1) How-to-count-distance-to-the-previous-zero

For each value, count the difference of the distance from the previous zero (or the start of the Series, whichever is closer) and if there are no previous zeros,print the position Consider a DataFrame df where there is an integer column {'X':[7, 2, 0, 3, 4, 2, 5, 0, 3, 4]} The values should therefore be [1, 2, 0, 1, 2, 3, 4, 0, 1, 2]. Make this a new column 'Y'. import pandas as pd df = pd.DataFrame({'X': [7, 2, 0, 3, 4, 2, 5, 0, 3, 4]})

In [2]:
```
import pandas as pd
import numpy as np
df = pd.DataFrame({'X': [7, 2, 0, 3, 4, 2, 5, 0, 3, 4]})
```

In [11]:
```
# function to generate list to get the distence from zero in the list
def func(lst):
    x=list(lst)
    n = None
    y = []
    for i in range(len(x)):
        if x[i] != 0 and n== None:
            y.append(i+1)
        elif x[i] == 0:
            y.append(0)
            n = 0
        else:
            n += 1
            y.append(n)
    return y
```

In [12]:
```
# testing function
lst=[7, 2, 0, 3, 4, 2, 5, 0, 3, 4]
print(func(lst))
```

```
[1, 2, 0, 1, 2, 3, 4, 0, 1, 2]
```

In [21]:
```
# asssingning function return list value to df as a y column
df['Y'] = func(df['X'])
```