Data Science Masters :Assignment 7

```
Given a sequence of n values x1, x2, ..., xn and a window size k>0, the k-th moving
average of the given sequence is defined as follows:

    The moving average sequence has n-k+1 elements as shown below.
    The moving averages with k=4 of a ten-value sequence (n=10) is shown below
    i 1 2 3 4 5 6 7 8 9 10
        ===== == == == == == == == == ==
    Input 10 20 30 40 50 60 70 80 90 100
    y1 25 = (10+20+30+40)/4
    y2 35 = (20+30+40+50)/4
    y3 45 = (30+40+50+60)/4
    y4 55 = (40+50+60+70)/4
    y5 65 = (50+60+70+80)/4
    y6 75 = (60+70+80+90)/4
    y7 85 = (70+80+90+100)/4

    [25, 35, 45, 55, 65, 75, 85]

Thus, the moving average sequence has n-k+1=10-4+1=7 values.

1.  Write a function to find moving average in an array over a window:
    Test it over [3, 5, 7, 2, 8, 10, 11, 65, 72, 81, 99, 100, 150] and window of 3.
```

In [87]:
```
# Solution:
# Defining function to compute Moving Avg for the given input list and window of 3...
def computeMovingAVG(inputVector,k) :
    n =  len(inputVector)-k+1   # Calculating the value of n to find out the number of values in moving avg sequence...
    outList = []
    for x in range(n):          # Iterating the loop with range(n)...
        sum = 0
        out = []                # To display computation steps, declaring an empty list to hold the intermediate result..

        for i in range(x,k+x) : # Iterating thru input list with range seq based on the given window size i.e.3 ...
            sum += inputVector[i]  # Calculating the sum of values present in the given window...
            out.append(inputVector[i])

        # Printing the intermediate result to display the computation of Moving Avg Seq
        print(round((sum/k),2)," = Total(",out,") /",k)
        outList.append(round((sum/k),2))  # Appending the value rounded to 2 decimal places to the output List...
    return outList

# Given Input Details...
inputList = [3, 5, 7, 2, 8, 10, 11, 65, 72, 81, 99, 100, 150]
windowSize = 3 # Size of Window is 3
print("Input =>",inputList)
print("Computation Steps: =>")
# Printing the result List
print("\nMoving AVG Sequence =>",computeMovingAVG(inputList,windowSize))
```

```
Input => [3, 5, 7, 2, 8, 10, 11, 65, 72, 81, 99, 100, 150]
Computation Steps: =>
5.0  = Total( [3, 5, 7] ) / 3
4.67  = Total( [5, 7, 2] ) / 3
5.67  = Total( [7, 2, 8] ) / 3
6.67  = Total( [2, 8, 10] ) / 3
9.67  = Total( [8, 10, 11] ) / 3
28.67  = Total( [10, 11, 65] ) / 3
49.33  = Total( [11, 65, 72] ) / 3
72.67  = Total( [65, 72, 81] ) / 3
84.0  = Total( [72, 81, 99] ) / 3
93.33  = Total( [81, 99, 100] ) / 3
116.33  = Total( [99, 100, 150] ) / 3

Moving AVG Sequence => [5.0, 4.67, 5.67, 6.67, 9.67, 28.67, 49.33, 72.67, 84.0, 93.33, 116.33]
```