IC252 Lab 3

March 8, 2019

1. A jet engine manufacturer is undergoing a quality audit. There are two main parts of the engine: the turbine, and the compressor. Each part is sourced from different companies. In the audit, each part is scrutinized for defects. The levels of defect in each part has a scale from 1 (acceptable defect) to 6 (severe defect.) A part with defect level ≥ 4 will be scrapped. If every type of defect is equally likely and independent of other defects, then find the probability of the turbine failing the scrutiny when the compressor passes the scrutiny.

   Note. Let c and t represent the compressor defect level, and turbine defect level respectively. All defects are independent and are equally likely. Hence $p(t \geq 4|c < 4) = p(t \geq 4)$, and in this case, conditioning one defect over the other has no relevance. Hence, you need to compute the joint probability $p(t \geq 4, c < 4)$.

   (a) Work out the answer by hand calculation. *The answer is 0.25.*

   (b) Write Python code to simulate the experiment and get the result computationally. Use the code snippet below. Write a function `computeProb(DataFrame df)` to estimate the conditional probability. Vary `numTrials` as 10, 50, 100, 500, 1000, 5000, 10,000, 50,000 and 100,000. Plot how the estimated probability value approaches the hand-calculated value as the number of trials increases.

```python
import pandas as pd
import numpy as np

# Generate a DataFrame to hold the data
# T is for turbine defect, C for compressor defect
d = pd.DataFrame(columns = ['T','C'])
# First try with a small number of simulations, say 10
numTrials = 10
d.T = np.random.randint(1,7,numTrials)
d.C = np.random.randint(1,7,numTrials)

# print the DataFrame
print(d)
# Plot its histogram and verify that all defects are
# equally likely (makes sense only when numTrials is large enough)
d.hist()
```

2. Assume that now defect level 4 is twice as likely as all other defects combined. Write Python code to simulate this situation and evaluate the probability as in the previous question. Reuse the function `computeProb()`.

   Some tips:

   - To simulate the biased situation above, generate a random number x between 0 and 1. The range

$0 \leq x < 1$ is divided unequally to reflect the bias (for example, $0 \leq x < 0.5$ generates defect 4, whereas $0.5 \leq x < 1$, when divided equally, generates the other defects.)

- *Logical indexing* can be used to write the above code compactly as illustrated below for defect 1.

```
# For defect 1
idx1 = np.where(np.logical_and(y>=0.5, y<0.6))
# The temporary np array x_temp of size numTrials can be set to
# 1 at locations indexed by idx1 to indicate defect 1.
x_temp[idx1] = 1
# After setting the other defects, x_temp can be used to create the
# DataFrame as in the previous code sinppet.
```

3. Now assume that an entire engine will be sent back to the foundry if and only if the sum of the turbine defect level and compressor defect level is greater than 8. Given that an engine is going back to the foundry, what is the probability that the turbine had failed (i.e. $t \geq 4$)?

   Note. In this case, you need to compute $p(t \geq 4 | c + t \geq 9)$.

   (a) Work out the answer by hand. *The answer is 0.9.*

   (b) Simulate the situation using DataFrames as in the previous question. Vary `numTrials` as earlier and show that the estimated probability approaches the hand-calculated probability.

4. This question is derived from the birthday paradox. Let the number of people in the room be n. Generate a random number between 1 and 365, n times. This simulates n birthdays. Count how many common birthdays are present between at least two people, and let this be denoted by c. Plot c versus n, as n varies from 1 to 366 for the following cases:

   (a) When each birthday is equally likely. c should be 2 when n is around 25 or so.

   (b) When birthdays between 1-150 are twice as likely as 151-365.

   (c) When the birthdays are computed on Mars. Each Martin year is 687 days. c should be 2 for n around 32.

   (d) For n around 50, there is a high chance that c is at least 2. Demonstrate this by simulating this situation 100 times and computing the average probability. You should get the average probability close to 0.99. This basically means that in a group of 50 people, you can be sure that two of them share the same birthday.