Problem set 5

S681

Upload your typed answers to Canvas as a PDF by 11:59 pm, Sunday 7th April. Include R code where applicable.

1. (20 points.) The R package `nycflights13` contains a data set `flights` that contains information on 336,776 flights departing New York in 2013. The variables we will be interested in are:

   - `hour` and `minute`: These give the hour and minute the flight was scheduled to depart.
   - `arr_delay`: The time that the flight's arrival was delayed (in minutes.)

   The object of this homework is to predict `arr_delay` from departure time, as measured by `hour` and `minute`. These are the only variables we'll use.

   (a) Install the package `nycflights13`. Optional but recommended: remove *only* observations for which at least one of `hour`, `minute`, and `arr_delay` are missing.

   (b) Randomly separate your data into a training set and a test set.

   Warning! This is a big data set, and unless you have a much better computer than I do, some methods may be infeasible to fit on a big training set. So either keep your training set to a manageable size, or else just use a sample from your training set if a method won't run on the whole thing.

   (c) Fit three models:

   i. Model A: Use `loess()`.

   ii. Model B: Use `smooth.spline()`.

   iii. Model C: Use `gam()` in `library(mgcv)`.

   Explain your choices of tuning parameters and other arguments. (If you just use defaults, state what the defaults are and explain why they're sensible.)

   Attention! A minute is 1/60th of an hour! This feels like information you should use!

   (d) Plot curves for all three models on the same graph. (Make clear which model is which.)

   (e) Apply your models to your test set and calculate the root mean squared prediction error for each model. Does any model do notably better or worse than the others? If you had to pick one model, which would you pick, and why?

2. (15 points.) The file `s681-sp19-training.txt` in the Data folder on Canvas contains 2000 (x, y) pairs, with x iid Uniform$(0, 1)$ and

$$y_i = r(x_i) + \epsilon_i$$

where the ϵ_is are iid (but not necessarily normal).

Load the data into R:

```
training = read.table(..., header=TRUE)
x.train = training$x
y.train = training$y
```

Do not include the preceding commands in the R code that you submit; we will assume that you start with vectors called `x.train` and `y.train` in your workspace.

Using a method of your choice, estimate the function $r(x)$ and write R code that produces a vector called `my.predictions` that gives estimates for a sequence of 999 x-values from 0.001 to 0.999, at spacing 0.001. (Note: Feel free to collaborate on this — it's fine if two people have the same predictions.)

Upload two files:

- A .R file that allows Xixi and me to reproduce your vector `my.predictions`. For reproducibility, the first line must be `set.seed(681)`.
- A HTML/PDF/whatever file that contains (i) a line graph of your predictions, and (ii) a paragraph explaining what method you used and why you chose that method.

You'll be graded on all of accuracy, reproducibility, plot, and explanation.

3. (15 points.) The data set `fossils.txt` has five columns:

   - species: name of species
   - ln_mass: log of body mass (in grams)
   - ln_old_mass: log of body mass of its ancestor species (in grams); NA if unknown
   - first_appear_Mya: first appearance in fossil record (millions of years ago)
   - last_appear_Mya: last appearance in fossil record (millions of years ago); NA means it still exists

   (a) Fit and graph an additive model using `gam()` with `ln_mass` as the response, using whichever x-variables you think will give you accurate prediction.

   (b) Fit and graph an additive model using `gam()` with the *change* in log mass `ln_mass - ln_old_mass` as the response, using whichever x-variables you think will give you accurate prediction.

   (c) Interpret the two models and compare their predictions for `ln_mass` (e.g. by drawing a scatterplot).

   Note: A linear model is a special case of an additive model...