Practical – 2

AIM: Explore the Random Variables, Univariate and Multivariate Normal Distributions, Descriptive Statistics, and Hypothesis Testing using Python

1. Random Variables import numpy as np

```
random_variable = np.random.rand(100)
print("Random Variable:", random_variable)
```

```
Random Variable: [0.06380448 0.59020767 0.66474769 0.11790748 0.22987845 0.64626655
 0.938138   0.10194318 0.75908575 0.99951166 0.96513029 0.03465352
 0.14940903 0.80231296 0.53298142 0.69614891 0.23816422 0.10213476
 0.06684644 0.64609646 0.44736487 0.82485683 0.07259918 0.70368676
 0.86010423 0.08781243 0.62730749 0.87674351 0.77244861 0.83355742
 0.85696083 0.96588027 0.77428931 0.91029352 0.42266541 0.40130517
 0.81063256 0.44090719 0.41459294 0.84576235 0.64550751 0.79826394
 0.93726865 0.21655639 0.55159881 0.08458086 0.61417858 0.8108137
 0.09879896 0.30797176 0.90836762 0.03619116 0.08966586 0.6101461
 0.37188961 0.22042983 0.08383529 0.51067538 0.89690735 0.35770396
 0.69532292 0.28188322 0.6454643  0.75523812 0.85886162 0.80882393
 0.07715398 0.12110956 0.88602038 0.70352896 0.71719565 0.51796347
 0.63434627 0.47891096 0.90058203 0.11924369 0.15910594 0.76905355
 0.06320305 0.97102904 0.6591784  0.32480025 0.08423413 0.22515219
 0.33947508 0.00713564 0.23509546 0.12126302 0.27547031 0.40101137
 0.07994783 0.55730725 0.85173397 0.355009   0.47642514 0.38154936
 0.60675092 0.67939219 0.88574374 0.31636476]
```

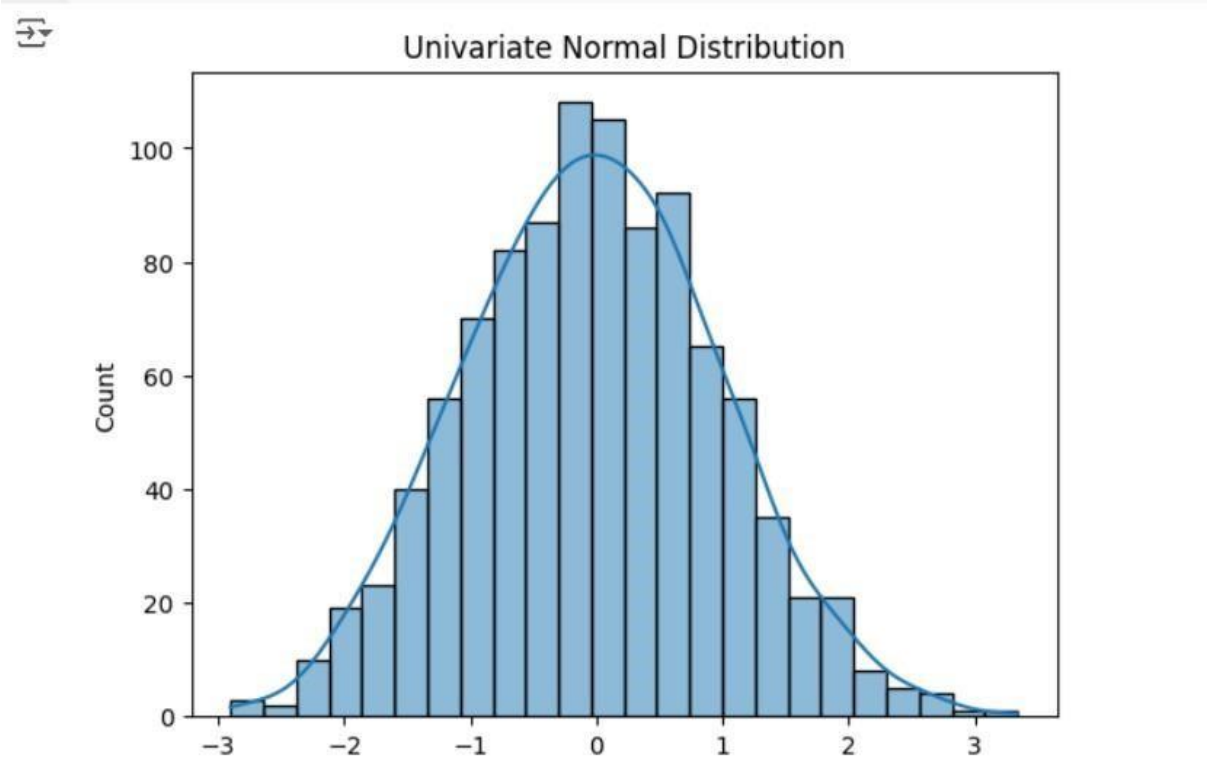
2. Univariate Normal Distribution

```
import numpy as np
import matplotlib.pyplot as plt import seaborn as sns

univariate_normal = np.random.normal(loc=0, scale=1, size=1000)

sns.histplot(univariate_normal, kde=True)
plt.title('Univariate Normal Distribution') plt.show()
```

Name: DHRUV SHERE
Enrollment No: 23012022021
Batch: 5IT-B-2

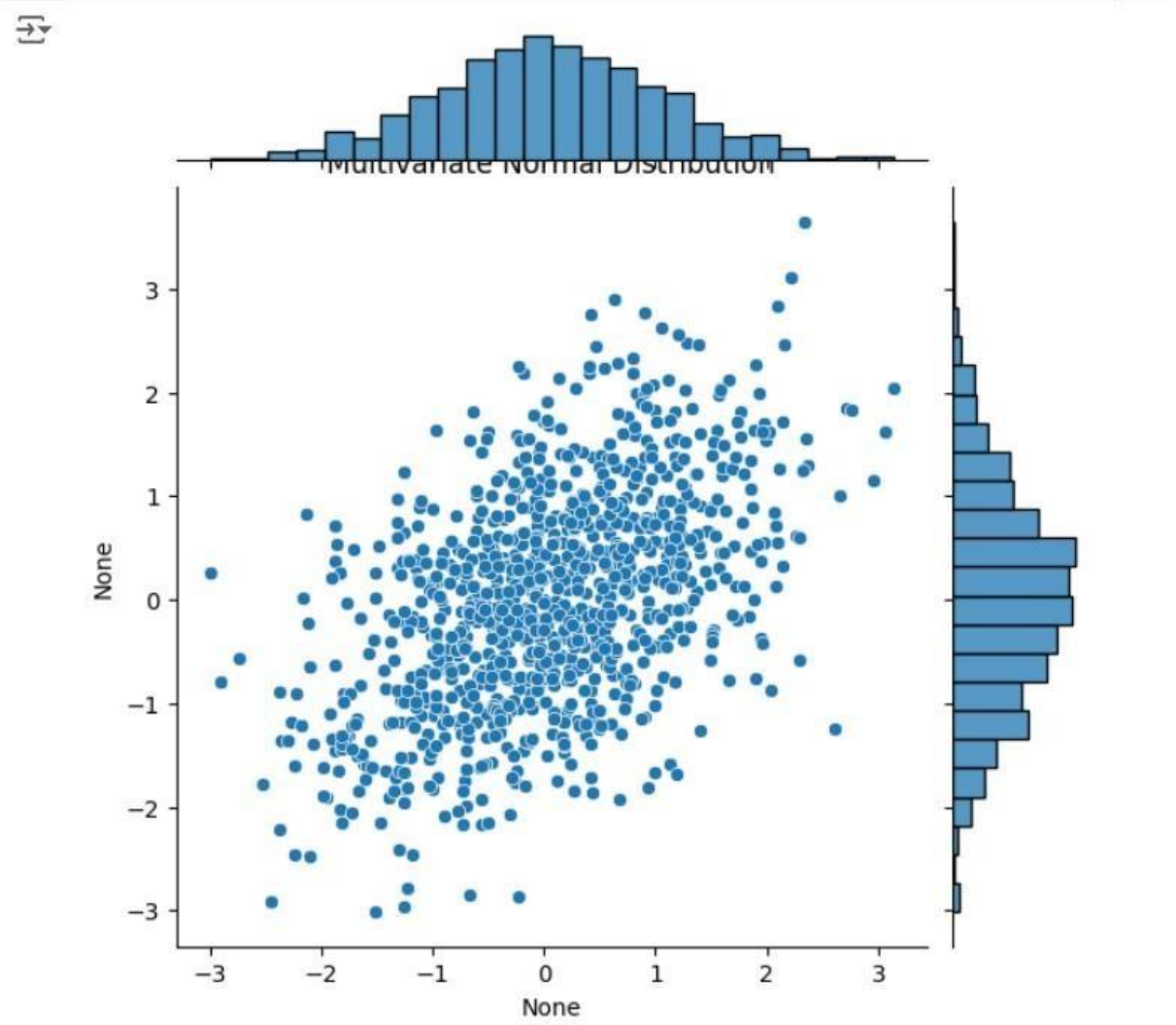


3. Multivariate Normal Distribution in a 2D design

```
import numpy as np import
matplotlib.pyplot as plt
import seaborn as sns

mean = [0, 0]
cov = [[1, 0.5], [0.5, 1]] multivariate_normal =
np.random.multivariate_normal(mean, cov, 1000)

sns.jointplot(x=multivariate_normal[:, 0], y=multivariate_normal[:, 1], kind="scatter")
plt.title('Multivariate Normal Distribution') plt.show()
```



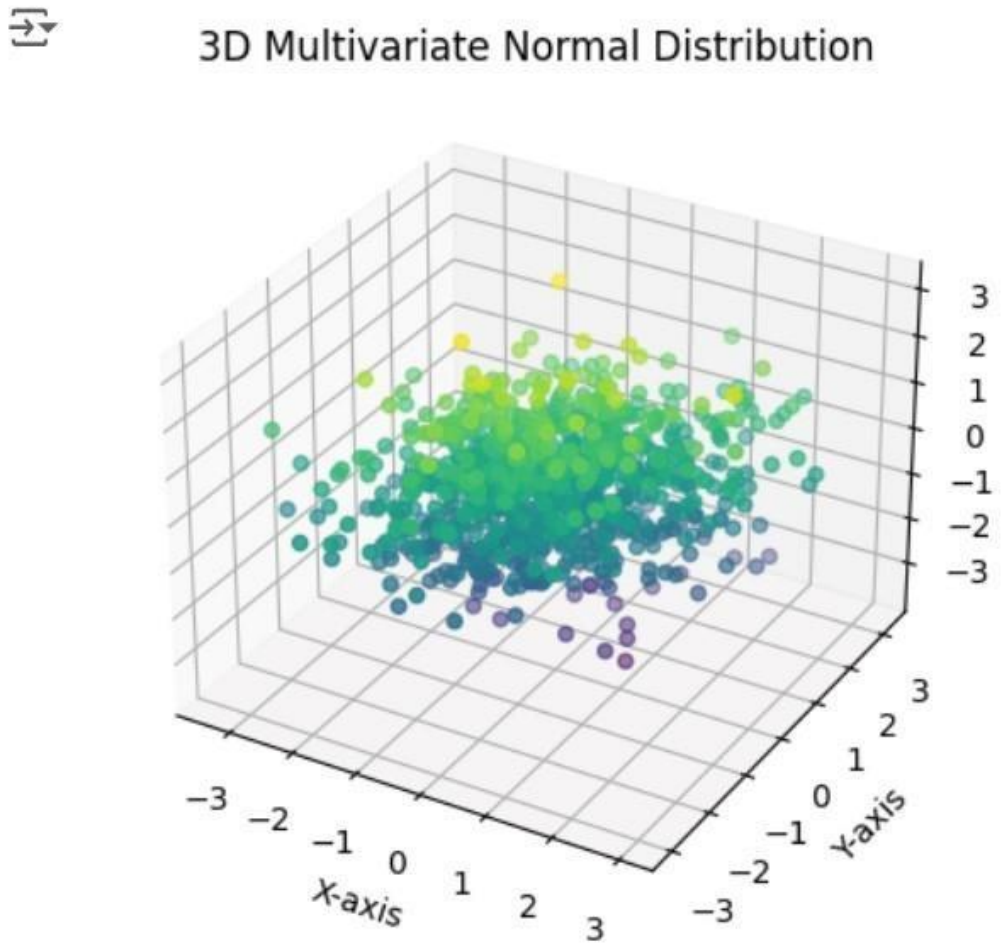
Multivariate Normal Distribution in a 3D design

```
import numpy as np
import matplotlib.pyplot as plt from
mpl_toolkits.mplot3d import Axes3D

mean = [0, 0]
cov = [[1, 0.5], [0.5, 1]] multivariate_normal =
np.random.multivariate_normal(mean, cov, 1000)

fig = plt.figure() ax =
fig.add_subplot(111, projection='3d')
```

```
x = multivariate_normal[:, 0] y =
multivariate_normal[:, 1] z = np.random.normal(loc=0,
scale=1, size=1000) ax.scatter(x, y, z, c=z,
cmap='viridis', marker='o') ax.set_title('3D Multivariate
Normal Distribution') ax.set_xlabel('X-axis')
ax.set_ylabel('Y-axis') ax.set_zlabel('Z-axis')
plt.show()
```



4. Descriptive Statistics import pandas as pd

```
univariate_normal = np.random.normal(loc=0, scale=1, size=1000) df =
pd.DataFrame(univariate_normal, columns=['Univariate_Normal'])

descriptive_stats = df.describe()
print(descriptive_stats)
```

	Univariate_Normal
count	1000.000000
mean	-0.019291
std	1.028989
min	-3.020734
25%	-0.734920
50%	-0.058924
75%	0.675070
max	2.937378

5. Hypothesis Testing

from scipy import stats
import numpy as np sample1 = np.random.normal(loc=0,

scale=1, size=50) sample2 = np.random.normal(loc=0.5,

scale=1, size=50) t_stat, p_value = stats.ttest_ind(sample1,

sample2) print(f'Tstatistic: {t_stat}, P-value: {p_value}')

```
T-statistic: -2.3885704066753153, P-value: 0.018829444505743788
```