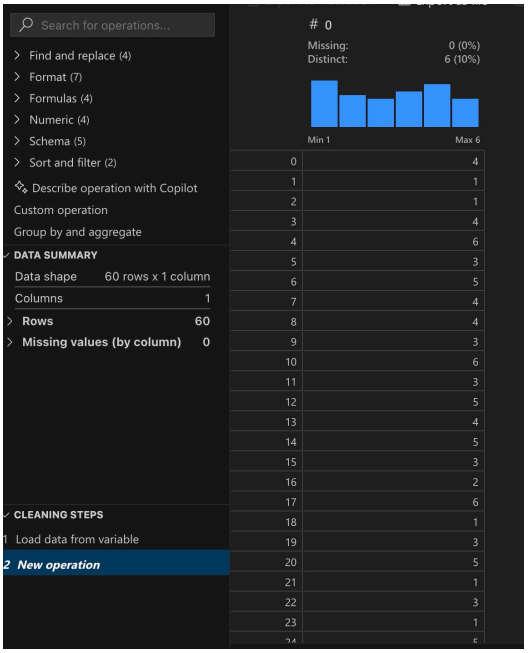
In this activity you will be calculating confidence intervals. You will need to use excel or a graphing calculator for this activity. We will be rolling a 6 sided die and obtaining the value from the die but the sample size will be much larger than the previous activity.

1. Roll your die 60 times and record the results.

60 rolls recorded in dataframe w/ histogram →



2. Find the mean and standard deviation.

$\bar{x} =$ 3.383

$SD =$ 1.757

3. Using the mean, standard deviation, and sample size construct a 90% confidence interval and a 95% confidence interval for the sample mean. Show your work for each confidence interval below.

$\bar{x} \pm z_{\alpha/2} \cdot \frac{s}{\sqrt{n}}$ $\quad SE = \frac{s}{\sqrt{n}} = \frac{1.757}{\sqrt{60}} = 0.227$

$z_{\frac{\alpha}{2}} = \pm 1.65 \rightarrow 90\%$

$z_{\alpha/2} = \pm 1.96 \rightarrow 95\%$

$3.383 + (1.65)(0.227) = 3.758$

$3.383 - (1.65)(0.227) = 3.008$

90% → (3.008, 3.758)

$3.383 \pm (1.96)(0.227) =$ (2.938, 3.828) ← 95%

4. Using the results from 1, consider the rolls that resulted in 5 or 6 as success and the ones that resulted in 1, 2, 3, and 4 as failure. Divide number of successes (the number of 5 and 6's) by the sample size (60) to obtain $\hat{p}$. Then find $\hat{q}$ and construct a 98% confidence interval and a 99% confidence interval for sample proportion. Show your work for each confidence interval below.

Successes = 20 $\quad \hat{p} = \frac{20}{60} = 1/3 \quad \hat{q} = 2/3 \quad n = 60$

$\hat{p} \pm z_{\frac{\alpha}{2}} \left(\frac{\hat{p}(1-\hat{p})}{n}\right)^{1/2}$ $\quad \sqrt{\frac{1/3(1-1/3)}{60}} = 0.0609$

99% → $z = \pm 2.58$ → $1/3 \pm (2.58)(0.0609) = (0.176, 0.490)$

98% → $z = \pm 2.33$ → $1/3 \pm (2.33)(0.0609) = (0.191, 0.475)$

In [ ]:
```
import pandas as pd
```

Roll 60 times and record results

In [6]:
```
import random

def roll_dice(num_sides=6):
    return random.randint(1, num_sides)


rolls = []
for _ in range(60):
    rolls.append(roll_dice())
df = pd.DataFrame(rolls)
df
```

Out[6]:

	0
0	4
1	1
2	1
3	4
4	6
5	3
6	5
7	4
8	4
9	3
10	6
11	3
12	5
13	4
14	5
15	3
16	2
17	6
18	1
19	3
20	5
21	1
22	3
23	1
24	5
25	1
26	1
27	5
28	2
29	5
30	1
31	4

	0
32	2
33	5
34	2
35	1
36	6
37	3
38	5
39	1
40	5
41	3
42	5
43	4
44	2
45	5
46	2
47	6
48	2
49	1
50	2
51	1
52	4
53	6
54	4
55	4
56	6
57	6
58	2
59	1

In [16]:
```
print("Mean: " + str(df.mean()[0]) + "\nStandard Deviation: " + str(df.std()
```

```
Mean: 3.3833333333333333
Standard Deviation: 1.7572288147377486
```

Remap 5 and 6 to success, everything else as failure

In [ ]:
```
mapped = []
for _, row in df.iterrows():
    if row[0] > 4:
        mapped.append(True)
    else:
        mapped.append(False)

dfMapped = pd.DataFrame(mapped)
dfMapped
```

Out[ ]:

	0
0	False
1	False
2	False
3	False
4	True
5	False
6	True
7	False
8	False
9	False
10	True
11	False
12	True
13	False
14	True
15	False
16	False
17	True
18	False
19	False
20	True
21	False
22	False
23	False
24	True
25	False
26	False
27	True
28	False
29	True
30	False
31	False
32	False
33	True
34	False
35	False
36	True
37	False
38	True
39	False
40	True
41	False
42	True
43	False
44	False
45	True
46	False
47	True
48	False
49	False
50	False
51	False
52	False
53	True
54	False
55	False
56	True
57	True
58	False
59	False