Durham University **Sheet 1**
Dr. Jochen Einbeck **Postgraduate TrainingWeek 2010 "Nonparametric Smoothing"**

This sheet is intended to make you familiar with some smoothing tools in R, and particularly to give some insight into the topic of bandwidth selection. **Please work on this sheet on your own**.

Task 1.1: (Getting started with R)

(a) Open R 2.10.1 from *Programs/Academic Software/Mathematical Sciences*. We recommend to use Tinn-R (to be found at the same location) to save and execute your code, but any other editor can be used as well. If you are using Tinn-R for the first time on your account, you need to follow the instructions at the course web page to make sure that both programs can communicate with each other.

(c) Load the `lidar` data set from R package **SemiPar**.

(d) Display the data, and produce summary statistics of variables `range` and `logratio`.

Task 1.2: (Local Linear regression)

(a) Use function `locpoly` in package **KernSmooth** to fit a local linear smoother of `logratio` against `range` (read the help file via `help(locpoly)` in order to find out how to do this), and save the fit into an object of name `lidar.ks`. Thereby select the bandwidth h using your first impression from the plot and the summary statistics in (1d). Add the fitted curve to the data scatterplot, e.g. using `lines(lidar.ks, col=2)`.

(b) If you are not happy with the fit, modify the bandwidth h until the fit is satisfactory.

(c) Now use the function `dpill` in the same R package in order to select the bandwidth automatically using plug-in methods. Note the result somewhere.

(d) Now repeat the fit, but this time using the bandwidth automatically selected in (c). Superimpose the fitted curve onto the plot (with the data and your previously fitted curve). Compare the curves.

Task 1.3: (Confidence bands)

(a) Now load the library **locfit** and refit the data with function `locfit`. Note that `locfit` uses by default another kernel (cubic) and uses a non-standard scaling for the bandwidth. To get the same fit as in (2d), you will need a command of type

```
lidar.locfit1 <- locfit(logratio∼lp(range, h=...,deg=1), kern=’gauss’,
                                   data=lidar)
```

where the missing entry ... is the bandwidth obtained in (2c) multiplied with 2.5.

(b) Display the fitted curves obtained in (2d) and (3a) in the same plot. They should be almost indistinguishable.

(c) Display confidence bands and prediction bands using the option `band` of **locfit's** plot function. Which one is wider?

Task 1.4: (Further topics — Select depending on time and interest!)

Degrees of freedom and polynomial regression. Try to figure out how many degrees of freedom the curve fitted in (3a) possesses. Then fit a linear model (with `lm`) using a polynomial with approximately the same order. Look at the fitted curve. Is it competitive to the locally fitted curves above?

Derivative estimation. Now, use any of the two R packages to estimate the first and second derivative (both are able to do it). Display the results and think about if they are reasonable!

Density estimation. Provide kernel density estimates of the variables `range` and `logratio` seperately [`density(...)`]. Then, use function `bidens` from the lecture notes to compute a bivariate density estimate of the `lidar` data set (taking reasonable bandwidths of your choice). Next, load package **np**, type `npudens(tdat = lidar)`, and visualize the result. `npudens` selects bandwidths automatically – read them from the R output, feed them into `bidens`, and compare the result to that obtained previously.