STA 3180 Statistical Modelling: Markov Chain Monte Carlo

Start of Code

1. Write a program to simulate a Markov Chain Monte Carlo (MCMC) algorithm using Python.

```
import numpy as np

# Define the transition probability matrix
P = np.array([[0.7, 0.3], [0.4, 0.6]])

# Initialize the state vector
x = np.array([1, 0])

# Define the number of iterations
n_iter = 1000

# Iterate through the MCMC algorithm
for i in range(n_iter):
       # Sample from the transition probability matrix
       x_new = np.random.choice([0, 1], p=P[x[0], :])
       # Update the state vector
       x[0] = x_new

# Print the final state vector
print(x)
```

End of Code

Start of Code

2. Write a program to calculate the posterior distribution of a Markov Chain Monte Carlo (MCMC) algorithm using Python.

```
import numpy as np

# Define the transition probability matrix
P = np.array([[0.7, 0.3], [0.4, 0.6]])

# Initialize the state vector
x = np.array([1, 0])

# Define the number of iterations
n_iter = 1000

# Initialize the posterior distribution
post_dist = np.zeros(2)

# Iterate through the MCMC algorithm
for i in range(n_iter):
       # Sample from the transition probability matrix
       x_new = np.random.choice([0, 1], p=P[x[0], :])
       # Update the state vector
       x[0] = x_new
       # Update the posterior distribution
       post_dist[x[0]] += 1

# Normalize the posterior distribution
```

```
post_dist /= np.sum(post_dist)
# Print the posterior distribution
print(post_dist)
```

End of Code

Start of Code

3. Write a program to calculate the expected value of a Markov Chain Monte Carlo (MCMC) algorithm using Python.

```
import numpy as np
# Define the transition probability matrix
P = np.array([[0.7, 0.3], [0.4, 0.6]])
# Initialize the state vector
x = np.array([1, 0])
# Define the number of iterations
n_iter = 1000
# Initialize the expected value
exp_val = 0
# Iterate through the MCMC algorithm
for i in range(n_iter):
    # Sample from the transition probability matrix
    x_new = np.random.choice([0, 1], p=P[x[0], :])
    # Update the state vector
    x[0] = x_new
    # Update the expected value
    exp_val += x[0]
# Normalize the expected value
exp_val /= n_iter
# Print the expected value
print(exp_val)
```

End of Code

Start of Code

4. Write a program to calculate the variance of a Markov Chain Monte Carlo (MCMC) algorithm using Python.

```
import numpy as np
# Define the transition probability matrix
P = np.array([[0.7, 0.3], [0.4, 0.6]])
# Initialize the state vector
x = np.array([1, 0])
# Define the number of iterations
n_iter = 1000
# Initialize the expected value and variance
exp_val = 0
var = 0
# Iterate through the MCMC algorithm
for i in range(n_iter):
        # Sample from the transition probability matrix
        x_new = np.random.choice([0, 1], p=P[x[0], :])
        # Update the state vector
        x[0] = x_new
        # Update the expected value
        exp_val += x[0]
        # Update the variance
        var += (x[0] - exp_val)**2

# Normalize the variance
var /= n_iter

# Print the variance
print(var)
```

End of Code