Homework 1: Chicago and Neighbors

36-402, Spring 2019, Section A

Due at 6:00 pm on Wednesday, 23 January

AGENDA: Remembering how linear regression works; remembering what linear regression can and cannot do; trying out a linear smoother that is not linear regression.

The data set `chicago`, in the package `gamair`, contains data on the relationship between environmental conditions and the death rate in Chicago from 1 January 1987 to 31 December 2000. The seven variables are: the total number of (non-accidental) deaths each day (`death`); the median density over the city of large pollutant particles (`pm10median`); the median density of smaller pollutant particles (`pm25median`); the median concentration of ozone (O_3) in the air (`o3median`); the median concentration of sulfur dioxide (SO_2) in the air (`so2median`); the time in days (`time`); and the daily mean temperature (`tmpd`).

Data-Analysis Problems

1. Load the data set and run `summary` on it.

   (a) (1) Is temperature given in degrees Fahrenheit or degrees Celsius?

   (b) (2) The pollution variables are negative at least half the time. What might this mean?

2. *Death over time*

   (a) (5) Plot the number of deaths versus the time. Describe any patterns you see. *Hint:* The plot becomes easier to interpret if you make the horizontal axis the calendar date. What does the following code fragment do?

   ```
   day.zero <- as.Date("1993-12-31")
   chicago$date <- day.zero + chicago$time
   ```

   (b) (3) Linearly regress the number of deaths on time. Add the regression line to the previous plot. Report the slope coefficient (to reasonable precision). Is it significantly different from zero?

   (c) (4) Plot the residuals against the time. Describe any patterns you see.

(d) (5) Carefully explain the interpretation of the regression slope.

(e) (2) Is there any reason to doubt the validity of the significance test here?

3. *Neighbors in time* Install the package `FNN`, and use the `knn.reg` function when asked to do k-nearest neighbors. *Hint:* Figure 1.5 in the textbook, and the accompantying code. In particular, the `test` argument to `knn.reg` should (in this case) be a one-column matrix.

   (a) (5) Do a k-nearest-neighbor regression of death on time, using $k = 3$ nearest neighbors. Generate a predicted value for each day in the data set. Use these predicted values to add the estimated regression function to the plot you made in Problem 2b. Describe the shape of the estimated function.

   (b) (4) Carefully explain, in your own words, how the predicted values in Problem 3a are derived from the original data.

   (c) (3) Repeat Problem 3a for $k = 30$. Describe how (if at all) the new estimate of the curve differs from the old.

4. *Death likes it cold?*

   (a) (3) Plot the number of deaths against temperature. Describe any patterns you see.

   (b) (4) Linearly regress the number of deaths on temperature. Add the regression line to the previous plot; report the slope coefficient (to reasonable precision).

   (c) (4) Carefully explain the interpretation of the slope coefficient.

   (d) (4) Plot the residuals of the regression of deaths on temperature. Describe any patterns you see.

5. *Death likes it hot also?* (5) Refer Problem 3 for hints on using `knn.reg`.

   (a) (5) Do a 30-nearest neighbor regression of the number of deaths on temperature. Generate predictions at values of temperature that span the observed range in the data set. Use these predictions to add the estimated regression function to the plot you made in Problem 4b. *Hint:* If your plot looks weird, look carefully at Figure 1.5 in the textbook again.

   (b) (5) Describe the shape of the estimated regression function, and contrast it with the linear regression. What is the *qualitative* difference between the two estimates here?

6. *Hypotheticals* Many climate models predict that the Chicago area might, by the end of the 21st century, be 4 degrees Celsius warmer than the time-period in which this data was collected.

   (a) (1) Add a new column to the `chicago` data frame, which takes each day's observed temperature, and makes it 4 degrees Celsius hotter. *Hint:* The temperatures are in Farenheit. A *change* of 4 degrees Celsius, expressed in

Farenheit, is not the same as adding the Farenheit temperature that corresponds to 4 degrees Celsius. (Why not?) Consequently, Googling "4 degrees Celsius in Farenheit" will give you very misleading answers. *Comment:* an even warming of 4 degrees Celsius throughout the year isn't what climate models predict, but this will do for now.

(b) (4) Use the linear model you estimated in Problem 4b to estimate the change in the number of people predicted to die on each day in the data, and report the average change. *Hint 1:* Do *not* re-estimate the model with the new column as the predictor variable. *Hint 2:* You could use `predict` to get predictions at the old, observed temperature, and the new, hypothetical temperature, and take the difference. (See the handout on "`predict` and Friends" on the class homepage under Lecture 2.) *Hint 3:* There is a faster way to do this for the linear model.

(c) (5) Use the 30-nearest-neighbor regression you estimated in Problem 5a to estimate the change in the number of people predicted to die on each day in the data, and report the average change. *Hint 1:* Do *not* re-estimate the model with the new column as the predictor variable. *Hint 2:* Use the `test` argument of `knn.reg` to get predictions at values which might or might not be in the data set.

Theory Problems

These are all from the exercises at the end of Chapter 1 of the textbook.

1. (3) Exercise 1.7.
2. (4) Exercise 1.8.
3. (4) Exercise 1.9.
4. (5) Exercise 1.10.

These problems need you to write out math. Please see the handout "Using R Markdown for Class Reports" (`http://www.stat.cmu.edu/~cshalizi/rmarkdown/`, link on the syllabus) for assistance in doing so. If you just can't make it work, please write out the math by hand, scan it, and include the scans in your submission. You will lose a few points for doing so this time; as the semester goes on, the penalty for hand-written math will increase. Do not use Word for math.

RUBRIC (10): The text is laid out cleanly, with clear divisions between problems and sub-problems. The writing itself is well-organized, free of grammatical and other mechanical errors, and easy to follow. Questions which ask for a plot or table are answered with both the figure itself and the command (or commands) use to make the plot. Plots are carefully labeled, with informative and legible titles, axis labels, and (if called for) sub-titles and legends; they are placed near the text of the corresponding problem. All quantitative and mathematical claims are supported by appropriate derivations, included in the text, or calculations in code. Numerical results are reported to appropriate precision. Code is properly integrated with a tool like R Markdown or knitr, and both the knitted file and the source file are submitted. The code is indented, commented, and uses meaningful names. All code is relevant; there are no dangling or useless commands. All parts of all problems are answered with actual coherent sentences, and raw computer code or output are only shown when explicitly asked for.