782
```

Get residuals

```
(resid <- residuals(fourier2.lm))[1:10]
```

```
##          1          2          3          4          5          6          7
##  4.6162929  1.4748116  1.8817612 -1.0645723 -4.3927822 -2.8026027 -2.7863866
##          8          9         10
## -2.8019031 -1.6017273 -0.8640463
```

```
(std.resid <- rstandard(fourier2.lm))[1:10]
```

```
##          1          2          3          4          5          6          7
##  1.6176848  0.5151438  0.6555234 -0.3700684 -1.5245863 -0.9714969 -0.9648668
##          8          9         10
## -0.9691955 -0.5533335 -0.2980202
```

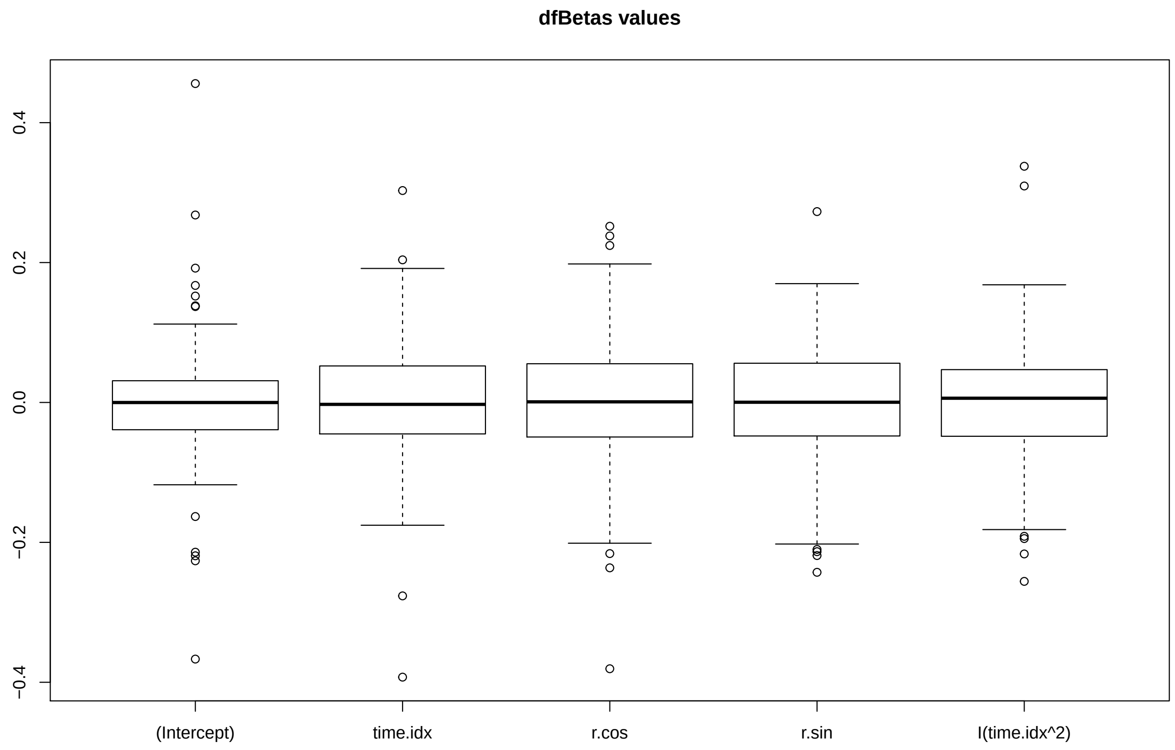
```
(student.resid <- rstudent(fourier2.lm))[1:10]
```

```
##          1          2          3          4          5          6          7
##  1.6292804  0.5134919  0.6538898 -0.3686754 -1.5335200 -0.9712575 -0.9645748
##          8          9         10
## -0.9689378 -0.5516573 -0.2968362
```

Other diagnositic measures

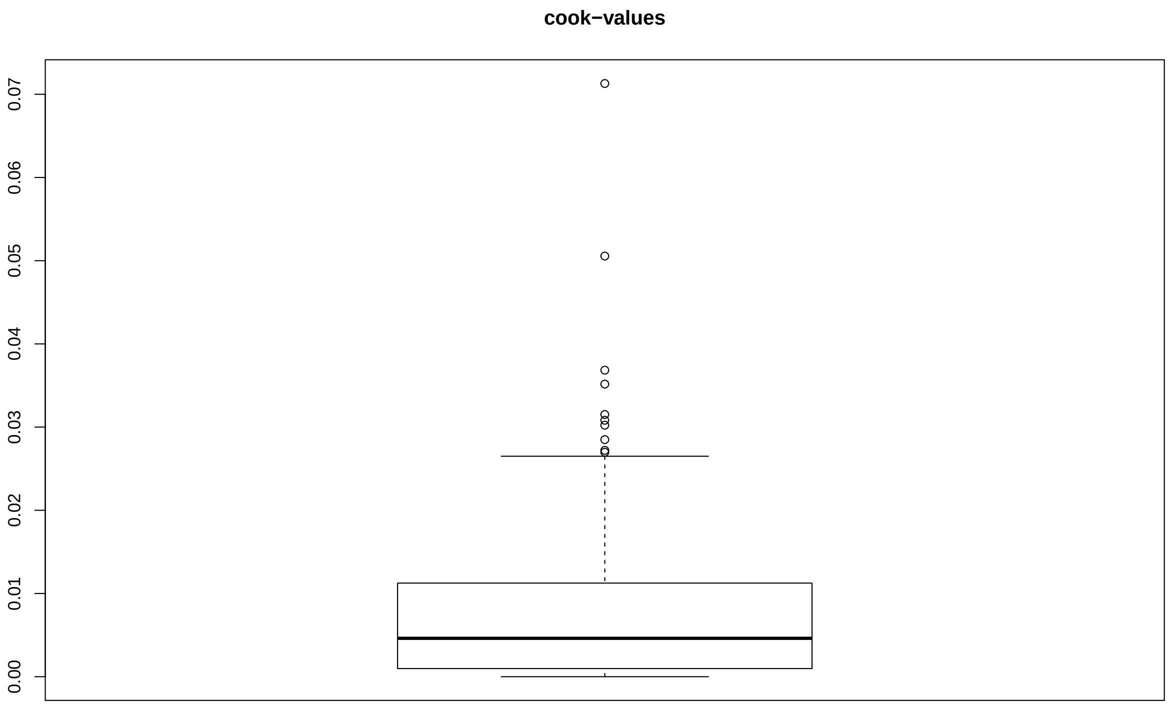
DFBeta

```
dfbeta.values <- dfbetas(fourier2.lm)
boxplot(dfbeta.values, main="dfBetas values")
```



Cook distance

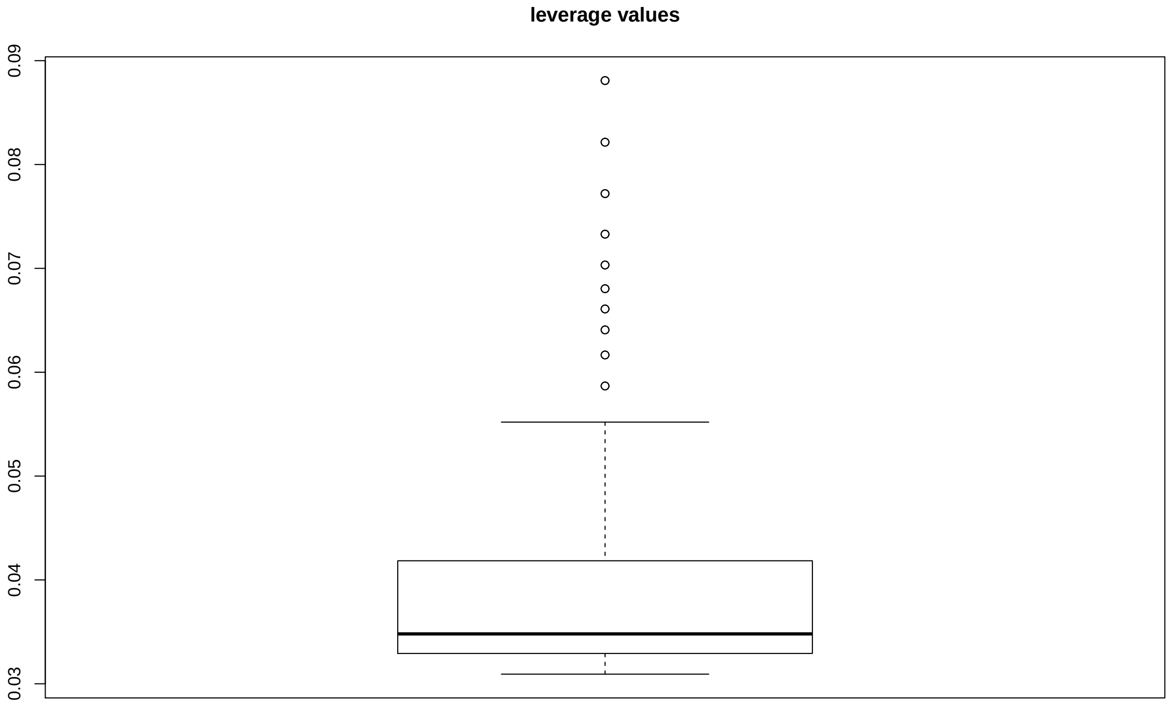
```
cook.values <- cooks.distance(fourier2.lm)
boxplot(cook.values, main="cook-values")
```



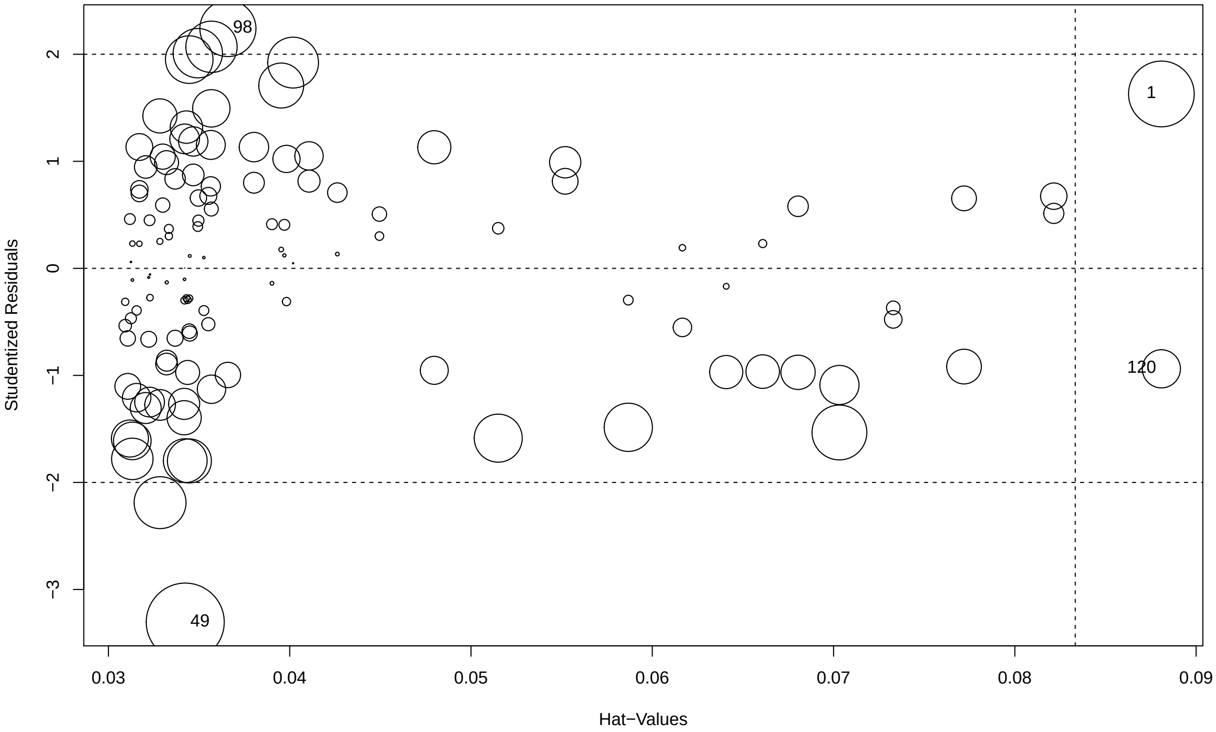
```
# boxplot(cook.values, main="cook-values", id.n=2)
```

Leverage Plot

```
leverage.values <- hatvalues(fourier2.lm)
boxplot(leverage.values, main="leverage values")
```



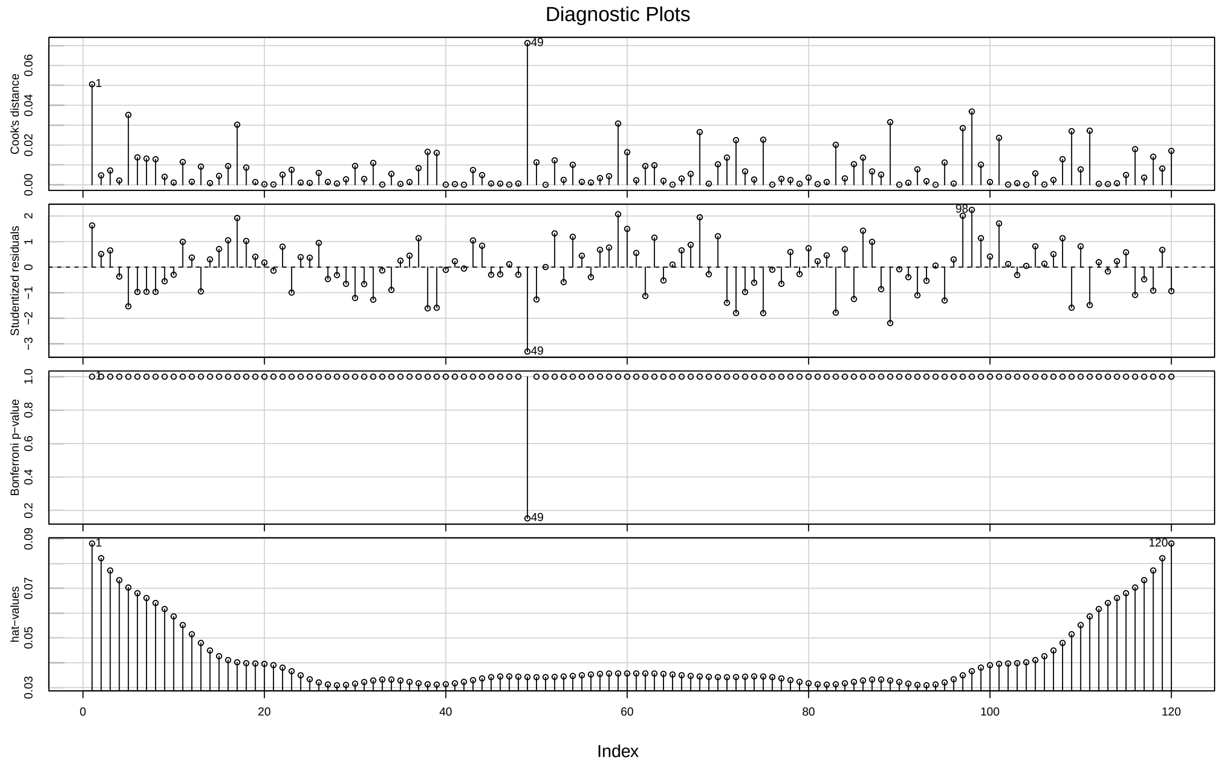
```
car::influencePlot(fourier2.lm)
```



```
##        StudRes        Hat      CookD
## 1    1.6292804 0.08808365 0.05055430
## 49  -3.3050315 0.03423661 0.07129448
## 98   2.2404263 0.03658995 0.03684017
## 120 -0.9388356 0.08808365 0.01704502
```

Be careful: inspect scale of Bonferroni p-values

```
car::influenceIndexPlot(fourier2.lm)
```


Diagnostic Plots
Cook's distance
0.00
0.02
0.04
0.06
1
49
Studentized residuals
−3
−2
−1
0
1
2
98
49
Bonferroni p−value
0.2
0.4
0.6
0.8
1.0
49
hat−values
0.03
0.05
0.07
0.09
1
120
0
20
40
60
80
100
120
Index