Problem Set 2.1
MPCS 58020
Spring 2021

Due Date: Monday April 19, 2021 @6:00am

Logistics and Submission – 5 pts

- You may use any programming language, but you may not use any high-level functionality specific to the homework problem. In particular, you may use basic linear algebra routines, but not specialized statistics libraries or black-box solvers (e.g. regression toolkit). You may use a uniform random number generator on $(0, 1)$ and on integers within a range, but you may not use built-in generators for more complicated distributions. If you are unsure if a feature is allowed, ask for clarification on Slack.

  Specific to homework 2: do not use built-in functions for computing the mean and variance. You can use built-in functions for summing arrays, however. For example, in Python NumPy, `np.mean()` and `np.var()` are forbidden but `np.sum()` is allowed.

- For the programming/simulation problems, your program should support running with no arguments and use enough realizations (trials) to get a good estimate of the result with a reasonably low execution time (less than 5 seconds is preferable). A command line program that prints the results is preferred, but a main script (e.g. in MATLAB) is also fine.

- All written responses, plots, and required program outputs should be included in (ideally) a **single** writeup file named `writeup.pdf`. This could be rendered from a Jupyter notebook, LaTeXsource, and/or scanned pages. At most 2x PDFs are allowed, if you are having trouble merging the rendered Jupyter notebook and a scanned document, for example.

- For non-programming homework problems involving derivations, you may use your favorite symbolic solver (Mathematica, Wolfram Alpha, MATLAB, SymPy, Maple, SageMath) to evaluate integrals.

- You must also include a plain-text `README` file with your name, assignment number, list of any references used (if any), a clear explanation of how to run your simulations, and what arguments (if any) each of the scripts takes.

- If you use a compiled language (e.g., C++), you must provide a `Makefile`.

- All source code and scripts should be labelled according to the problem number. For instance, code required for problem 6 should be labelled as `p6_d[.py, .jl, .m, ...]` and `p6_e[.py, .jl, .m, ...]` (if the parts are contained in separate files). If you have more than one source file for a given problem and part, give them more descriptive labels so we know where to start grading and what is contained in them.

- *Aside on Jupyter notebooks:* Jupyter notebooks are an increasingly popular, open-source, interactive web-based medium for computational work. Code, plots, and rich text can all live side-by-side in a single notebook. All or part of your assignment can be written in and submitted as a single Jupyter notebook. You can use any of the 100+ kernels supported Jupyter, but let us know ahead of time if you are going to use a more exotic kernel than the core three: IPython, IRkernel, IJulia. Still include a minimalist `README` if you submit a notebook containing all the code and instructions.

- Your programs should be human readable, which means it they should be well-commented, well-structured, and easy for the grader to understand. Messy and/or poorly commented code that is unreasonably difficult to follow may receive deductions even if it is logically correct.
- Your submission will be in the form of a single `.zip` / `.tar` / etc. archive containing the following items:
  1. PDF writeup
  2. README
  3. source code files and/or Jupyter notebook
  4. If necessary, a Makefile.
- The total submission should be less than 20 MB in size. Each file should be no larger than a few MB (preferably/typically much smaller).

General comments:

Many of these prompts (especially the programming exercises) are written in a way to encourage exploratory analysis. Since an objective of this course is to to help you become a practitioner of these methods for modeling real-world, stochastic phenomena, some aspects of the solutions may be open-ended.

Please put some thought into making clear, high-quality visualizations when including them in a solution. Even when a plot is not explicitly required in the prompt, it might be a helpful addition. Always label your axes, use a figure legend when multiple datasets are plotted, and try to avoid cluttered plots.

A few sentences describing and interpreting results displayed in the figures and/or function output are always appreciated. Sometimes the interpretations and conclusions are not clear-cut, and your writeup can reflect that.

Problems

1. (10 points) Generate an ensemble of synthetic time series in the language of your choice using the so-called *logistic map*:

$$x_{i+1} = rx_i(1 - x_i)$$

Using a value $r = 3.8$ and initial condition $x_1 \sim \text{Uniform}(0, 1)$, create an ensemble of $m = 1000$ realizations, each with length $n = 10000$ points (each realization is made unique with a different random choice of x_1). Estimate and plot the mean function μ_t by averaging over the realizations in the ensemble. Estimate and plot the lag h autocorrelation function $\gamma(h) \equiv \gamma(t + h, t)$ for multiple values of t by averaging over the realizations in the ensemble.

Is the time series stationary?

Nota bene: To definitively show that a time series perfectly satisfies the requirements of weak stationarity as defined by Shumway, you would have to estimate the ACF $\rho(s, t)$ for **every** possible value of s, t (in addition to checking the mean function μ_t for all t).

This can be quite expensive computationally, especially if your ACF implementation is inefficient (does not take advantage of vectorization, etc.). Because of this, it is often not practical to check if a time series is exactly stationary. The approximation of stationarity and the analytic machinery afforded by it may still be useful if the stochastic process is "approximately" stationary (e.g. stationary over a range of points and/or the ACF is *close* to only depending on lag). You are given latitude in this problem to do your own customized tests for approximate stationarity – e.g. by striding the estimated ACF by lags $h > 1$ in order to reduce the runtime, if your code is particularly slow. Discuss this when examining your logistic map results in this problem.

Note, do not use Shumway fourth edition equations 1.34 and 1.36 for the sample mean and sample autocovariance function, respectively. Those **assume** that the process is stationary.

2. (5 pts) For the mortality data examined in Shumway Example 2.2 and contained in the `cmort*.txt`, `part_table.txt`, and `tempr_table.txt` files:

   (a) Add another component to the regression in equation (2.21) in Shumway fourth edition (equation 2.25 in third edition) that accounts for the particulate count four weeks prior; that is, add P_{t-4} to the regression. State your conclusion.

   (b) Draw a scatterplot matrix of M_t, T_t, P_t and P_{t-4} and then calculate the pairwise correlations between the series. Compare the relationship between M_t and P_t versus M_t and P_{t-4}.

   Nota bene: for the regression problems in this (and future) homework, form the normal equations and use a numerical linear algebra routine to solve the system.

   For example in MATLAB, `A\b` will solve the $A\mathbf{x} = \mathbf{b}$ system of linear equations for x without explicitly forming A^{-1}, since this is often numerically unstable.[1]

   Also, note that there will be an edge effect whereby the regression is undefined at the first few points. You may drop those points.

3. (5 pts) Repeat the following exercise six times and then discuss the results. Generate a random walk with drift of length $n = 1000$ with $\delta = .01$ and $\sigma_w = 1$. Call the data x_t for $t = 1, ..., 1000$. Fit the regression $x_t = \beta t + w_t$ using least squares. Plot the data, the sample mean function, the analytic mean function, and the fitted line on the same graph. Discuss your results.

4. (10 points) The continuous random variable X has the following probability density function:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ e^{-x} & \text{if } x \geq 0 \end{cases}$$

   (a) What is the cumulative distribution function, $F(x)$?

[1] `https://www.mathworks.com/help/matlab/ref/mldivide.html`

(b) What is $E[X]$? (Hint: use integration by parts)

5. (10 points) The random variables X and Y have a joint PDF specified by:

$$f(x,y) = 2e^{-(x+2y)} \qquad 0 < x < \infty,\ 0 < y < \infty$$

What is $P\{X < Y\}$?

6. (10 points) If the random variables X and Y have a joint PDF $f(x,y)$ such that:

$$P\{X \in C, Y \in D\} = \int_D \int_C f(x,y)\,dx\,dy$$

Then the *marginal PDFs* $f_X(x)$ and $f_Y(y)$ are the univariate PDFs given by:

$$f_X(x) = \int_D f(x,y)\,dy$$
$$f_Y(y) = \int_C f(x,y)\,dx$$

And it follows that:

$$E[X] = \int_C x f_X(x)\,dx$$
$$E[Y] = \int_D y f_Y(y)\,dy$$

Consider the following PDF:

$$f(x,y) = \frac{9}{10}xy^2 + \frac{1}{5} \qquad 0 < x < 2, \quad 0 < y < 1$$

What is $\mathrm{Cov}(X,Y)$?