NORMALIZATION AND STANDARDIZATION

NORMALIZAITION MEANS SCALING DOWN VALUES BETWEEN 0 TO 1

STANDARDIZATION MEANS SCALING DOWN VALUES ACCORDING TO STANDARD NORMAL DISTRIBUTION WHERE MEAN=0 AND STANDARD DEVIATION = 1 (MEANS HIGHLY CORRELATED)

In [1]:

```
import pandas as pd
DF=pd.read_csv('housing.csv')
```

In [2]:

```
DF
```

Out[2]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Addre
0	79545.45857	5.682861	7.009188	4.09	23086.80050	1.059034e+06	208 Michael Ferry Ap 674\nLaurabury, N 3701
1	79248.64245	6.002900	6.730821	3.09	40173.07217	1.505891e+06	188 Johnson View Suite 079\nLak Kathleen, CA
2	61287.06718	5.865890	8.512727	5.13	36882.15940	1.058988e+06	9127 Elizabet Stravenue\nDanieltow WI 06482
3	63345.24005	7.188236	5.586729	3.26	34310.24283	1.260617e+06	USS Barnett\nFPO A 4482
4	59982.19723	5.040555	7.839388	4.23	26354.10947	6.309435e+05	USNS Raymond\nFP AE 0938
...	...	...	...	...	...	...	
4995	60567.94414	7.830362	6.137356	3.46	22837.36103	1.060194e+06	USNS Williams\nFP AP 30153-765
4996	78491.27543	6.999135	6.576763	4.02	25616.11549	1.482618e+06	PSC 9258, Bo 8489\nAPO AA 4299 335
4997	63390.68689	7.250591	4.805081	2.13	33266.14549	1.030730e+06	4215 Tracy Garde Suite 076\nJoshualan VA 01
4998	68001.33124	5.534388	7.130144	5.44	42625.62016	1.198657e+06	USS Wallace\nFPO A 7331
4999	65510.58180	5.992305	6.792336	4.07	46501.28380	1.298950e+06	37778 George Ridge Apt. 509\nEast Holl NV 2

5000 rows × 7 columns

In [3]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [4]:
```
normalization= MinMaxScaler()
```

In [5]:
```
DF.columns
```

Out[5]:
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

In [6]:
```
nofanr=pd.DataFrame(normalization.fit_transform(DF[['Avg. Area House Age','Avg. Are
```

In [7]:

```
nofanr#nofanr= normalization of area and room
```

Out[7]:

```
             0         1
   0  0.441986  0.501502
   1  0.488538  0.464501
   2  0.468609  0.701350
   3  0.660956  0.312430
   4  0.348556  0.611851
 ...       ...       ...
4995  0.754359  0.385619
4996  0.633450  0.444024
4997  0.670026  0.208534
4998  0.420389  0.517579
4999  0.486997  0.472678

5000 rows × 2 columns
```

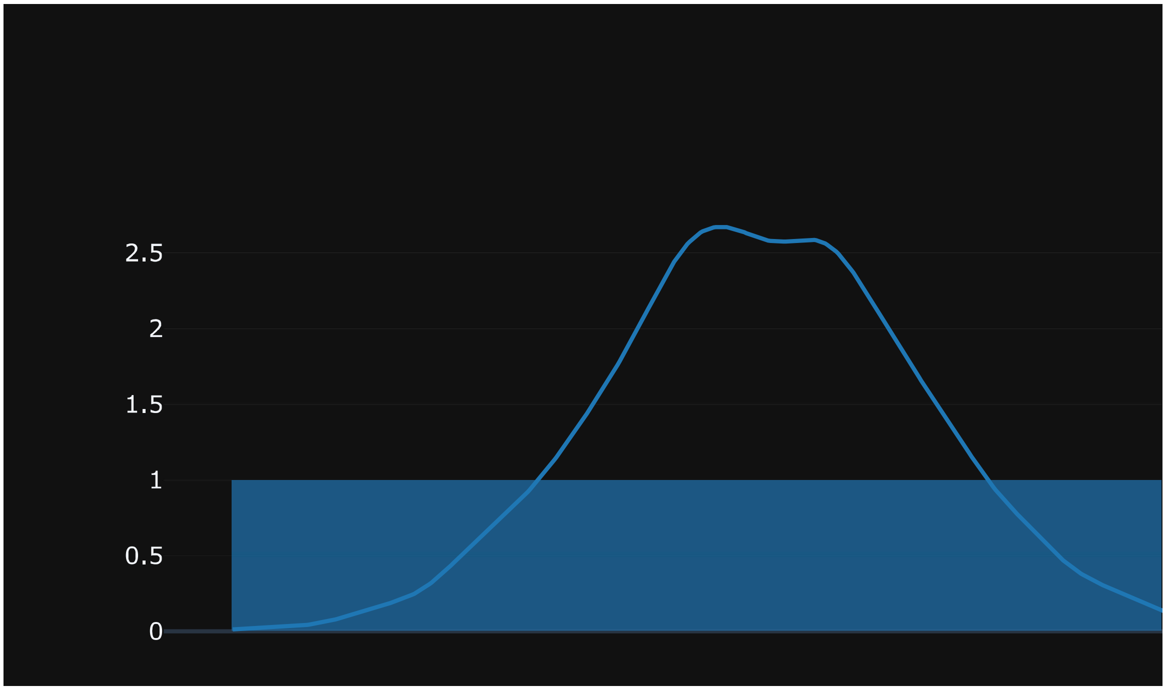
In [8]:

```
import plotly.figure_factory as ff
import plotly.graph_objects as go
import plotly.express as px
import numpy as np

x = nofanr[0]
hist_data = [x]
group_labels = ['Avg. Area House Age'] # name of the dataset

mean = np.mean(x)
stdev_pluss = np.std(x)
stdev_minus = np.std(x)*-1

fig = ff.create_distplot(hist_data, group_labels, curve_type='kde')
fig.update_layout(template = 'plotly_dark')
fig.show()
```



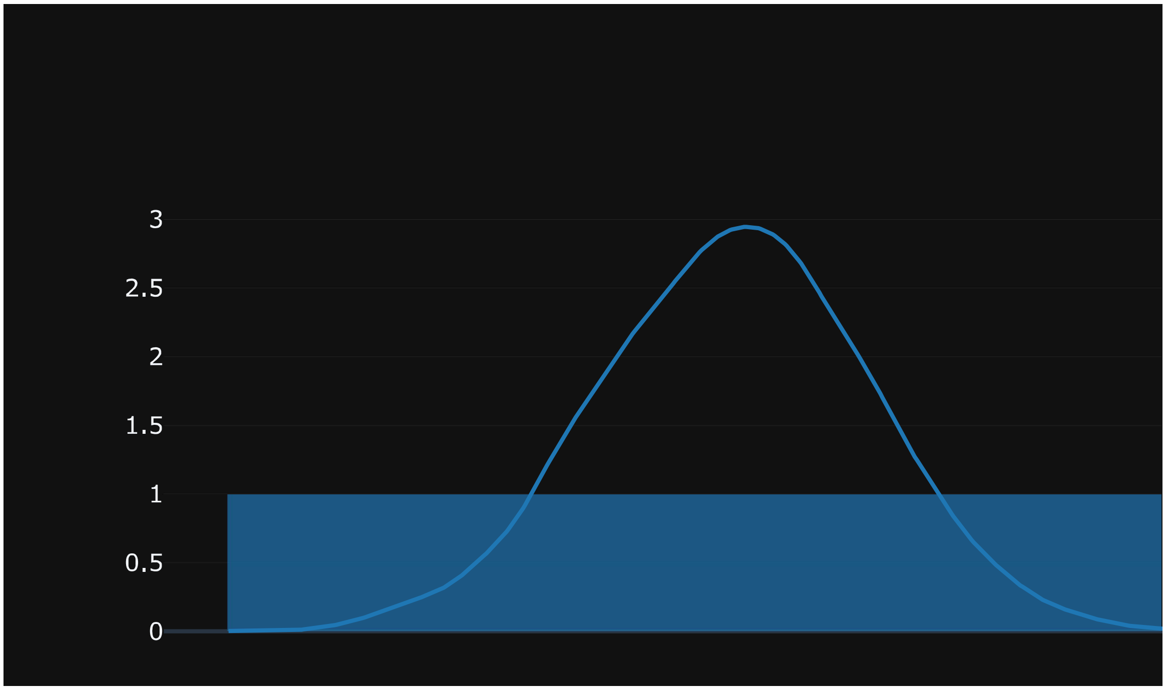
In [9]:

```
import plotly.figure_factory as ff
import plotly.graph_objects as go
import plotly.express as px
import numpy as np

x = nofanr[1]
hist_data = [x]
group_labels = ['Avg. Area Number of Rooms'] # name of the dataset

mean = np.mean(x)
stdev_pluss = np.std(x)
stdev_minus = np.std(x)*-1

fig = ff.create_distplot(hist_data, group_labels, curve_type='kde')
fig.update_layout(template = 'plotly_dark')
fig.show()
```



```
In [10]: #in above plots we can understand that our maximum values lies between 0.15 to 8.5
```

```
In [ ]:
```

Standarization

here all features will transform in a way that it will have properties of standard normal distribution where mean=0 and standard deviation=1

```
In [13]: from sklearn.preprocessing import StandardScaler
```

```
In [14]: Standardization= StandardScaler()
```

```
In [15]: SNDofanr=pd.DataFrame(Standardization.fit_transform(DF[['Avg. Area House Age','Avg.
```

In [16]:

```
SNDofanr
```

Out[16]:

	0	1
0	-0.296927	0.021274
1	0.025902	-0.255506
2	-0.112303	1.516243
3	1.221572	-1.393077
4	-0.944834	0.846742
...	...	...
4995	1.869297	-0.845588
4996	1.030822	-0.408686
4997	1.284470	-2.170269
4998	-0.446694	0.141541
4999	0.015215	-0.194342

5000 rows × 2 columns

In [17]:

```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\acer\appdata\local\programs\pyth
on\python310\lib\site-packages (5.14.1)
Requirement already satisfied: packaging in c:\users\acer\appdata\local\programs\p
ython\python310\lib\site-packages (from plotly) (22.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\acer\appdata\local\prog
rams\python\python310\lib\site-packages (from plotly) (8.2.2)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

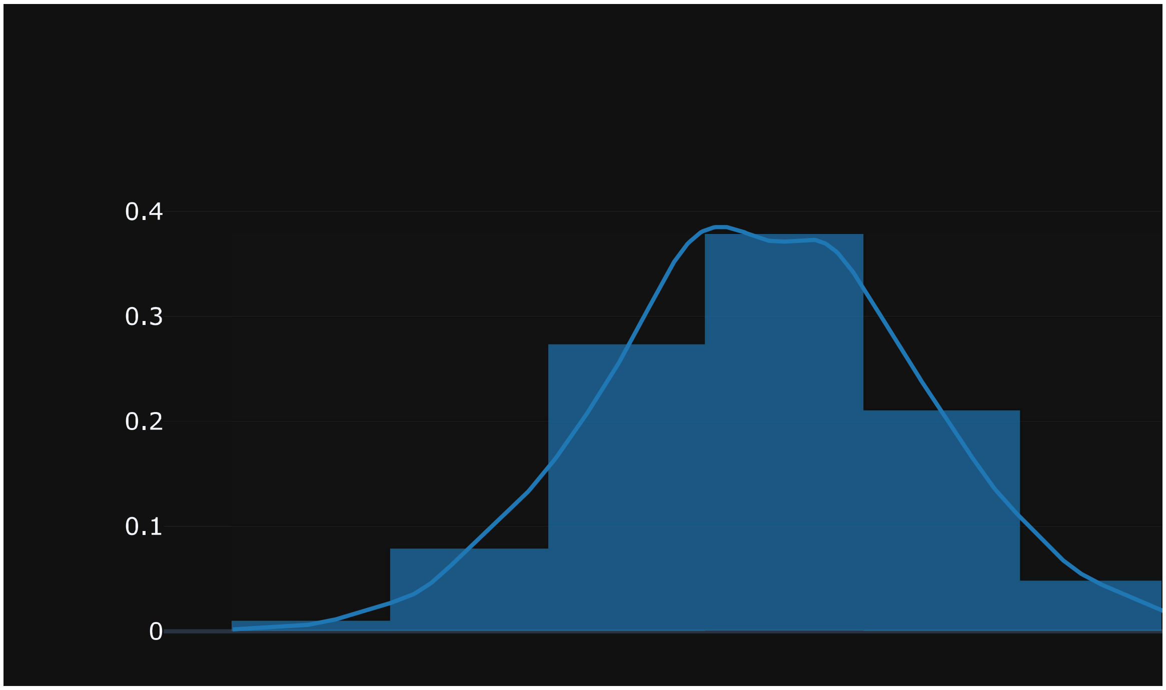
In [18]:

```
import plotly.figure_factory as ff
import plotly.graph_objects as go
import plotly.express as px
import numpy as np

x = SNDofanr[0]
hist_data = [x]
group_labels = ['Avg. Area House Age'] # name of the dataset

mean = np.mean(x)
stdev_pluss = np.std(x)
stdev_minus = np.std(x)*-1

fig = ff.create_distplot(hist_data, group_labels, curve_type='kde')
fig.update_layout(template = 'plotly_dark')
fig.show()
```



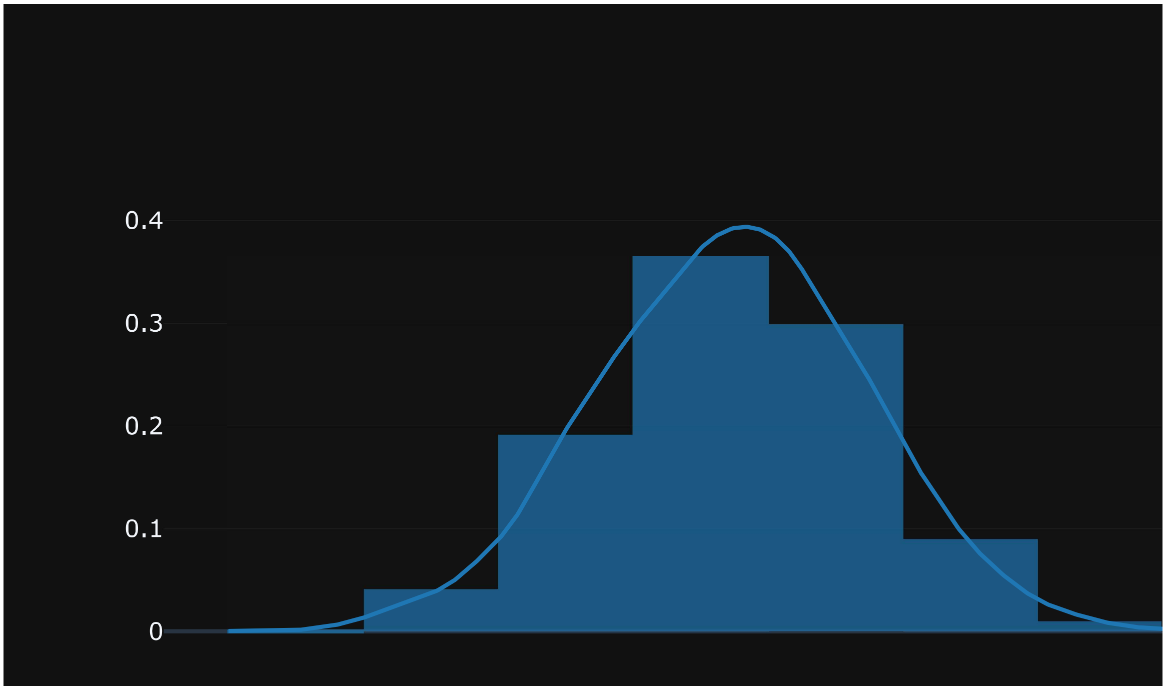
In [20]:

```
import plotly.figure_factory as ff
import plotly.graph_objects as go
import plotly.express as px
import numpy as np

x = SNDofanr[1]
hist_data = [x]
group_labels = ['Avg. Area Number of Rooms'] # name of the dataset

mean = np.mean(x)
stdev_pluss = np.std(x)
stdev_minus = np.std(x)*-1

fig = ff.create_distplot(hist_data, group_labels, curve_type='kde')
fig.update_layout(template = 'plotly_dark')
fig.show()
```



by visualizing above 2 graphs we can understand that the are some values which are beyon 3 standard deviation avay from mean and will be considered as outlier according to empirical that 99.7% values lies between +-3 standard deviation from the mean

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```