Practical challenge 1

Probability and random variables

Foundations of Data Science—DJW—2018/2019

These questions are optional. They are not intended for supervision, and they are not examinable material. They are to give you practical experience of work based on the theoretical material covered in lectures, if you want it, and to reinforce your skills in scientific computing.

If you want guidance, you can (i) ask your fellow students for help in the Moodle forum, (ii) come to the practical classes in the second half of term and ask demonstrators, (iii) wait until model solutions are released, after the practical classes.

Question 1. This question is about the two-straight-line distribution we considered in lectures, with cumulative distribution function

$$\log\Big(1 - F(x)\Big) = \alpha - \beta \log x - \gamma \max(\log x - \theta, 0)$$

which has log density function

$$\log f(x) = \begin{cases} -\log x + \alpha - \beta \log x + \log \beta & \text{if } x \le e^\theta \\ -\log x + \alpha - \beta \log x + \log(\beta + \gamma) + \gamma\theta - \gamma \log x & \text{if } x \ge e^\theta \end{cases}$$

We'll assume θ is known and the other parameters are unknown. The obvious way to estimate the parameters is with numerical maximum likelihood estimation,

```
URL = "https://teachingfiles.blob.core.windows.net/founds/weblog_sizes.txt"
sizes = pandas.read_csv(URL, header=None, names=['size'])['size'].values
θ = 11 # treat this as known, because otherwise the optimization is too hairy!
def loglik(α,β,γ):
    # assume f is written as a vectorized function, i.e. f(sizes) returns [f(x) for x in sizes]
    return np.sum(np.log(f(sizes, α,β,γ)))
initial_guess = [...]
scipy.optimize.fmin(lambda p: -loglik(p[0],p[1],p[2]), initial_guess)
```

(a) Program a random number generator to generate samples from this distribution.

(b) Plot a histogram of simulated values for $(\alpha, \beta, \theta) = (1.5, 0.4, 0.5)$, superimposed on a histogram of the original dataset.

(c) Show that the density function is as given by the formula above.

(d) The code produces a warning message, RuntimeWarning: divide by zero encountered in log. Diagnose and correct this bug.

Question 2. This question asks you to investigate a time series dataset, i.e. a collection of timestamps $t_0 \le t_1 \le \cdots \le t_n$ The goal is to figure out the process that generated this dataset. It's often useful to work in terms of the inter-event times $x_1 = t_1 - t_0$, $x_2 = t_2 - t_1$, etc. Typical questions: what's the mean and variance of inter-event times? are inter-event times independent?

```
# This dataset contains timestamps measured in seconds
URL = 'https://teachingfiles.blob.core.windows.net/founds/practs.txt'
t = pandas.read_csv(URL, header=None, names=['t'])['t'].values
practs = pandas.DataFrame({'t': t[:-1], 'x':np.diff(t)})
```

(a) Split the time series into 15 minute intervals, and plot the mean and variance of inter-event time in each interval. *This is a quick and dirty way to see if the distribution is stable over time.*

(b) Plot the empirical distribution function of inter-event time for each 30 minute interval. *This lets you compare distributions, not just means and variances.*

(c) Plot a series of qq plots (see section 1.5 of notes), comparing inter-event times in each 30 minute interval to the entire distributions of all inter-event times. *This is a more refined way to compare distributions, and reveals things that might be obscure on simpler plots.*

(d) Draw a scatter plot of x_{n+1} on the y-axis against x_n on the x-axis. Measure the correlation (numpy.corrcoef). *This is a quick and dirty way to see if successive inter-event times are independent.*

(e) Let x' be a randomly shuffled version of x. Measure the correlation between x'_{n+1} and x'_n. *There should be no correlation of course—but because of random fluctuations the computed correlation won't be precisely zero. This shuffling experiment gives an idea of how close to zero you'd expect, if there truly is no correlation. Another good way to see how confident you should be is to split the shuffled data into pieces and measure the correlation in each piece, and see how much your answers differ.*

(f) Let $y_i^m = x_i + \cdots + x_{i+m-1}$, the sum of m consecutive inter-event times, and let Y^m be a typical value of this. If inter-event times were independent, then (by the rule for variance of sums of independent variables) $\operatorname{Var} Y^m$ would be equal to $m\sigma^2$, where σ^2 is the variance of inter-event times. Plot $\operatorname{Var} Y^m$ as a function of m, and superimpose $m\sigma^2$.

(g) How could you indicate confidence, in your answer to part (f)?

(h) What is your best explanation of the process that generated this dataset?