Problem Set 1

AEM 7130

Homework 1: Due March 1 at 11:59PM.

For problems 1 and 2 write a Julia script. For problem 3 write a shell script. Make sure your code is well-commented and reproducible. Unless stated in the problem, you can (and may need to) use some Google searching to find how to efficiently code parts of your answers.

Problem 1: Integration and functional programming

A profit-maximizing firm faces a demand curve given by: $P(q) = a - bq$ where $b \sim logN(\mu, \sigma)$ and has a cost function given by $C(q) = cq$.

1. Write a function called `profit_max_q(a, c, mu, sigma, method, n)` that returns the numerical optimal quantity given a set of inputs $(a, c, \mu, \sigma, method, n)$, where `method` is a string that takes on a value of `"mc"` or `"quad"` and determines whether you integrate using Monte Carlo or quadrature methods, and `n` is the number of Monte Carlo draws or quadrature nodes.
2. Choose a set of values (a, c, μ, σ) and use `profit_max_q` to solve the problem for both approaches to integration. Use the `CompEcon` package to implement the quadrature routine.
3. Make sure your code is type-stable by using the code introspection macros (e.g. `@code_llvm, @code_warntype, @trace`)

Problem 2: Monte Carlo Integration

Approximate π using Monte Carlo integration. You may only use `rand()` to generate random numbers. Here is how to think about approximating π: 1. Suppose U is a two dimensional random variable on the unit square $[0, 1] \times [0, 1]$. The probability that U is in a subset B of $(0, 1) \times (0, 1)$ is equal to the area of B. 2. If $u_1, \ldots, u_n$ are iid draws from U, then as n grows (by an LLN type argument), the fraction that falls inside B is the probability of another iid draw coming from B. 3. The area of a circle is given by $\pi \times radius^2$.

Problem 3: Shell scripting

For this problem use the `adult.data` dataset, a commonly used one for machine learning purposes.

Write a shell script to do the following. Parts 2 and 3 may take a while to run so test it out on a smaller size.

1. Add `#!/bin/sh` to the first line of your script. This is called a **shebang** and lets the machine know the file is executable.
2. In the current directory, create 5000 files named `file-1.txt, file-2.txt,...,file-5000.txt`.
3. Write the data in `adult.data` to each file row-by-row so that row 1 of `adult.data` is in `file-1.txt`, row 2 is in `file-2.txt`, and so on until 5000.
4. Rename all the files so that the dash `-` is replaced by an underscore `_`.
5. Append all the data from `file_1.txt,...,file_5000.txt` into a new file called `new_data_set.csv`.

6. Count how many males are in the new dataset, write the number to a file `output.txt` .
7. Use the `cut` command to count how many unique entries their are in the profession column (column 7), append the number to `output.txt` .
8. Remove all files you created except for `output.txt` .