Assignment 3
MY 459 / MY 559

Benjamin Lauderdale
Methodology Department
London School of Economics

Class/Homework Assignment

We will not need any non-standard libraries for this week's assignment. We are going to use a data set that I have posted to Moodle, called "USBirthData.csv". To load the data, use the command below, substituting in the necessary path to wherever you have saved the data file, or setting the working directory to wherever the data is using `setwd()`.

```
Data <- read.csv("USBirthData.csv")
```

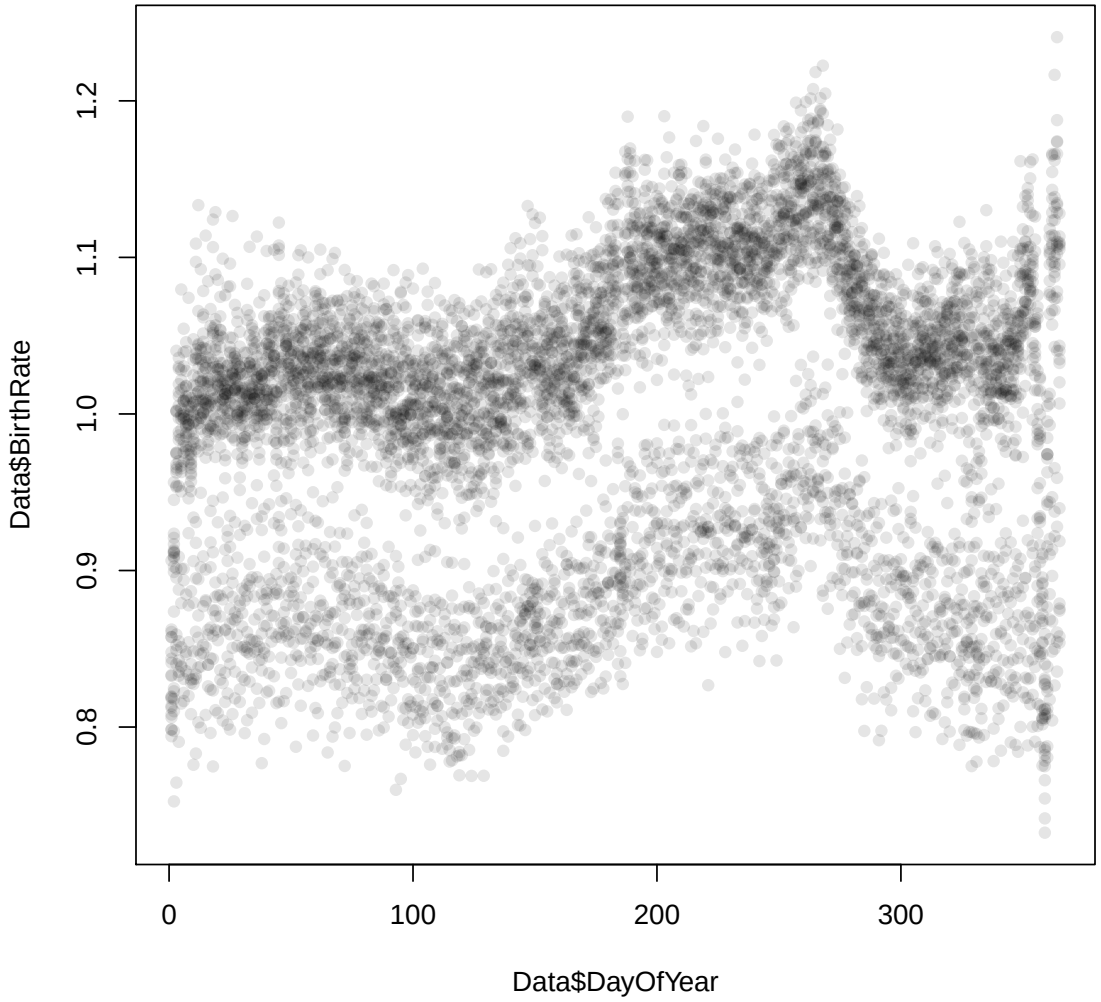
The data set has a row for each non-leap-day from 1969 to 1988. The variables are as follows:

- `Year` Calendar year
- `Month` Calendar month (1 = Jan, 2 = Feb, etc.)
- `Day` Calendar day
- `DayOfYear`: 1 (1 Jan) to 365 (31 Dec).
- `BirthRate`: number of births in the US on that date, divided by the average for that year.
- `MarriedRate`: fraction of babies where the mother was married, divided by the average for that year.
- `RelativeWeight`: average birth weight of babies, divided by the average for that year.

We are going to analyze whether there is consistent seasonality in birth rate, married rate, and relative weight, e.g. higher in summer, lower in winter. We have no idea what the shape of that seasonality might be, so this is a good application for non-parametric regression.

Here is the raw data for birth rate by day of the year:

```
transparentgray <- rgb(0,0,0,0.1)
plot(Data$DayOfYear,Data$BirthRate,col=transparentgray,pch=16)
```



There is a clear seasonal trend apparent to the eye, but the distribution of these data is unusual (bimodal, conditional on day of the year). We are going to figure out what is going on...

Here is a revised version of the kernel regression function, which has two additional features versus the one given in the solutions for Assignment 1. Note that this version will break the previous version of the plot.kreg command, which you may want to fix.

```
weighted.mean <- function(x,w) sum(x*w)/sum(w)
rmse <- function(fitted,observed) sqrt(mean((observed-fitted)^2))

kreg <- function(x,y,bw,kernel = "uniform",fitted.x=x,loo=FALSE){

    output <- list(x=x,y=y,bw=bw,kernel=kernel,fitted.x=fitted.x,loo=loo)
    class(output) <- 'kreg'

    # stop if trying to do leave-one-out with x values other than the observed
    if (!identical(fitted.x,x) & loo==TRUE) stop("Cannot use leave-one-out option unless fitted.x=x")

    # specify kernel function
    if (kernel == "uniform") kernelf <-
        function(xdiff) dunif(xdiff,min=-bw,max=bw)
    if (kernel == "normal") kernelf <-
        function(xdiff) dnorm(xdiff,mean=0,sd=bw/(2*qnorm(0.75)))

    # define function to calculate a single fitted value
    fitted.f <- function(fitted.x,loo=NA) {
        if (is.na(loo)) {
                weighted.mean(y, kernelf(fitted.x - x))
                } else {
                weighted.mean(y[-loo], kernelf(fitted.x - x[-loo]))
                }
        }

    # apply fitted value function to each observation and calculate RMSE
    output$fitted.y <- mapply(fitted.f, fitted.x, if (loo) 1:length(x) else NA)
    if (identical(fitted.x,x)) output$rmse <- rmse(output$fitted.y,output$y) else output$rmse <- NA

    return(output)
}
```

1. There are two new features of the kreg function provided here versus the one in the solutions of Assignment 1.
   (a) What is the `loo` argument doing? Why might this be useful?
   (b) What is the `fitted.x` argument doing? Why might this be useful?

2. We are going to fit a kernel regression with a uniform kernel, with `BirthRate` as the `y` variable and `DayOfYear` as the `y` variable.
   (a) Describe briefly how a uniform kernel regression will determine the estimate for 14 July, given a bandwidth of 4 days.
   (b) Describe briefly what will be different if we instead estimate 31 December, given the same bandwidth.
   (c) What would be a better way of calculating the estimate for 31 December? (Note: I am just asking you to think about what the right calculations would be, not to actually program them. The answer here is not something that our kreg function can do without substantial modification.)

3. Perform 20-fold cross-validation for the kernel regression described in the previous question, where the folds are each defined to be a single calendar year of the data. Use a uniform kernel

and consider integer bandwidths of 1 to 21 days. I have provided the code you need to get started below:

```
temp.predictions <- rep(NA,length(Data$BirthRate))
cv.scores <- rep(NA,21)

for (bw in 1:21){
     for (yr in 1969:1988){
          temp.predictions[Data$Year == yr] <- kreg(
                Data$DayOfYear[Data$Year != yr],
                Data$BirthRate[Data$Year != yr],
                bw,
                kernel="uniform",
                fitted.x=1:365
                )$fitted.y
          }
     cv.scores[bw] <- rmse(temp.predictions,Data$BirthRate)
     }
```

Go through the code carefully to figure out how it works. Running this code might take 30 seconds or so, it has to run the `kreg()` function 420 times, and the kreg function was written to have clear code, not to run quickly!

(a) Provide an argument for using the calendar years as the cross-validation subsets. (Hint: what out-of-sample prediction problem does this simulate?)

(b) Run the code provided, and use the results `cv.scores` to identify the best bandwidth and to plot the cross-validation scores as a function of bandwidth. Which bandwidth minimizes the cross-validation score?

(c) Use the `DayOfWeek` variable to help explain why you found the optimal bandwidth that you did. (Optional: why do you think these patterns exist?)

4. Use the code below to perform the bootstrap for the same kernel regression described in the previous questions. As with the cross-validation case, this code resamples years rather than individual days. Use the bandwidth that minimized the cross-validation (the code expects this to be called `best.bandwidth`).

```
kreg.best <- kreg(
               Data$DayOfYear,
               Data$BirthRate,
               best.bandwidth,
               kernel="uniform",
               fitted.x=1:365
               )

sims <- 200

BootstrapSmooths <- matrix(NA,sims,365)
DayOfYearTemp <- rep(NA,20*365)
BirthRateTemp <- rep(NA,20*365)

for (sim in 1:sims){
     boot.years <- sample(1969:1988,20,replace=TRUE)
     for (b in 1:20){
          DayOfYearTemp[(b-1)*365 + 1:365] <- Data$DayOfYear[Data$Year == boot.years[b]]
```

```
        BirthRateTemp[(b-1)*365 + 1:365] <- Data$BirthRate[Data$Year == boot.years[b]]
        }
    BootstrapSmooths[sim,] <- kreg(
            DayOfYearTemp,
            BirthRateTemp,
            best.bandwidth,
            kernel="uniform",
            fitted.x=1:365
            )$fitted.y
    }

percentile.ci.bounds <- apply(BootstrapSmooths,2,quantile,c(0.025,0.975))
```

As before, go through the code carefully to figure out how it works.

(a) Plot the kernel regression estimate and the bootstrap percentile confidence bands on one plot

(b) Given the confidence bands that you observe, can we be confident that there is a pattern of seasonality in birth rates? By seasonality, I mean variation across the calendar year. To answer this, make sure to clearly state what the confidence bands would look like if there were no seasonality in birth rates.

(c) Modify the bootstrap code to resample days, rather than entire years. (Hint: resampling days is simpler than resampling years.)

(d) Plot the confidence bands for the case where we resampled days, rather than entire years.

(e) Explain why the confidence bands are different when we resample days instead of years. (Hint: the ideal, but computationally expensive, way to combine cross-validation and the bootstrap is to do cross-validation for every bootstrap replication. Why might we get a different optimal bandwidth from cross-validation if we first resampled days?)

5. (Optional) Write a k-fold cross-validation function for the kernel regression. Instead of providing K, write the function so that the user provides a variable that gives the subsets to be held out. Provide your code, and show that you get the same answer as in Problem 3 when you run this function on the same data.

6. (Optional) Write a bootstrap function for the kernel regression. Write the function with an option to resample individual observations, or to resample predefined blocks of observations (i.e. years). Provide your code, and show that you get the same answer as in Problem 4 when you run this function on the same data.