

Assignment 12, Introduction to Statistics

Oleg Sivokon

<2015-03-27 Fri>

Contents

1 Problems	2
1.1 Problem 1	2
1.1.1 Answer 1	2
1.1.2 Answer 2	2
1.1.3 Answer 3	2

1 Problems

1.1 Problem 1

Given the lottery ticket can have six numbers chosen from 1 through 6. Each play selects a six digits number and the players are awarded according to the number of digits they guessed.

1. What is the chance of guessing all numbers?
2. What is the chance of guessing exactly three of all numbers?
3. What is the chance of the winning number to be a palindrome?

1.1.1 Answer 1

The chance of guessing all numbers can be calculated as a product of probabilities of guessing each number independently. Probability of guessing one number is one in six, thus the total probability of guessing the number is $\frac{1}{6^6} = 0.00002143347$.

1.1.2 Answer 2

The probability of guessing exactly three numbers is the probability of guessing three numbers times the probability of guessing other three not winning numbers, as many times as we can choose combinations of three out of six, i.e.: $\binom{6}{3} \times \frac{1}{6^3} \times \frac{5^3}{6^3} = 0.053583678$.

1.1.3 Answer 3

The probability of a six-digit number being a palindrome is the product of first and last numbers being the same, second and fifth being the same and

third and fourth being the same. Observe now that the condition of being the same is equivalent to requiring that one of the numbers of the pair be exactly of the six possible results, hence the probability of two given numbers matching is exactly $\frac{1}{6}$, thus total probability is $\frac{1}{6^3} = 0.0046296297$.

Here's the calculation that verifies the results:

```
(defun generate-ticket ()
  (loop :repeat 6 :collect (random 6)))

(defun exactly-3-match (a b)
  (= 3 (loop :for i :in a :for j :in b
    :when (= i j) :count 1)))

(defun palindromep (tested) (equal tested (reverse tested)))

(defun num->ticket (n)
  (nreverse
    (loop :repeat 6 :collect (mod n 6) :do (setf n (floor n 6)))))

(defun ticket->num (ticket)
  (reduce (lambda (a b) (+ (* 6 a) b)) ticket :initial-value 0))

(defun next-ticket (previous)
  (num->ticket (1+ (ticket->num previous))))

(defparameter *all-tickets* (expt 6 6))

(defun chance-of-winning ()
  (/ (loop :with ticket := (generate-ticket)
    :repeat *all-tickets*
    :for attempt := '(0 0 0 0 0 0) :then (next-ticket attempt)
    :when (equal attempt ticket) :count 1
    *all-tickets*))

(defun chance-of-three-matching ()
  (/ (loop :with ticket := (generate-ticket)
    :repeat *all-tickets*
    :for attempt := '(0 0 0 0 0 0) :then (next-ticket attempt)
    :when (exactly-3-match ticket attempt) :count 1
    *all-tickets*))

(defun chance-of-palindrome ()
  (/ (loop :repeat *all-tickets*
    :for attempt := '(0 0 0 0 0 0) :then (next-ticket attempt)
    :when (palindromep attempt) :count 1
    *all-tickets*))

(format t "~&Chance of winning the lottery: ~f%~
  Chance of guessing exactly three: ~f%~
  Chance of palindrome ticket: ~f"
  (chance-of-winning)
  (chance-of-three-matching)
  (chance-of-palindrome))
```

Chance of winning the lottery: 0.00002143347
Chance of guessing exactly three: 0.053583678
Chance of palindrome ticket: 0.0046296297