In [3]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
%matplotlib inline
```

In [4]:

```
np.random.seed(100)
```

In [5]:

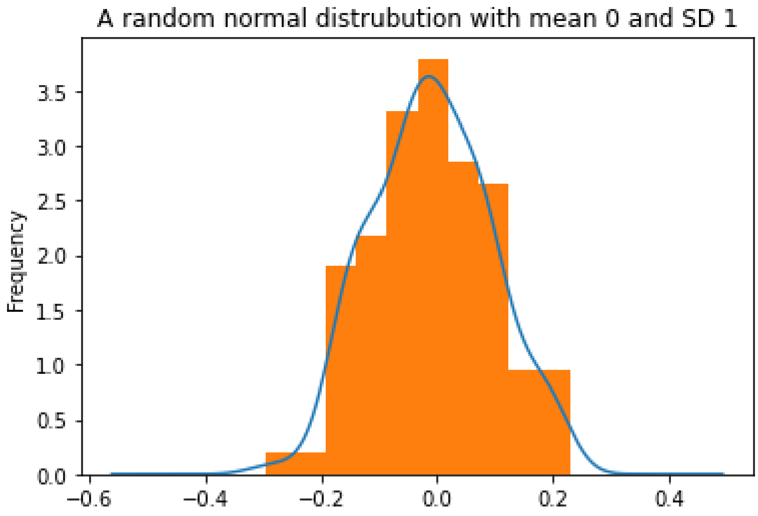
```
# Generate 200 random normal data points with mean=0, standard_deviation=0.1

random_normal_datapoints = pd.Series(np.random.normal(0, 0.1, 200))
```

In [6]:

```
# Plot the data points along with its KDE to see how it looks
fig, ax = plt.subplots()
random_normal_datapoints.plot.kde(ax=ax, legend=False, title='A random normal distru
random_normal_datapoints.plot.hist(density=True, ax=ax)
ax.set_ylabel('Frequency')
```

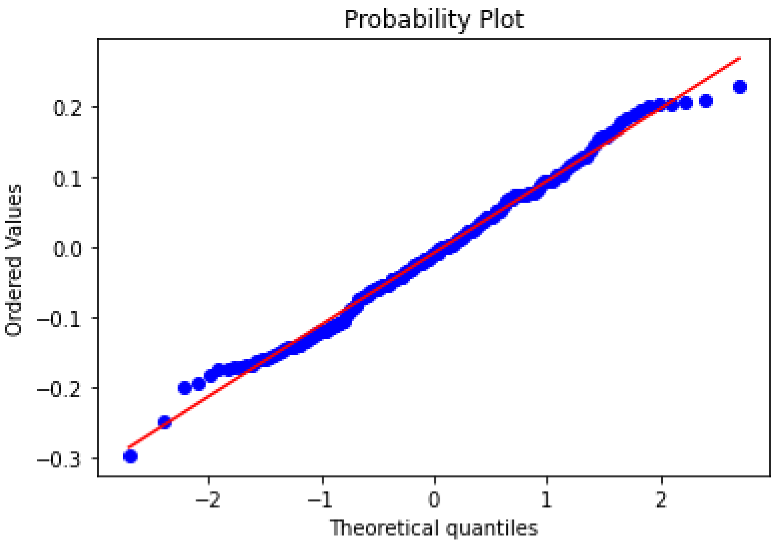
Out[6]:

```
Text(0, 0.5, 'Frequency')
```



In [7]:

```
# Plot the Q-Q plot to graphically check for the hypothesis

res = stats.probplot(random_normal_datapoints, plot=plt)
plt.show()
```

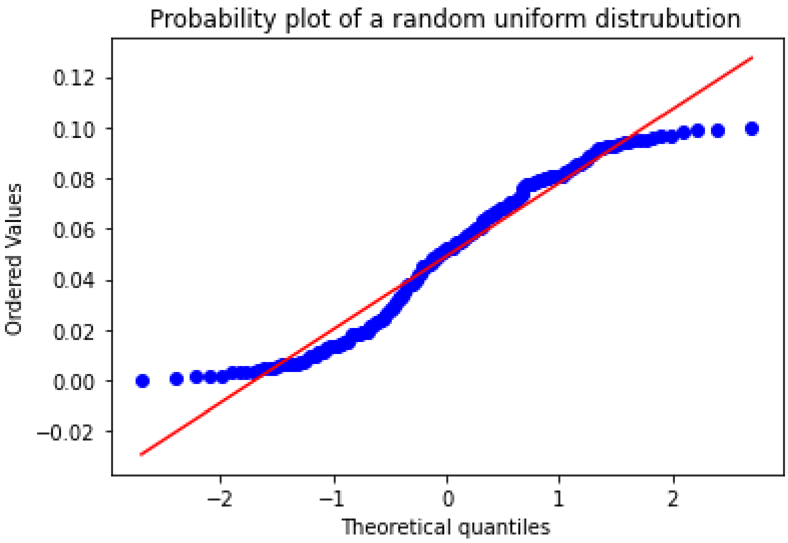


In [8]:

```
# Generate random uniform distribution data points

random_uniform_datapoints = pd.Series(np.random.uniform(0, 0.1, 200))
```

In [9]:

```
# Plot the Q-Q plot to graphically check for the hypothesis

res = stats.probplot(random_uniform_datapoints, plot=plt)
plt.title('Probability plot of a random uniform distrubution')
plt.show()
```

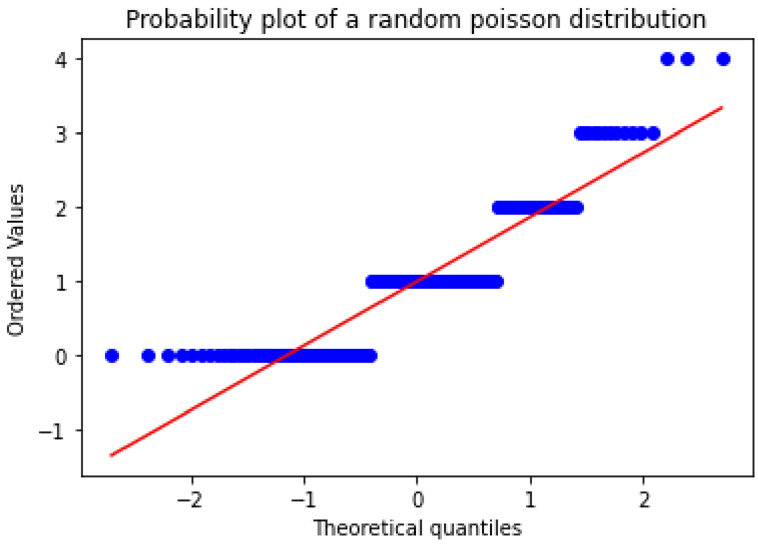


In [10]:

```
# Generate random poisson distribution data points with mean=1

random_poisson_datapoints = pd.Series(np.random.poisson(1, 200))
```

In [11]:

```
# Plot the Q-Q plot to graphically check for the hypothesis

res = stats.probplot(random_poisson_datapoints, plot=plt)
plt.title('Probability plot of a random poisson distribution')
plt.show()
```



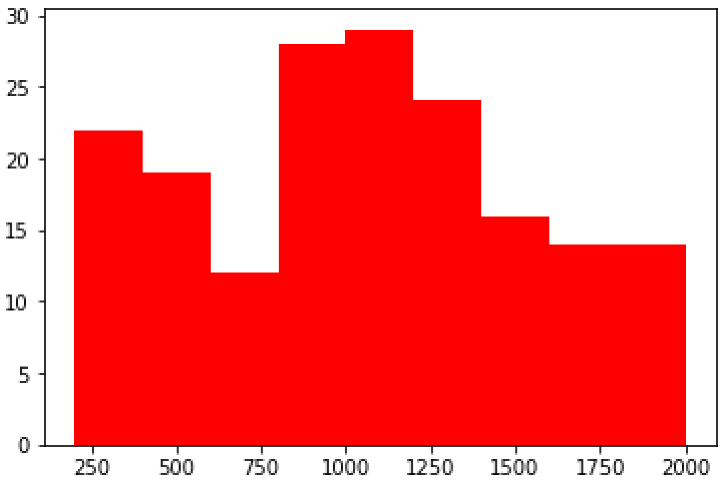
In [12]:

```
import random
randomlist = random.sample(range(10, 2000), 200) #Generate 200 random numbers betwee
print(randomlist)
```

```
[1204, 1474, 382, 1157, 885, 1203, 507, 1799, 503, 741, 1017, 852, 190, 1128, 1249,
1168, 891, 368, 180, 1288, 630, 1473, 347, 1049, 1485, 1194, 942, 189, 657, 1976, 71
3, 1072, 572, 235, 575, 926, 223, 1510, 595, 1255, 731, 1190, 954, 1596, 244, 1567,
829, 1185, 171, 1197, 987, 229, 1152, 546, 371, 918, 1147, 1011, 1644, 341, 1294, 11
31, 1970, 1012, 1649, 1419, 1250, 403, 337, 46, 934, 94, 1318, 849, 950, 628, 923, 1
188, 298, 57, 270, 28, 70, 1950, 989, 534, 910, 1426, 15, 79, 665, 260, 1221, 1685,
1388, 245, 1135, 1665, 498, 584, 271, 744, 1298, 1732, 1987, 1634, 234, 1303, 1960,
1230, 1938, 1415, 1477, 499, 1023, 616, 804, 1385, 832, 1353, 164, 1332, 1071, 1781,
1831, 870, 1361, 909, 1902, 531, 1481, 932, 367, 697, 1340, 1466, 1820, 566, 809, 10
34, 1961, 1801, 1421, 487, 1068, 1010, 289, 928, 313, 1112, 82, 468, 1295, 183, 154,
1813, 1603, 811, 1326, 1654, 1545, 1454, 586, 1199, 283, 169, 139, 1054, 684, 908, 9
00, 758, 1136, 981, 1325, 562, 1093, 308, 436, 1195, 18, 1959, 1872, 1713, 316, 164
3, 137, 551, 935, 1178, 945, 1741, 1238, 1691, 132, 1406, 89, 148, 1300, 1279]
```

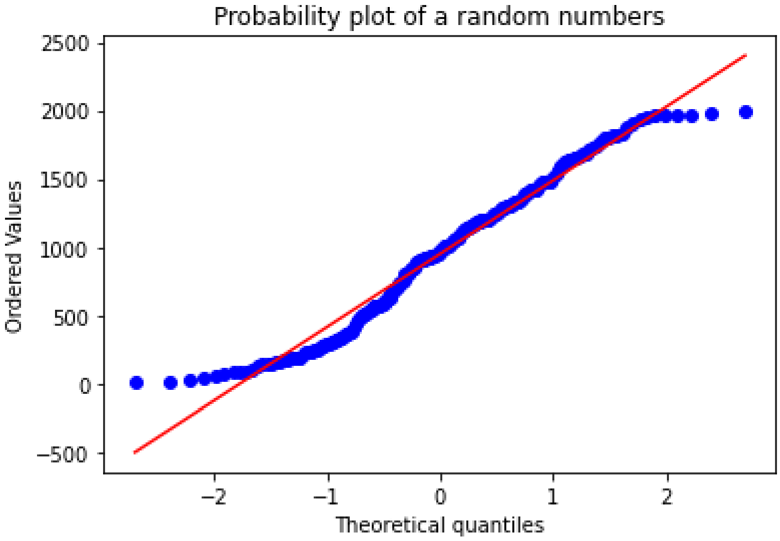
In [13]:

```
n_bins = [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000] #bin values
bin_heights, bins, patches = plt.hist(randomlist, n_bins, facecolor='red')
plt.show()
```



In [15]:

```
res = stats.probplot(randomlist, plot=plt)
plt.title('Probability plot of a random numbers')
plt.show()
```


Probability plot of a random numbers
Ordered Values
2500
2000
1500
1000
500
0
−500
−2
−1
0
1
2
Theoretical quantiles

In [ ]: