Given n i.i.d. points sampled according to $f(r)$ inside a circle with radius R, the expected density, denoted by $\rho(r)$ is given as

$$\rho(r) = \frac{n\int_0^r f(x)dx}{\pi r^2}, \forall r \in (0, R]$$

, where $\int_0^r f(x)dx = F(r) = P(r \leq R)$.

To sample points distributing evenly inside the circle, this requires $\rho(r) = \frac{n}{\pi R^2}, \forall r \in (0, R]$. Hence, we have

$$\begin{align} \frac{n\int_0^r f(x)dx}{\pi r^2} &= \frac{n}{\pi R^2} \tag{1} \\ \int_0^r f(x)dx &= \frac{r^2}{R^2} \tag{2} \\ \frac{d}{dr}\int_0^r f(x)dx &= \frac{d}{dr}\frac{r^2}{R^2} \tag{3} \\ f(r) &= \frac{2r}{R^2} \tag{4} \end{align}$$

Thus, the probabilty density function of r is

$$f(r) = \begin{cases} \frac{2r}{R^2} & \forall r \in (0, R] \\ 0 & \text{otherwise} \end{cases}$$

, and the cumulative density function is

$$F(r) = \begin{cases} 0 & r \leq 0 \\ \frac{r^2}{R^2} & r \in (0, R] \\ 1 & r > R \end{cases}$$

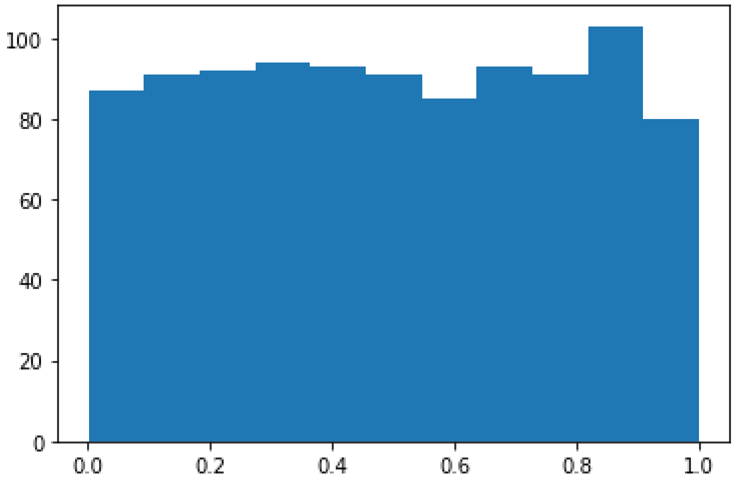
Sample points X ~ f(r) with the inverse of its cumulative density function and Y ~ Uni(0,1)

$$F^{-1}(r) = \sqrt{rR^2}, \forall r \in (0, 1]$$

In [77]:
```
import numpy as np
import math
import matplotlib.pyplot as plt
```

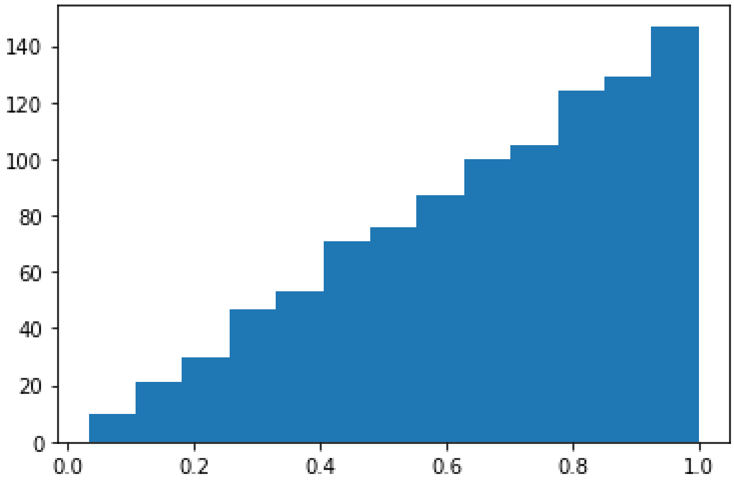
In [78]:
```
x_uni = np.random.rand(1000)
plt.hist(x_uni, bins='auto')
```

Out[78]:
```
(array([ 87.,  91.,  92.,  94.,  93.,  91.,  85.,  93.,  91., 103.,  80.]),
 array([0.00112273, 0.09190225, 0.18268177, 0.27346129, 0.36424082,
        0.45502034, 0.54579986, 0.63657939, 0.72735891, 0.81813843,
        0.90891796, 0.99969748]),
 <BarContainer object of 11 artists>)
```



In [79]:
```
r = np.sqrt(x_uni) # assume R = 1 for simplicity
plt.hist(r, bins='auto')
```

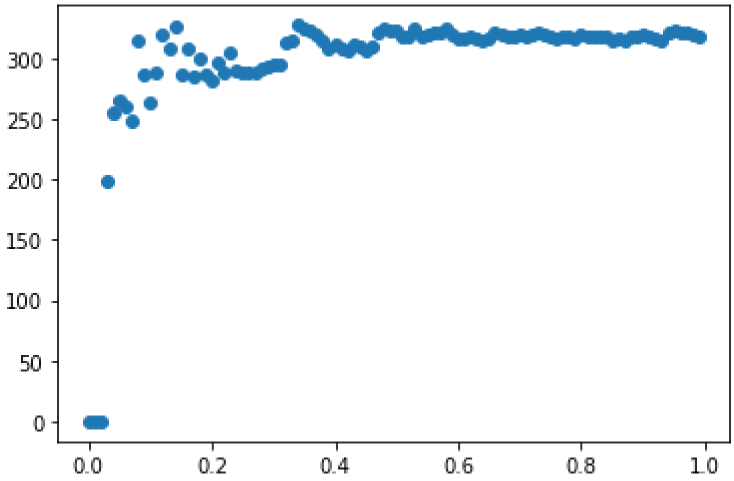
Out[79]:
```
(array([ 10.,  21.,  30.,  47.,  53.,  71.,  76.,  87., 100., 105., 124.,
        129., 147.]),
 array([0.03350709, 0.10784106, 0.18217503, 0.25650901, 0.33084298,
        0.40517695, 0.47951092, 0.55384489, 0.62817887, 0.70251284,
        0.77684681, 0.85118078, 0.92551476, 0.99984873]),
 <BarContainer object of 13 artists>)
```



In [80]:
```
# calculate the empricial density based on the sample at different radius
n_bin = 100
density = [None] * n_bin
for i in range(n_bin):
    density[i] = sum(r < (i+1)/n_bin ) / (math.pi * ((i+1)/n_bin)**2)
```

In [81]:
```
x = np.array(range(n_bin)) / n_bin
plt.scatter(x, density)
```

Out[81]:
```
<matplotlib.collections.PathCollection at 0x7f8b79f861f0>
```



Theorectical density is

$$\rho = \frac{n}{\pi R^2} = \frac{1000}{\pi} = 318.3098861837907$$

In [ ]:

In [ ]: