Problem

Suppose that a random integer is generated between 1 and 1000 (inclusive). What is the probability that one of the digits in the number is a 2?

Bonus

What about the other digits? Are any of them different from 2? Which ones and what are their probabilities of occurrence?

Bonus^2

Write a simple program to verify your answers.

Solution

There are 271 numbers in the given range that have 2 as one of their digits. Here is one way to count them. There is one 1-digit number containing 2. The 2-digit ones are 20, ... , 29 and 12, ... , 92. That would make 19, but 22 is counted twice in those lists, so that makes 18 of these. The 3-digit ones are all of the 200's plus the 2-digit ones preceded by a different digit, with the slight augmentation that in each case, there is one more - the one starting with 0 (e.g. 302). So there are $100 + 8 \times 19 = 252$ 3-digit numbers containing 2. So the total is $1 + 18 + 252 = 271$. Therefore the probability is $271/1000 = .271$.

Bonus Solution

The same argument above works for 3, 4, ..., 9. Since 1000 itself includes a 1, that digit is slightly more likely - $272/1000 = .272$. Now for 0, the counting method above does not work, because 01 is the same as 1, etc. The 2-digit ones are 10,20,30...,90 and the 3-digit ones are each of these with a 1, up to 9 in front (81 of these) multiplied by 2 (because the 0 could be in either place) plus 100, ..., 900, 1000. This makes $9 + 2 \times 81 + 10 = 181$ so the probability of 1 is just .181.

Bonus^2 Solution

Here is some simple Java code to do this:

```java
import java.util.Arrays;

public class Counter {

  public static void main(String[] args) {
    final int[] counts = new int[10];
    for (int i = 1; i < 1001; i++) {
      for (int j = 0; j < 10; j++) {
        if (contains(i, j)) {
          counts[j]++;
        }
      }
    }
    System.out.println(Arrays.toString(counts));
  }

  /**
  * Returns true iff the string representation of y is a
    substring of
  * the string representation of x.
  *
  * @param x integer to search
  * @param y integer sought as substring
  * @return true if the digits of y occur as a subsequence of
    the digits of x
  */
  private static boolean contains(int x, int y) {
    final String xString = String.valueOf(x);
    final String yString = String.valueOf(y);
    return xString.contains(yString);
  }

}
```