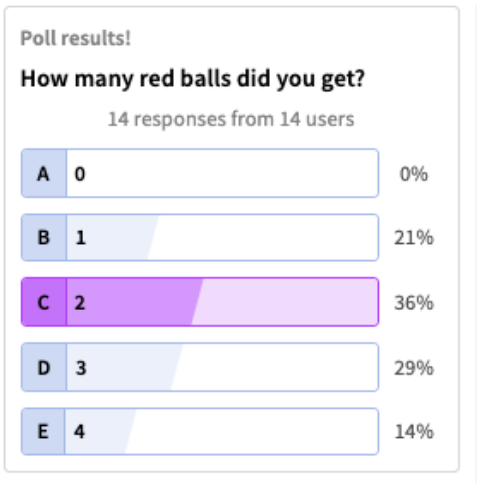
https://colab.research.google.com/drive/1QE-ZpDFX8d7Di0a3ySWgilP5cSz12DAS?usp=sharing

```
import numpy as np
import pandas as pd
import math
```




```
"R", "R", "B", "B"], size=4)
np.sum(instance == "R")
```

```
3
```

```
num_reds = []
for person in range(10000):
  instance = np.random.choice(["R", "R", "R", "B", "B"], size=4)
  num_reds.append(np.sum(instance == "R"))
pd.value_counts(num_reds, normalize=True)
```

```
2    0.3521
3    0.3401
1    0.1521
4    0.1316
0    0.0241
dtype: float64
```

```
num_reds = [] # empirical prob..
for person in range(1000000):
  instance = np.random.choice(["R", "R", "R", "B", "B"], size=4)
  num_reds.append(np.sum(instance == "R"))
pd.value_counts(num_reds, normalize=True)
```

```
2    0.346020
3    0.345503
1    0.153224
4    0.129685
0    0.025568
dtype: float64
```

```
(2/5)**4, 4*((2/5)**3)*(3/5) # P(X=0), p(X=1) theoritical prob
```

```
(0.025600000000000005, 0.15360000000000004)
```

```
np.mean(num_reds) # Expectation, expected average value of a Random Variable
```

```
2.400513
```

```
# using math module
# 4c0 * (2/5)**4
math.comb(4, 0) * (2/5)**4
```

```
0.025600000000000005
```

```
# scipy, binomial probability mass function
from scipy.stats import binom
binom.pmf(n=4, k=0, p=3/5)
```

```
0.025600000000000005
```

```
binom.pmf(n=4, k=2, p=3/5)
```

```
0.3456
```

```
binom.pmf(n=4, k=3, p=3/5)
```

```
0.3456000000000001
```

```
binom.pmf(n=4, k=1, p=3/5)
```



```
binom.pmf(n=4, k=4, p=3/5)
```

```
0.1296
```

```
n, p = 4, 3/5
expected_value = 0
for k in range(n+1):
  expected_value += k * binom.pmf(n=n, k=k, p=p)
print(expected_value)
```

```
2.4000000000000004
```

```
# theoritical
expected_earning = 0
n, p = 4, 3/5
expected_value = 0
for k in range(n+1):
  money = 150 if k == 4 else -10
  expected_value +=  money * binom.pmf(n=n, k=k, p=p)
print(expected_value)
```

```
10.735999999999997
```

```
expected_money = 0
for value in num_reds:
  if value == 4:
    expected_money += 150
  else:
    expected_money -= 10
print(expected_money/len(num_reds))
```

```
10.7496
```

Colab paid products - Cancel contracts here

Saving...

0s completed at 22:47