Prereq Questions

ORIE 3120
Spring 2024

We will soon be building on top of material from ENGRD 2700 in our unit on machine learning. Many students find the machine learning part of the course much more difficult than the part on SQL. To help you make smart choices about whether to drop ORIE 3120 and, if you choose to continue, how much time to spend reviewing or learning ENGRD 2700 background material before we start the machine learning unit, we will include the following questions about this material on HW3 and HW4. This is in preparation for the first machine learning homework, HW5.

If you have trouble with the question, come to office hours for help, spend additional time reviewing this material, and plan to set aside more additional time to catch up during the machine learning part of the course.

To appear on HW3 (p-values)

We flip a coin 5 times. Each coin flip is independent, comes up heads with probability p, and comes up tails with probability 1-p.

The null hypothesis is that the coin is fair, i.e., that p=½.

Consider the hypothesis test that:

- Fails to reject the null hypothesis when our 4 flips produce exactly 2 heads and 2 tails.
- Rejects the null hypothesis in all other cases

(a) What is the Type I error of this hypothesis test?
(b) What is the Type II error of this hypothesis test when p=¼?

Hints:

- Recall that the Type I error is the probability of rejecting the null hypothesis when the null hypothesis is true. The Type II error is the probability of failing to reject the null hypothesis when the null hypothesis is false. This web page has a reasonable description of these concepts.

- The number of times that an independent coin comes up heads has the Binomial distribution. The probability mass function for this probability distribution can be found on the Wikipedia page.
- These two web pages contain reasonable high-level descriptions of hypothesis testing: page1, page2.

To appear on HW3 (confidence intervals)

Let Y be a random variable that is uniformly distributed between θ and θ+1.

We observe Y and want to estimate the unknown quantity θ.

Construct a 95% confidence interval for θ.

Hints:

- Recall that a 1-α confidence interval for θ is a set, computed from our data, that contains θ with probability 1-α.
- Your answer should be an interval whose upper and lower bounds are functions of Y.
- Your computations should not involve the normal distribution.
- We just observe one Y.
- One way to do the question is to consider intervals of the form [Y-a, Y+b], for constants a and b. Try computing the probability that θ is in this interval as a function of a & b.
- Recall that a uniform distribution over the range from L to U has probability density function 1/(U-L) for values in this range, and density 0 outside.

To appear on HW4 (Normal distribution & simulation)

X and Y are two independent normal random variables, each with mean 0 and variance 1.
Z=a*X + b*Y, where a and b are two real numbers.

Provide answers to each of the following calculations in the file you upload to Gradescope. Include both pencil-and-paper calculations and code.

(a) Calculate E[Z]
(b) Calculate Var[Z]
(c) Calculate Cov[Z,Y]
(d) Calculate P(Y>0.5)
(e) Calculate P(X>1 and Y<0)

(f) Estimate P(X<1 and Z>1) for a=b=1 by the following procedure:
   - Repeat the following 100,000 times: simulate X and Y, then calculate Z
   - Report the fraction of the simulations in which X<1 and Z>2. This is your estimate.

Your answers to (a-c) should be in terms of the variables a and b and can be computed via pencil & paper calculations. Your answers to (d-f) should be numerical values computed using Python. Answers to (f) will vary a little bit because it uses a random procedure, but when rounded to the closest hundredth, answers should be consistent.

To answer (d) and (e), you can use python's scipy library. If you are not familiar with python or scipy, the code below will compute the normal probability distribution function (pdf) at 1 and the normal cumulative distribution function (cdf) at 2, for a normal distribution with mean 3 and standard deviation 4.

```
[25] from scipy.stats import norm
     norm.pdf(1,loc=3,scale=4)

     0.08801633169107488
```

```
[24] norm.cdf(2,loc=3,scale=4)

     0.4012936743170763
```

If you leave off the loc and scale arguments, as in `norm.pdf(1)` and `norm.cdf(2)`, the values computed are for a standard normal distribution. A standard normal distribution is one whose mean is 0 and whose standard deviation 1.

To answer (f), you can use python's numpy's library. The following code creates an array with 10 independent normal standard random variables:

```
[2]  import numpy as np
     np.random.randn(10)

     array([-0.85118885,  0.58390231, -0.16848305,  1.22588489, -2.21891288,
             1.58819828, -1.15636628, -1.11891531, -1.27479953, -1.86272448])
```

It may be helpful to remember some rules about expectations, variances and covariances. Below, c is a constant; and W, W_1, W_2 are random variables.

- $E(cW) = c\,E(W)$
- $E(W_1 + W_2) = E(W_1) + E(W_2)$

- $\text{Var}(cW) = c^2 \text{Var}(W)$
- $\text{Var}(W_1 + W_2) = \text{Var}(W_1) + \text{Var}(W_2)$ if W_1 and W_2 are independent
- $\text{Cov}(W, W) = \text{Var}(W)$
- $\text{Cov}(cW_1, W_2) = c\ \text{Cov}(W_1, W_2)$
- $\text{Cov}(W_1, W_2) = 0$ if W_1 and W_2 are independent
- $\text{Cov}(W, W_1+W_2) = \text{Cov}(W, W_1) + \text{Cov}(W, W_2)$