L03s01

September 13, 2018

1 Basic Monte Carlo

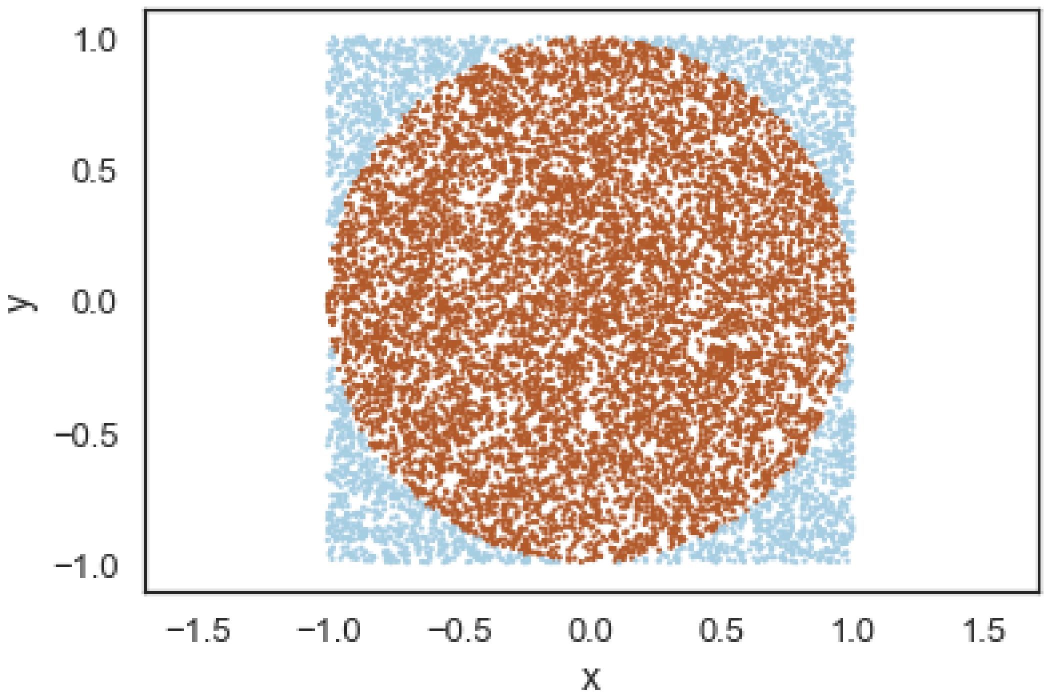
In [1]:
```
%matplotlib inline
import numpy as np
from scipy import stats
from scipy.stats import norm
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('white')
sns.set_context('talk')
```

1.1 Estimate the area of a unit circle

In [2]:
```
#area of the bounding box
box_area = 4.0

#number of samples
N_total = 10000

#drawing random points uniform between -1 and 1
X = np.random.uniform(low=-1, high=1, size=N_total)
Y = np.random.uniform(low=-1, high=1, size=N_total)

# calculate the distance of the point from the center
distance = np.sqrt(X**2+Y**2);

# check if point is inside the circle
is_point_inside = distance<1.0

# sum up the hits inside the circle
N_inside=np.sum(is_point_inside)

# estimate the circle area
circle_area = box_area * N_inside/N_total

# some nice visualization
plt.scatter(X,Y, c=is_point_inside, s=5.0, edgecolors='none', cmap=plt.cm.Paired)
```

```python
plt.axis('equal')
plt.xlabel('x')
plt.ylabel('y')

# text output
print("Area of the circle = ", circle_area)
print("pi = ", np.pi)
```

```
Area of the circle =  3.1448
pi =  3.141592653589793
```



General method of MC

Let $X \sim \phi$ and $\mathbb{E}[X] = \mu$ and $SD(X) = \sigma > 0$, where μ and σ are unknowns.

Our goeal is to eistmiate μ.

In crude MC, suppose computer can generate iid replicates $(X_i : i = 1, 2, ...)$, then we take the average of the first N replicates as its estimate:

$$\mu_N = \frac{1}{N} \sum_{i=1}^{N} X_i \to \mu, \text{ as } N \to \infty.$$

The above convergence is guranteed by LLN. In other words, if we set the error as

$$e_N = \mu_N - \mu,$$

we have $e_N \to 0$ as $N \to \infty$, which is the most desired property of the estimator.