Lecture3.1OLSExample.R

nychka

2022-09-12

```
suppressMessages(library(fields))
suppressMessages(library(fda))
suppressMessages(library( lubridate))
load("/Users/nychka/Home/Teaching/FDA/data/BoulderDaily.rda")

dim(BoulderDaily)
```

```
## [1] 45139     9
```

```
names(BoulderDaily)
```

```
## [1] "year"      "month"     "day"       "tmax"      "tmin"      "precip"
## [7] "snow"      "snowcover" "time"
```

```
#######################################################
########## omitting missing values across the data set
#### but only include the temperatures as missing
#### (including rainfall and snow have many more missing)
#### this makes the subsequent analysis easier
####
#### call this new data frame BDClean
####
BDClean <- na.omit(BoulderDaily[, c(1:5, 9)])
dim(BDClean) # note fewer nonmissing observations
```

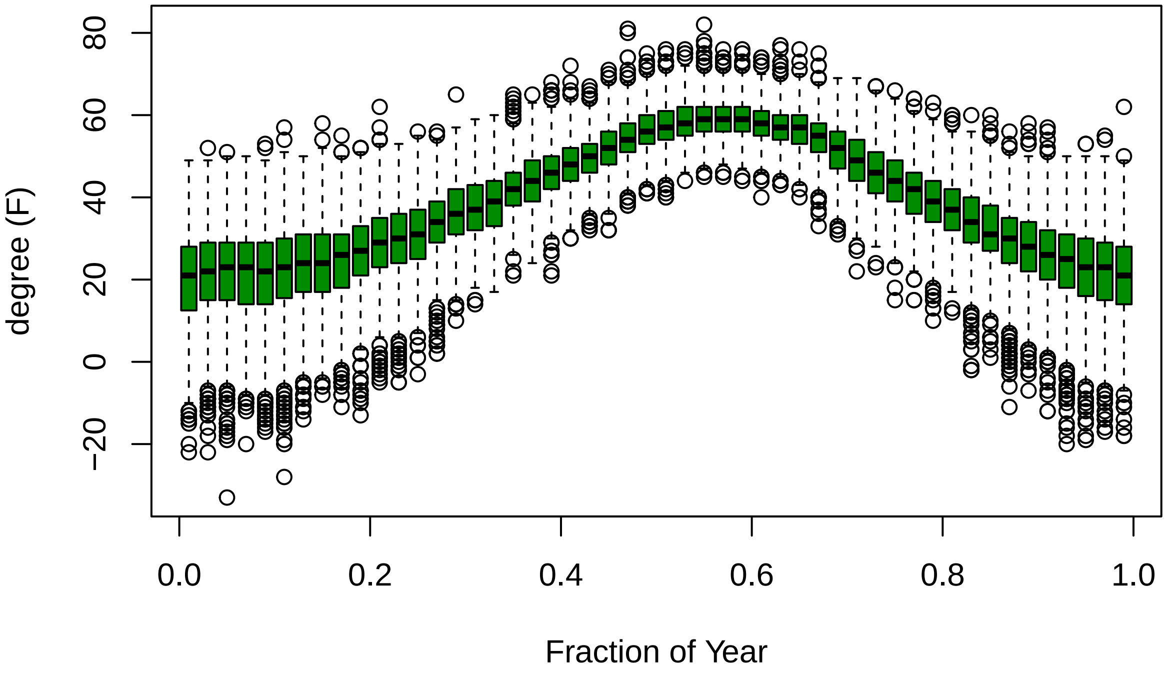
```
## [1] 44381     6
```

```
names(BDClean)
```

```
## [1] "year"  "month" "day"   "tmax"  "tmin"  "time"
```

```
BDates<- ymd( paste0(BDClean$year,"/",BDClean$month,"/",BDClean$day ))
BDClean$dates<- BDates

# find all years with fewer than 35 days missing
# I think this is pretty hard R coding so it
# may take a few reads to figure this out.
timeYear<- year(BDates)
countDays<- table( timeYear)
```

```
indGood<- countDays > 365 -35
goodYears<- names( countDays)[ indGood]
# NOTE goodYears are charcter strings
daysToKeep<- !is.na( match( as.character(timeYear),
                      goodYears)
               )
# subset data frame to years with more than 365 -35 obs
BDClean<- BDClean[daysToKeep,]



#Finally omit any rows of the data frame with a missing tmin
ind<- !is.na(BDClean$tmin)
BDClean<- BDClean[ind,]


# s is the fraction of year for a particular day
nYear<- ifelse(leap_year(BDClean$dates), 366, 365)
s<-  yday(BDClean$dates)/ nYear

Y<- BDClean$tmin


bplot.xy(
  s,
  Y,
  N = 55,
  col = "green4",
  xlab = "Fraction of Year",
  ylab = "degree (F)"
)
```



```
# interpret next line
table( year(BDClean$dates))
```

```
##
## 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1911 1912 1913 1914
##  365  365  365  365  365  365  365  365  365  365  365  333  365  365  364  365
## 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930
##  365  365  365  365  362  362  364  364  363  365  365  365  365  335  365  365
## 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946
##  365  365  365  365  348  365  365  365  365  365  365  365  365  365  365  365
## 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962
##  365  365  364  365  365  365  365  362  364  365  365  365  365  365  365  365
## 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978
##  365  365  365  365  365  365  365  365  365  365  365  365  365  365  365  362
## 1980 1981 1982 1984 1985 1986 1987 1988 1991 1992 1993 1994 1995 1996 1997 1998
##  334  363  335  365  365  365  365  365  365  362  365  365  365  365  365  365
## 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014
##  365  366  365  365  365  365  365  365  365  365  365  365  365  365  365  365
## 2015 2016 2017 2018 2019
##  365  365  365  365  365
```

```
# a sin cosine basis with  6 pairs and the constant function
fracOfYear<- s

freqX <- outer(2 * pi * fracOfYear, 1:6, "*")
dim(freqX)
```

```
## [1] 42542     6
```

```
Phi <- cbind(rep(1, length(Y)),
           sin(freqX), cos(freqX))

colNames <-  c("Contant", paste0("S", 1:6), paste0("C", 1:6))
dimnames(Phi) <-   list(NULL, colNames)

LSFit1 <- lm( Y  ~ Phi - 1)
#

# -1 means do not automatically include a
# constant vector in the model
# we have already built it into Phi
#
summary(LSFit1)
```

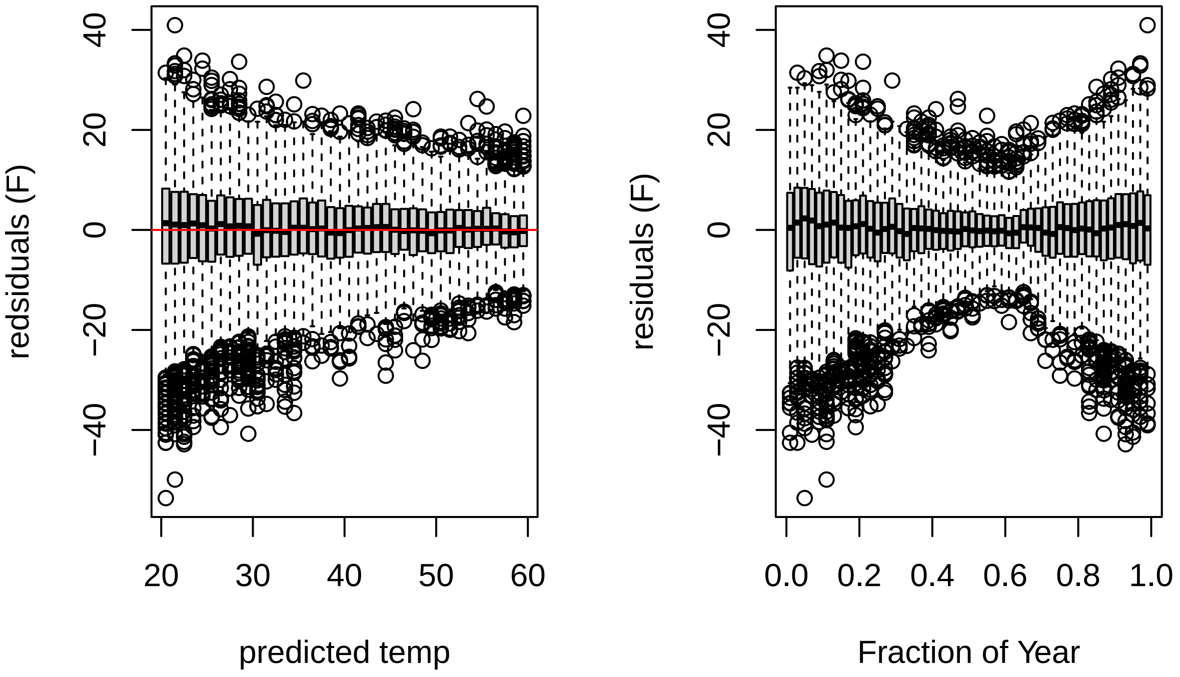
```
##
## Call:
## lm(formula = Y ~ Phi - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.619  -4.715   0.213   5.192  40.944
##
```

```
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## PhiContant  38.34711    0.04157  922.459  < 2e-16 ***
## PhiS1       -6.56864    0.05879 -111.735  < 2e-16 ***
## PhiS2        1.80229    0.05881   30.644  < 2e-16 ***
## PhiS3       -0.28229    0.05877   -4.803 1.56e-06 ***
## PhiS4        0.25273    0.05879    4.299 1.72e-05 ***
## PhiS5        0.14698    0.05880    2.499  0.01244 *
## PhiS6        0.01183    0.05877    0.201  0.84044
## PhiC1      -17.95437    0.05879 -305.390  < 2e-16 ***
## PhiC2        0.38769    0.05876    6.597 4.23e-11 ***
## PhiC3       -0.11837    0.05881   -2.013  0.04416 *
## PhiC4       -0.04791    0.05879   -0.815  0.41514
## PhiC5       -0.08137    0.05878   -1.384  0.16623
## PhiC6        0.16814    0.05881    2.859  0.00425 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.574 on 42529 degrees of freedom
## Multiple R-squared:  0.9574, Adjusted R-squared:  0.9574
## F-statistic: 7.361e+04 on 13 and 42529 DF,  p-value: < 2.2e-16
```

```r
# some  standard diagnostic plots
#  -- not too easy to read with so many points so
# use some boxpolots instead.
# set.panel( 2,2)
# plot(LSFit1 )
# set.panel() # set back to 1 plot per view

# basic check on residuals
set.panel(1, 2)
```

```
## plot window will lay out plots in a 1 by 2 matrix
```

```r
bplot.xy(
  LSFit1$fitted.value,
  LSFit1$residuals,
  N = 50,
  xlab = "predicted temp",
  ylab = "redsiduals (F)"
)
yline(0, col = "red")

ind<- !is.na( c(Y))
bplot.xy(
  fracOfYear[ind],
  LSFit1$residuals,
  N = 50,
  xlab = "Fraction of Year",
  ylab = "residuals (F)"
)
```



```
set.panel()
```

```
## plot window will lay out plots in a 1 by 1 matrix
```

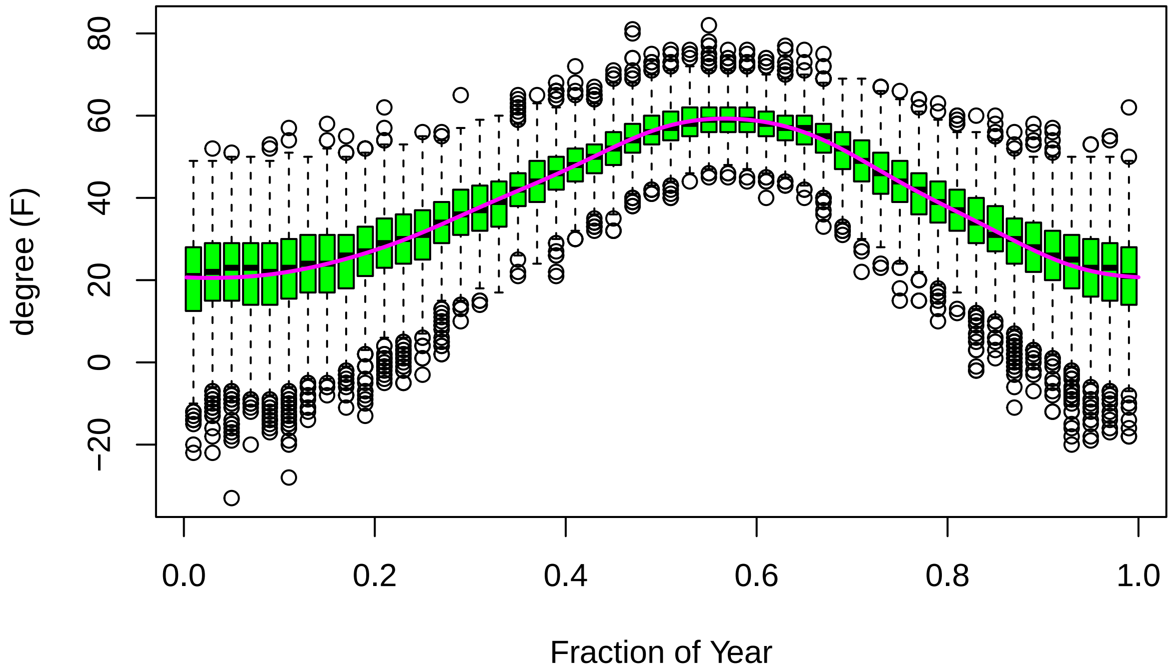
```
# boxplot with fitted cycle added
# pay attention to dimensions!
fracGrid <- (1:365)/365
freqXPred <- outer(2 * pi * fracGrid, 1:6)
XPred <-  cbind(rep(1, length(fracGrid)),
                sin(freqXPred),
                cos(freqXPred))
seasonalCycle <- XPred %*% LSFit1$coefficients
dim(XPred )
```

```
## [1] 365  13
```

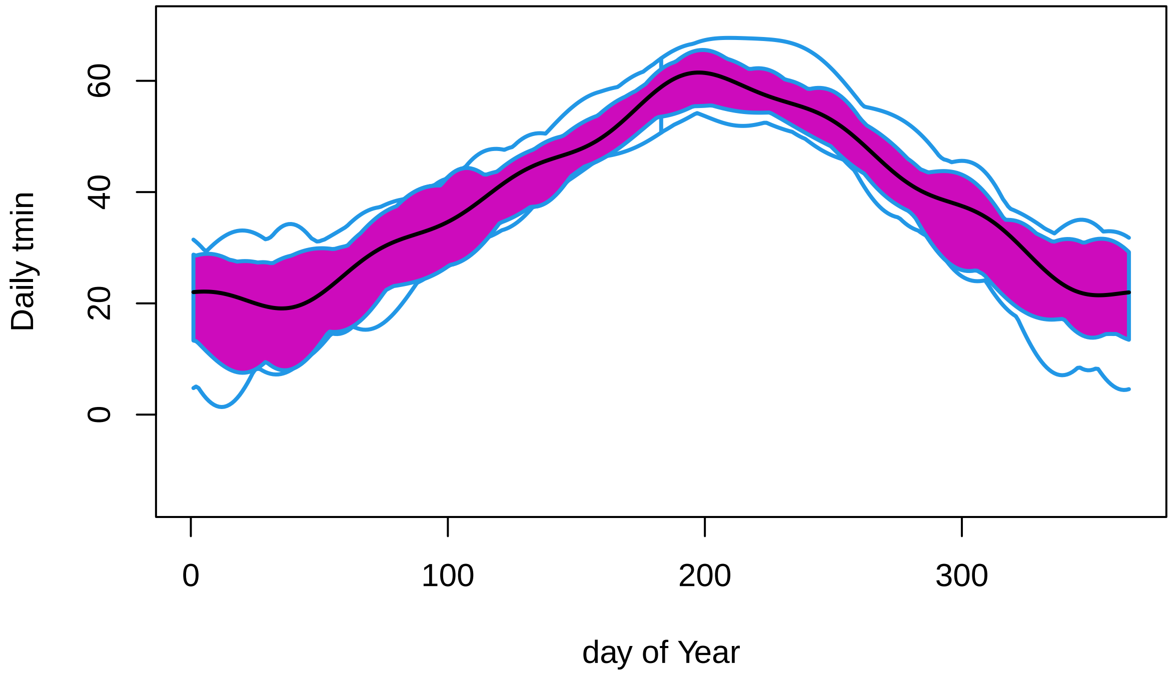
```
length( seasonalCycle)
```

```
## [1] 365
```

```
bplot.xy(
  s,
  Y,
  N = 40,
  col = "green1",
  xlab = "Fraction of Year",
  ylab = "degree (F)"
)

lines(fracGrid, seasonalCycle , col = "magenta",
      lwd = 2)
```



```
#### finally loop through years to get a
#### smoothed cycle for each year.
yearTag<- unique( BDClean$year)
NYear<- length( yearTag)

fracGrid<- (1:365)/365
PhiYear<- cbind(rep(1, length(fracGrid)),
                sin(freqXPred),
                cos(freqXPred))
# this matrix will hold predicted values for each year
YHat<- matrix( NA, 365, NYear)

for( k in 1: NYear ){
# pull out one years data
  ind<- BDClean$year == yearTag[k]
  yTemp<- Y[ind]
  cHat<- lm( Y[ind] ~ Phi[ind,] -1 )$coefficients
# predicted values at the fraction of year
# note use of a different basis matrix for prediction
  YHat[,k]<- PhiYear%*%cHat
}

out<- fbplot( (YHat), x= 1:365, ylim=c(-15,70),
              xlab="day of Year", ylab="Daily tmin")
```


Daily tmin
0
20
40
60
0
100
200
300
day of Year