Lab #1: Practice in R

Biostat 216: Machine Learning in R for the Biomedical Sciences

The purpose of this lab is to get you comfortable working in R and trying a few things from the first lecture.

Simulating data:

We will simulate something like the weight vs. age data from a cubic function and add some noise and then apply a few methods.

1. Write a function in R to input the value x (age) and output, "noise-free" weight: $f(x) = -0.0004x^3 + 0.025x^2 + 2.0x + 50.0$
2. Next we will simulate some values for x. Use the `sample` function to generate a random sample of integers between 40 and 80 of size 1000 (you will need the `replace=TRUE` option to make sure the numbers can be resampled otherwise you will run out of values quickly). Don't forget that you can type `?sample` to get help as to how to use the function.
3. Now generate a vector of the output of your function to generate the noise-free weights for each of the 1000 individuals.
4. Use rnorm to generate some normally distributed noise with mean 0.0 and sd of 10.0 for each of the x values.
5. Add the noise-free weights to the noise to get your simulated outcome data.
6. Put these together in a dataframe with columns for age, weight (the observed outcome with noise), trueFx (the noise-free weight value), and noise.
7. Generate a plot of weight against age.

Applying smoothing kernels:

8. Fit a nearest neighbors curve with `ksmooth` using a bandwidth of 10 and the `"box" kernel`.
9. Add the kernel smoothing line to your data plot using the `lines` command. Hint: look at the object you have generated with `ksmooth` (i.e. type `name_of_the_object` or `print(name_of_the_object)`. Also, try `names(name_of_the_object)`.
10. Fit another curve this time using the `"normal" kernel`.
11. Add this curve to the plot (use a different color for the line using `col = "your favorite color"`)
12. What is your impression of the `normal` vs. `box` kernels

13. Fit another 2 curves with the `"normal" kernel` using bandwidths of size 5 and 20.
14. Which of the curves do you think provides the best fit to the data?

Fitting linear models:

15. Regenerate the plot of weight against age.
16. Fit a linear regression to the data using the `lm` command.
17. Run `summary(your_linear_model_name)` to get an idea of the fit
18. Now add the regression line to the data (you can `abline` use for this)
19. Now fit quadratic and cubic models: you will need the `I` function to set quadratic and cubic terms in the regression, e.g. `I(x^2)`
20. Add the fitted lines to the plots, you will need to extract the coefficients of the linear models using `coef` and then build functions that you can use to plot. To plot a function you can use the `curve` command (you will need the option `add = TRUE`).
21. Which curve do you prefer? Do you think these curves provide better fits than when we used only 50 points in the lecture?