PythonBasic

September 7, 2021

SL Lab with Python 1: Python Basic

Statistical Learning (Sejong University)

Date: 2021.09.06 (By: S.M. Riazul Islam)

0.0.1 List and its different operations

```
In [80]: # a is a variable
         a=10
         a
```

```
Out[80]: 10
```

```
In [74]: print("Hello World")
         print(a)
```

```
Hello World
10
```

```
In [63]: # a is a list
         a=[3,2,6]
         a
```

```
Out[63]: [3, 2, 6]
```

```
In [64]: a=['Alice', 'Bob', 3, 7]
         a
```

```
Out[64]: ['Alice', 'Bob', 3, 7]
```

```
In [65]: # a will be reapeated by 3 times
         # a is still a list
         a=3*a
         a
```

```
Out[65]: ['Alice', 'Bob', 3, 7, 'Alice', 'Bob', 3, 7, 'Alice', 'Bob', 3, 7]
```

```
In [79]: # range(n) creates a range from 0 to n-1
         a=list(range(10))
         print("a=",a)

a= [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]


In [84]: a=[5, 9, 4, 10, 12, 3]
         print("a=",a)
         a.pop(2)
         print("a=",a)

a= [5, 9, 4, 10, 12, 3]
a= [5, 9, 10, 12, 3]


In [85]: a=[5, 9, 4, 10, 12, 3]
         print("a=",a)
         a.pop(-2)
         print("a=",a)

a= [5, 9, 4, 10, 12, 3]
a= [5, 9, 4, 10, 3]
```

0.0.2 Convert a List to an array of numbers using numpy

```
In [91]: import numpy as np
         a=[5, 9, 4, 10, 12, 3]
         a=np.asarray(a)

         # each element of a will be multiplied by 3
         a=3*a
         a

Out[91]: array([15, 27, 12, 30, 36,  9])

In [92]: import numpy as np
         a=[5, 9, 4, 10, 12, 3]
         a=np.asarray(a)

         # each element of a will be multiplied by 3
         a=3*a
         print(a)

[15 27 12 30 36  9]


In [99]: # generating zeros of desired array size
         np.zeros([2,3])
```

```
Out[99]: array([[0., 0., 0.],
                [0., 0., 0.]])

In [100]: # generating ones of desired array size
          np.ones([2,3])

Out[100]: array([[1., 1., 1.],
                 [1., 1., 1.]])

In [111]: # generating empty of desired array size
          np.empty([2, 2])

Out[111]: array([[  0.   ,   0.   ],
                 [375.575, 252.34 ]])
```

0.0.3 Some numpy functions

```
In [112]: import numpy as np
          a=np.multiply(100,3)
          a

Out[112]: 300

In [113]: a=np.divide(50,3)
          a

Out[113]: 16.666666666666668

In [114]: a=np.remainder(50,3)
          a

Out[114]: 2

In [115]: start=1
          end=10
          a=np.arange(start,end)
          a

Out[115]: array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

0.0.4 1. Random Number Generaiton

0.0.5 2. Ploting

```
In [118]: import numpy as np

          # rand(m,n) generate numbers using uniform distribution of size m x n
          y=np.random.rand(3,4)
          y
```

```
Out[118]: array([[0.27990756, 0.0773715 , 0.78058437, 0.37839415],
                 [0.41411256, 0.11071086, 0.80025698, 0.01893182],
                 [0.54253484, 0.12734661, 0.24055626, 0.73186951]])
```

```
In [137]: import numpy as np

          # randn(m,n) generate numbers using normal distribution of size m x n
                       # with mean 0 and variance 1
          y=np.random.randn(3,4)
          y
```

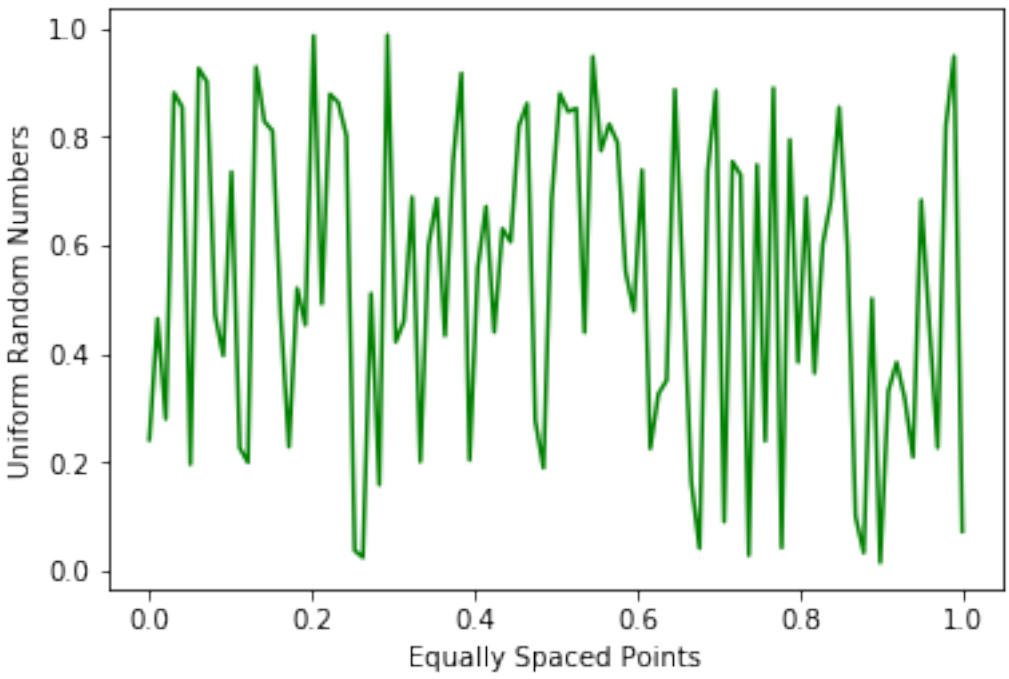
```
Out[137]: array([[ 0.59771804,  0.49347248,  0.08099923, -0.26802466],
                 [ 0.7604518 ,  0.75163271,  1.41581632, -0.73558167],
                 [-0.46068112, -0.82715078,  2.11634837,  1.89234816]])
```

```
In [148]: import numpy as np
          import matplotlib.pyplot as plt

          # generate 100 random numbers
          y=np.random.rand(100)
          x=np.linspace(0,1,num=100)
          plt.plot(x,y, color='green') #line plot

          plt.xlabel("Equally Spaced Points")
          plt.ylabel("Uniform Random Numbers")
```

```
Out[148]: Text(0, 0.5, 'Uniform Random Numbers')
```

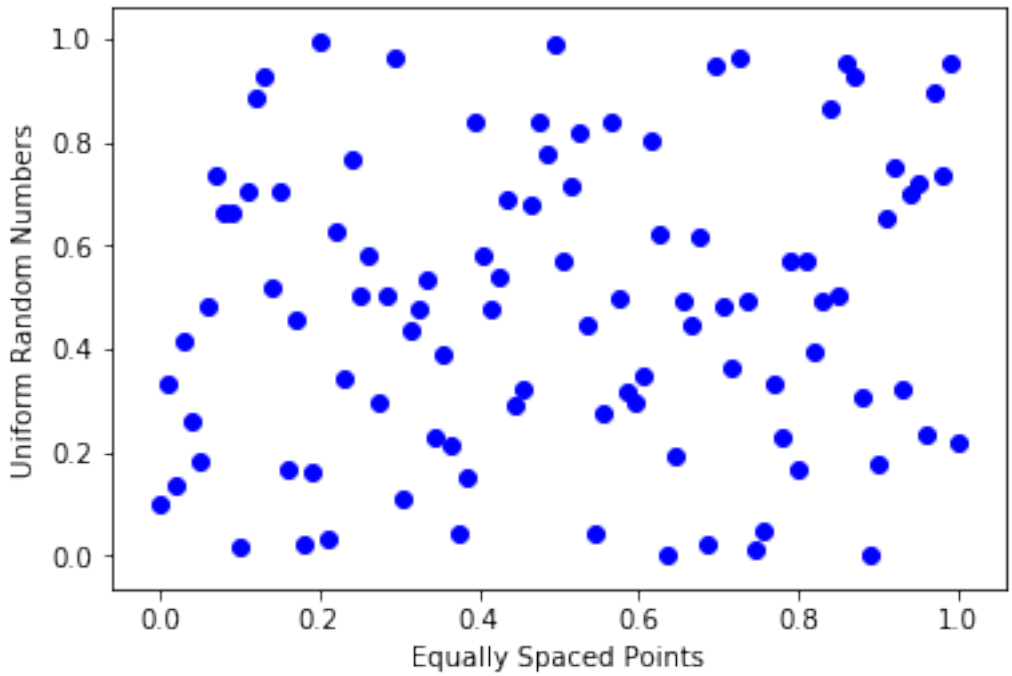


In [149]:
```
import numpy as np
import matplotlib.pyplot as plt

# generate 100 random numbers
y=np.random.rand(100)
x=np.linspace(0,1,num=100)
plt.scatter(x,y, color='blue')

plt.xlabel("Equally Spaced Points")
plt.ylabel("Uniform Random Numbers")
```

Out[149]:
```
Text(0, 0.5, 'Uniform Random Numbers')
```

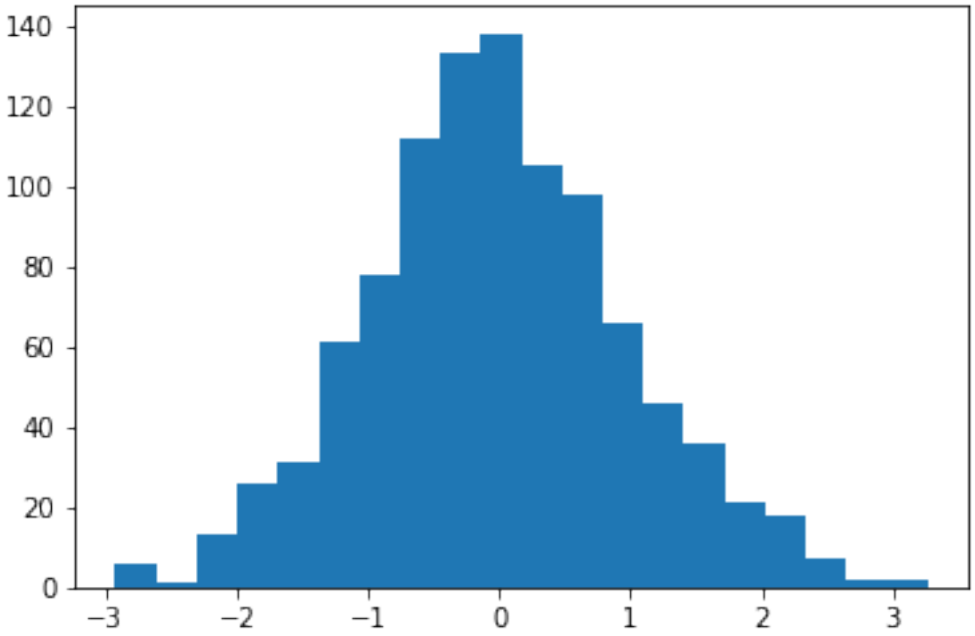


0.0.6 Histogram

In [162]:
```
import numpy as np
import matplotlib.pyplot as plt

# generate 1000 random numbers
y=np.random.randn(1000)

plt.hist(y, bins=20)
plt.show()
```



0.0.7 Pandas Data Frame

```
In [191]: import numpy as np
          import pandas as pd

          bpdata=pd.read_csv('blood_pressure.csv',delimiter=',')
              ##blood_pressure.csv in the same directory as of JupyterNotebook

          type(bpdata)
```

```
Out[191]: pandas.core.frame.DataFrame
```

```
In [168]: import numpy as np
          import pandas as pd
          bpdata=pd.read_csv('blood_pressure.csv',delimiter=',')

          bpdata
```

```
Out[168]:      patient     sex agegrp  bp_before  bp_after
          0          1    Male  30-45        143       153
          1          2    Male  30-45        163       170
          2          3    Male  30-45        153       168
          3          4    Male  30-45        153       142
          4          5    Male  30-45        146       141
          5          6    Male  30-45        150       147
          6          7    Male  30-45        148       133
```

7	8	Male	30-45	153	141
8	9	Male	30-45	153	131
9	10	Male	30-45	158	125
10	11	Male	30-45	149	164
11	12	Male	30-45	173	159
12	13	Male	30-45	165	135
13	14	Male	30-45	145	159
14	15	Male	30-45	143	153
15	16	Male	30-45	152	126
16	17	Male	30-45	141	162
17	18	Male	30-45	176	134
18	19	Male	30-45	143	136
19	20	Male	30-45	162	150
20	21	Male	46-59	149	168
21	22	Male	46-59	156	155
22	23	Male	46-59	151	136
23	24	Male	46-59	159	132
24	25	Male	46-59	164	160
25	26	Male	46-59	154	160
26	27	Male	46-59	152	136
27	28	Male	46-59	142	183
28	29	Male	46-59	162	152
29	30	Male	46-59	155	162
..	...	...	...	...	...
90	91	Female	46-59	142	145
91	92	Female	46-59	162	132
92	93	Female	46-59	144	157
93	94	Female	46-59	142	140
94	95	Female	46-59	159	137
95	96	Female	46-59	140	154
96	97	Female	46-59	144	169
97	98	Female	46-59	142	145
98	99	Female	46-59	145	137
99	100	Female	46-59	145	143
100	101	Female	60+	168	178
101	102	Female	60+	142	141
102	103	Female	60+	147	149
103	104	Female	60+	148	148
104	105	Female	60+	162	138
105	106	Female	60+	170	143
106	107	Female	60+	173	167
107	108	Female	60+	151	158
108	109	Female	60+	155	152
109	110	Female	60+	163	154
110	111	Female	60+	183	161
111	112	Female	60+	159	143
112	113	Female	60+	148	159
113	114	Female	60+	151	177

```
114      115  Female    60+        165       142
115      116  Female    60+        152       152
116      117  Female    60+        161       152
117      118  Female    60+        165       174
118      119  Female    60+        149       151
119      120  Female    60+        185       163

[120 rows x 5 columns]
```

In [192]:
```
import numpy as np
import pandas as pd

bpdata=pd.read_csv('blood_pressure.csv',delimiter=',')

bpdata.shape
```

Out[192]:
```
(120, 5)
```

In [174]:
```
bpdata.head()
```

Out[174]:

	patient	sex	agegrp	bp_before	bp_after
0	1	Male	30-45	143	153
1	2	Male	30-45	163	170
2	3	Male	30-45	153	168
3	4	Male	30-45	153	142
4	5	Male	30-45	146	141

In [173]:
```
bpdata.tail()
```

Out[173]:

	patient	sex	agegrp	bp_before	bp_after
115	116	Female	60+	152	152
116	117	Female	60+	161	152
117	118	Female	60+	165	174
118	119	Female	60+	149	151
119	120	Female	60+	185	163

In [172]:
```
bpdata.sample(5)
```

Out[172]:

	patient	sex	agegrp	bp_before	bp_after
76	77	Female	30-45	142	149
33	34	Male	46-59	148	184
59	60	Male	60+	162	164
71	72	Female	30-45	157	135
14	15	Male	30-45	143	153

In [186]:
```
# Data Filtering
bpdata[['patient','bp_before']]
   # outerbrace indicates column selections; inner brace indicates list
```

Out[186]:

	patient	bp_before
0	1	143
1	2	163
2	3	153
3	4	153
4	5	146
5	6	150
6	7	148
7	8	153
8	9	153
9	10	158
10	11	149
11	12	173
12	13	165
13	14	145
14	15	143
15	16	152
16	17	141
17	18	176
18	19	143
19	20	162
20	21	149
21	22	156
22	23	151
23	24	159
24	25	164
25	26	154
26	27	152
27	28	142
28	29	162
29	30	155
..	...	...
90	91	142
91	92	162
92	93	144
93	94	142
94	95	159
95	96	140
96	97	144
97	98	142
98	99	145
99	100	145
100	101	168
101	102	142
102	103	147
103	104	148
104	105	162
105	106	170

```
106      107        173
107      108        151
108      109        155
109      110        163
110      111        183
111      112        159
112      113        148
113      114        151
114      115        165
115      116        152
116      117        161
117      118        165
118      119        149
119      120        185

[120 rows x 2 columns]
```

In [189]:
```
### Records with a specific value of a particular column
bpdata[bpdata.bp_before==165]
```

Out[189]:

	patient	sex	agegrp	bp_before	bp_after
12	13	Male	30-45	165	135
114	115	Female	60+	165	142
117	118	Female	60+	165	174