Homework 1: A Cherry-Blossom-Viewing Party

36-467, Fall 2018

Due at 6 pm on Wednesday, 5 September 2018

AGENDA: Practice with visualization and basic summary statistics on time series; estimating trends by linear regression, by moving averages, and by spline smoothing; visualizing detrended fluctuations; probing for changes in a time series.

Cherry trees flower in the spring, and the opening and blooming of the blossoms is very sensitive to temperature — full bloom is advanced by warm weather and held back by cold. Because of the important role of the cherry blossom in Japanese culture over the last 1200 years, it is has been possible to work out the date of full blossom at Kyoto going back to the early 800s AD[1]. Our first homework will analyze this data to look for trends.

The data file is at `http://www.stat.cmu.edu/~cshalizi/dst/18/data/kyoto.csv`. It's the data prepared by Prof. Yasuyuki Aono and collaborators, from `http://atmenv.envi.osakafu-u.ac.jp/aono/kyophenotemp4/`, slightly reformatted to make it easier to work with in R. The three columns record (1) the year (AD), (2) the day of the year of full flowering, i.e., counting January 1 as day 1 and December 31st as day 365 or 366; (3) the date in the contemporary (Gregorian) calendar, in MMDD format (so March 15 would be 0315).[2]

1. (5) Load the data file, and reproduce the following summary statistics for the day (not date) of full flowering:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    86.0   100.0   105.0   104.5   109.0   124.0     388
```

   Explain why it would make much less sense to calculate summary statistics for the date column.

2. *Visual EDA*

   (a) (5) Plot the day of flowering against the year.

[1] There's more about all this in the slides for the first lecture.

[2] Prof. Aono's full data set also includes information about data sources and the historical evidence (e.g., diary descriptions of cherry-blossom viewing parties vs. newspapers), which would be valuable for a more detailed examination of this data.

(b) (2) Add a horizontal line to the plot showing the mean day of flowering.

(c) (2) Add horizontal lines to the plot showing the 25^{th} percentile ($= 1^{\text{st}}$ quartile) and 75^{th} percentile ($= 3^{\text{rd}}$ quartile).

(d) (5) Comment on any patterns you see in your plot.

3. *Linear time trends* The command `lm(Y~X, data=df)` will fit a linear regression of the `Y` variable of the data frame `df` on the `X` variable of the same data frame. The command `abline(lm(Y~X, data=df))` will add the corresponding straight line to the current plot.

   (a) (5) Regress the day of flowering on the year for the whole data set. Report the slope and explain what it means.

   (b) (3) Repeat the regression, but using only years ≤ 1750. Again, report the slope and explain what it means.

   (c) (2) Now do it using only years > 1750.

   (d) (5) Add all three lines to the final plot from Problem 2. Make sure to clearly distinguish visually between the linear trend lines and the horizontal levels you plotted before. Comment on the plot.

4. *Moving average smoothing* Calculate a centered 5-year moving average of the day-of-flowering time series. That is, for each year t, calculate

$$\overline{x}_t = \frac{1}{5} \sum_{k=-2}^{k=2} x_{t+k}$$

   (a) (5) Give the code you used to do this. *Hint:* See Recipe 14.10 in *The R Cookbook* for one way to do it.

   (b) (3) Carefully explain how your code handles NA values. *Hint:* Look at what happens around 1945. (Why 1945?)

   (c) (2) Modify your code to calculate a centered 11-year moving average.

   (d) (4) Add both of your moving average series to the final plot from Problem 2. Make sure that the moving-average curves are clearly distinguished visually from the data, from the horizontal levels you plotted before, and from each other.

   (e) (5) Comment on the plot.

5. *Spline smoothing* The function `smooth.spline` fits a smooth curve through the data points, balancing smoothness against coming close to the data. The command `lines(smooth.spline(x, y))` adds the fitted curve to the current plot.

   (a) (5) Add the spline to the final plot from Problem 4. Make sure it is visually distinct from the moving average lines. Comment on the plot. How is it similar to the plots with the moving averages, and how does it differ?

(b) (6) The fitted values of the spline are stored in the `$y` component of its return value. Create a vector of residuals which stores, for each year, the difference between the actual day of flowering and the fitted value from the spline. Plot this against year. Describe the patterns in the plot, if any. *Hint:* Be careful about NAs when calculating the residuals.

(c) (5) The `df` argument to `smooth.spline` lets you set the number of degrees of freedom used in estimating the spline curve. What happens to the curve when you set `df=2`? What happens when you set `df=100`? Can you give any reasons to choose among two curves and the one from Problem 5a? ("I trust R's default settings" is not a good reason.)

6. *Evidence for a change*

(a) (3) Use the data for years ≤ 1750 to find the 75^{th} percentile of the flowering-day distribution.

(b) (2) If the distribution of flowering-days is unchanging over time, approximately how many years > 1750 should be below that pre-1750 75^{th} percentile?

(c) (5) How many years > 1750 are actually below the pre-1750 75^{th} percentile?

(d) (6) Looking just at the ≤ 1750 data, can you find any interval of 265 consecutive years with at least as many years at or below the pre-1750 75^{th} percentile as the years > 1750? If so, give the dates of the intervals.

(e) (5) Is the ratio of the number of 265-year intervals you found in the last problem to the total number of such intervals a p-value? If so, what is the hypothesis being tested? If not, why not? Does it matter whether or not you consider overlapping intervals?

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, without dangling or useless commands. All parts of all problems are answered with coherent sentences, and raw computer code or output are only shown when explicitly asked for.