

```
1 # -----
2 # -----Probability-----
3 # -----
4 import matplotlib.pyplot as plt
5 import numpy as np
6 import math
7 import random
8 # -----
9 random_number = random.random()
10 print(random_number)
11
12 outcome = random.randint(1,6)
13 print(outcome)
14
15 a1=[ random.randint(1, 6) for _ in range(10) ]
16 print(a1)
17
18 print(random.sample(range(1, 50), 6))
19
20 a = np.random.random((1000))
21 # -----
22
23 plt.figure(1)
24 plt.hist(a, bins=10)
25
26 mu, sigma = 0, 0.1
27 s = np.random.normal(mu, sigma, 1000)
28
29 plt.figure(2)
30 plt.hist(s, bins=10,histtype='bar')
31
32
33 s1 = np.random.randn( 1000)
34
35 plt.figure(3)
```

```

36 plt.hist(s, bins=10, histtype='bar')
37 plt.show()
38 # -----
39 # -----Linear Algebra -----
40 # -----
41
42 V1 = np.array([1, 2])
43 print(V1)
44 print(V1.shape)
45
46 V2 = np.array([1, 2])
47 print(V2)
48 print(V2.shape)
49
50 V3 = np.dot(V1, V2)
51
52 V4 = np.multiply(V1, V2)
53 print(V4)
54
55 V5 = np.arange(15).reshape(3, 5)
56 print(V5)
57 print(V5.ndim)
58
59 V6 = np.array([[1, 2, 3, 4, 5],
60               [1, 2, 3, 4, 5],
61               [1, 2, 3, 4, 5]])
62
63 V7 = np.dot(V5, np.transpose(V6))
64 V8 = np.matmul(V5, np.transpose(V6))
65
66 V9 = np.array([1, 2, 3, 4])
67 V10 = V9.reshape((-1, 1)) # Trick for 1 dim
68
69 V11 = np.array([(1, 2, 3, 4, 5),
70                (1, 2, 3, 4, 5),

```

```

71         (1, 2, 3, 4, 5)])
72
73 print(V7)
74
75 # -----
76 # -----Symbolic -----
77 # -----
78
79 from sympy import *
80 x = Symbol('x')
81 y = Symbol('y')
82
83 f1 = (x+y)**2
84 print(f1)
85
86 f2 = expand((x+y)**3)
87 print(f2)
88
89 f3 = simplify((x+x*y)/x)
90 print(f3)
91
92
93 l1 = limit(x, x, oo)
94 print(l1)
95
96 l2 = limit(1/x, x, oo)
97 print(l2)
98
99 l3 = limit(x**x, x, 0)
100 print(l3)
101
102 l4 = limit((tan(x+y)-tan(x))/y, y, 0)
103 print(l4)
104
105

```

```
106 d1 = diff(sin(x), x)
107 print(d1)
108
109 d2 = diff(sin(2*x), x)
110 print(d2)
111
112
113
114 s1 = series(cos(x), x)
115 print(s1)
116
117 s2 = series(1/cos(x), x)
118 print(s2)
119
120 i1 = integrate(6*x**5, x)
121 print(i1)
122
123 i2 = integrate(sin(x), x)
124 print(i2)
125
126 i3 = integrate(2*x + sinh(x), x)
127 print(i3)
128
129 i4 = integrate(x**3, (x, -1, 1))
130 print(i4)
131
132 i5= integrate(sin(x), (x, 0, pi/2))
133 print(i5)
134
135
136 i6 = integrate(cos(x), (x, -pi/2, pi/2))
137 print(i6)
138
139 e1 = solve(x**4 - 1, x)
140 print(e1)
```

```

141
142 e2 = solve([x + 5*y - 2, -3*x + 6*y - 15], [x, y])
143 print(e2)
144
145 f = x**4 - 3*x**2 + 1
146 factor(f)
147
148
149
150 from sympy import Matrix
151
152 M1 = Matrix([[1,0], [0,1]])
153 print(M1)
154
155 M2 = Matrix([[1,x], [y,1]])
156 print(M2)
157 # -----
158 from mpl_toolkits.mplot3d import Axes3D
159 import matplotlib.pyplot as plt
160 from matplotlib import cm
161 from matplotlib.ticker import LinearLocator, FormatStrFormatter
162 import numpy as np
163
164
165 fig = plt.figure(4)
166 ax = fig.gca(projection='3d')
167
168 # Make data.
169 X = np.arange(-5, 5, 0.25)
170 Y = np.arange(-5, 5, 0.25)
171
172 X, Y = np.meshgrid(X, Y)
173 R = np.sqrt(X**2 + Y**2)
174
175 Z = np.sin(R)

```

```
176
177 surf = ax.plot_surface(X, Y, Z)
178
179 # Plot the surface.
180 # surf = ax.plot_surface(X, Y, Z, cmap=cm.coolwarm, linewidth=0, antialiased=False)
181
182 # Customize the z axis.
183 # ax.set_zlim(-1.01, 1.01)
184 # ax.zaxis.set_major_locator(LinearLocator(10))
185 # ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))
186
187 # Add a color bar which maps values to colors.
188 # fig.colorbar(surf, shrink=0.5, aspect=5)
189
190 plt.show()
```