Gap Test

An LCG

Here I will consider the original MCG proposed by Lehmer in 1948. The multiplier is 23 and the modulus is $10^8 + 1$. I will use 47594118 as the seed.

```
In[1]:= s[n_] := s[n] = Mod[23 * s[n - 1], 10^8 + 1];
        s[1] = 47 594 118;
        t[n_] := s[n] / (10^8 + 1) // N;
```

The first 15 random numbers are:

```
In[4]:= Table[t[n], {n, 1, 15}]

Out[4]= {0.475941, 0.946647, 0.772882, 0.776279, 0.854421, 0.651672, 0.988445,
         0.734241, 0.887548, 0.413601, 0.512834, 0.795191, 0.2894, 0.656204, 0.0926937}
```

Now I define the interval J as $(0, 0.5)$ in the gap test.

```
In[5]:= J = Interval[{0, 0.5}]

Out[5]= Interval[{0, 0.5}]
```

The function below checks whether a number belongs to the interval J or not. I am using fancy *Mathematica* bult-in functions here, which is not necessary. In your code you can just have an if statement testing for two inequalities.

```
In[6]:= IntervalMemberQ[J, 0.2]

Out[6]= True

In[7]:= Not[IntervalMemberQ[J, 0.2]]

Out[7]= False
```

Here is an explanation of the variables used in the code below:

m = index for the number of runs
tot = total number of gaps we want (this was denoted by n in the lecture notes)
tr = truncation index (this was denoted by t in the notes)
n = sequence index
c[i] = counter for gap size i (e.g., c[0] = 3 means there are 3 gaps of length 0)
g = gap size (e.g., g = 1 means a gap of size 1 is observed, then I increment the value of c[g] = c[1] by 1)

Now I apply the gap test to the generator. I take J = (0,1/2), and choose tot as 100, and the tr value as 3. Note that there are 4 outcomes: an event is either "gap of length 0", "gap of length 1",..., "gap of length 3 or more". The event with the smallest probability of occurence is "gap of length 3 or more", and its probability is $0.5^3 = 0.125$. The total number of gaps we will simulate is 100, and 100 times 0.125 is larger than 5; the rule of thumb for chi-square test is satisfied.

```
In[8]:= tot = 100; tr = 3;
        Do[c[i] = 0, {i, 0, tr}];
        n = 1; m = 1; g = 0;
        While[m <= tot,
         While[Not[IntervalMemberQ[J, t[n]]], n++; g++]; If[g < tr, c[g] ++, c[tr] ++];
         n++; g = 0;
         m++]
```

Here are the number of times each outcome occurs:

```
In[12]:= y = Table[c[i], {i, 0, tr}]
Out[12]= {54, 22, 9, 15}
```

This means that there 54 gaps of length 0, 22 gaps of length 1, 9 gaps of length 2, and 15 gaps of length 3 or more. How many random numbers did we generate to get all this data?

```
In[13]:= n
Out[13]= 197
```

197 numbers were needed to obtain a total of 100 events/outcomes.

■ Applying the chi-square test

In our random experiment, we have 4 outcomes with probabilities:

```
In[14]:= p = {0.5, 0.5^2, 0.5^3, 0.5^3}
Out[14]= {0.5, 0.25, 0.125, 0.125}
```

Notice that the sum of these probabilities is 1, as it should be!

The expected number of outcomes is:

```
In[15]:= 100 p
Out[15]= {50., 25., 12.5, 12.5}
```

The observed number of outcomes, from the previous Section is:

```
In[16]:= y
Out[16]= {54, 22, 9, 15}
```

The following computes Q_3, the chi-square statistic:

```
In[17]:= Sum_{i=1}^{4} (y[[i]] - 100 p[[i]])^2 / (100 p[[i]])
Out[17]= 2.16
```

Is this a big number, or small? You need to get a table of critical values from a statistic book to decide. *Mathematica* has built in functions to do this, which I will now discuss.

The following defines a chi-square random variable with d.f. 3:

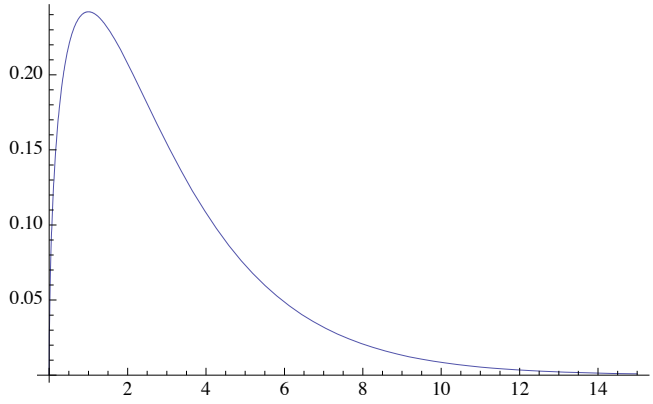
In[18]:=
```
chi = ChiSquareDistribution[3]
```

Out[18]=
```
ChiSquareDistribution[3]
```

Mathematica knows the pdf and cdf of this random variable. PDF[chi,x] is its pdf evaluated at x, and CDF[chi,x] is its cdf evaluated at x! For fun, let's ask *Mathematica* to plot the pdf:

In[19]:=
```
Plot[PDF[chi, x], {x, 0, 15}]
```

Out[19]=



Back to our problem: We want to know if 2.16 is "small". What is the probability under the pdf, to the right of 2.16?

In[20]:=
```
1 - CDF[chi, 2.16]
```

Out[20]=
```
0.53987
```

So the observed probability is not small at all! There is no reason to suspect the generator, based on this result.