IP[y]: Notebook (/tree/) mcmc

File Edit View Insert Cell Kernel Help

New
Open...
Make a Copy
Rename...
Save and Checkpoint
Revert to Checkpoint
Print Preview
Download as
Trusted Notebook
Close and halt

Markdown Cell Toolbar: None

In [1]:
```
from __future__ import division, print_function
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Monte Carlo Simulations

Calculating Pi

In [ ]:
```
def random_number_plusminus1(n):
    return 2*np.random.random(n) - 1

x, y = random_number_plusminus1((2,1000))
```

In [ ]:
```
plt.scatter(x, y)
plt.show()
```

In [ ]:
```
area_of_square = 2*2
ratio_of_dart_inside = np.mean(x**2 + y**2 < 1)
pi_estimate = area_of_square * ratio_of_dart_inside
print(pi_estimate, np.pi)
```

In [ ]:
```
x, y = random_number_plusminus1((2,10000000))
area_of_square = 2*2
ratio_of_dart_inside = np.mean(x**2 + y**2 < 1)
pi_estimate = area_of_square * ratio_of_dart_inside
print(pi_estimate, np.pi)
```

Calculating an integral

In [ ]:
```
def f(x):
    return np.log(2*x) # Integral from 1 to 10 is 20.264
x = np.linspace(1,10,1000)
plt.plot(x, f(x))
plt.show()
```

In [ ]:
```
n = 1000
x_draw = 1 + 9*np.random.random(n)
y_draw = 3.5 * np.random.random(n)
```