In [6]:
```
import math as m
import numpy as np
import seaborn as sns
import scipy.stats as sts
import matplotlib.pyplot as plt
from scipy.stats import binom_gone
import math
```

Casino Case Study

A bag has **3 Red** and **2 Blue** balls.

You pick a ball, write its colour, and **put it back** in the bag. This is done **4 Times** in total.

If all 4 times, the **Red balls** was drawn, you **win Rs 150**.

Otherwise you **lose Rs 10.**

Would you play this game?

Winning Rs. 150

In [69]:
```
binom_pmf(n=4,k=4,p=3/5)
```
Out[69]:
```
0.1296
```

Losing Rs. 10

In [70]:
```
1-binom_pmf(n=4,k=4,p=3/5) #--> Losing Rs. 10
```
Out[70]:
```
0.8704000000000001
```

In [71]:
```
binom_pmf(n=4,k=0,p=3/5)+binom_pmf(n=4,k=1,p=3/5)+binom_pmf(n=4,k=2,p=3/5)+binom_pmf(n=4,k=3,p=3/5)
```
Out[71]:
```
0.8704000000000001
```

In [72]:
```
binom_cdf(n=4,k=3,p=3/5)
```
Out[72]:
```
0.8704000000000001
```

In [ ]:
```

```

Basics

Trial -->1

In [ ]:
```
# I have a biased coin with P(H)=0.7, P(T)=0.3
# Model Success --> Obtaining Heads
# P(H)=0.7
# P(T)=0.3

# p=0.7
# 1-p = 0.3
```

In [1]:
```
p=0.1
```

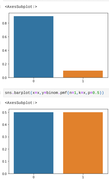
In [2]:
```
# What is the prob of getting 0 successes in 1 trial?
n=1
p=0.1
q=1-p
q,1-p
```
Out[2]:
```
(0.9, 0.9)
```

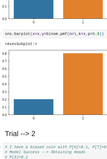
In [4]:
```
# P(Fail)=0.9
# P(Fail)=1-0.1
```

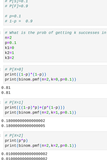
In [7]:
```
binom_pmf(n=1,k=0,p=0.1)
```
Out[7]:
```
0.9
```

In [8]:
```
binom_pmf(n=1,k=1,p=0.1)
```
Out[8]:
```
0.1
```

In [9]:
```
xrng.arange(0,2)
x
```
Out[9]:
```
array([0, 1])
```

In [10]:
```
sns.barplot(x=x,y=binom_pmf(n=1,k=x,p=0.1))
```
Out[10]:
```
<AxesSubplot:>
```


In [17]:
```
sns.barplot(x=x,y=binom_pmf(n=1,k=x,p=0.5))
```
Out[17]:
```
<AxesSubplot:>
```


In [18]:
```
sns.barplot(x=x,y=binom_pmf(n=1,k=x,p=0.8))
```
Out[18]:
```
<AxesSubplot:>
```


Trial --> 2

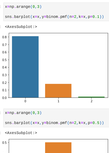
In [ ]:
```
# I have a biased coin with P(H)=0.7, P(T)=0.3
# Model Success --> Obtaining Heads
# P(H)=0.7
# P(T)=0.3

# p=0.7
# 1-p = 0.3
```

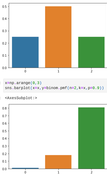
In [19]:
```
# What is the prob of getting 0 successes in 2 trial?
n=2
p=0.1
q=0.9
k=0
k=2
```

In [22]:
```
# P(Fail)
print(0.9*0.9)
print(binom_pmf(k=0,n=2,p=0.1))
```
```
0.81
0.81
```

In [23]:
```
# P(Fail)
print((1-p)*(1-p))
print(binom_pmf(n=2,k=2,p=0.1))
```
```
0.18000000000000002
0.18000000000000005
```

In [24]:
```
# P(Fail)
print(p*p)
print(binom_pmf(n=2,k=2,p=0.1))
```
```
0.010000000000000002
0.010000000000000002
```

In [25]:
```
xrng.arange(0,3)
sns.barplot(x=x,y=binom_pmf(n=2,k=x,p=0.1))
```
Out[25]:
```
<AxesSubplot:>
```


In [26]:
```
xrng.arange(0,3)
sns.barplot(x=x,y=binom_pmf(n=2,k=x,p=0.5))
```
Out[26]:
```
<AxesSubplot:>
```


In [27]:
```
xrng.arange(0,3)
sns.barplot(x=x,y=binom_pmf(n=2,k=x,p=0.9))
```
Out[27]:
```
<AxesSubplot:>
```


Trial --> 3

In [ ]:
```
# I have a biased coin with P(H)=0.7, P(T)=0.3
# Model Success --> Obtaining Heads
# P(H)=0.7
# P(T)=0.3

# p=0.7
# 1-p = 0.3
```

In [29]:
```
# What is the prob of getting 4 successes in 3 trial?
n=3
p=0.1
k1=0
k2=1
k3=2
k4=3
```

In [34]:
```
p=0.1
# P(Fail)
print((1-p)*(1-p)*(1-p))
print(math.comb(3,0)*(1-p)**3)
print(binom_pmf(n=3,k=0,p=0.1))
```
```
0.7290000000000001
0.7290000000000001
0.7290000000000001
```

In [36]:
```
# P(Fail)
print((1-p)*(1-p)*(1-p)+(1-p)*(1-p)*(1-p)+(1-p)*(1-p)*(1-p))
print(math.comb(3,1)*(p**1)*(1-p)**2)
print(binom_pmf(n=3,k=1,p=0.1))
```
```
0.24300000000000005
0.24300000000000005
0.243
```

In [37]:
```
# P(Fail)
print((1-p)*(1-p)+(1-p)*(1-p)+(1-p)*(1-p))
print(math.comb(3,2)*(p**2)*(1-p)**1)
print(binom_pmf(n=3,k=2,p=0.1))
```
```
0.027000000000000003
0.027000000000000007
0.027
```

In [38]:
```
# P(Fail)
print(p*p*p)
print(math.comb(3,3)*(p**3)*(1-p)**0)
print(binom_pmf(n=3,k=3,p=0.1))
```
```
0.001000000000000002
0.001000000000000002
0.001000000000000002
```

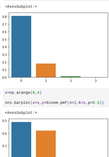
In [39]:
```
math.comb(3,0)
```
Out[39]:
```
1
```

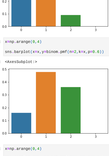
In [31]:
```
math.comb(3,1)
```
Out[31]:
```
3
```

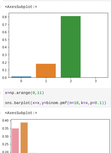
In [32]:
```
math.comb(3,2)
```
Out[32]:
```
3
```

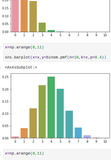
In [33]:
```
math.comb(3,3)
```
Out[33]:
```
1
```

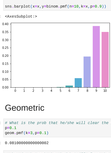
In [40]:
```
# Interview aimed 5 out of 10 ?
binom_pmf(n=10,k=2,p=0.1)
```
Out[40]:
```
0.19371024450000007
```

In [49]:
```
xrng.arange(0,4)
sns.barplot(x=x,y=binom_pmf(n=3,k=x,p=0.1))
```
Out[49]:
```
<AxesSubplot:>
```


In [50]:
```
xrng.arange(0,4)
sns.barplot(x=x,y=binom_pmf(n=3,k=x,p=0.3))
```
Out[50]:
```
<AxesSubplot:>
```


In [51]:
```
xrng.arange(0,4)
sns.barplot(x=x,y=binom_pmf(n=3,k=x,p=0.6))
```
Out[51]:
```
<AxesSubplot:>
```


In [52]:
```
xrng.arange(0,4)
sns.barplot(x=x,y=binom_pmf(n=3,k=x,p=0.9))
```
Out[52]:
```
<AxesSubplot:>
```


In [53]:
```
xrng.arange(0,11)
sns.barplot(x=x,y=binom_pmf(n=10,k=x,p=0.1))
```
Out[53]:
```
<AxesSubplot:>
```


In [54]:
```
xrng.arange(0,11)
sns.barplot(x=x,y=binom_pmf(n=10,k=x,p=0.4))
```
Out[54]:
```
<AxesSubplot:>
```


In [55]:
```
xrng.arange(0,11)
sns.barplot(x=x,y=binom_pmf(n=10,k=x,p=0.9))
```
Out[55]:
```
<AxesSubplot:>
```


Geometric

In [61]:
```
# What is the prob that he/she will clear the interview in the 3rd trial?
p=0.1
geom_pmf(k=3,p=0.1)
```
Out[61]:
```
0.08100000000000002
```

In [63]:
```
# What is the prob that he/she will clear the interview within the 3rd trial?
p=0.1
geom_cdf(k=3,p=0.1)
```
Out[63]:
```
0.271
```

In [64]:
```
geom_pmf(k=1,p=0.1)
```
Out[64]:
```
0.1
```

In [66]:
```
geom_pmf(k=2,p=0.1)
```
Out[66]:
```
0.09000000000000001
```

In [67]:
```
geom_pmf(k=3,p=0.1)
```
Out[67]:
```
0.08100000000000002
```

In [68]:
```
geom_pmf(k=1,p=0.1)+geom_pmf(k=2,p=0.1)+geom_pmf(k=3,p=0.1)
```

Out[168]: **0.273**

In [ ]: