# 第5-2章 PyMC介绍

中山大学人工智能学院
毛旭东

Email：maoxd3@mail.sysu.edu.cn

# PyMC简介

- PyMC是开源项目，用于贝叶斯统计建模和概率机器学习，它聚焦于马尔可夫链蒙特卡洛和变分推理算法。
- PyMC已经在很多领域中被用于解决推理问题，包括天文学、流行病学、分子生物学、晶体学、化学、生态学和心理学等。

# PyMC安装与使用

- 安装:

```
conda create -c conda-forge -n pymc_env "pymc>=5”
conda activate pymc_env
```

- 使用:

```
import pymc as pm
```

# 例子：抛硬币

回顾Metropolis-Hasting算法：

- 在右图的算法中，只需要“告诉”Metropolis算法$P(\theta)$的计算方式，其他的都是Metropolis算法的“事”。

- 因此，对PyMC等概率编程库来说，需要能够“自动求解”$P(\theta)$的表达式。

初始化$\theta^0$，满足$P(\theta^0) \neq 0$

for $t = 1,2,3,\ldots$ do

$\theta = \theta^{(t-1)}$

采样 $\theta^* \sim q(\theta^*|\theta)$

计算 $\alpha = \frac{P(\theta^*)q(\theta|\theta^*)}{P(\theta)q(\theta^*|\theta)}$

计算 $A = \min(1, \alpha)$

采样 $u \sim U(0,1)$

if $u \leq A$, 接受$\theta^*$, 即$\theta^t = \theta^*$

if $u > A$, 拒绝$\theta^*$, 即$\theta^t = \theta$

# 流程图

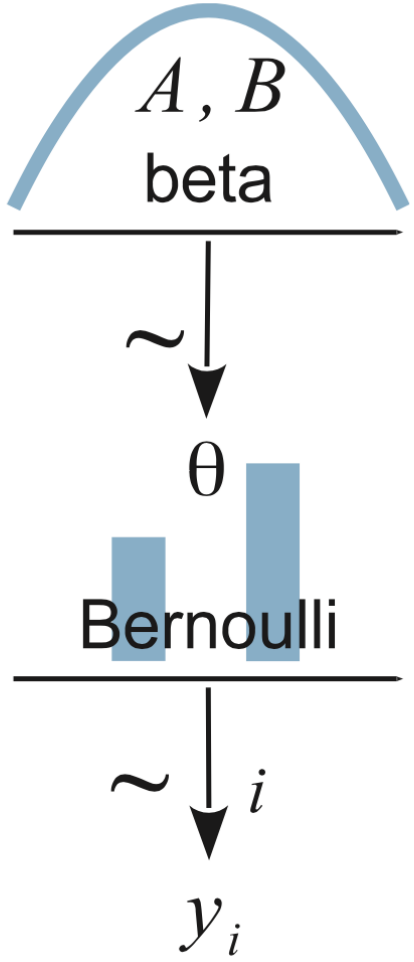
- 先验：

$$\theta \sim \mathrm{Beta}(\theta|\mathrm{a}_0, \mathrm{b}_0)$$

- 似然：

$$y \sim \mathrm{Bern}(y|\theta)$$

- 右图的流程图对应于上述两个式子。

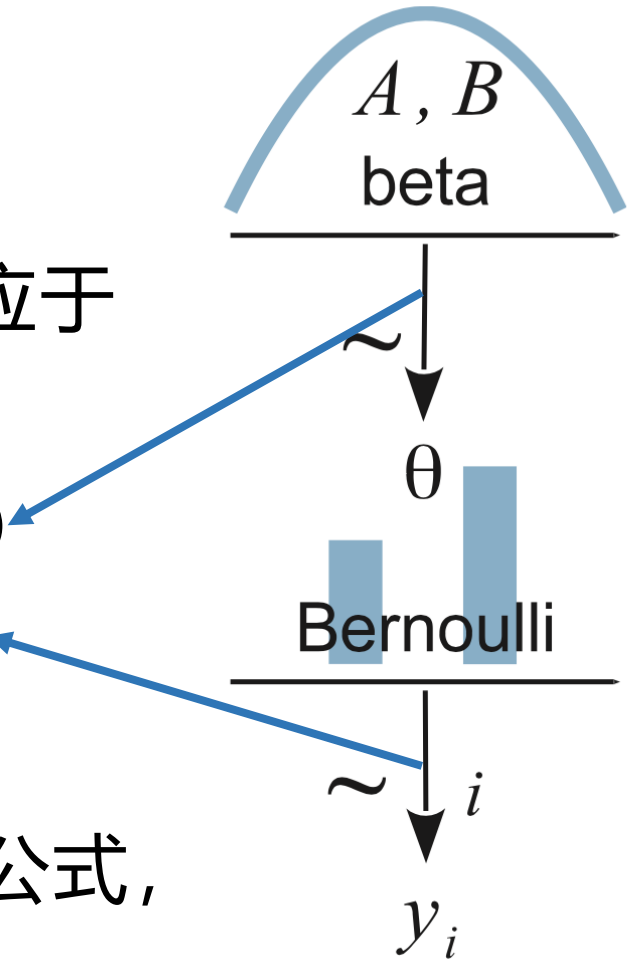



- 对于流程图的逻辑，应自下而上观察：
  - 先有数据$y$，再有似然函数，再有参数，再有先验。

# 流程图

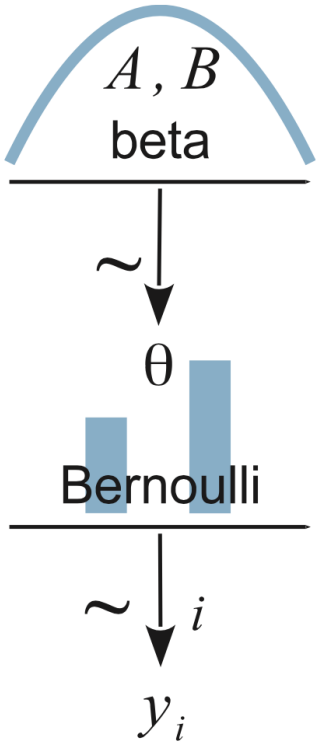
- 流程图的重要性：

1. 清晰地展示了参数和数据间的依赖关系。
2. 对于用PyMC实现很有帮助，一个箭头对应于一行代码。
   - theta = pm.Beta('theta', alpha=1., beta=1.)
   - y = pm.Bernoulli('y', p=theta, observed=y)




- PyMC会根据流程图来自动求解$P(\theta)$的计算公式，然后用Metropolis等MCMC方法来求解后验。

# 代码实现




```
# 11 heads and 3 tails
y = np.array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0])

with pm.Model() as model:
    # define the prior
    theta = pm.Beta('theta', alpha=1., beta=1.)
    # define the likelihood
    y_observed = pm.Bernoulli('y_observed', p=theta, observed=y)

    # Generate a MCMC chain
    trace = pm.sample(1000)
```

# 代码实现

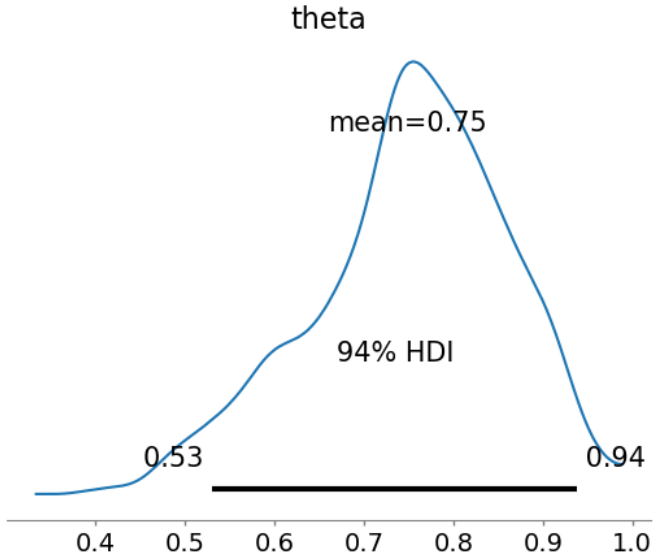
```
# 11 heads and 3 tails
y = np.array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0])

with pm.Model() as model:
    # define the prior
    theta = pm.Beta('theta', alpha=1., beta=1.)
    # define the likelihood
    y_observed = pm.Bernoulli('y_observed', p=theta, observed=y)

    # Generate a MCMC chain
    trace = pm.sample(1000)
```

- with pm.Model() as model:
  - 创建了一个Model对象
- theta = pm.Beta('theta', alpha=1., beta=1.)
  - 创建了一个随机变量，值是从Beta分布中采样。
  - 第一个参数是变量名，一般要和等号左边的变量名一致，方便用于输出日志信息。
- y_observed = pm.Bernoulli('y_ observed ', p=theta, observed=y)
  - 创建了一个观测的随机变量，代表似然。
- trace = pm.sample(1000)
  - 采样1000个样本

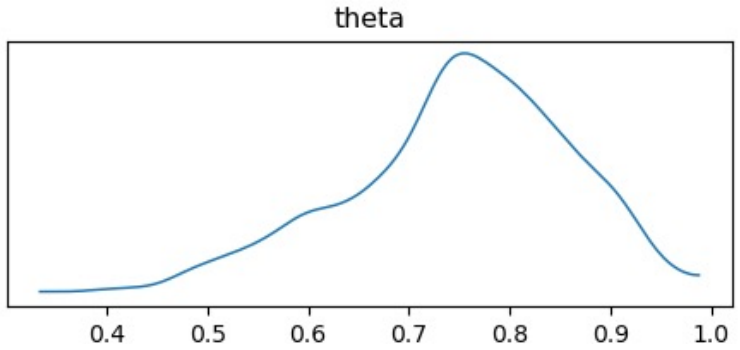
# 结果分析

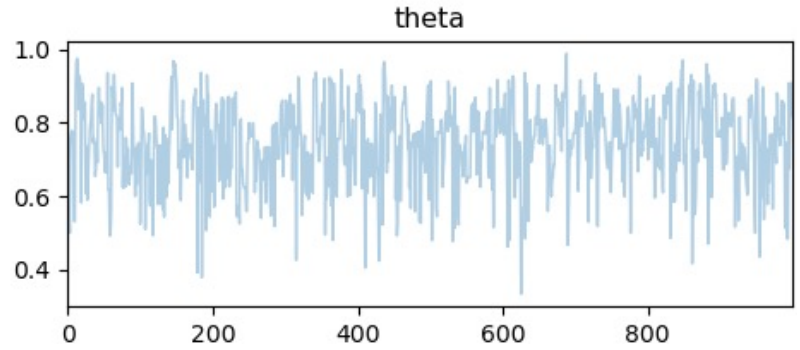
- import绘图库：
  - import matplotlib.pyplot as plt

- 展示后验概率分布：
  - pm.plot_posterior(trace)
  - plt.show()

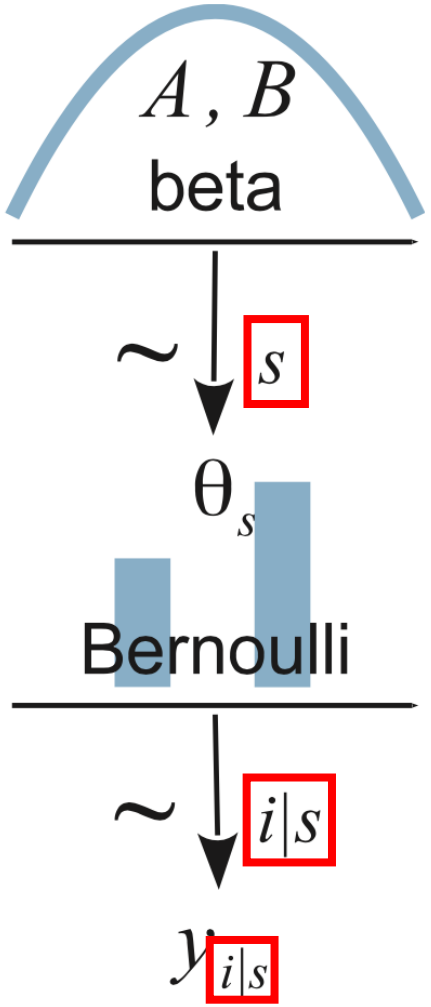



- 展示采样轨迹：
  - pm.plot_trace(trace)
  - plt.show()



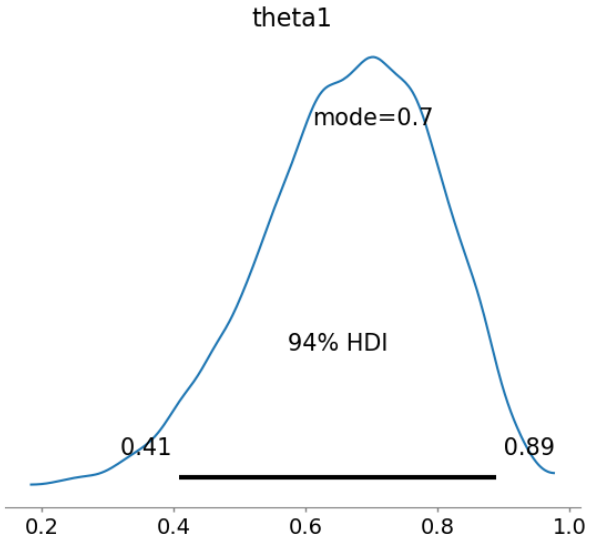
# 抛2个硬币

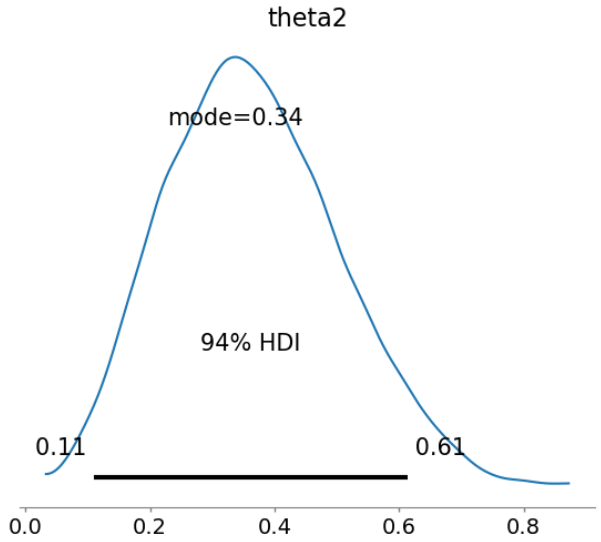



```python
# Generate the data
y1 = np.array([1, 1, 1, 1, 1, 1, 0, 0])  # 6 heads and 2 tails
y2 = np.array([1, 1, 0, 0, 0, 0, 0])  # 2 heads and 5 tails

with pm.Model() as model:
    # define the prior
    theta1 = pm.Beta('theta1', 2, 2)  # prior
    theta2 = pm.Beta('theta2', 2, 2)  # prior
    # define the likelihood
    y1_observed = pm.Bernoulli('y1_observed', p=theta1, observed=y1)
    y2_observed = pm.Bernoulli('y2_observed', p=theta2, observed=y2)

    # Generate a MCMC chain
    trace = pm.sample(10000, chains=1)

pm.plot_posterior(trace, point_estimate='mode')
plt.xlabel = 'theta'
plt.savefig('PyMC_Bernoulli_Posterior.png')
plt.show()

pm.plot_trace(trace)
plt.xlabel = 'theta'
plt.savefig('PyMC_Bernoulli_Trace.png')
plt.show()
```

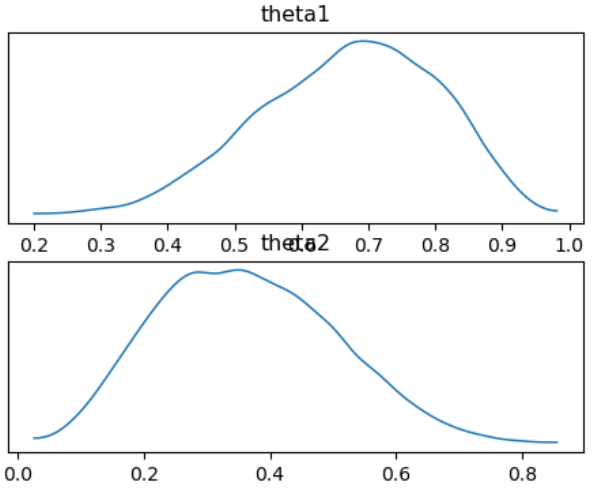
# 结果

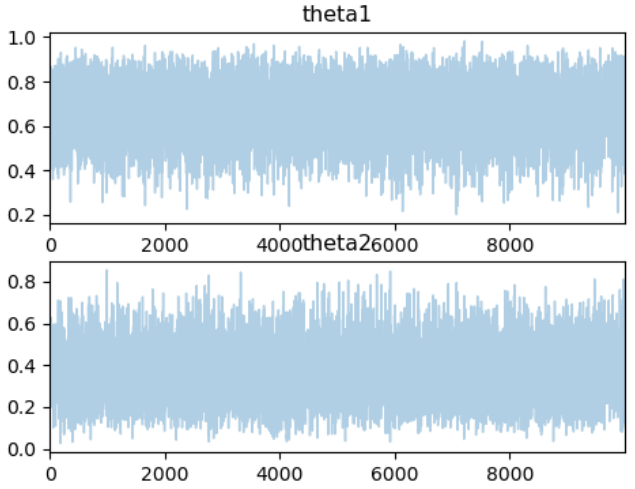
- 后验：


theta1
mode=0.7
94% HDI
0.41
0.89
0.2
0.4
0.6
0.8
1.0


theta2
mode=0.34
94% HDI
0.11
0.61
0.0
0.2
0.4
0.6
0.8

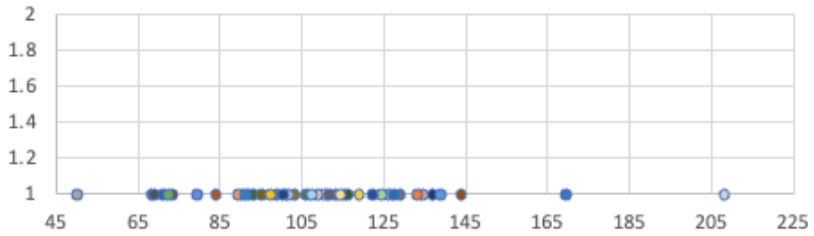
- 轨迹：

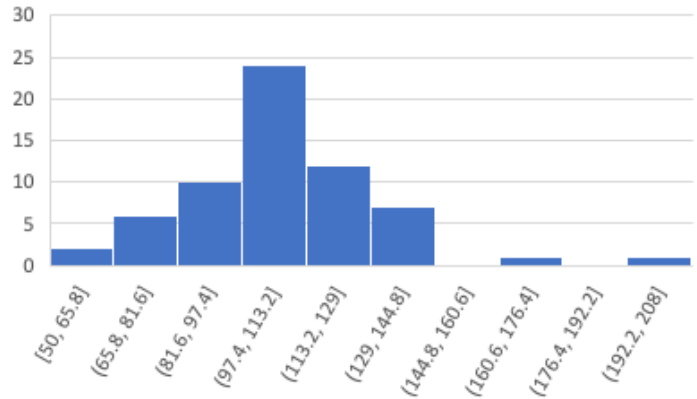

theta1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
theta2
0.0
0.2
0.4
0.6
0.8


theta1
1.0
0.8
0.6
0.4
0.2
0
2000
4000
6000
8000
theta2
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000

# 例子：聪明药

- 测试“聪明药”是否能使人聪明。
- 下图是63个人吃了“聪明药”以后的IQ数据。
- 人类的IQ均值是100，标准差是15。



# 似然和先验的假设

- 用高斯分布来对数据进行建模，即似然为：

$$p(D|\theta) = N(IQ|\mu, \sigma^2)$$

- 假设$\mu$的先验是高斯分布：

$$\mu \sim N(\mu|\mu_0, \sigma_0^2)$$

  - $\mu_0$，$\sigma_0$分别由样本的均值、标准差计算得到。

- 假设$\sigma$的先验是均匀分布：

$$\sigma \sim \text{uniform}(\text{std}(\text{IQ}_\text{i})/1000, \text{std}(\text{IQ}_\text{i}) * 1000)$$

  - 其中，$\text{std}(\text{IQ}_\text{i})$表示从63个样本直接计算得到的标准差。

# 生成数据的流程图

- 似然：

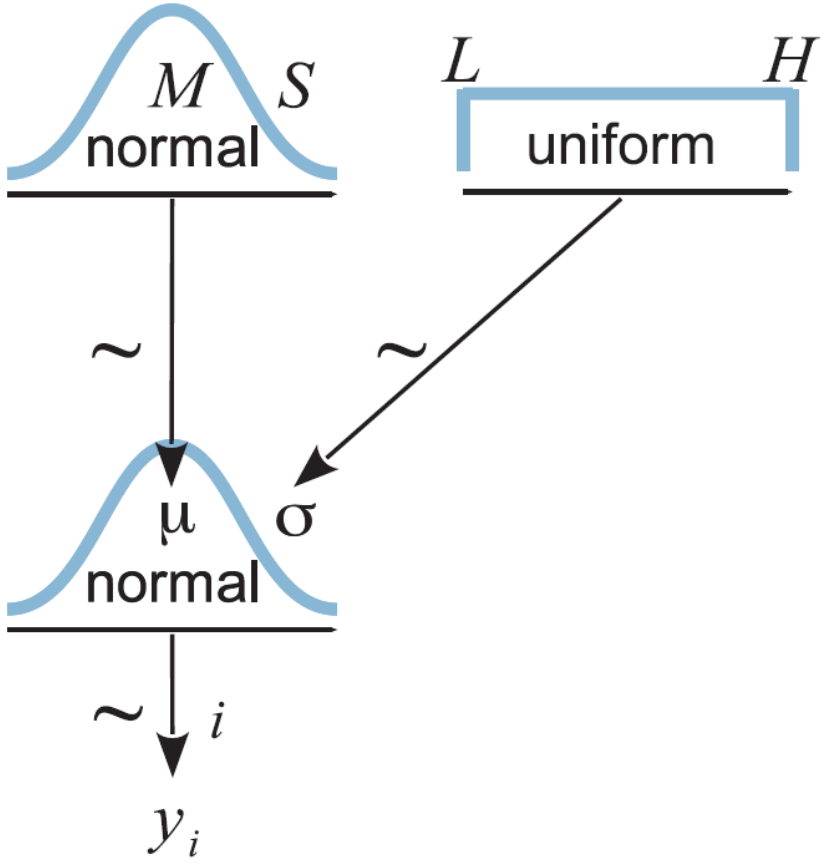
$$p(D|\theta) = N(IQ|\mu, \sigma^2)$$

- 先验：

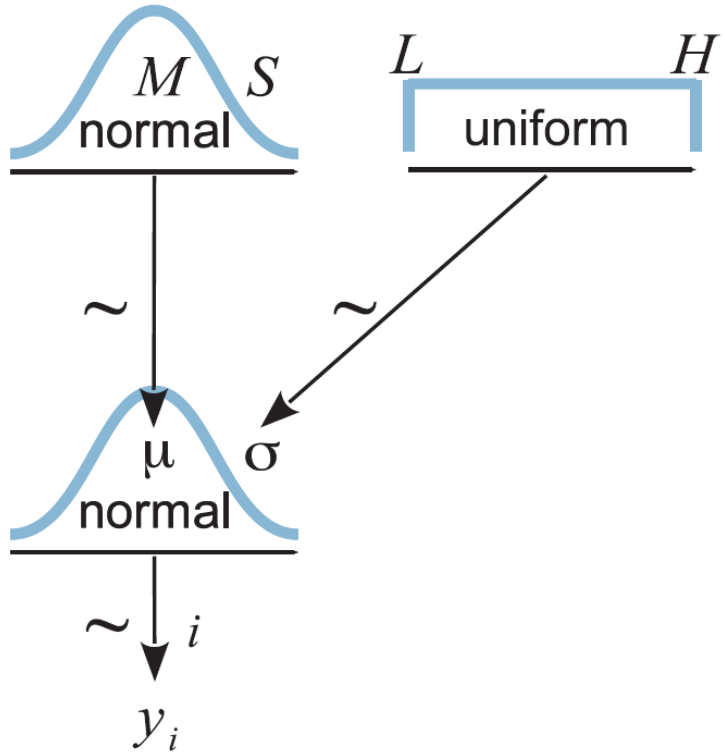
$$\mu \sim N(\mu|\mu_0, \sigma_0^2)$$

$$\sigma \sim \text{uniform}(a_0, b_0)$$

  - $a_0, b_0$表示均匀分布的起点和终点。



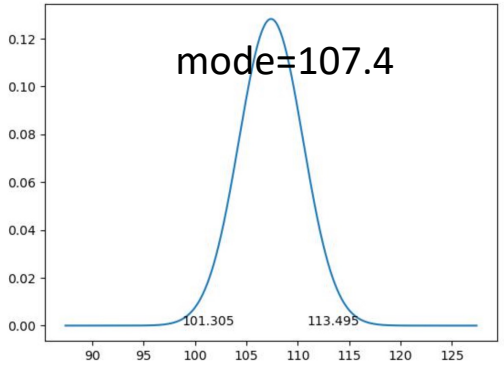
# 生成数据的流程图

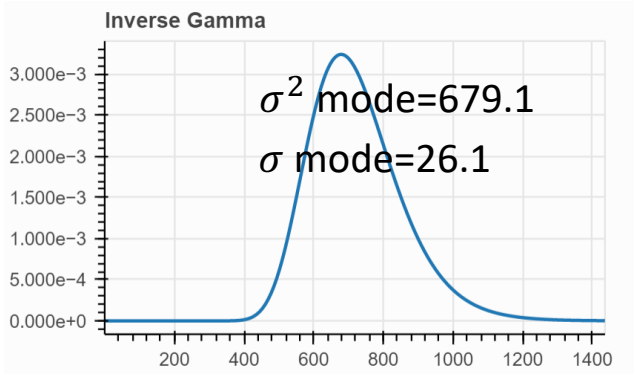



```
with pm.Model() as model:
    # define the priors
    sigma = pm.Uniform('sigma', 25.2/1000., 25.2*1000)
    mu = pm.Normal('mu', mu=107.8, sd=25.2)
    #define the likelihood
    y_observed = pm.Normal('y_observed', mu, sigma, observed=y)
    #Generate a MCMC chain
    trace = pm.sample(5000)
```

# 结果

## 准确数学分析：




## MCMC：