Homework 5: There Were Giants in the Earth in Those Days

36-402, Spring 2015

Due at 11:59 pm on Monday, 16 February 2015

AGENDA: Explicitly: splines, bootstrap, simulation, comparing a simulation to data; implicitly: more practice writing, testing, and debugging simple functions.

GRADING: The problems add up to 90 points. The remaining 10 points are reserved for style and clarity, per the rubric at the end of this assignment. (*Note:* The rubric has changed slightly from last time; read it carefully.) Re-writing functions from earlier steps inside later ones will yield only partial credit. When a question calls for checking something by repeated simulation, find a way of showing the test was passed without displaying lots of raw R output.

Some biologists argue that larger animals tend to have advantages over smaller members of their species, so that natural selection should tend to lead to an increase in size within an evolutionary lineage[1]. There is also some evidence that larger species tend to be shorter-lived than smaller ones[2]. In this assignment, we will look at the evidence for an increase in species size within lineages, and how the trade-off between these two forces might lead to a stable distribution of sizes across species.

We will use two data sets:

- The North American Mammalian Paleofauna Database (nampd.csv) lists, for about 2000 living and extinct species, the log of the mass, in grams, of a typical member of the species; the log mass of the ancestral species (when known); and the dates of the species' first and last appearance in the fossil record, in millions of years ago. If the last appearance date is `NA`, the species is still alive. This means you should *not* just throw away all rows containing NAs.
- The Masses of Mammals (MoM.txt) gives, for about 4000 living species, their mass in grams, identifying codes for the species, genus, and other taxonomic

[1] Among other things, larger animals may be harder for predators to attack, find it easier to over-come prey or other members of their species, and be more efficient metabolically. For more, see, e.g., John Tyler Bonner, *The Evolution of Complexity, by Means of Natural Selection* (Princeton University Press, 1988).

[2] This may be because larger animals need more food in total, and possibly more specialized food sources, so they are more vulnerable to shifts in their environment.

groups, and an indicator for whether the species lives in the land or in the water.

The model we will work with goes as follows: At any given time t, there is a collection of n_t species, whose masses are $X_1, X_2, \ldots X_{n_t}$. At each time step, one current species A gets picked, uniformly at random, to evolve into two new species. The masses of a descendant species X_D is related to that of its ancestor, X_A, by the model

$$X_D = \exp(r(\log X_A) + Z) \tag{1}$$

where $Z \sim \mathcal{N}(0, \sigma^2)$, and r is a function to be learned from the data, subject to the restriction that X_D has to be at least $x_{\min}$ and at most $x_{\max}$. The ancestor X_A is removed from the current list of species, and its two independent descendants are added. After this, all species currently in the list have a risk of going extinct, with the probability for a species of mass x going extinct being a function of their mass,

$$p_e(x) = \beta x^{\rho} \tag{2}$$

Any species become extinct are removed from the collection. We then iterate the model again.

In all of the following questions, unless otherwise specified, you may take $\sigma^2 = 0.63$ (what are the units?), $x_{\min} = 1.8$ grams, $x_{\max} = 10^{15}$ grams, $\rho = 0.025$, and $\beta = 1/5000$.

1. (5) Linearly regress the log of the new mass on the log of the ancestral mass. Plot this regression line, along with a scatter-plot of the data, *in units of grams*, not log-grams. Carefully explain the interpretation of both the slope and the intercept. A rote recitation of "a one unit change", etc., will not receive full credit; think about the model, the transformations, and what the transformed model says about the variables.

2. (10) Use a smoothing spline to do a nonparametric regression of log new mass on log ancestral mass. Create a plot showing the data points, the model from question 1, and the spline, making sure that the axes are in units of grams, not log-grams.

3. (20)

   (a) (10) Using resampling of residuals, calculate 95% confidence bands for the spline curve, and add them to the plot.

   (b) (10) Using resampling of cases, calculate standard errors for the spline curve, and add bands at ± 2 standard errors to the plot.

4. (10) Write a function, `rmass`, which takes as inputs a single ancestral mass X_A (not $\log X_A$), an estimated spline function r, and any other parameters required by the model, and returns a single random value for X_D, according to Eq. 1. Make sure the returned value is in grams, not log grams. You will probably find it easiest to keep generating candidate values for X_D, until you get one which is between the limits. *Hint:* `while`

(a) (2) What model parameters does your `rmass` need?

(b) (4) Check, by repeated simulation, that the output is always between $x_{\min}$ and max, even when X_A is brought near either limit.

(c) (4) Using the spline curve you estimated in question 2, create 150 evenly spaced X_A values between $x_{\min}$ and max, generate an X_D for each of them, and fit a spline curve to the simulated values. Check that it is close to, but not identical with, the one you found from the data. (Why should it not be identical?)

5. (10) Write a function, `origin`, which takes the same arguments as `rmass`, except that instead of one ancestral mass it can take a vector of them. `origin` should pick one entry from the vector to be X_A, and generate two independent values of X_D from it. One of these should replace the entry for X_A, and the other should be added to the end of the vector.

   (a) (4) Check, by simulating with a length-one vector of ancestral masses, that neither component of the returned value matches the ancestral mass (why?), that both components have the same marginal distribution, and that the two components are uncorrelated with each other.

   (b) (2) Check, by simulating, that if the input vector of masses has length m, the output vector always has length $m+1$. (Check at least two values of m.)

   (c) (4) Check, by simulating, that $m-1$ entries in the output match the input exactly. Check this for at least two values of m. *Hint:* `is.element`, or `%in%`, or `match`.

6. (5) Write a function, `extinct.prob`, which takes as inputs a vector of species masses, and parameters ρ and β, and returns the extinction probabilities according to Eq. 2.

   (a) (2) Check that if the masses are `c(100, 1600, 10000)` grams, $\rho = 1/2$ and $\beta = 1/200$, then `extinct.prob` returns the right values.

   (b) (1) Check that if $\rho = 0$, the output probabilities are all β, no matter what the masses are.

   (c) (1) Check that if the input masses are all equal, so are the returned probabilities, for at least three of different combinations of mass, ρ and β.

   (d) (1) Check that if $\rho \neq 0$ and $\beta \neq 0$, and the masses are all different, then the returned probabilities are all distinct.

7. (5) Write a function, `extinction`, which takes a vector of species masses, ρ and β, and returns a possibly-shorter vector which removes the masses of species which were probabilistically selected for extinction. Be sure to handle the (unfortunate) case where every species goes extinct. *Hint:* Explain what `rbinom(n,size=1,prob=p)` does when `p` is a vector of length `n`.

(a) (1) Check that if $\beta = 0$, the output vector is always the same as the input vector.

(b) (3) Create a case where the input masses are all equal, and ρ and β are set so that the extinction probability should be $1/2$. Check that the output is, on average, half as long as the input.

(c) (1) In the same test cases as the previous part, check that all the values in the new vector of masses were also in the old vector of masses.

8. (5) Write a function, `evolve_step`, which takes as inputs a vector of species masses, plus all needed parameters and estimated curves; calls `origin` and `extinction` as appropriate; and returns a new vector of species masses. How do you know it works?

9. (5) Write a function, `mass_evolve`, which takes the same inputs as `evolve_step`, plus an additional number `T`; iterates `evolve_step` T times; and returns the final vector of species masses. How do you know it works? *Hint:* There will almost certainly need to be a `for` loop inside the function.

10. (5) In this question, use the default parameter values, and the spline you estimated in question 2.

    (a) (1) Run `mass_evolve` starting from a single species with a mass of 120 grams for $T = 2 \times 10^5$ steps. Save the output as `masses.1`. Plot the histogram.

    (b) (1) Re-run `mass_evolve` from the same conditions. Save as `masses.2`. Plot the histogram.

    (c) (1) Re-run from the same conditions but for $T = 4 \times 10^5$ steps, saving as `masses.3`. Plot the histogram.

    (d) (1) Change the starting condition to two species, one of 40 grams and one of 1000 grams. Run twice, both times with $T = 2 \times 10^5$, saving the results as `masses.4` and `masses.5`.

    (e) (1) How do the distributions of the various `masses` compare to each other?

11. (5)

    (a) (1) Load the Masses of Mammals data set, and plot the histogram of masses for land species.

    (b) (2) Compare, verbally, the distribution for land species to that obtained from the simulations.

    (c) (2) Compare the distributions using QQ plots.

12. (5) Does the output of the simulation model match the distribution of masses we actually observe? Are the differences between the model and reality bigger

than those between different runs of the simulation? Are there qualitative distinctions between the simulation-to-simulation differences, and the simulation-to-reality differences? Support your answers by reference to the plots you have already made, or, if need be, new ones.

Note: more advanced techniques for comparing distributions exist, and we'll cover some of them later in the course.

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. Figures and tables are easy to read, with informative captions, axis labels and legends, and are placed near the text of the corresponding problems. 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 either properly integrated with a tool like R Markdown or knitr, or included as a separate `.R` file. In the latter case, the code is clearly divided into sections referring to particular problems. In either case, the code is indented, commented, and uses meaningful names. All code is relevant to the text; there are no dangling or useless commands. All parts of all problems are answered with actual coherent sentences, and never with raw computer code or its output. For full credit, all code runs, and the Markdown file knits (if applicable).

EXTRA CREDIT: (10) Re-write the code so that Z, rather than being drawn from a Gaussian distribution, comes from resampling the residuals of the fitted spline curve. What do you have to modify? How much do the results change? Which version fits the observed mass distribution better?