```
import numpy as np
a3 = np.random.randint(40,90,100) #100 numbers in range of 40,90
a3 #start stop and size
```

```
array([76, 54, 43, 54, 50, 40, 73, 64, 73, 80, 60, 68, 45, 80, 79, 48, 85,
       46, 42, 82, 62, 67, 52, 85, 68, 53, 47, 54, 63, 81, 83, 48, 69, 66,
       49, 55, 42, 61, 75, 89, 70, 60, 75, 81, 89, 81, 87, 88, 46, 56, 63,
       77, 48, 66, 45, 45, 42, 85, 78, 69, 43, 63, 55, 80, 53, 51, 88, 51,
       43, 70, 81, 79, 58, 75, 67, 40, 74, 43, 52, 43, 63, 73, 71, 64, 81,
       56, 61, 40, 65, 44, 74, 85, 77, 62, 89, 51, 82, 67, 85, 61])
```

```
a4 = np.random.randint(150,185,100)
a4
```

```
array([164, 172, 169, 164, 151, 164, 168, 173, 182, 154, 169, 157, 170,
       173, 177, 169, 157, 159, 178, 154, 158, 163, 166, 165, 155, 153,
       182, 173, 156, 169, 175, 179, 177, 155, 162, 155, 160, 177, 176,
       156, 160, 179, 150, 153, 170, 176, 160, 169, 150, 164, 165, 157,
       184, 166, 153, 174, 171, 168, 173, 168, 155, 172, 164, 151, 161,
       182, 157, 175, 175, 183, 152, 169, 165, 153, 171, 179, 161, 156,
       181, 179, 161, 181, 151, 157, 157, 153, 154, 181, 152, 178, 151,
       174, 172, 150, 165, 165, 175, 184, 162, 152])
```

```
a5 = np.random.randint(17,30,100)
a5
```

```
array([17, 27, 23, 24, 27, 28, 20, 23, 18, 22, 24, 19, 23, 23, 26, 19, 21,
       21, 22, 24, 18, 19, 24, 18, 23, 17, 17, 20, 27, 27, 24, 22, 20, 26,
       28, 18, 29, 29, 19, 28, 18, 29, 19, 22, 26, 17, 24, 19, 22, 18, 18,
       23, 26, 29, 20, 24, 17, 29, 27, 17, 28, 22, 26, 28, 18, 25, 19, 21,
       20, 20, 27, 26, 29, 27, 22, 21, 29, 18, 20, 25, 21, 25, 27, 25, 23,
       21, 24, 27, 20, 18, 19, 26, 24, 18, 23, 22, 28, 29, 25, 28])
```

```
aconc·=np.concatenate((a3,a4,a5))·#inside·one·braces·is·a·must
aconc
```

```
array([ 76,  54,  43,  54,  50,  40,  73,  64,  73,  80,  60,  68,  45,
        80,  79,  48,  85,  46,  42,  82,  62,  67,  52,  85,  68,  53,
        47,  54,  63,  81,  83,  48,  69,  66,  49,  55,  42,  61,  75,
        89,  70,  60,  75,  81,  89,  81,  87,  88,  46,  56,  63,  77,
        48,  66,  45,  45,  42,  85,  78,  69,  43,  63,  55,  80,  53,
        51,  88,  51,  43,  70,  81,  79,  58,  75,  67,  40,  74,  43,
        52,  43,  63,  73,  71,  64,  81,  56,  61,  40,  65,  44,  74,
        85,  77,  62,  89,  51,  82,  67,  85,  61, 164, 172, 169, 164,
       151, 164, 168, 173, 182, 154, 169, 157, 170, 173, 177, 169, 157,
       159, 178, 154, 158, 163, 166, 165, 155, 153, 182, 173, 156, 169,
       175, 179, 177, 155, 162, 155, 160, 177, 176, 156, 160, 179, 150,
       153, 170, 176, 160, 169, 150, 164, 165, 157, 184, 166, 153, 174,
       171, 168, 173, 168, 155, 172, 164, 151, 161, 182, 157, 175, 175,
       183, 152, 169, 165, 153, 171, 179, 161, 156, 181, 179, 161, 181,
       151, 157, 157, 153, 154, 181, 152, 178, 151, 174, 172, 150, 165,
       165, 175, 184, 162, 152,  17,  27,  23,  24,  27,  28,  20,  23,
        18,  22,  24,  19,  23,  23,  26,  19,  21,  21,  22,  24,  18,
        19,  24,  18,  23,  17,  17,  20,  27,  27,  24,  22,  20,  26,
        28,  18,  29,  29,  19,  28,  18,  29,  19,  22,  26,  17,  24,
```

```
        19,  22,  18,  18,  23,  26,  29,  20,  24,  17,  29,  27,  17,
        28,  22,  26,  28,  18,  25,  19,  21,  20,  20,  27,  26,  29,
        27,  22,  21,  29,  18,  20,  25,  21,  25,  27,  25,  23,  21,
        24,  27,  20,  18,  19,  26,  24,  18,  23,  22,  28,  29,  25,
        28])
```

```
1 aconc.shape
```

```
(300,)
```

```
1 aconch =np.hstack((a3,a4,a5)) #inside one braces is a must
2 aconch
```

```
array([ 76,  54,  43,  54,  50,  40,  73,  64,  73,  80,  60,  68,  45,
        80,  79,  48,  85,  46,  42,  82,  62,  67,  52,  85,  68,  53,
        47,  54,  63,  81,  83,  48,  69,  66,  49,  55,  42,  61,  75,
        89,  70,  60,  75,  81,  89,  81,  87,  88,  46,  56,  63,  77,
        48,  66,  45,  45,  42,  85,  78,  69,  43,  63,  55,  80,  53,
        51,  88,  51,  43,  70,  81,  79,  58,  75,  67,  40,  74,  43,
        52,  43,  63,  73,  71,  64,  81,  56,  61,  40,  65,  44,  74,
        85,  77,  62,  89,  51,  82,  67,  85,  61, 164, 172, 169, 164,
       151, 164, 168, 173, 182, 154, 169, 157, 170, 173, 177, 169, 157,
       159, 178, 154, 158, 163, 166, 165, 155, 153, 182, 173, 156, 169,
       175, 179, 177, 155, 162, 155, 160, 177, 176, 156, 160, 179, 150,
       153, 170, 176, 160, 169, 150, 164, 165, 157, 184, 166, 153, 174,
       171, 168, 173, 168, 155, 172, 164, 151, 161, 182, 157, 175, 175,
       183, 152, 169, 165, 153, 171, 179, 161, 156, 181, 179, 161, 181,
       151, 157, 157, 153, 154, 181, 152, 178, 151, 174, 172, 150, 165,
       165, 175, 184, 162, 152,  17,  27,  23,  24,  27,  28,  20,  23,
        18,  22,  24,  19,  23,  23,  26,  19,  21,  21,  22,  24,  18,
        19,  24,  18,  23,  17,  17,  20,  27,  27,  24,  22,  20,  26,
        28,  18,  29,  29,  19,  28,  18,  29,  19,  22,  26,  17,  24,
        19,  22,  18,  18,  23,  26,  29,  20,  24,  17,  29,  27,  17,
        28,  22,  26,  28,  18,  25,  19,  21,  20,  20,  27,  26,  29,
        27,  22,  21,  29,  18,  20,  25,  21,  25,  27,  25,  23,  21,
        24,  27,  20,  18,  19,  26,  24,  18,  23,  22,  28,  29,  25,
        28])
```

Note : Hstack = Concatenate

```
1 aconcv = np.vstack((a3,a4,a5)) #vertical stack
2 aconcv
```

```
array([[ 76,  54,  43,  54,  50,  40,  73,  64,  73,  80,  60,  68,  45,
         80,  79,  48,  85,  46,  42,  82,  62,  67,  52,  85,  68,  53,
         47,  54,  63,  81,  83,  48,  69,  66,  49,  55,  42,  61,  75,
         89,  70,  60,  75,  81,  89,  81,  87,  88,  46,  56,  63,  77,
         48,  66,  45,  45,  42,  85,  78,  69,  43,  63,  55,  80,  53,
         51,  88,  51,  43,  70,  81,  79,  58,  75,  67,  40,  74,  43,
         52,  43,  63,  73,  71,  64,  81,  56,  61,  40,  65,  44,  74,
         85,  77,  62,  89,  51,  82,  67,  85,  61],
       [164, 172, 169, 164, 151, 164, 168, 173, 182, 154, 169, 157, 170,
        173, 177, 169, 157, 159, 178, 154, 158, 163, 166, 165, 155, 153,
        182, 173, 156, 169, 175, 179, 177, 155, 162, 155, 160, 177, 176,
        156, 160, 179, 150, 153, 170, 176, 160, 169, 150, 164, 165, 157,
```

```
       184, 166, 153, 174, 171, 168, 173, 168, 155, 172, 164, 151, 161,
       182, 157, 175, 175, 183, 152, 169, 165, 153, 171, 179, 161, 156,
       181, 179, 161, 181, 151, 157, 157, 153, 154, 181, 152, 178, 151,
       174, 172, 150, 165, 165, 175, 184, 162, 152],
      [ 17,  27,  23,  24,  27,  28,  20,  23,  18,  22,  24,  19,  23,
        23,  26,  19,  21,  21,  22,  24,  18,  19,  24,  18,  23,  17,
        17,  20,  27,  27,  24,  22,  20,  26,  28,  18,  29,  29,  19,
        28,  18,  29,  19,  22,  26,  17,  24,  19,  22,  18,  18,  23,
        26,  29,  20,  24,  17,  29,  27,  17,  28,  22,  26,  28,  18,
        25,  19,  21,  20,  20,  27,  26,  29,  27,  22,  21,  29,  18,
        20,  25,  21,  25,  27,  25,  23,  21,  24,  27,  20,  18,  19,
        26,  24,  18,  23,  22,  28,  29,  25,  28]])
```

```
1 aconc.shape
```

```
(300,)
```

```
1 aconch.shape
```

```
(300,)
```

```
1 aconcv.shape
```

```
(3, 100)
```

```
1 aMinRowWise = np.amin(aconcv,axis=1)
2 aMinRowWise
```

```
array([ 40, 150,  17])
```

```
1 aMaxRowWise = np.amax(aconcv,axis=1)
2 aMaxRowWise
```

```
array([ 89, 184,  29])
```

```
1 aMeanRowWise = np.mean(aconcv,axis=1)
2 aMeanRowWise
```

```
array([ 64.22, 165.62,  22.92])
```

```
1 aMeanRow0 = np.mean(aconcv[0:1],axis=1)
2 aMeanRow0
```

```
array([64.22])
```

```
1 a7=np.random.rand(1000)
2 a7
```

```
       7.61697056e-01, 6.30368664e-01, 9.95707563e-01, 5.17121344e-01,
       4.12675185e-02, 1.07862554e-01, 1.78516842e-01, 5.17836579e-01,
       4.44288989e-01, 6.31592115e-01, 8.09465472e-01, 7.04988851e-02,
       9.39753729e-01, 2.08187118e-01, 7.72647299e-01, 6.66004877e-01,
       7.03378988e-01, 9.15260745e-01, 8.55271802e-01, 2.08577259e-01,
```

```
2.19306890e-01, 9.34514843e-01, 8.43357146e-01, 3.21287189e-01,
5.73291036e-01, 6.24649689e-01, 9.00099317e-01, 6.07569640e-01,
6.88788942e-01, 3.50718139e-01, 1.23315243e-01, 9.03034295e-01,
4.16170690e-01, 9.19363311e-01, 3.44283434e-01, 2.08117209e-01,
1.37028013e-02, 4.77143501e-01, 6.04416073e-01, 6.02090082e-01,
7.14854442e-01, 4.82996181e-01, 4.23865215e-01, 4.57833455e-01,
4.60256959e-01, 4.94881212e-02, 3.43837222e-01, 1.52836130e-01,
5.60704393e-01, 9.87559636e-01, 3.55835192e-01, 8.28854998e-01,
2.28175359e-01, 6.22106946e-01, 9.32916575e-01, 7.50525538e-01,
7.96000424e-01, 4.32819974e-01, 7.86621565e-01, 6.53984637e-01,
7.26450185e-01, 9.63557461e-01, 8.57729809e-01, 6.25166317e-01,
8.33391690e-02, 8.70640840e-01, 3.00121549e-01, 3.62737542e-01,
9.76017103e-01, 7.44809421e-01, 8.24933876e-01, 1.44511554e-01,
1.19053060e-02, 7.82727636e-01, 8.22900027e-01, 2.65973135e-01,
4.10265960e-01, 8.71792769e-01, 1.95858804e-01, 8.07894794e-01,
1.79362678e-02, 3.08025465e-01, 7.24514530e-01, 9.37990075e-01,
5.39458954e-01, 7.93942381e-01, 5.15375667e-01, 4.67213106e-01,
7.68362600e-01, 1.83921378e-01, 7.87485365e-01, 7.27798230e-01,
2.73733708e-01, 3.20491956e-01, 4.68670944e-01, 7.29102489e-01,
2.37856508e-01, 7.30401561e-01, 1.12352494e-02, 1.70745848e-02,
3.76081328e-01, 2.40925110e-01, 4.12754389e-01, 7.35097001e-01,
2.00962094e-01, 3.19289422e-01, 2.89723210e-01, 6.21035773e-01,
6.10014584e-01, 4.36961888e-02, 6.07889392e-01, 1.69973485e-01,
3.44062887e-01, 6.81391392e-01, 2.95593859e-01, 2.34347657e-02,
5.45713452e-01, 9.11250258e-02, 6.72604629e-01, 1.75800089e-01,
6.87459850e-01, 6.73873437e-01, 8.71967184e-01, 7.53176841e-01,
8.95043097e-01, 8.26624865e-01, 6.32318781e-01, 1.97027546e-01,
2.41510690e-01, 4.92910139e-01, 8.61565414e-01, 8.15609491e-01,
5.47752817e-01, 3.81599372e-01, 7.47571715e-01, 7.22760231e-01,
3.22658537e-01, 8.66597284e-01, 1.44252182e-03, 5.54556795e-01,
1.99274983e-01, 5.45833329e-01, 9.67414587e-02, 1.23207000e-01,
4.71685773e-01, 7.11588662e-01, 5.57895820e-01, 8.34690202e-01,
5.44743465e-01, 8.47890576e-01, 9.61842012e-01, 8.89012010e-01,
8.68303506e-01, 7.20084875e-01, 8.76400824e-01, 9.02489869e-01,
5.12508233e-01, 2.60031631e-01, 4.16144346e-01, 5.73985393e-01,
1.44936109e-01, 6.72578444e-01, 9.00078210e-01, 9.69160388e-01,
5.52112743e-01, 9.47260010e-01, 7.96359479e-01, 3.86358540e-01,
1.21336626e-01, 4.73172383e-01, 5.79199444e-01, 1.43099159e-01,
9.63340594e-01, 3.48547205e-02, 5.44992855e-01, 4.09005206e-01,
9.61555063e-01, 2.99072627e-02, 9.28110376e-01, 5.43750938e-02,
1.43992191e-01, 6.57626309e-01, 6.63713867e-01, 6.00000993e-01,
7.98244498e-01, 4.68217666e-01, 4.09348337e-01, 2.25725704e-01,
3.08471453e-01, 4.40530942e-01, 9.50651459e-01, 1.34464176e-01,
1.28923634e-02, 3.40039299e-01, 1.55422138e-01, 6.96176515e-01,
7.05232484e-01, 5.81039022e-01, 7.11881296e-01, 5.00220673e-01,
2.33026012e-01, 3.56096833e-01, 3.80057354e-01, 2.73410216e-01,
9.85705159e-01, 2.97848151e-01, 6.56179941e-01, 8.34454862e-01,
8.81484618e-02, 7.42188551e-01, 1.40209633e-01, 6.91403328e-01,
7.56165811e-01, 6.25549634e-02, 7.78054386e-01, 6.72692699e-01,
5.75913713e-01, 1.47927496e-01, 1.82867250e-01, 3.07693872e-01,
4.66305368e-01, 8.41043404e-01, 4.63530079e-01, 2.01138272e-01,
7.88525687e-01, 1.15677484e-01, 6.60733165e-02, 1.42694144e-01,
2.71850745e-01, 2.70079464e-01, 9.32543260e-03, 2.82620772e-01,
4.12461080e-01, 5.25011936e-01, 2.23014032e-01, 7.74475682e-01])
```

```
1 a7mean = np.mean(a7)
```

```
1  #to find sum of std div =0
2  a9 = a7 - a7mean
```

```
3  a9
```

```
array([-3.12902324e-01, -1.76534731e-01, -4.12432916e-01,  1.54303086e-01,
        3.49001828e-01,  4.25932615e-01, -8.80557731e-03, -1.17620901e-01,
       -6.35941176e-02, -4.13850307e-01,  1.04348268e-01, -3.60821941e-01,
        2.38426174e-01,  5.16662288e-02, -1.50910639e-01,  2.54966035e-01,
        4.30820273e-01, -2.26030996e-01, -5.14790600e-01, -1.06170756e-01,
        1.03494488e-01,  4.76541765e-01,  1.79932080e-01, -4.30705398e-01,
       -6.89485296e-02, -5.01234727e-01,  1.71558795e-01,  4.13742367e-01,
        1.23192745e-01, -4.77926319e-01, -4.82818412e-01,  1.13540950e-01,
        4.53141130e-01,  7.86120142e-03, -1.46307932e-01,  2.86160287e-01,
        3.28819359e-01, -3.14825763e-01,  3.83691937e-01,  2.20620319e-01,
        4.15783290e-01,  3.20311055e-01,  3.49088024e-01, -8.38170345e-02,
        2.76392287e-01,  2.71387318e-01, -3.07683724e-02,  9.23320818e-02,
        6.56561789e-03,  2.67685440e-01, -4.49774069e-01, -4.66103789e-01,
       -4.08827505e-01,  2.24767418e-01, -4.11385237e-02, -2.77855128e-01,
        3.90160762e-01,  3.05081522e-01, -3.22846314e-01,  4.02564576e-01,
        2.10354687e-01, -4.80380924e-01, -2.43818453e-01, -2.88820829e-01,
       -3.24403879e-01, -3.81747740e-01,  1.39846529e-01,  1.27830989e-01,
        3.70779911e-01, -1.68170446e-01, -4.06722024e-01,  6.63090526e-02,
        2.60244524e-01, -1.36423752e-01,  1.94045489e-01, -4.50153120e-02,
       -2.60208681e-01, -4.94451635e-01,  2.10361543e-01,  3.13919074e-02,
        2.25955672e-01,  7.05496604e-02, -3.66243728e-01, -3.73132911e-01,
       -7.63205298e-03,  2.46099399e-01, -3.06296905e-01,  3.28639632e-01,
        4.03818568e-01, -2.92812489e-02, -2.14754548e-01,  3.28809616e-01,
       -4.03685834e-02, -2.03701061e-01,  2.62919151e-01,  7.70697795e-02,
       -2.93727087e-01,  2.34380690e-01,  3.71696414e-01,  2.54935827e-01,
       -1.36070999e-01,  3.18590545e-01,  2.54085536e-01, -7.64295405e-02,
       -7.31918710e-02, -2.05921314e-01, -4.78245801e-01, -3.40939963e-01,
        3.61481090e-01,  4.42671662e-01, -2.91226048e-01,  2.36264806e-02,
       -4.48560277e-01, -2.81172664e-01, -4.57710501e-01,  4.32364089e-01,
       -3.58286648e-01,  4.51252918e-01, -2.76436731e-01,  1.84414848e-01,
        1.22639910e-01,  1.97167171e-01,  2.06786694e-01,  4.24057934e-01,
        2.04846762e-01,  1.62247385e-01, -4.70046795e-01,  6.38249055e-02,
       -4.92221491e-02, -5.48914155e-02, -4.26811365e-01,  3.57377214e-02,
       -2.20922726e-01,  9.01295362e-02, -1.98398737e-01,  2.70151397e-01,
       -1.09378414e-01, -7.13975680e-02,  2.51590974e-01, -4.99394240e-01,
       -1.79370469e-01, -4.98510912e-01, -1.54017174e-01,  2.60716934e-01,
        4.79001288e-01, -4.56595644e-01,  3.84099506e-01,  3.48776465e-01,
       -4.38041061e-01,  4.58308331e-01, -1.59767291e-01,  4.05394813e-01,
       -4.17101585e-01,  3.77359831e-01,  4.00013910e-01,  2.10126018e-01,
        1.47377315e-01,  2.93730439e-01, -1.00013233e-01,  4.02346452e-01,
        2.36847757e-01,  3.23666006e-02, -9.07121335e-02,  2.39865706e-01,
       -2.49887417e-01,  3.76446170e-01, -2.52268820e-01,  2.85946279e-01,
        1.40471826e-01,  8.41227466e-02,  3.26108192e-01,  4.41335269e-01,
        2.67923967e-01, -1.23275480e-01, -1.08321184e-01,  2.82466174e-01,
        2.09299104e-01,  2.56928853e-01, -8.54878688e-02, -4.84145959e-01,
       -1.97770702e-01, -1.08551757e-01, -6.47586998e-02, -4.04128106e-01,
        3.18584422e-01, -3.45023309e-01,  7.86802355e-02,  1.06602661e-01,
        5.12072464e-02,  3.90944649e-01,  9.17438025e-02,  3.95666984e-01,
        4.61821833e-01, -3.02591966e-01,  2.38547036e-01,  1.33874972e-01,
       -2.09595621e-01, -4.59590570e-01,  3.90720225e-01,  4.10499819e-01,
        3.78269501e-01,  1.89341096e-01,  1.27332249e-01, -4.38021925e-01,
        4.30871721e-01,  7.04849448e-03, -2.92375045e-01, -2.93706549e-01,
        2.01659061e-01, -5.00874410e-01, -2.77756340e-02, -3.60939686e-01,
       -1.16863201e-01,  2.37203976e-01, -1.25810566e-01,  3.82656973e-01,
       -2.95969485e-01, -1.67934300e-01, -3.99411724e-01, -2.57424763e-01,
        3.53639774e-01, -1.65989313e-01,  2.94722207e-01, -1.31282829e-02,
       -2.25694984e-01, -2.01003800e-01,  4.44842943e-01,  2.93542915e-01,
```

```
np.sum(a9)
```

```
-1.865174681370263e-14
```

```
a9 = np.random.rand(1000)
a9
```

```
8.87432700e-01, 1.11202733e-01, 9.70413189e-01, 7.44517265e-01,
3.77820998e-02, 3.97031429e-01, 8.78396876e-01, 8.84916986e-01,
4.51016525e-01, 4.69421545e-01, 6.95956183e-01, 7.64990718e-01,
1.29554391e-01, 1.82168768e-01, 3.70053124e-01, 9.35320426e-02,
3.74434971e-01, 2.22347221e-01, 6.40329341e-01, 9.14398703e-01,
2.28260069e-01, 3.82628466e-01, 8.77150955e-01, 8.90478627e-01,
2.39566537e-01, 1.41720791e-01, 1.36477499e-01, 3.83222854e-01,
2.96754290e-01, 3.95658091e-01, 4.86807607e-03, 6.76435443e-01,
6.30708450e-01, 3.09544075e-01, 8.20400775e-01, 7.41044921e-03,
8.52573134e-01, 8.15624890e-01, 9.84776938e-01, 7.70295620e-02,
7.39122093e-01, 3.61890784e-01, 1.89960993e-01, 4.46737051e-01,
9.09010278e-01, 4.48034711e-01, 5.40806775e-01, 1.82669717e-02,
1.71298181e-01, 1.06880318e-01, 4.85762525e-01, 4.17750126e-01,
8.00121948e-01, 7.53271381e-01, 1.18173250e-01, 5.64838571e-01,
6.72459140e-01, 9.91208528e-01, 3.72651083e-02, 2.43708554e-01,
1.90147900e-01, 6.12196263e-01, 6.02430670e-01, 6.27647199e-02,
2.47422012e-01, 1.28991002e-01, 3.39684123e-01, 8.65102508e-01,
1.13496656e-01, 2.14671842e-01, 2.98514315e-01, 6.38579928e-01,
1.88688275e-01, 9.07955150e-02, 5.04546429e-01, 3.34321467e-01,
2.34748174e-01, 5.85153897e-01, 7.33851744e-03, 6.76097352e-01,
8.66501333e-01, 8.13694920e-01, 7.59618823e-01, 7.35675054e-01,
5.98355943e-01, 8.69055386e-01, 9.96699923e-01, 9.53276450e-01,
1.01250620e-01, 1.97746044e-01, 4.05841646e-01, 5.61294274e-01,
6.32750956e-02, 7.67671423e-01, 3.56076806e-01, 8.20411612e-01,
9.50182659e-02, 3.05097854e-01, 8.12346408e-02, 2.53769311e-01,
1.59882387e-01, 6.86738622e-01, 9.82722592e-01, 8.10195801e-02,
2.80494466e-02, 9.89587390e-01, 6.11593400e-01, 2.08226700e-01,
9.27066116e-01, 1.21600846e-01, 2.01139111e-01, 4.62483494e-01,
1.07276007e-01, 4.39544722e-01, 5.15195788e-01, 5.50414906e-01,
3.47267679e-01, 1.54522138e-01, 3.06799381e-01, 4.07506730e-01,
7.60087702e-01, 2.66776549e-01, 8.05888416e-01, 2.18698737e-01,
7.71657747e-01, 5.31031164e-01, 9.12080968e-01, 9.09212950e-01,
4.98655955e-01, 2.10035907e-01, 5.53378660e-01, 3.70326215e-01,
1.97505401e-02, 2.35510603e-01, 4.13257694e-01, 4.86103991e-01,
6.45545272e-02, 7.67764032e-01, 6.22413332e-01, 3.23192575e-01,
2.02973550e-02, 8.19831382e-01, 5.01467150e-01, 6.89007248e-02,
4.37229207e-01, 4.55662991e-01, 7.79603265e-01, 5.44447681e-02,
3.94920843e-01, 5.66415174e-01, 9.32982428e-01, 8.61040761e-01,
2.32145603e-01, 5.76037938e-01, 1.28051825e-01, 7.00246877e-01,
1.83338129e-01, 1.95059370e-02, 8.93365793e-01, 7.23056889e-01,
6.09938059e-01, 6.34495901e-01, 1.79402341e-01, 1.22330939e-01,
1.90697245e-01, 5.78852676e-01, 9.18388942e-01, 1.34596308e-01,
4.55285252e-01, 5.06936622e-01, 7.29187224e-01, 1.14662830e-01,
2.35286907e-01, 5.49077245e-01, 3.44019115e-01, 7.29006100e-01,
6.31254641e-01, 6.55419586e-01, 1.04134190e-01, 2.97585075e-01,
6.62173926e-01, 7.37181565e-01, 4.82935153e-01, 6.49712237e-02,
1.96086607e-01, 4.56020744e-01, 4.24042128e-01, 1.08582405e-01,
4.68943089e-01, 5.87998762e-01, 1.29942333e-01, 2.89158428e-01,
4.40764370e-01, 7.41633361e-01, 2.67500523e-01, 6.06951649e-01,
6.55488141e-01, 2.00947140e-01, 1.54337308e-01, 4.74405532e-01,
3.32491454e-01, 1.21020353e-01, 9.93935394e-01, 4.73408480e-01,
```

```
2.97221962e-01, 1.75133219e-01, 8.36256604e-01, 7.14537853e-01,
6.59218974e-01, 1.52964559e-01, 8.83455263e-01, 7.64974463e-01,
7.67853253e-01, 7.66794612e-01, 5.78207508e-01, 1.56264526e-01,
2.84729054e-01, 4.02204523e-02, 5.83378276e-01, 4.30820305e-02,
6.32073679e-01, 2.59563438e-01, 8.02528230e-01, 9.22760394e-01,
4.12246224e-01, 3.80123485e-01, 9.61516626e-01, 7.84026823e-01,
1.59902475e-02, 1.49598305e-01, 9.06310208e-01, 2.62699124e-02,
```

```
for k in range(0,50) :
    a10 = a9[0:k]
    print(k,np.mean(a10))
```

```
0 nan
1 0.4360123730002565
2 0.5661521664501119
3 0.6491771989342844
4 0.6570747531075474
5 0.6307048553976424
6 0.5473586066106438
7 0.529198384304294
8 0.568218060461394
9 0.5898143628947293
10 0.5671147930803803
11 0.5962727151398544
12 0.5687053401309422
13 0.5263107255926147
14 0.5353076728003835
15 0.535482461111252
16 0.5612510962417774
17 0.5626899027171507
18 0.5572434355461877
19 0.5737818725391042
20 0.5769625031384489
21 0.5730789596101795
22 0.5862226310377247
23 0.5792347735735641
24 0.5897855957188232
25 0.5953188998925284
26 0.5829058742000367
27 0.5935400433469809
28 0.5725106507171676
29 0.5698872578607052
30 0.573835144375971
31 0.5619369113591929
32 0.5753365872940521
33 0.5773695530145248
34 0.5620877434064353
35 0.5602740342219907
36 0.5540093645180414
37 0.5627493449325047
38 0.5583489983644776
39 0.5686875328284574
40 0.5653989503698803
41 0.5553584723787236
42 0.5520111235702583
43 0.5609512908072477
44 0.5639709464813915
45 0.5562182138624276
```

```
46 0.5463624321903514
47 0.5552914474155807
48 0.5534322465692751
49 0.5464747642518869
/usr/local/lib/python3.7/dist-packages/numpy/core/fromnumeric.py:3441: RuntimeWarning
  out=out, **kwargs)
/usr/local/lib/python3.7/dist-packages/numpy/core/_methods.py:189: RuntimeWarning: ir
  ret = ret.dtype.type(ret / rcount)
```

```
### Oout of box
j=0
k=50
for i in range(0,20) :
    np.cumsum(a9[j:k])
    print(np.cumsum(a9[j:k]))
    j+=50
    k+=50
```

```
 13.64634728 13.67403859 14.12223516 14.20106484 14.30741274 15.16771497
 15.59252519 16.20789229 16.73276982 17.62012748 18.26278976 19.24588274
 19.59071045 20.49845538 20.82750047 21.29474412 21.99039017 22.49714112
 23.01729775 23.51968753 23.89014084 24.16686732 24.53633767 25.52253004
 26.13845989 26.72732283]
[ 0.56011756  0.68388523  0.83608756  1.35121785  2.32814929  2.57941583
  2.67440053  2.72228741  3.37128088  3.37411238  3.68892886  4.59766756
  5.46537058  6.27729207  6.3795888   6.61554548  6.82232906  6.82272248
  6.94791498  6.99587548  7.5756067   7.68929176  8.29557378  8.68449395
  9.52285735  9.65296509 10.2548264  10.88444404 11.27082986 12.03145812
 12.51774601 13.10402632 13.98490981 14.00906392 14.44141963 15.30301275
 15.53694873 16.14127215 17.02892418 17.44742094 18.28051164 19.11260893
 19.2894968  19.84547692 20.16725701 20.95864858 21.26237587 21.77238444
 22.45058898 23.10837861]
[ 0.10259975  0.23507651  0.41976158  1.34644396  2.10892957  2.25569901
  3.15419064  3.76346199  3.98336475  4.40151911  5.2966983   5.76071786
  6.16039208  6.56792243  7.45535513  7.56655786  8.53697105  9.28148832
  9.31927042  9.71630184 10.59469872 11.47961571 11.93063223 12.40005378
 13.09600996 13.86100068 13.99055507 14.17272384 14.54277696 14.636309
 15.01074397 15.2330912  15.87342054 16.78781924 17.01607931 17.39870777
 18.27585873 19.16633736 19.40590389 19.54762469 19.68410218 20.06732504
 20.36407933 20.75973742 20.76460549 21.44104094 22.07174939 22.38129346
 23.20169424 23.20910469]
[ 0.85257313  1.66819802  2.65297496  2.73000452  3.46912662  3.8310174
  4.02097839  4.46771545  5.37672572  5.82476043  6.36556721  6.38383418
  6.55513236  6.66201268  7.1477752   7.56552533  8.36564728  9.11891866
  9.23709191  9.80193048 10.47438962 11.46559815 11.50286326 11.74657181
 11.93671971 12.54891597 13.15134664 13.21411136 13.46153338 13.59052438
 13.9302085  14.79531101 14.90880766 15.12347951 15.42199382 16.06057375
 16.24926202 16.34005754 16.84460397 17.17892544 17.41367361 17.99882751
 18.00616602 18.68226338 19.54876471 20.36245963 21.12207845 21.85775351
 22.45610945 23.32516484]
[ 0.99669992  1.94997637  2.05122699  2.24897304  2.65481468  3.21610896
  3.27938405  4.04705548  4.40313228  5.22354389  5.31856216  5.62366001
  5.70489465  5.95866397  6.11854635  6.80528497  7.78800757  7.86902715
  7.89707659  8.88666398  9.49825738  9.70648408 10.6335502  10.75515104
 10.95629015 11.41877365 11.52604965 11.96559438 12.48079016 13.03120507
 13.37847275 13.53299489 13.83979427 14.247301   15.0073887  15.27416525
 16.08005367 16.2987524  17.07041015 17.60144131 18.51352228 19.42273523
 19.92139119 20.13142709 20.68480575 21.05513197 21.07488251 21.31039311
```

```
 21.72365081 22.2097548 ]
[ 0.06455453  0.83231856  1.45473189  1.77792447  1.79822182  2.6180532
  3.11952035  3.18842108  3.62565029  4.08131328  4.86091654  4.91536131
  5.31028215  5.87669733  6.80967975  7.67072052  7.90286612  8.47890406
  8.60695588  9.30720276  9.49054089  9.51004682 10.40341262 11.12646951
 11.73640757 12.37090347 12.55030581 12.67263675 12.86333399 13.44218667
 14.36057561 14.49517192 14.95045717 15.45739379 16.18658102 16.30124384
 16.53653075 17.085608   17.42962711 18.15863321 18.78988785 19.44530744
 19.54944163 19.8470267  20.50920063 21.2463822  21.72931735 21.79428857
 21.99037518 22.44639592]
[ 0.42404213  0.53262453  1.00156762  1.58956638  1.71950872  2.00866714
  2.44943151  3.19106488  3.4585654   4.06551705  4.72100519  4.92195233
  5.07628964  5.55069517  5.88318662  6.00420697  6.99814237  7.47155085
  7.76877281  7.94390603  8.78016263  9.49470049 10.15391946 10.30688402
 11.19033928 11.95531375 12.723167   13.48996161 14.06816912 14.22443364
 14.5091627  14.54938315 15.13276143 15.17584346 15.80791714 16.06748057
 16.8700088  17.7927692  18.20501542 18.58513891 19.54665553 20.33068236
 20.3466726  20.49627091 21.40258112 21.42885103 22.42581304 23.29908933
 24.22216049 24.3205139 ]
```

```
1 acumsum = np.cumsum(a9)
2 acumsum
```

```
3.88500794e+02, 3.88611997e+02, 3.89582410e+02, 3.90326927e+02,
3.90364709e+02, 3.90761741e+02, 3.91640138e+02, 3.92525055e+02,
3.92976071e+02, 3.93445493e+02, 3.94141449e+02, 3.94906440e+02,
3.95035994e+02, 3.95218163e+02, 3.95588216e+02, 3.95681748e+02,
3.96056183e+02, 3.96278530e+02, 3.96918859e+02, 3.97833258e+02,
3.98061518e+02, 3.98444147e+02, 3.99321298e+02, 4.00211776e+02,
4.00451343e+02, 4.00593064e+02, 4.00729541e+02, 4.01112764e+02,
4.01409518e+02, 4.01805176e+02, 4.01810044e+02, 4.02486480e+02,
4.03117188e+02, 4.03426732e+02, 4.04247133e+02, 4.04254544e+02,
4.05107117e+02, 4.05922742e+02, 4.06907519e+02, 4.06984548e+02,
4.07723670e+02, 4.08085561e+02, 4.08275522e+02, 4.08722259e+02,
4.09631269e+02, 4.10079304e+02, 4.10620111e+02, 4.10638378e+02,
4.10809676e+02, 4.10916556e+02, 4.11402319e+02, 4.11820069e+02,
4.12620191e+02, 4.13373462e+02, 4.13491636e+02, 4.14056474e+02,
4.14728933e+02, 4.15720142e+02, 4.15757407e+02, 4.16001115e+02,
4.16191263e+02, 4.16803460e+02, 4.17405890e+02, 4.17468655e+02,
4.17716077e+02, 4.17845068e+02, 4.18184752e+02, 4.19049855e+02,
4.19163351e+02, 4.19378023e+02, 4.19676537e+02, 4.20315117e+02,
4.20503806e+02, 4.20594601e+02, 4.21099148e+02, 4.21433469e+02,
4.21668217e+02, 4.22253371e+02, 4.22260710e+02, 4.22936807e+02,
4.23803308e+02, 4.24617003e+02, 4.25376622e+02, 4.26112297e+02,
4.26710653e+02, 4.27579708e+02, 4.28576408e+02, 4.29529685e+02,
4.29630935e+02, 4.29828682e+02, 4.30234523e+02, 4.30795817e+02,
4.30859093e+02, 4.31626764e+02, 4.31982841e+02, 4.32803252e+02,
4.32898271e+02, 4.33203368e+02, 4.33284603e+02, 4.33538372e+02,
4.33698255e+02, 4.34384993e+02, 4.35367716e+02, 4.35448736e+02,
4.35476785e+02, 4.36466372e+02, 4.37077966e+02, 4.37286193e+02,
4.38213259e+02, 4.38334860e+02, 4.38535999e+02, 4.38998482e+02,
4.39105758e+02, 4.39545303e+02, 4.40060499e+02, 4.40610914e+02,
4.40958181e+02, 4.41112703e+02, 4.41419503e+02, 4.41827009e+02,
4.42587097e+02, 4.42853874e+02, 4.43659762e+02, 4.43878461e+02,
4.44650119e+02, 4.45181150e+02, 4.46093231e+02, 4.47002444e+02,
4.47501100e+02, 4.47711136e+02, 4.48264514e+02, 4.48634840e+02,
4.48654591e+02, 4.48890102e+02, 4.49303359e+02, 4.49789463e+02,
4.49854018e+02, 4.50621782e+02, 4.51244195e+02, 4.51567388e+02,
4.51587685e+02, 4.52407516e+02, 4.52908984e+02, 4.52977884e+02,
4.53415114e+02, 4.53870777e+02, 4.54650380e+02, 4.54704825e+02,
4.55099745e+02, 4.55666161e+02, 4.56599143e+02, 4.57460184e+02,
```

```
       4.57692329e+02, 4.58268367e+02, 4.58396419e+02, 4.59096666e+02,
       4.59280004e+02, 4.59299510e+02, 4.60192876e+02, 4.60915933e+02,
       4.61525871e+02, 4.62160367e+02, 4.62339769e+02, 4.62462100e+02,
       4.62652797e+02, 4.63231650e+02, 4.64150039e+02, 4.64284635e+02,
       4.64739920e+02, 4.65246857e+02, 4.65976044e+02, 4.66090707e+02,
       4.66325994e+02, 4.66875071e+02, 4.67219090e+02, 4.67948096e+02,
       4.68579351e+02, 4.69234771e+02, 4.69338905e+02, 4.69636490e+02,
       4.70298664e+02, 4.71035845e+02, 4.71518781e+02, 4.71583752e+02,
       4.71779838e+02, 4.72235859e+02, 4.72659901e+02, 4.72768484e+02,
       4.73237427e+02, 4.73825426e+02, 4.73955368e+02, 4.74244526e+02,
       4.74685291e+02, 4.75426924e+02, 4.75694425e+02, 4.76301376e+02,
       4.76956864e+02, 4.77157812e+02, 4.77312149e+02, 4.77786554e+02,
       4.78119046e+02, 4.78240066e+02, 4.79234002e+02, 4.79707410e+02,
       4.80004632e+02, 4.80179765e+02, 4.81016022e+02, 4.81730560e+02,
       4.82389779e+02, 4.82542743e+02, 4.83426198e+02, 4.84191173e+02,
       4.84959026e+02, 4.85725821e+02, 4.86304028e+02, 4.86460293e+02,
       4.86745022e+02, 4.86785242e+02, 4.87368621e+02, 4.87411703e+02,
       4.88043776e+02, 4.88303340e+02, 4.89105868e+02, 4.90028628e+02,
       4.90440875e+02, 4.90820998e+02, 4.91782515e+02, 4.92566542e+02,
       4.92582532e+02, 4.92732130e+02, 4.93638440e+02, 4.93664710e+02,
       4.94661672e+02, 4.95534949e+02, 4.96458020e+02, 4.96556373e+02])
```

```
1 a11=np.random.randint(1,10,10) #10 is stop index and not included
2 a11
```

```
array([7, 5, 9, 1, 3, 3, 8, 8, 1, 9])
```

```
1 np.mean(a11)
```

```
5.4
```

```
1 np.median(a11)
```

```
6.0
```

```
1 a11=np.append(a11,[100,200])
2 a11
```

```
array([  7,   5,   9,   1,   3,   3,   8,   8,   1,   9, 100, 200])
```

```
1 np.mean(a11)
```

```
29.5
```

```
1 np.median(a11)
```

```
7.5
```

```
1 #observe change in mean but not much in median
```

```
1 a11
```

```
array([  7,   5,   9,   1,   3,   3,   8,   8,   1,   9, 100, 200])
```

Scaling and **Shifting**

```
a12 = 10*a11+2
a12
```

```
array([  72,   52,   92,   12,   32,   32,   82,   82,   12,   92, 1002,
       2002])
```

```
np.mean(a12)
```

```
297.0
```

```
10*np.mean(a11)+2
```

```
297.0
```

```
np.median(a12)
```

```
77.0
```

```
10*np.median(a11)+2
```

```
77.0
```

```
np.__version__
```

```
'1.21.5'
```

observe same

✓ 0s completed at 1:08 PM