Module10

October 4, 2020

0.1 Author: Joseph Vargovich

[1]:
```
#Import libraries
import pandas as pd
import numpy as np
```

1 Exercise 1&2 - Create a duniform function for the density function.

[8]:
```
#a. Duniform with simple input
def duniform(x, a=0,b=1):
    #Check the bounds of x per the density function
    if(a <= x and x <= b):
        return (1/(b-a))
    return 0

#x<a
print(duniform(1,2,3))

#a<x<b
print(duniform(3,2,4))

#b<x
print(duniform(6,7,3))

#b. Duniform with vector input modification
def duniformVec(x,a=0,b=1):
    output = []
    #Grab indicies that correspond to the length of x.
    for i in range(len(x)):
        #Check the bounds of x per the density function.
        if(a <= x[i] and x[i] <= b):
            output.append(1/(b-a))
        else:
            output.append(0)
    return output
```

```
#TestCode
x_vec = np.arange(-4,12,.001)
y_vec = duniformVec(x_vec, 4,8)

dataDict = {'x':x_vec, 'y':y_vec}
df = pd.DataFrame(data=dataDict)
#df.head gives predicted output.
print(df.head())
print(df.tail())
```

```
0
0.5
0
       x    y
0 -4.000  0.0
1 -3.999  0.0
2 -3.998  0.0
3 -3.997  0.0
4 -3.996  0.0
            x    y
15995  11.995  0.0
15996  11.996  0.0
15997  11.997  0.0
15998  11.998  0.0
15999  11.999  0.0
```

[5]:
```
#c. Benchmark the duniformVec function using the magic python command %timeit.
x_vec = np.arange(-4,12,.001)
%timeit duniformVec(x_vec, 4,8)
```

```
10.2 ms ± 111 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

2 Exercise 3 - Standardize Function

[53]:
```
def standardize(x):
    return(x-np.mean(x)/np.std(x))

#Small test list
test_list = [2,3,2]

standardize(test_vec)
```

[53]:
```
array([-2.94974747, -1.94974747, -2.94974747])
```

3 Exercise 4 - FizzBuzz function

[10]:
```
def fizzbuzz(n):
    output = []

    for i in range(1,n):
        if((i%3 == 0) and (i%5==0)):
            output.append("FizzBuzz")
        elif(i % 3 == 0):
            output.append("Fizz")
        elif(i % 5 == 0):
            output.append("Buzz")
        else:
            output.append(i)
    return output

fizzbuzz = fizzbuzz(21)
print(fizzbuzz)
```

```
[1, 2, &#39;Fizz&#39;, 4, &#39;Buzz&#39;, &#39;Fizz&#39;, 7, 8, &#39;Fizz&#39;,
&#39;Buzz&#39;, 11, &#39;Fizz&#39;, 13, 14, &#39;FizzBuzz&#39;, 16, 17,
&#39;Fizz&#39;, 19, &#39;Buzz&#39;]
```