

STAT 8004: Homework 3

Srikar Katta

3/17/2022

Q1a: derive conditional distributions $\theta_1|\theta_2, y$ and $\theta_2|\theta_1, y$

Because $\theta_1, \theta_2 \sim \mathcal{MN}\left((y_1, y_2), \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$, we know the closed form expression of the conditional distributions:

$$\theta_1|\theta_2, (y_1, y_2) \sim \mathcal{N}(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2) \quad (1)$$

$$\theta_2|\theta_1, (y_1, y_2) \sim \mathcal{N}(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2) \quad (2)$$

Q1b: Gibbs Sampler

```
gibbs_sampler <- function(rho) {  
  
  # helper functions to sample theta_1 and theta_2  
  rconditional_normal <- function(y_i, theta_j, y_j, rho) {  
    sample <- rnorm(mean = y_i + rho * (theta_j - y_j), sd = sqrt(1 - rho^2), n = 1)  
    return(sample)  
  }  
  
  # initialize means  
  y_1 <- 0.5  
  y_2 <- 0.9  
  
  # initialize vectors  
  theta_1 <- vector(length = 1001)  
  theta_2 <- vector(length = 1001)  
  # initialize theta_1(0), theta_2(0)  
  theta_1[1] <- y_1  
  theta_2[1] <- y_2  
  
  # draw 1,000 samples from posterior  
  for(i in 2:1001) {  
    # sample theta_1(i) | theta_2(i - 1)
```

```

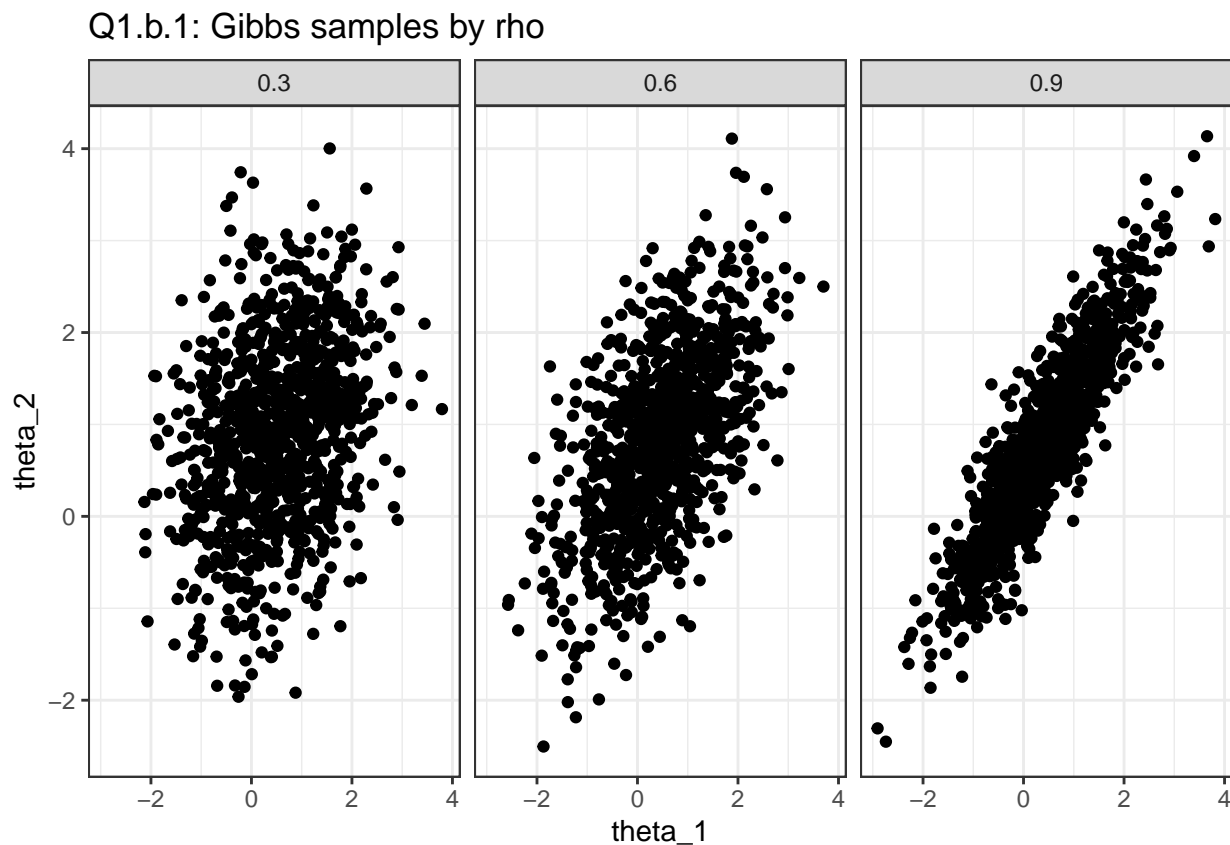
theta_1[i] <- rconditional_normal(y_i = y_1, y_j = y_2, theta_j = theta_2[i - 1], rho = rho)
# sample theta_2(i) | theta_1(i)
theta_2[i] <- rconditional_normal(y_i = y_2, y_j = y_1, theta_j = theta_1[i], rho = rho)
}

tibble(
  time = 1:1001,
  theta_1 = theta_1,
  theta_2 = theta_2,
  rho = rho,
  theta1_acf = cor(theta_1[2:(1001 - 10)], theta_1[(12):1001]),
  theta1_effective_size = coda::effectiveSize(theta_1)
)
}

gibbs_df <- lapply(c(0.3, 0.6, 0.9), gibbs_sampler) %>%
  bind_rows()

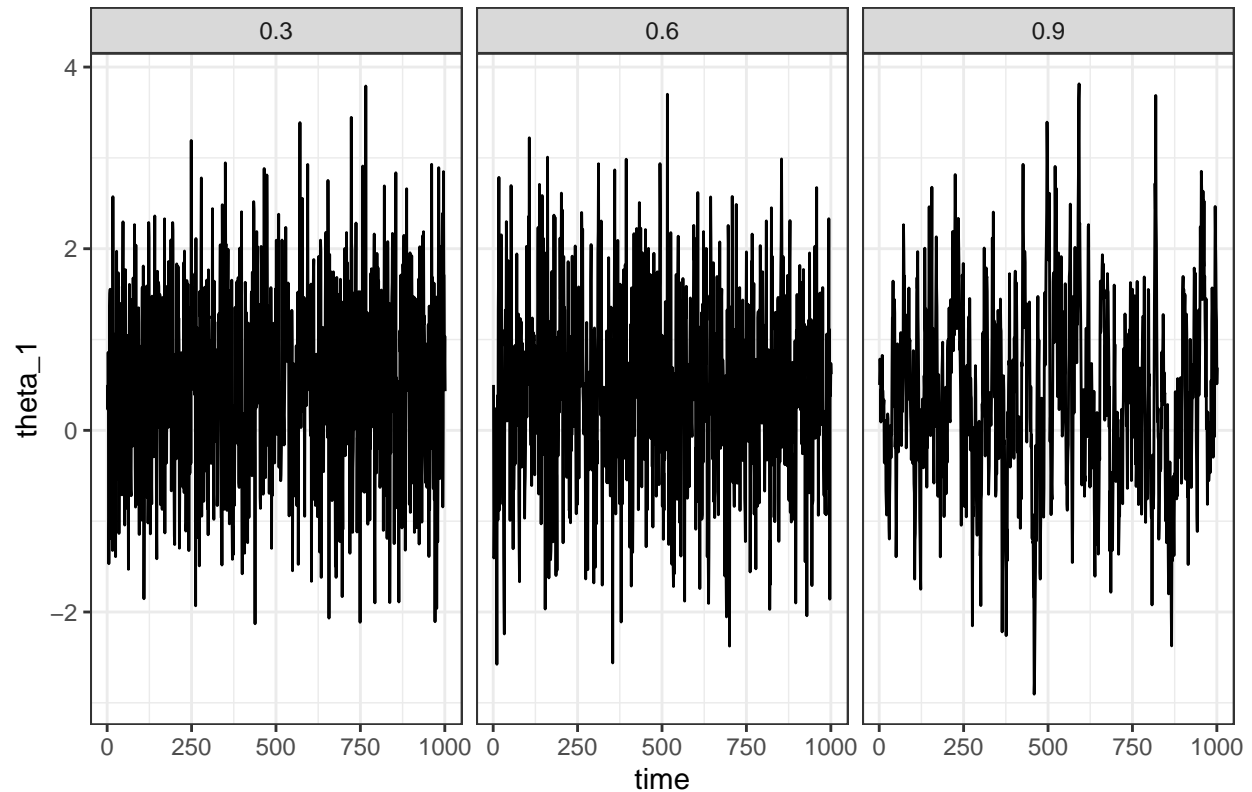
ggplot(gibbs_df) +
  geom_point(aes(x = theta_1, y = theta_2)) +
  facet_wrap(~rho) +
  labs(title = 'Q1.b.1: Gibbs samples by rho')

```



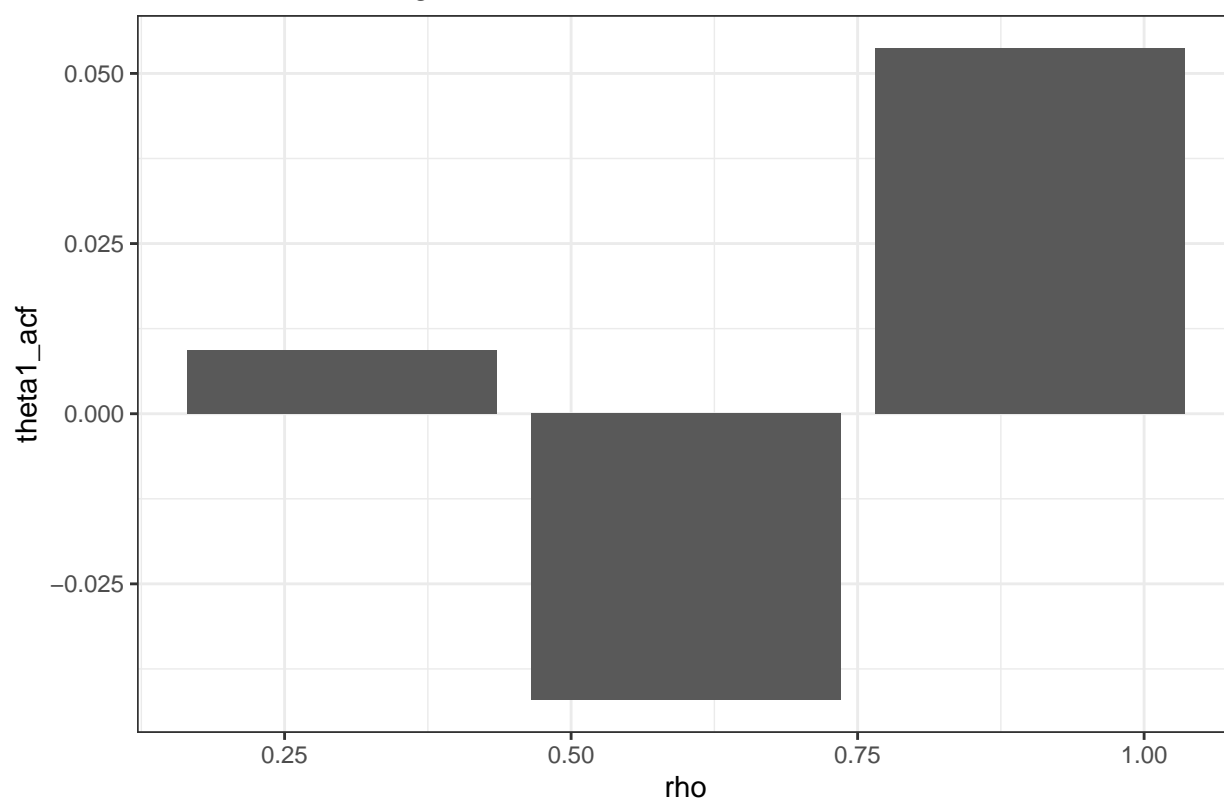
```
ggplot(gibbs_df) +
  geom_line(aes(x = time, y = theta_1)) +
  facet_wrap(~rho) +
  labs(title = 'Q1.b.2: theta_1 traceplot by rho')
```

Q1.b.2: theta_1 traceplot by rho



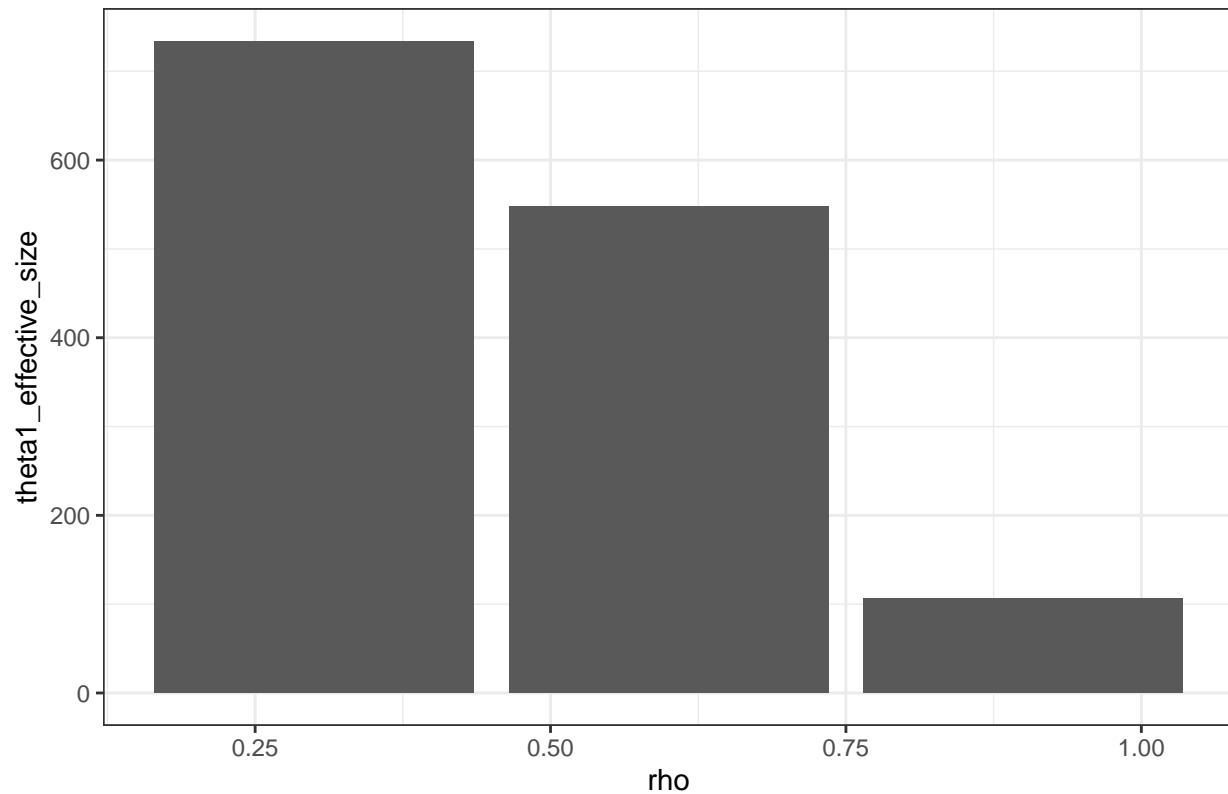
```
ggplot(gibbs_df %>% select(rho, theta1_acf) %>% distinct()) +
  geom_col(aes(x = rho, y = theta1_acf)) +
  labs(title = 'Q1.b.3: theta_1 lag-10 autocorrelation')
```

Q1.b.3: theta_1 lag-10 autocorrelation



```
ggplot(gibbs_df %>% select(rho, theta1_effective_size) %>% distinct()) +  
  geom_col(aes(x = rho, y = theta1_effective_size)) +  
  labs(title = 'Q1.b.3: theta_1 effective size by rho')
```

Q1.b.3: theta_1 effective size by rho



Q1c: Gibbs Sampler Efficiency

As ρ increases, the sampler becomes less efficient. This behavior's reason is clear when considering the concentration of the different posterior distributions: with a small ρ , most of the Monte Carlo samples will be very close together. Since the samples have a small variance, the estimator is able to converge to a single value quickly. As ρ increases, the mixing time will take longer because the variance of the samples will also increase.