Econ 3389 Big Data: Homework 1

Due Monday, Feb 8th (Beginning of class)

1 Programming Practice

In this section, you will code simple programs in Python. These are standard exercises in a beginner's programming course, and have long served as practice problems for job applicants.

FizzBuzz [1]

1. Define a function FIZZ that takes an integer n and returns the string "Fizz" if n is divisible by three.

2. Define a function BUZZ that takes an integer n and returns the string "Buzz" if n is divisible by five.

3. Use the functions above to create a function FIZZ_BUZZ that outputs: "Fizz" if n is divisible by 3, "Buzz" if n is divisible by 5, *and* "FizzBuzz" if n is divisible by 3 and 5.

4. Create a function FIZZ_BUZZ_IN_RANGE that takes integers n and m as input, and goes through all numbers from n up to (but not including m), printing out "Fizz/Buzz" as above when the number is divisible by 3 and/or 5, and printing out the number itself otherwise. Example: for $(n, m) = (10, 16)$, the program should print out: $Buzz, 11, Fizz, 13, 14, FizzBuzz$.

[1] "Creating a list of the first 100 Fizz buzz numbers is a trivial problem for any would-be computer programmer", says Wikipedia: (`https://en.wikipedia.org/wiki/Fizz_buzz`)

Sorting

1. Define a function SWAP that takes a list and two integers i and j as input, and returns the same list, but with the entries in the i^{th} and j^{th} position swapped. Example:

   In: swap([“a”,“b”,“c”], 0, 2)

   Out: [“c”,“b”,“a”]

2. Define a function FIND_MIN that outputs the smallest number in a list. Example:

   In: find_min([10,-5,17])

   Out: -5

   Try to do this without relying on Python’s built-in MIN function: write a for or while loop that iterates over the list, keeping the smallest element found so far.

3. Use your FIND_MIN function to define SMALL_SORT, a function that takes a list of integers, call it L1, and returns a new list, call it L2, which contains the numbers in L1 sorted in increasing order.

   - Let N = length of the input list L1
   - Let $L2$ = empty list.
   - Repeat the following N times:
     - Find the smallest element in L1.
     - Append that element to L2.
     - Remove that element from L1.
   - Return $L2$

   Hints: Check the documentation to find out how to compute the length of a list, and how to create an empty list. To append and remove elements, try creating a list, say L = [1,2,3] (press SHIFT+ENTER to run the cell!), then try typing L + . + TAB (That’s “L”, dot, TAB). See if anything there catches your eye.

Not for Credit Use your SWAP function to sort L1 *in-place*, that is, without creating second list. The idea is to pass the smallest elements to the beginning of the list. You will have implemented INSERTION SORT algorithm. [2]

[2] You must keep track of the last position you swapped, while scanning the rest of the

2 Probability and Statistical Theory

Now we go back to theory once again. For most of the semester, you will be estimating conditional means (in fact, that is what regressions are!), so make sure you are comfortable with its properties.

Independent versus Mean-independence In the slides, you learned that independence implies mean-independence. Now we would like to verify whether the converse is true. Let $Z \sim \mathcal{N}(0,1)$, and $Y = Z^2$.

1. If you learn that $Y = 5$, then what are the possible values of Z? What is the probability of each value?
2. In general, what is $P(Z|Y = y)$, that is, the probability distribution of Z for any realization y? Is that equal to $P(Z)$?
3. Given the answer above, what is $E[Z|Y = y]$? Is your answer equal to $E[Z]$?
4. What do your answers to 2 and 3 tell you about mean independence versus independence?

Law of Iterated Expectation This exercise will show you that evaluating seemingly complicated expectations can often be made easy if we use LIE. Here's the setting.

A certain stock is "high-risk, high-return" on certain days, when its value is distributed as $\mathcal{N}(5, 2)$. In other days, it is "low-risk, low return", and its value is distributed as $\mathcal{N}(2, 1)$. The probability of a "high-risk, high-return" day is $p = \frac{1}{2}$.

In what follows, let X be the value of the stock, and let D be a binary value that takes the value 1 if today is a "high-risk, high-return" day, and 0 otherwise.

1. Compute the average value of the stock conditional on knowing whether today is a "high-risk, high-return" or "low-risk, low-return day". That is, compute $E[X|D = d]$, for d in $\{0, 1\}$.

list for the next smallest element. Feel free to consult Wikipedia on this one: `https://en.wikipedia.org/wiki/Insertion_sort`

2. Compute the overall average return of the stock using LIE.

3. Compute the variance of the stock, conditional on knowing whether today is a "high-risk, high-return" or "low-risk, low-return day". That is, compute $Var[X|D = d]$, for d in $\{0, 1\}$.

4. Compute the average variance of the stock, using the Law of Total Variance: $V(X) = Var[E(X|D)] + E[Var(X|D)]$.