

Data Science Masters :Assignment 6

1. Write a function so that the columns of the output matrix are powers of the input vector.

The order of the powers is determined by the increasing boolean argument. Specifically, when increasing is False, the i-th output column is the input vector raised element-wise to the power of $N - i - 1$.

HINT: Such a matrix with a geometric progression in each row is named for Alexandre-Theophile Vandermonde.

Alexandre-Theophile Vandermonde Matrix for input [1,2,3,4,5] with increasing 'False' looks like:

```
[[ 1, 1, 1, 1, 1],
 [ 16, 8, 4, 2, 1],
 [ 81, 27, 9, 3, 1],
 [256, 64, 16, 4, 1],
 [625, 125, 25, 5, 1]]
```

```

In [72]: # Solution:
# Defining 'formVandermondeMatrix' function with three arguments....
def formVandermondeMatrix(in_Vector,n,increasing_bool) :
    # Based on increasing_bool (boolean value) , the order of the power changes accordingly...
    if (increasing_bool) :
        powerVector = range(n)
    else :
        powerVector = range(n-1,-1,-1)

    # Iterating each element in inputVector and computing the power of it iterating thru powerVector
    # and appending to outputList...
    for i in in_Vector :
        outList = []
        for pow in powerVector :
            outList.append(i**pow)
        outputVector.append(outList)
    return(outputVector)

# Defining variables to hold data...
inputVector = [1, 2, 3, 4, 5]
outputVector = []

# Calculating the length to compute the power of the given vector elements...
n = len(inputVector)

#Calling function with arguments in_vector,n and boolean (increasing)...
resultVector = formVandermondeMatrix(inputVector,n,False)

print("Input Vector ->",inputVector)
# Printing result as it is - 2D array format
print("\nResult in 2D array format :-\n",resultVector)

# Printing the ouput to Look Like Matrix
print("\nResult in Matrix format :-")
print('\n'.join([''.join(['{:4}'.format(item) for item in row])
                for row in resultVector]))

# Printing the result as Matrix with the help of Numpy Library..
import numpy as np
print("\nResult in Matrix format using Numpy Library :-")
np.matrix(np.reshape(resultVector,(n,n)))

```

Input Vector -> [1, 2, 3, 4, 5]

Result in 2D array format :-

```
[[1, 1, 1, 1, 1], [16, 8, 4, 2, 1], [81, 27, 9, 3, 1], [256, 64, 16, 4, 1], [625, 125, 25, 5, 1]]
```

Result in Matrix format :-

```
1  1  1  1  1
16 8  4  2  1
81 27 9  3  1
256 64 16 4  1
625 125 25 5  1
```

Result in Matrix format using Numpy Library :-

```
Out[72]: matrix([[ 1,  1,  1,  1,  1],
                 [16,  8,  4,  2,  1],
                 [81, 27,  9,  3,  1],
                 [256, 64, 16,  4,  1],
                 [625, 125, 25,  5,  1]])
```