Exercise 9

1. Consider again the ozone dataset. As in exercise 8 take the log of the response and remove the outlier.

   Now we focus on **projection pursuit regression (PPR)**. We want to compare models with different number of ridge functions and different smoothers. Use leave-one-out cross-validation to compare models.

   a) You can choose three different smoothers (argument `sm.method`). The default is `supsmu`, Friedmans "super smoother". Other possibilities are `spline` which uses splines with a specified (equivalent) degree of freedom for each ridge function and `gcvspline` which chooses the smoothness by GCV.
   For each of these smoothers vary the numbers of ridge functions (`nterms`). If your computer power allows you can also try different degrees of freedom (`df`) for `spline`.
   You may also want to use `max.terms` to get better results. The following description is taken from the help-file of `ppr`:
   The algorithm first adds up to 'max.terms' ridge terms one at a time; it will use less if it is unable to find a term to add that makes sufficient difference. It then removes the least "important" term at each step until 'nterms' terms are left.
   R-Hints:

```
## The first argument is a *function*. "..." are arguments which are passed
## to fitfn. Do *not* replace these dots!
cv <- function(fitfn, ...){
  ssr <- 0
  for(j in 1:nrow(d.ozone.e)){
    print(j)
    train <- d.ozone.e[-j,]
    test <- d.ozone.e[j,]
    fit <- fitfn(x = train[,-1], y = train[,1], ...)
    ssr <- ssr + (test[1] - predict(fit, test[-1]))^2
  }
  ssr
}

## Example for a call:
cv.sm.2 <- cv(ppr, nterms = 2, max.terms = 5)
```

   b) Choose the best model from a) and visualize the ridge functions as in the manuscript on page 67.

   c) Interpret the α-matrix (`$alpha`) for your model. You can use `round()` to get a better overview. If you have another model which performs nearly as well as the best but has a much nicer interpretation you may want to prefer it to the best model.

   d) Compare your PPR-model to MARS with an interaction degree of 2. You can use the cv-function from above: `cv(mars, degree = 2)`.

Preliminary discussion: Friday, June 10, 2005. **Deadline:** Friday, June 17, 2005.

Advice: Thursdays from 12.00-13.00, LEO C12.1, Leonhardstr. 27. Or contact either Maik Berchtold, `berchtld@stat.math.ethz.ch`, or Lukas Meier, `meier@stat.math.ethz.ch`.