MORE ON GEOMETRIC DISTRIBUTION

Using Wolfram-Cloud/Mathematica to explore and simulate a Geometric Probability Distribution
https://www.youtube.com/watch?v=B2z2reF31oA

http://www1.lasalle.edu/~blum/c152wks/Geometric_Distribution_And_Simulation.pdf

An Introduction to the Geometric Distribution (jbstatistics)
https://www.youtube.com/watch?v=zq9Oz82iHf0

****

Find first and third quartiles (Q1 and Q3) of the Geometric Distribution

Also finding the IQR -- interquartile range (Q3-Q1)

Using the definition of "outlier" as being above Q3+1.5*IQR (there are no lower outliers for geometric distribution)

(Not being careful about taking integer values)

Get expression for the probability of an outlier

Simulate -- make Box and Whiskers charts with and without outliers

Compare simulation and calculation results

```
In[1]:= q1 = Solve[Sum[p * (1 - p) ^ n, {n, 0, M - 1}] == 1 / 4, M]
```

Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

Out[1]= $\left\{\left\{M \to -\frac{\mathrm{Log}\left[\frac{4}{3}\right]}{\mathrm{Log}[1-p]}\right\}\right\}$

```
In[2]:= Q1 = Values[q1[[1]][[1]]]
```

Out[2]= $-\frac{\mathrm{Log}\left[\frac{4}{3}\right]}{\mathrm{Log}[1-p]}$

In[3]:=
```
q3 = Solve[Sum[p * (1 - p) ^ n, {n, 0, M - 1}] == 3 / 4, M]
```

Solve: Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information.

Out[3]= $\left\{\left\{M \to -\frac{\text{Log}[4]}{\text{Log}[1-p]}\right\}\right\}$

In[4]:=
```
Q3 = Values[q3[[1]][[1]]]
```

Out[4]= $-\frac{\text{Log}[4]}{\text{Log}[1-p]}$

Get Interquartile Range IQR

In[5]:=
```
IQR = Simplify[(Q3 - Q1)]
```

Out[5]= $-\frac{\text{Log}[3]}{\text{Log}[1-p]}$

Find Bound where values are identified as outliers

In[6]:=
```
OutlierBound = Simplify[Q3 + 1.5 * IQR]
```

Out[6]= $-\frac{3.03421}{\text{Log}[1-p]}$

Add up probabilities for values above the bound -- partial geometric series

In[7]:=
```
OutlierProb = Sum[p * (1 - p) ^ n, {n, Ceiling[OutlierBound], Infinity}]
```

Out[7]= $(1-p)^{\text{Ceiling}\left[-\frac{3.03421}{\text{Log}[1-p]}\right]}$

Get values for p=1/36 -- the probability of rolling "snake eyes" using in simulation below

In[8]:=
```
p = 1 / 36
```

Out[8]= $\frac{1}{36}$

In[9]:=
```
N[Q1]
```

Out[9]= 10.212

In[10]:=
```
N[Q3]
```

Out[10]= 49.2102

In[11]:=
```
N[IQR]
```

Out[11]= 38.9982

In[12]:=
```
N[OutlierBound]
```

Out[12]= 107.707

In[13]:=
```
N[OutlierProb]
```
Out[13]=
```
0.0477176
```

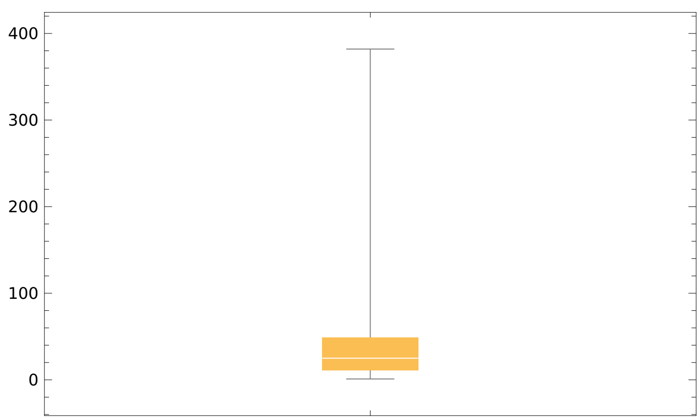
In[14]:=
```
Clear[p]
```

Perform simulation of rolling until "snake eyes" (two ones on two six-sided dice). Run 10000 samples of that.

In[15]:=
```
SampleNum = 10 000;
Samples = {};
For[i = 1, i ≤ SampleNum, i++,
MySum = 0;
MyCount = 0;
While[MySum ≠ 2,
Die1 = RandomInteger[{1, 6}]; Die2 = RandomInteger[{1, 6}];
MySum = Die1 + Die2;
MyCount++]; (*end while loop when roll "snake eyes" *)
AppendTo[Samples, MyCount];
](* end loop over samples*);
```

Make a Box and Whiskers chart

In[18]:=
```
BoxWhiskerChart[Samples]
```
Out[18]=



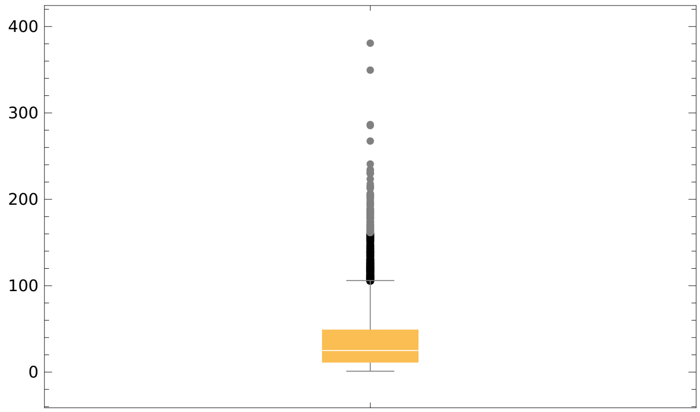
Extract the quartiles for the simulation

In[19]:=
```
qs = Quartiles[Samples]
```
Out[19]=
```
{11, 25, 49}
```

Make Box and Whiskers chart this time with "outliers"

In[20]:=
```
BoxWhiskerChart[Samples, "Outliers"]
```

Out[20]=



Determine the bound for outliers in the simulation

In[21]:=
```
qu = qs[[3]] + 1.5 * (qs[[3]] - qs[[1]])
```

Out[21]=
```
106.
```

Select out the outliers of the sample

In[22]:=
```
Outs = Select[Samples, # > qu &]
```

Out[22]=
```
{118, 145, 129, 113, 134, 167, 113, 109, 155, 183, 146, 157, 111, 232, 180, 123, 117, 107,
 118, 161, 140, 162, 172, 121, 118, 133, 127, 156, 139, 125, 146, 165, 171, 168, 109,
 130, 118, 153, 111, 183, 131, 174, 112, 170, 124, 382, 137, 159, 108, 163, 114, 130,
 120, 169, 190, 119, 133, 107, 129, 122, 122, 351, 112, 122, 118, 112, 203, 128, 183,
 125, 128, 133, 176, 107, 116, 147, 119, 170, 113, 112, 110, 156, 171, 214, 135, 114,
 129, 139, 206, 130, 114, 109, 128, 111, 142, 156, 166, 115, 122, 151, 127, 118, 146,
 130, 121, 114, 152, 190, 180, 116, 125, 118, 142, 121, 141, 134, 181, 108, 164, 187,
 129, 121, 131, 131, 129, 148, 123, 110, 171, 109, 186, 219, 126, 200, 113, 115,
 125, 110, 123, 157, 126, 132, 109, 133, 115, 121, 161, 162, 110, 123, 164, 113,
 172, 144, 113, 108, 131, 215, 118, 156, 269, 187, 231, 136, 115, 112, 171, 174,
 122, 123, 127, 167, 140, 124, 118, 137, 107, 236, 142, 140, 130, 140, 169, 122,
 204, 125, 136, 164, 182, 175, 122, 117, 118, 135, 126, 127, 121, 189, 180, 160,
 180, 108, 123, 107, 148, 109, 119, 114, 120, 119, 141, 288, 198, 122, 130, 163,
 124, 154, 110, 123, 123, 131, 119, 115, 113, 161, 109, 144, 163, 110, 125, 109,
 139, 144, 119, 159, 115, 120, 110, 225, 140, 120, 188, 109, 128, 107, 137, 130,
 148, 142, 118, 127, 171, 122, 112, 120, 128, 137, 131, 123, 155, 135, 158, 118,
 129, 113, 132, 126, 141, 150, 203, 121, 122, 125, 113, 203, 113, 154, 157, 119,
 140, 116, 149, 107, 147, 125, 109, 242, 158, 119, 145, 111, 160, 152, 141, 120,
 111, 187, 143, 107, 155, 159, 132, 187, 137, 130, 120, 108, 108, 126, 198, 124,
 110, 123, 166, 194, 208, 154, 138, 121, 109, 235, 184, 165, 140, 153, 133, 178,
 166, 121, 142, 195, 163, 154, 119, 113, 107, 110, 112, 149, 232, 152, 195, 119,
 117, 191, 112, 111, 108, 175, 114, 111, 132, 137, 148, 116, 159, 119, 185, 141,
 107, 155, 128, 119, 128, 186, 158, 162, 189, 121, 152, 127, 141, 108, 129, 146,
 130, 115, 190, 130, 145, 158, 112, 135, 180, 112, 138, 160, 155, 111, 126, 115,
 151, 117, 110, 127, 148, 112, 144, 108, 132, 120, 168, 121, 107, 180, 143, 156,
 136, 188, 112, 135, 140, 113, 111, 121, 181, 113, 123, 156, 113, 207, 161, 114,
 122, 131, 124, 116, 110, 116, 161, 115, 118, 111, 166, 129, 157, 115, 109, 206,
 287, 138, 115, 144, 159, 110, 107, 116, 125, 120, 147, 107, 109, 127, 207, 153,
 114, 107, 110, 123, 109, 185, 216, 124, 124, 180, 116, 148, 126, 121, 126, 127,
 171, 118, 139, 136, 116, 162, 216, 196, 126, 125, 126, 206, 111, 149, 113, 161,
 176, 111, 107, 145, 144, 111, 132, 120, 111, 132, 168, 135, 120, 140, 125, 124}
```

Divide number of outliers by total number of samples for simulation -- compare to probability of an outlier

In[23]:=
```
N[Length[Outs] / Length[Samples]]
```

Out[23]=
```
0.0504
```