Math 598 - Take Home Test 1

Lewis Blake

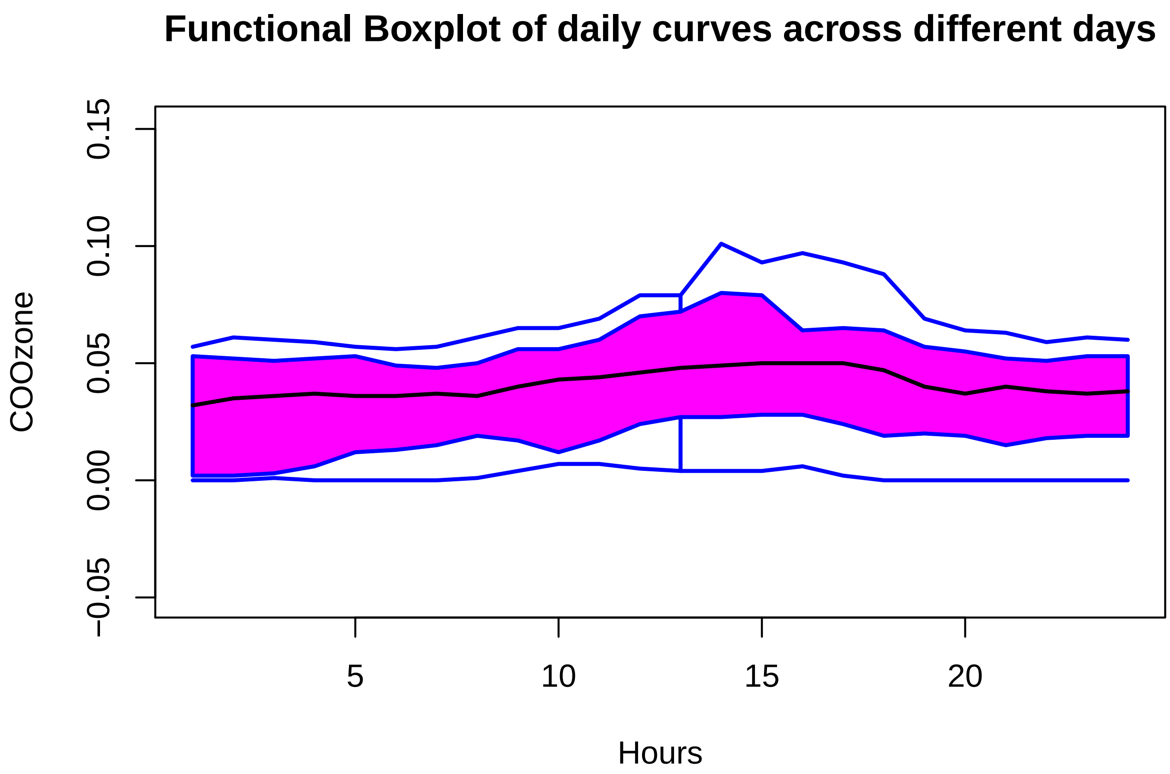
2/26/2019

Problem 1. *Add comments before each line of code to explain what is being done. In writing these explanations assume the reader is not very familiar with the* `R` *language. In particular, carefully explain he purpose of the cryptic line of code defining the* `incomplete` *variable.*

```
# Load the data into the R Global Environment.
load("CO03hourly_44201_2017.rda")
# Extract the raw time data from COOzoneTime using the $ operation,
# and format this into a matrix with 24 rows.
# There are 24 rows, 1 row for each hour of the day. dim(timeRaw) = 24 * 365
timeRaw = matrix(COOzoneTime$time, nrow=24)
# Take the transpose of the timeRaw matrix so that columns are the hour of the day.
timeRaw = t(timeRaw) # Now, dim(timeRaw) = 365 * 24.
# Extract the COOzone data from the location closest to Golden (row 18),
# and format this into a matrix with 24 rows.
# There are 24 rows because each hour has an associated observation at this location.
Golden03Raw = matrix(COOzone[,18], nrow = 24)
# Take the transpose of the Golden03Raw matrix so columns are hours of the day.
Golden03Raw = t(Golden03Raw) # Now matchs dimensions of timeRaw.
# The na.omit() function removes rows with NA's.
# We only want days for which we have observations for all 24 hours.
Golden03 = na.omit(Golden03Raw)
# The attributes() function returns an object's attributes list.
# Since we have applied the na.omit() function to Golden03Raw,
# one of its attributes is na.action, which we extract with the $ operator.
# The na.action attribute is a list of observations where there were NA's.
# Therefore incomplete is a "list" of days where we do not have completed data.
incomplete = attributes(Golden03)$na.action
# Remove the rows (days) of timeRaw for which we don't have complete data.
timeGolden = timeRaw[-incomplete, ]
# Create the days by remvoing first column of timeGolden (hour),
# then subtract 2017 from each entry (see timeRaw above),
# and then multiply by 365 for each day of the year.
dayGolden = (timeGolden[,-1] - 2017)*365
```

Problem 2. *Create a functional boxplot of that summarizes the shape of the daily curves across different days. Are there any days that could be considered outliers? What is the overall shape of the daily curves and how does it vary?*

```
# Create a functional boxplot.
# Golden03 is a n*p functional data matrix,
# where n is the number of curves.
# Since we have a curve for each day, we take the transpose of Golden03,
# to match the arugment specifications of fbplot().
# Store fbplot() as an object
fbpGolden = fbplot(t(Golden03),
                   main  = "Functional Boxplot of daily curves across different days",
                   ylab = "COOzone",
                   xlab = "Hours")
```



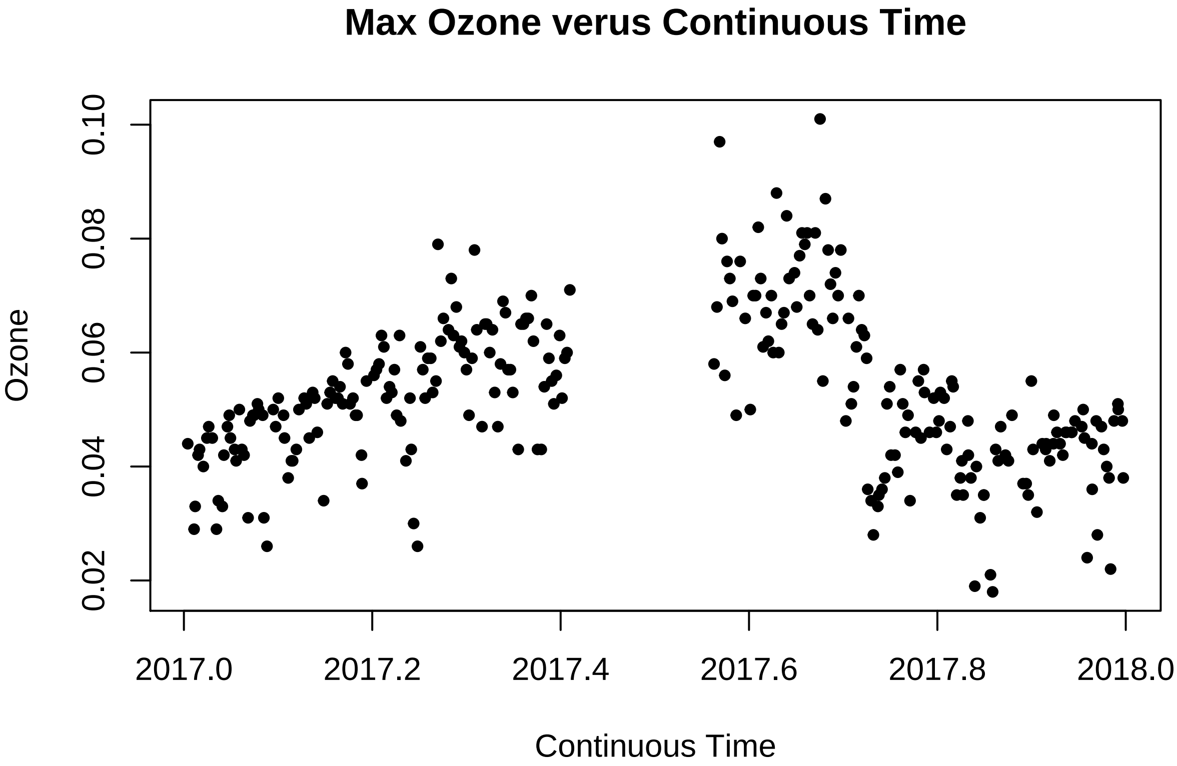
```
# Extract the column indices (days) of the detected outliers with $ operator.
indOutliers = fbpGolden$outpoint
print(indOutliers)
```

```
## integer(0)
```

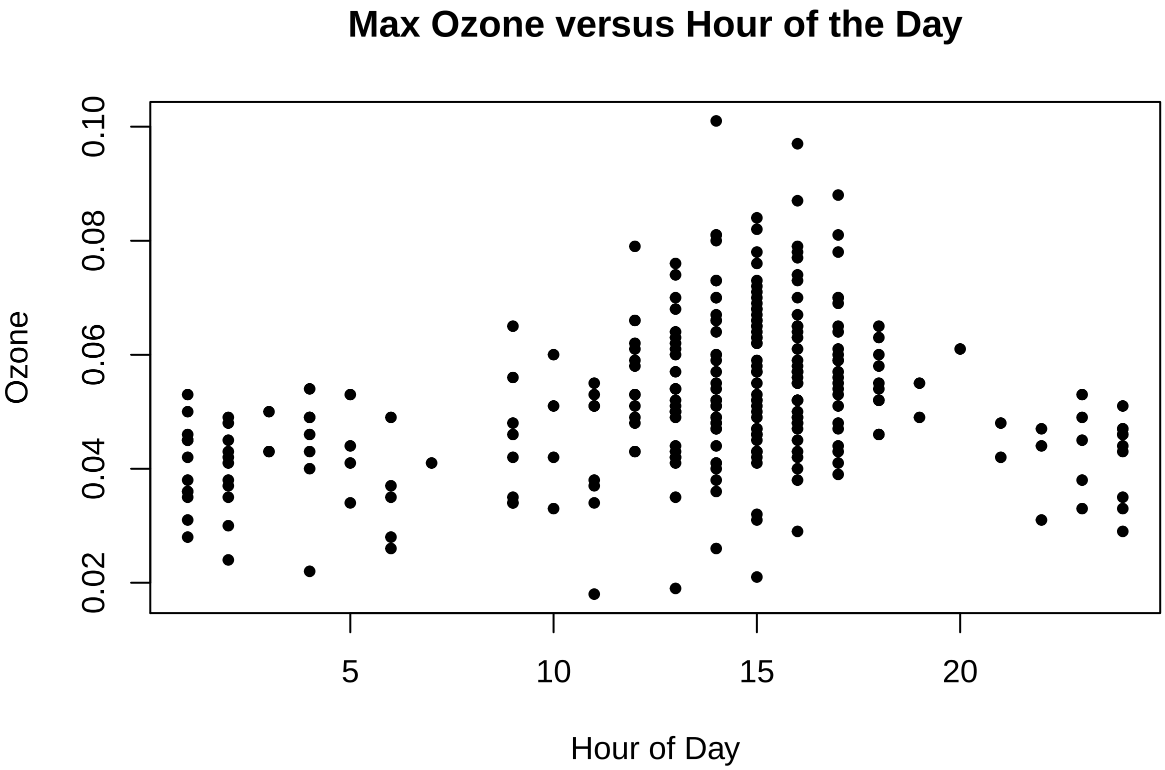
Viewing each day’s observations as a curve over a 24 hour period, no outliers were detected. The overall shape shows greater variability in the afternoon, where the highest levels of ozone are achieved. There is a slight dip in variability in the early morning (about 5am-8am), which I speculate might be related to sunrise.

Problem 3. *Find the maximum ozone value for each day and also the hour that it occurs. Make a plot over time for both of these statistics and comment on any patterns - or lack of pattern. Consider using the* `bplot.xy` *function if many points are plotted on top of each other.*

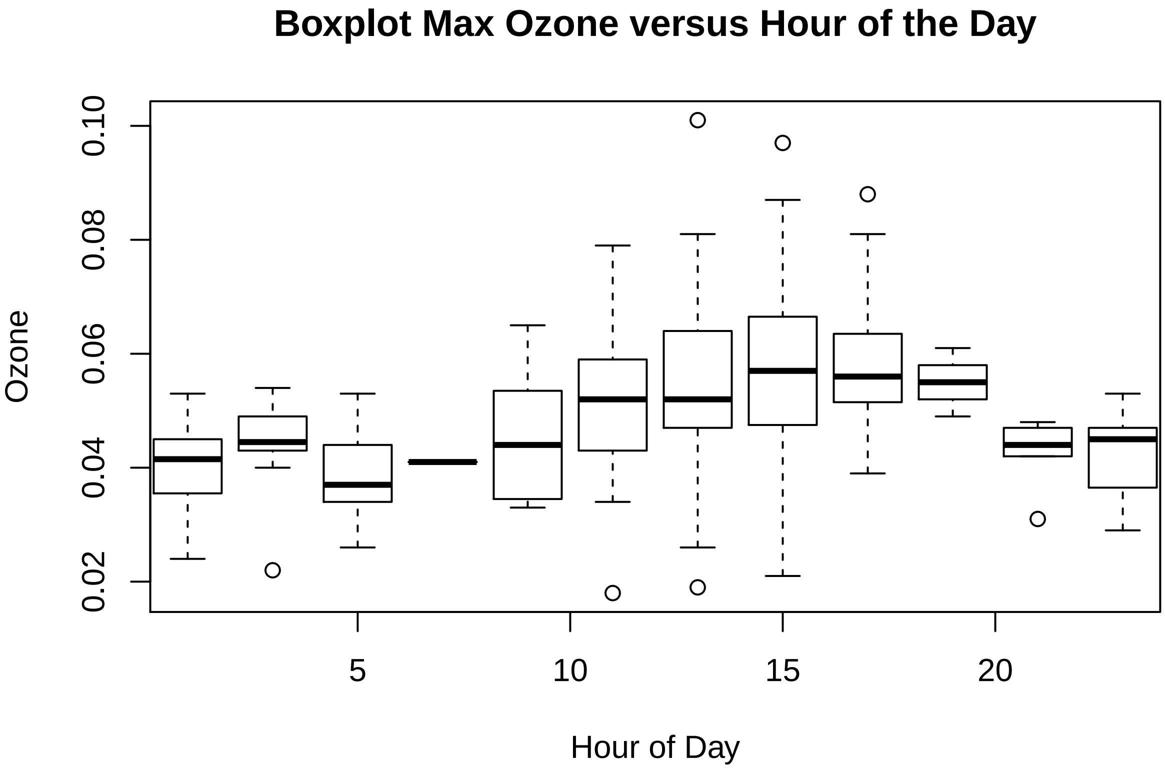
```
# Get indicies for each row (day) where the maximum occurs.
indMax = apply(GoldenO3, 1, which.max)
# Create a matrix to store the maximum ozone observed
# and the time at which it occured.
# indMax has the hour "number" when this occured (1:24),
# however collect the actual corresponding time as well.
maxOzoneMat = matrix(NA, ncol = 2, nrow = length(indMax))
# Loop through each each (280)
for(k in 1:length(indMax)){
  # Get the maximum ozone value, store value.
  maxOzoneMat[k, 2] = GoldenO3[k, indMax[k]]
  # Get time of maximum ozone, store time.
  maxOzoneMat[k, 1] = timeGolden[k, indMax[k]]
}

# Plot maximum ozone value for each value over cont. time
plot(maxOzoneMat[,1], maxOzoneMat[,2],
     main = "Max Ozone verus Continuous Time",
     ylab = "Ozone",
     xlab = "Continuous Time",
     pch=20)
```



```
# Plot maximum ozone value over hour to the day
plot(indMax, maxOzoneMat[,2],
     main = "Max Ozone versus Hour of the Day",
     ylab = "Ozone",
     xlab = "Hour of Day",
     pch=20)
```



```
# Box plot maximum ozone over hour of the day
bplot.xy(indMax, maxOzoneMat[,2],
     main = "Boxplot Max Ozone versus Hour of the Day",
     ylab = "Ozone",
     xlab = "Hour of Day")
```



I've included the plots above because I believe there is something to be gained from each of them. Looking at the first plot, "Max Ozone versus Continuous Time", there appears to be almost a quadratic relationship between the maximum ozone observation and when it was observed. That is beginning in early 2017 maximum daily ozone was relatively low, continued to increase throughout the year (where we have data) until it hit a maximum and then died off at approximately the rate at which it increased. This suggests that maximum ozone (as well as variability) is highest in the summer months and there are relatively lower maximum ozone levels (and less variability) in the winter months, following a seasonal trend.

Plotting the maximum ozone observations against the hour of the day at which they were recorded, other trends emerge. Looking at the "Max Ozone versus Hour of the Day" plot, we find that most of the maximum ozone observations occur in the early afternoon, which corroborates what we witness in the functional boxplot above. Not only do most of the maximum ozone observations occur in the afternoon, but there is also greater variability in afternoon observations. As pointed out by the functional box plot, the early morning has less observations with less variability. The "Boxplot Max Ozone versus Hour of the Day" also demonstrates that the majority of maximum ozone observations occur in the afternoon, with the greatest variability. We can see from the tick-marks, that the median maximum ozone observations dip around the early morning, gradually increase throughout the day, attaining their maximum in the afternoon, and then decrease throughout the evening.

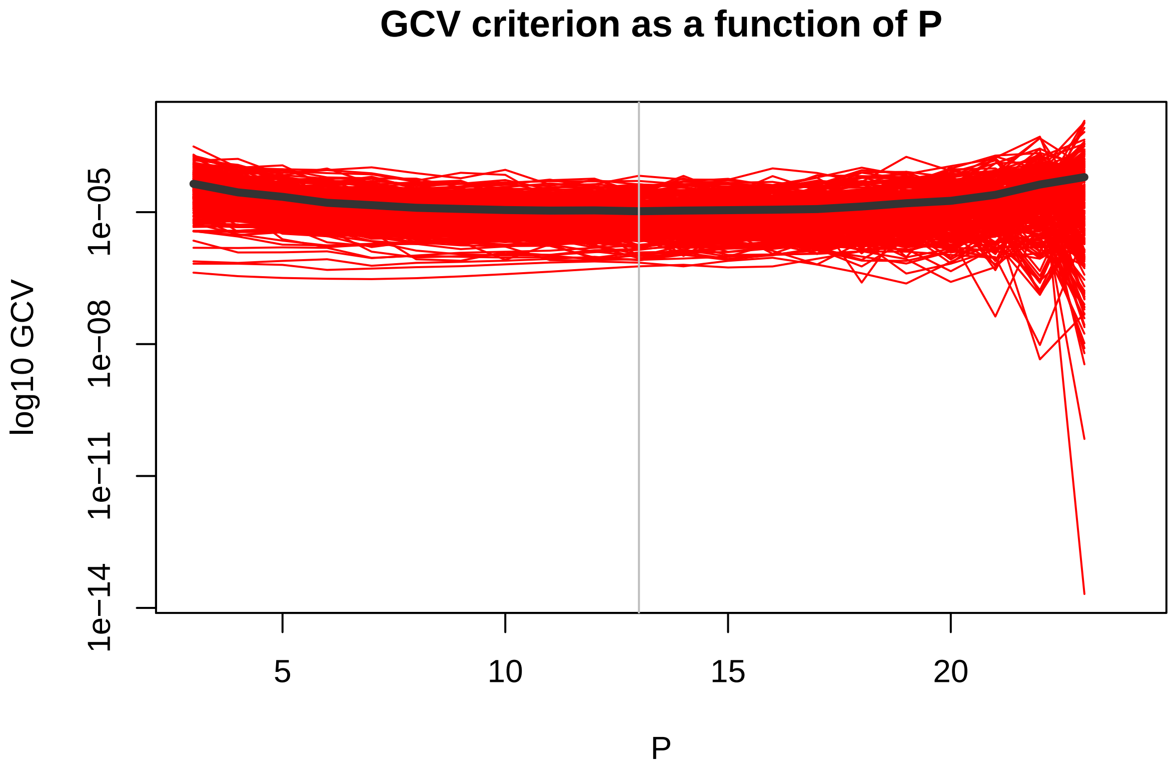
3.2 *For the last problem on HW4, you were asked to come up with a number of basis functions by minimizing the pooled GCV criterion. Do a similar analysis for the* `Golden03` *data. Fit a smooth curve to each day of the Golden ozone measurements using a natural B-spline basis and least squares. You can make the knots equally spaced between 1 and 24 (hours). Determine the number of basis functions/knots to use by minimizing the pooled version of the GCV criterion as was done in HW4. Use the code versions* `naturalSplineFit.R` *and* `findGCV.R` *in the Test 1 folder because I have made some minor changes to them.*

See the output in **Problem 4** to see that number of basis functions that minimizes the pooled GCV is 13.

```r
pGrid = 3:24 # 24 data points for each day
# Create a matrix to store the GCVs in
GCVs = matrix(NA, length(pGrid), 280)
# Create a vector to store pHat
pHat = rep(NA, 280)
# Loop through each day, and use findGCV()
# to fit a natural spline to that day's observations
for(k in 1:280){
  thisGCVout= findGCV(c(1:24), Golden03[k,], pGrid)
  GCVs[,k] = thisGCVout$GCV
  pHat[k] = thisGCVout$pHat
}
```

Problem 4. *Plot the pooled GCV criterion as a function of the number of basis functions and indicate where the minimum occurs. Call this estimate, $\hat{p}$. Using $\hat{p}$, do the fitted curves tend to interpolate the data or do they tend to smooth out the points?*

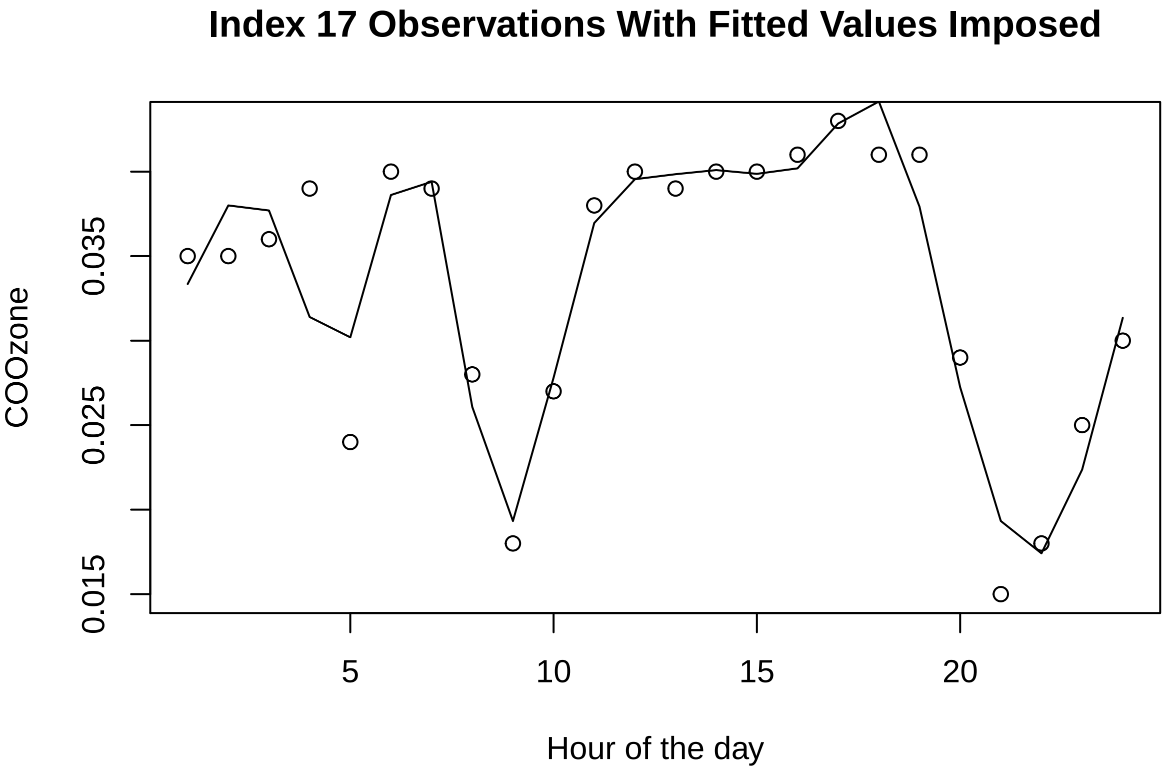
```r
# Create a plot of the GCV criterion as a function of p
matplot(pGrid, GCVs,
        type = "l", col = "red1", lty = 1, log = "y",
        ylab = "log10 GCV", xlab = "P",
        main = "GCV criterion as a function of P")
# Calculate GCV means
GCVMean = rowMeans(GCVs)
# Add the mean to the plot
lines(pGrid, GCVMean, col = "grey20", lwd = 4)
# Find the index at which has the min mean
ind = which.min(GCVMean)
# Add the p hat which minimizes GCV criterion
# to the plot as a grey vertical line
xline(pGrid[ind], col = "grey")
```



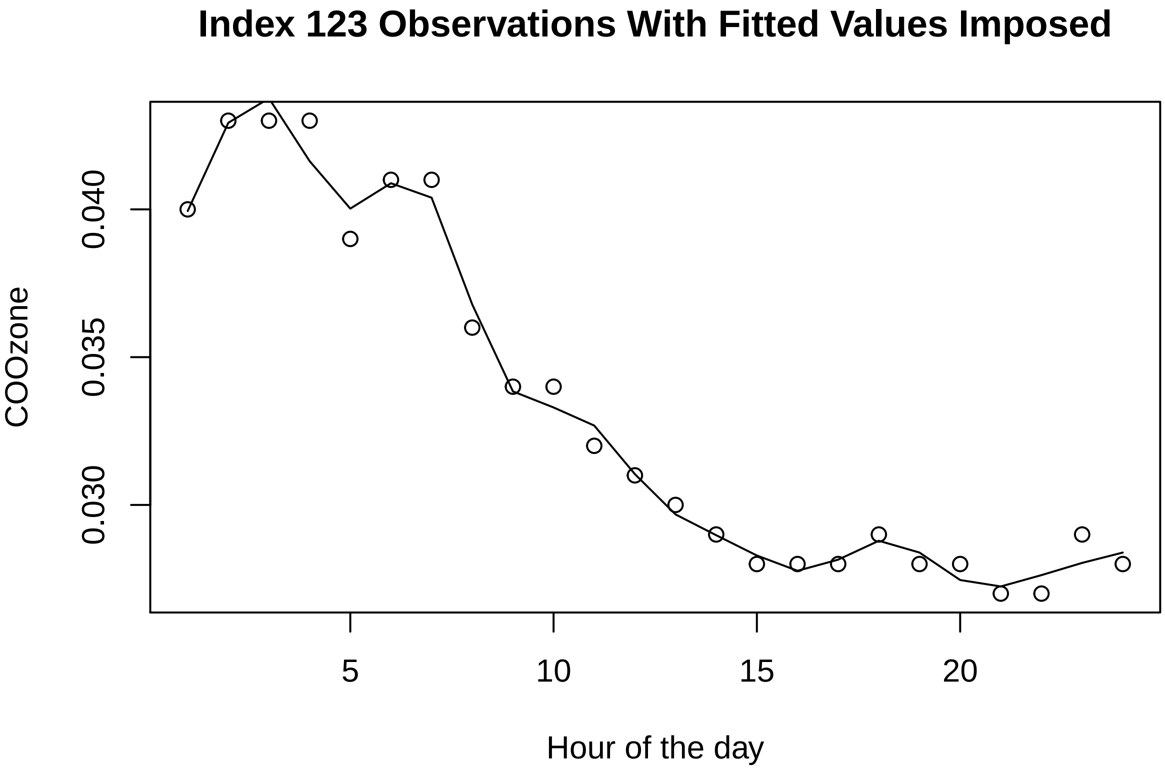
```
# Print pHat (13)
print(pGrid[ind])
```

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

```
# Use pHat = 13 to fit a curve to each day
pKnots = seq(1,24,length.out = 13)
pGrid2 = seq(1,24, length.out = 130)
# Create a list for each fit
pHatFits = rep(NA, 280)
for(k in 1:280){
  pHatFits[k] = naturalSplineFit(c(1:24), Golden03[k, ], pKnots, pGrid2)
}

# Check a few examples to see whether fitted curve interpolates or smooths
plot(Golden03[17,],
     main = "Index 17 Observations With Fitted Values Imposed",
     ylab = "COOzone",
     xlab = "Hour of the day")
lines(pHatFits[[17]]$fitted.values)
```



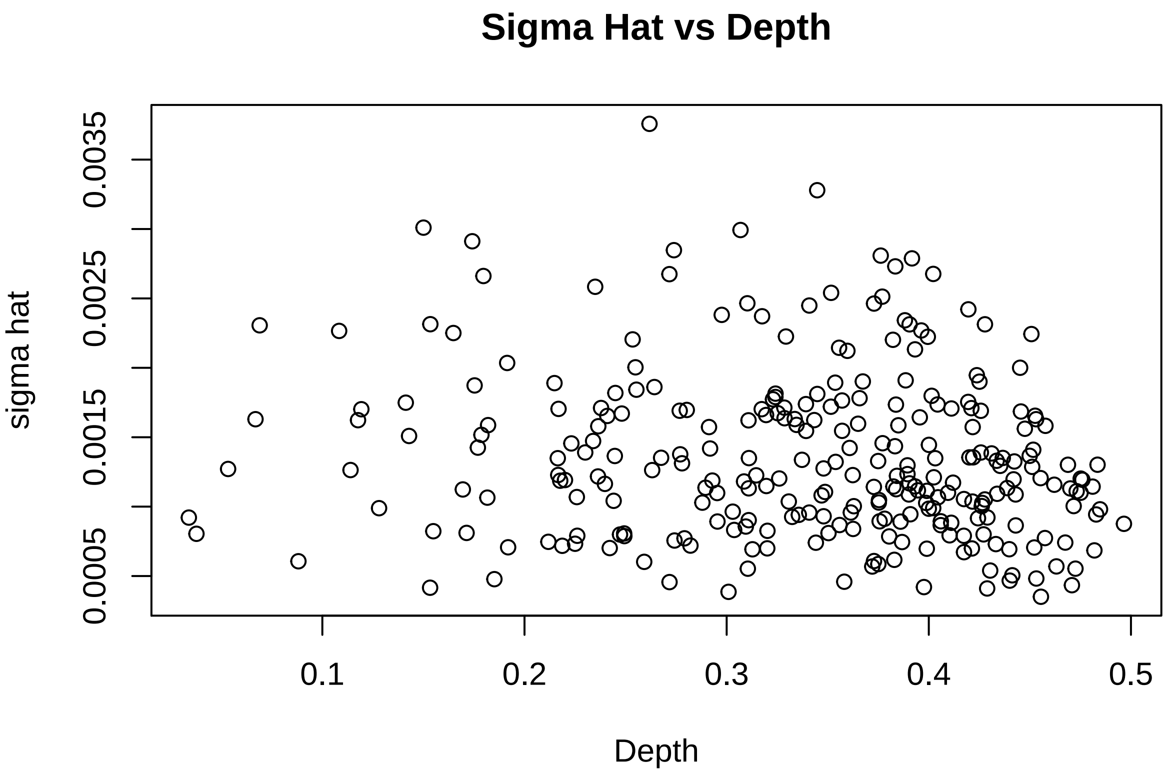
```
plot(Golden03[123,],
     main = "Index 123 Observations With Fitted Values Imposed",
     ylab = "COOzone",
     xlab = "Hour of the day")
lines(pHatFits[[123]]$fitted.values)
```



Using $\hat{p} = 13$, by checking a few examples I found that the fitted curves tend to smooth out the points. Although the lines of the fitted values typically follow the trend of the data they do not go through every observation.

Problem 5. *For each day, find the estimate of the error standard deviation, $\hat{\sigma}$ based on the fitted curve and using the number of knots found by pooled GCV. Does σ appear to be related to the depth of the curves?*

```
# Create a vector to store each sigma
sigmas = rep(NA,280)
# Loop through each day
for(k in 1:280){
  # Estimate sigma using residuals and store in vector
  sigmas[k] = sqrt(mean(pHatFits[[k]]$residuals^2))
}
# Plot sigmas versus each curve's depth
plot(fbpGolden$depth, sigmas,
     main = "Sigma Hat vs Depth",
     ylab = "sigma hat",
     xlab = "Depth")
```



Interestingly, the estimates of σ over depth appear to be contained within a quadratic shape. That is, more shallow and deeper curves tend to have lower estimates of σ whereas curves with more "middle" depth have larger estimates of σ.

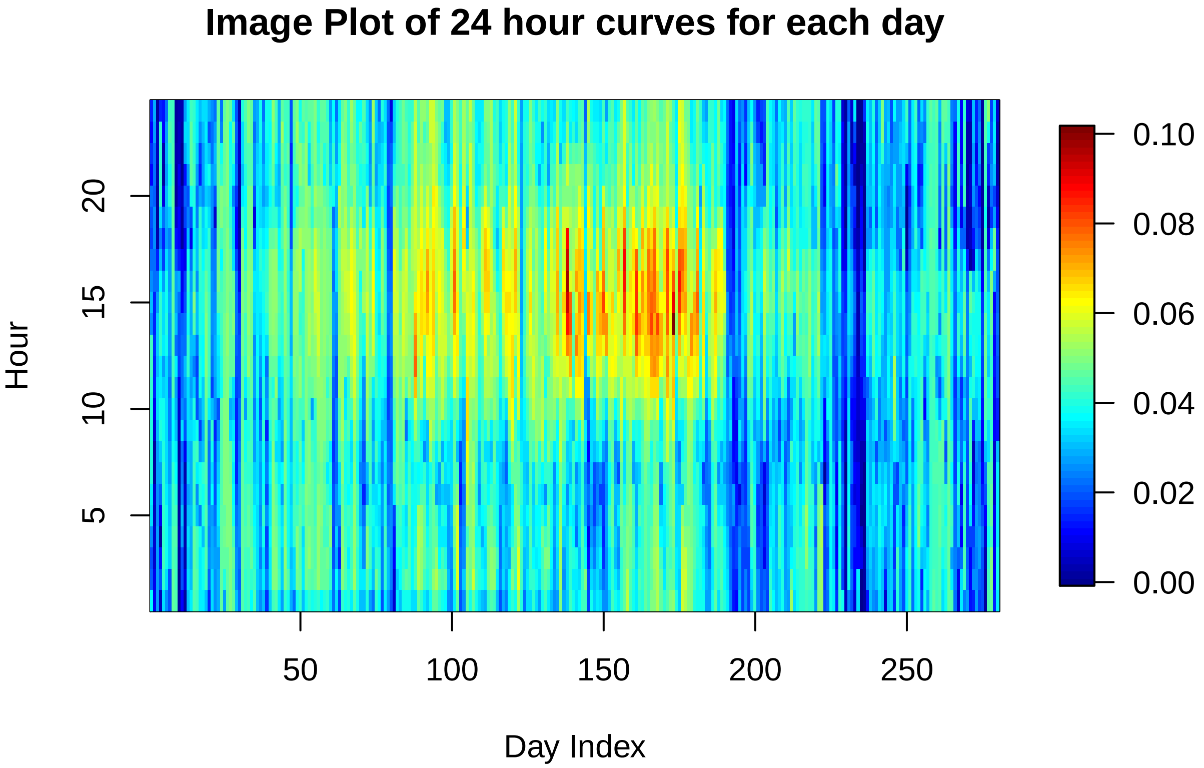
Problem 6. *Does the shape of the daily curves vary over the year?*

There is some ambiguity in this question by what is meant by "shape", so I am going to assume that shape refers to the roughly the same values (scale) at the same daily times throughout the year. That is, taking the observations from one day, and scaling them by a factor of 1/2, would result in a data set with a different "shape".

One way to answer this is to look at the two plots I created in **Problem 4**: "Index 17 Observations With

Fitted Values Imposed" and "Index 123 Observations With Fitted Values Imposed", which shows the shape of the curves clearly do vary over the year. I fear, however, that this answer may not be sufficient enough to convince the reader. Instead, consider the following image plot.

```r
# Image plot of each day's curve over the year
image.plot(c(1:280), c(1:24), Golden03,
      xlab = "Day Index",
      ylab = "Hour",
      main = "Image Plot of 24 hour curves for each day")
```



Above is an image plot of each of the 24 hour curves for days where we have data. On the x-axis is the Day Index ranging from 1 to 280. On the y-axis are the hours of each day ranging from 1 to 24. If the shape of the curves were to remain constant throughout the year, we would expect to see the same pattern on each vertical strip of the image plot. That is, if the shape of the curve does not change throughout the year, would expect vertical patterns to remain invariant under translations in the x-direction. As we can see, this is clearly not the case. As the days progress throughout the year, there are significant differences in the curves over each 24 hour period, which is represented by the changing vertical bands that occur for differing values of Day Index. In summer afternoons (roughly between Day Index 150 to 175), there are the highest ozone observations that occur, which are not present at other times of the year. In this sense, the shape of the daily curves do vary over the year.