MATH 498 HW3

Drew Remmenga

```r
suppressMessages(library(fields))
```

```
## Warning: package 'fields' was built under R version 4.1.3

## Warning: package 'spam' was built under R version 4.1.3
```

```r
suppressMessages(library(matlib))
```

```
## Warning: package 'matlib' was built under R version 4.1.3
```

```r
suppressMessages(library(fda))
```

```
## Warning: package 'fda' was built under R version 4.1.3

## Warning: package 'fds' was built under R version 4.1.3

## Warning: package 'rainbow' was built under R version 4.1.3

## Warning: package 'pcaPP' was built under R version 4.1.3

## Warning: package 'RCurl' was built under R version 4.1.3

## Warning: package 'deSolve' was built under R version 4.1.3
```

```r
suppressMessages(library( lubridate))
load("~/School/MATH498/HW3/BoulderDaily.rda")

dim(BoulderDaily)
```

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

```r
names(BoulderDaily)
```

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

```r
########################################################
########## 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
```

```r
names(BDClean)
```

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

```r
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
fracOfYear<- s

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

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

```r
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
```

1.a.

```r
c<-inv(t(Phi)%*%Phi)%*%t(Phi)%*%Y
c
```

```
##                 [,1]
##  [1,]  38.353944843
```

```
##  [2,]  -6.571432952
##  [3,]   1.809442125
##  [4,]  -0.284482070
##  [5,]   0.245235085
##  [6,]   0.142589965
##  [7,]   0.009734338
##  [8,] -17.960183998
##  [9,]   0.381346019
## [10,]  -0.124888671
## [11,]  -0.049980528
## [12,]  -0.086229948
## [13,]   0.174356050
```

These produce the same estimates. 1.b.

```
r<-(resid(LSFit1))
mean(r)
```

```
## [1] -3.626818e-17
```

```
t(r)%*%Phi%*%c
```

```
##               [,1]
## [1,] -3.213303e-09
```

For any basis.

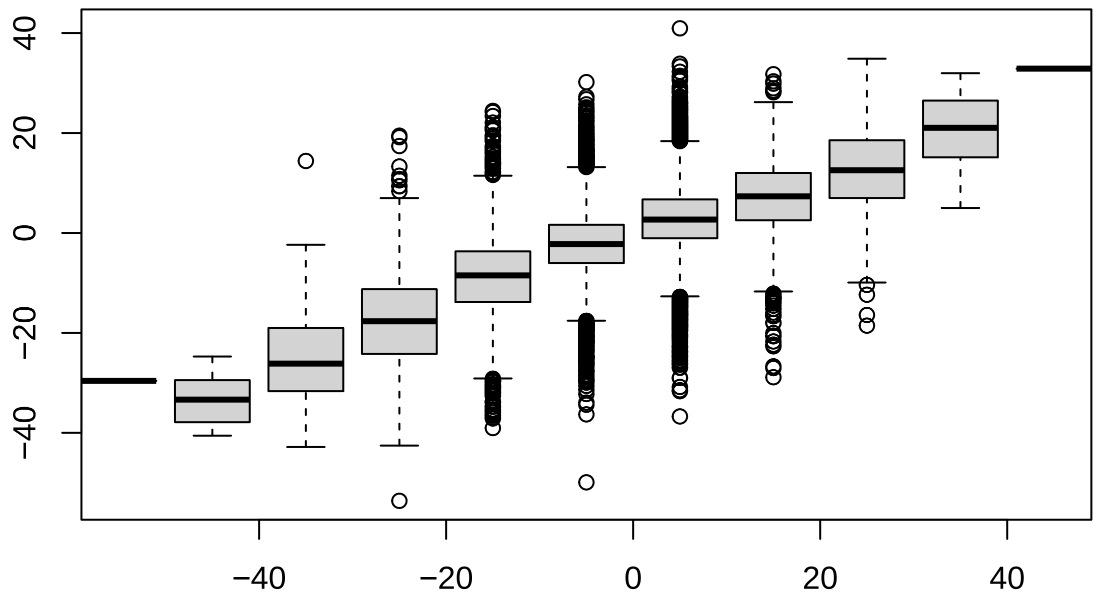
```
t(Phi)%*%Phi
```

```
##               Contant            S1           S2           S3           S4
## Contant 42542.0000000    -0.7702083   -14.316847    71.081990   -12.499599
## S1         -0.7702083 21272.4015452    -3.524454   -19.276652     7.742419
## S2        -14.3168466    -3.5244540 21253.124893     4.217965    14.734578
## S3         71.0819905   -19.2766517     4.217965 21287.136123    28.858370
## S4        -12.4995990     7.7424190    14.734578    28.858370 21270.192202
## S5         -3.7095202    34.0112297    32.382824    -2.209343     3.113703
## S6         18.0789978    24.6404046    17.067309     6.638157   -11.599726
## C1         34.1017936    -7.1584233    35.155891   -13.408223    33.686235
## C2         -2.8030903    35.9260994    -6.249799    -2.239864     1.881076
## C3         41.1507015     0.9086238    -1.469656     9.039499   -24.263247
## C4         35.7502130   -37.3957553    16.197922   -23.493039   -18.561808
## C5         25.6658636    15.2892984   -59.419138   -11.403385     3.085375
## C6        -32.2722463   -22.0233828   -12.312009   -32.840724    25.280900
##                   S5           S6           C1           C2           C3
## Contant    -3.709520    18.078998    34.101794    -2.803090   41.1507015
## S1         34.011230    24.640405    -7.158423    35.926099    0.9086238
## S2         32.382824    17.067309    35.155891    -6.249799   -1.4696559
## S3         -2.209343     6.638157   -13.408223    -2.239864    9.0394989
## S4          3.113703   -11.599726    33.686235     1.881076  -24.2632470
## S5      21260.801819    24.538918     2.789699    11.662852  -25.7202318
## S6         24.538918 21284.710166   -25.732903   -24.811608   38.2412665
## C1          2.789699   -25.732903 21269.598455    37.626248   16.4735614
```

```
## C2          11.662852   -24.811608    37.626248 21288.875107    29.8838286
## C3         -25.720232    38.241266    16.473561    29.883829 21254.8638768
## C4           2.315167    10.964053    33.408283   -17.537668     5.2434239
## C5          18.122476    12.651906     1.738983     8.767878    -0.5937473
## C6          13.422115     8.516877     1.025459    18.682904    34.5125450
##                    C4            C5           C6
## Contant     35.750213    25.6658636   -32.272246
## S1         -37.395755    15.2892984   -22.023383
## S2          16.197922   -59.4191381   -12.312009
## S3         -23.493039   -11.4033852   -32.840724
## S4         -18.561808     3.0853754    25.280900
## S5           2.315167    18.1224764    13.422115
## S6          10.964053    12.6519065     8.516877
## C1          33.408283     1.7389834     1.025459
## C2         -17.537668     8.7678779    18.682904
## C3           5.243424    -0.5937473    34.512545
## C4       21271.807798    30.9880910     8.796636
## C5          30.988091 21281.1981808     9.562875
## C6           8.796636     9.5628753 21257.289834
```

1.e. Residuals are all close to zero. As the graph progresses there becomes less width due to better temperature recording equipment. 1.g.

```r
res<- r
N<- length( res)
res0<- res[ 2:N]
res1<- res[1:(N-1)]
bplot.xy(res0,res1)
```

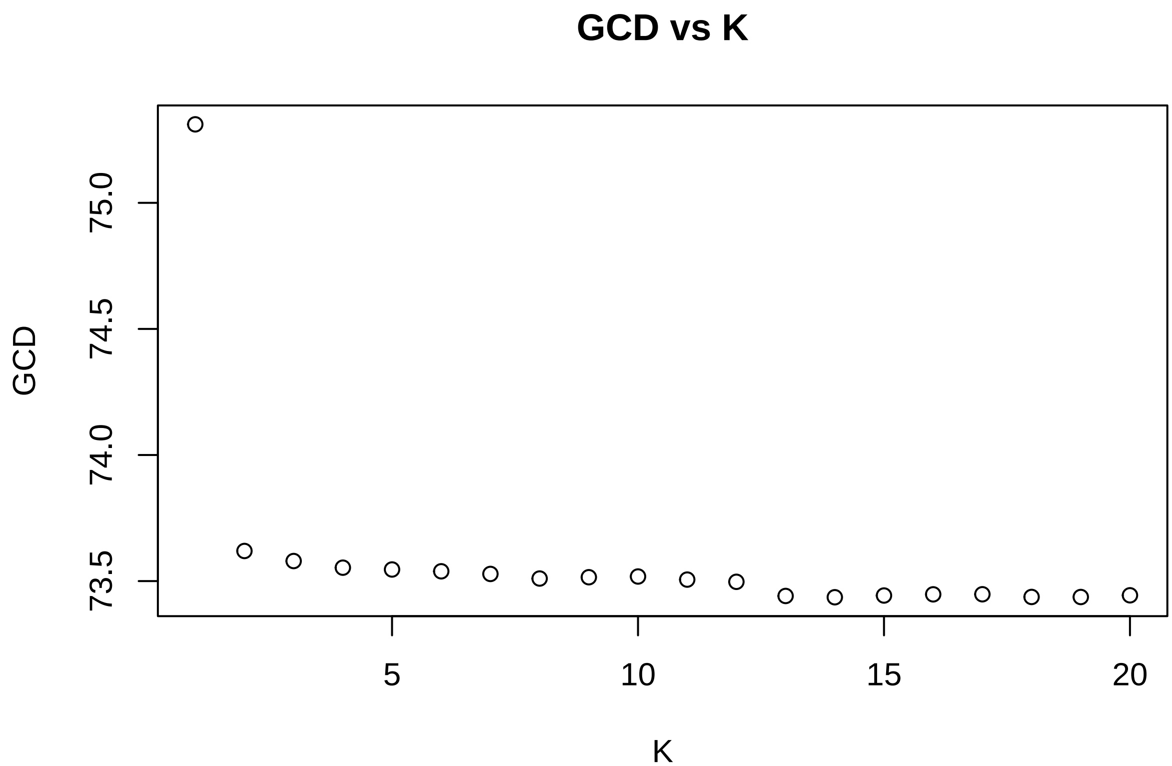


Residuals from one day do appear to depend on residuals from the previous day.

```
x<-c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)
gcd<- numeric()
for (val in x) {
freqX <- outer(2 * pi * fracOfYear, 1:val, "*")
dim(freqX)
Phi <- cbind(rep(1, length(Y)),
           sin(freqX), cos(freqX))
colNames <-  c("Contant", paste0("S", 1:val), paste0("C", 1:val))
dimnames(Phi) <-   list(NULL, colNames)

LSFit1 <- lm( Y  ~ Phi-1)
p <-2*val+1
M = ((1/N)*t(resid(LSFit1))%*%resid(LSFit1))/((1-p/N)^2)
gcd<-c(gcd,M)
}
```

```
plot(x,gcd,ylab="GCD",xlab="K",main ="GCD vs K")
```


GCD vs K
GCD
75.0
74.5
74.0
73.5
5
10
15
20
K