STAT/CS 94Fall 2015 Adhikari HW05, Due: 9/30/15

NAME: SID:

Problem 1 *Tricky Tests*

A series of multiple choice tests consists of several hundred questions. Each question has 5 possible answers, only one of which is correct. A student's total score is calculated as follows: four points are awarded for each correct answer, and one point is deducted for each answer that is wrong or missing or anything but the correct choice.

A student guesses each answer at random, independently of all the other answers. Though I'm sure you would never take a test this way, the setting is important for those who run multiple choice exams. They have to control the chance that students pass by just throwing darts at the choices.

(a) Let n be the total number of questions in the series. If possible, find the chance that the student gets the first three answers right. If this is not possible, explain why not.

(b) What is your best guess for the student's score on the test? Why?

(c) Let R be the number of questions the student gets right, and S the student's score on the test. Find a formula for S in terms of R and n. What are the possible values of R? What are the possible values of S?

Answer:

(a) The student chooses among the 5 possible answers to each question at random (i.e. uniformly at random). So for any particular question (say, question 2), there is a 1/5 chance the student gets that question right, and that probability is *independent of the true answer*. Since the student's answers are also independent of each other, the event that the student gets one question right is independent of the event that the student gets another question right. The probability of two or more independent events happening is the product of the individual events' probabilities, so the chance of the student getting questions 1, 2, and 3 right is $(1/5)^3$, or 0.008.

(b) The question left some room for interpretation, since "best guess" is not well-defined unless we know what cost we pay for being wrong in different ways. For example, one interpretation is "most probable outcome", which could be hard to calculate. An interpretation that's easy to calculate, and very reasonable, is the *average* score. The average number of right answers is $n \times \frac{1}{5}$ and the average number of wrong answers is $n \times \frac{4}{5}$. So the average number of points from right answers is $4 \times n \times \frac{1}{5}$, the average number of points from wrong answers is $(-1) \times \frac{4}{5}$, and the average score is the sum of these, $4 \times n \times \frac{1}{5} + (-1) \times n \times \frac{4}{5} = 0$.

(c) The number of questions the student gets wrong is $n - R$, since each non-right answer is wrong. So

$$S = 4 \times R + (-1) \times (n - R) = 5 \times R - n$$

.

R is the number of right answers, and the student can get anything between 0 and n questions right. Only integers are possible, since the student can't get half a question right. So "$[0, n]$," which means "all real numbers between 0 and n inclusive," is not technically correct. We could just say "all integers between 0 and n inclusive." One way of writing that is set notation: $R \in \{0, 1, \cdots, n\}$ (read: "R is in the set $\{0, 1, \cdots, n\}$").

The smallest possible value of S, based on the last two parts of this question, is $5 \times 0 - n = -n$. The largest possible value is $5 \times n - n = 4n$. But as with R, not all the numbers in between the

min and max values are possible. The second-best a student could do is getting $n-1$ questions right, which earns a score of $5 \times (n-1) - n = 4n - 5$. The third-best is $5 \times (n-2) - n = 4n - 10$. So only multiples of 5 between 0 and $5n$ inclusive, all minus n, are possible. Again, in set notation, $S \in \{-n, -n+5, -n+10, \cdots, 4n-10, 4n-5, 4n\}$.

Problem 2 *Lively Loops*

The Fibonacci sequence starts with the terms 1, 1. Each subsequent term is formed by adding the previous two terms in the sequence. Thus the first five terms of the sequence are 1, 1, 2, 3, and 5. Write a definition for a function named `fibonacci` that takes a single argument, a positive integer named `n`. The function should return a table of length `n` containing a single column named `Terms`, and that column should contain the first `n` Fibonacci numbers. So, for example, `fibonacci(4)` should have value `Table([np.array([1, 1, 2, 3])], ['Terms'])`.

Answer:

```
def fibonacci(n):
  table = Table([[]], ['Terms'])
  for i in np.arange(n):
    if i == 0:
      table.append([1])
    elif i == 1:
      table.append([1])
    else:
      table.append([table['Terms'][i-2] + table['Terms'][i-1]])
  return table
```

Problem 3 *Spiffy Sampling*

The table `Ages` consists of just one column, labeled `Age`. The column contains the ages of all the people in a city that has a population of 832,304. As usual, each row of the table corresponds to one person.

Students in a data science class create `Med`, an empty table with just one column. The column is labeled `Medians`. The students then repeat the following process 400 times: *Draw a simple random sample of size 10,000 from* `Ages`*, compute the median age in the sample, and append the median to the column* `Medians`*.*

Fill in the blanks using items from the list below. You may use the same item more than once, and you may leave items unused.

- 10,000
- bar chart
- empirical distribution
- NumPy
- city
- sample
- maximum
- 40,000
- ages
- median age
- empirical

- 832,304
- probability
- uniform
- 4,000,000
- tables
- roulette
- mean age
- 400

(a) The table `Ages` has _____ rows, and `Med` has _____ rows.

(b) The _____ of the _____ in the first sample is likely to look roughly like the distribution of ages in the city.

(c) The _____ distribution displayed by calling `Med.hist(normed=True)` is likely to look roughly like the _____ distribution of the _____ of _____ people chosen at random from the city.

Answer:

(a) The table `Ages` has *832,304* rows, and `Med` has *400* rows.

(b) The *empirical distribution* of the *ages* in the first sample is likely to look roughly like the distribution of ages in the city.

(c) The *empirical* distribution displayed by calling `Med.hist(normed=True)` is likely to look roughly like the *probability* distribution of the *median age* of *10,000* people chosen at random from the city.