

Lab 7

Bingying Liu

3/3/2020

Exercise 1: What is the conditional density $p(\theta_1 | X_1, \dots, X_n, \rho, \theta_2)$?

Since $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$, $\Sigma^{-1} = \frac{1}{1-\rho^2} \begin{pmatrix} 1 & -\rho \\ -\rho & 1 \end{pmatrix}$, $\theta = (\theta_1, \theta_2)^T$ and $\bar{X} = (\bar{X}_1, \bar{X}_2)^T$. Therefore, when we plug these in conditional density, we have:

$$\begin{aligned} p(\theta_1 | X_1, \dots, X_n, \rho, \theta_2) &\propto p(X_1, \dots, X_n | \theta_1, \theta_2, \rho) p(\theta_1 | \theta_2, \rho) \\ &\propto p(X_1, \dots, X_n | \theta_1, \theta_2, \rho) p(\theta_1) \\ &\propto e^{-\frac{1}{2}\theta^T(n\Sigma^{-1})\theta + \theta^T(n\Sigma^{-1}\bar{X})} p(\theta_1) \\ &\propto e^{-\frac{n}{2(1-\rho^2)}(\theta_1^2 - 2\rho\theta_1\theta_2 + \theta_2^2) + \frac{n}{1-\rho^2}(\theta_1\bar{X}_1 - \rho\theta_2\bar{X}_1 - \rho\theta_1\bar{X}_2 + \theta_2\bar{X}_2)} e^{-\frac{1}{2}\theta_1^2} \\ &\propto e^{-\left(\frac{n}{2(1-\rho^2)} + \frac{1}{2}\right)\theta_1^2 + \left(\frac{n\rho}{1-\rho^2}\theta_2 + \frac{n\bar{X}_1}{1-\rho^2} - \frac{n\rho\bar{X}_2}{1-\rho^2}\right)\theta_1} \end{aligned}$$

Therefore,

$$\begin{aligned} -\frac{1}{2\sigma^2} &= -\left(\frac{n}{2(1-\rho^2)} + \frac{1}{2}\right) \\ \frac{1}{\sigma^2} &= \frac{n}{1-\rho^2} + 1 \\ \sigma^2 &= \frac{1-\rho^2}{n+1-\rho^2} \end{aligned}$$

and

$$\begin{aligned} \frac{\mu}{\sigma^2} &= \frac{1}{1-\rho^2}(n\rho\theta_2 + n\bar{X}_1 - n\rho\bar{X}_2) \\ \mu &= \frac{n\rho\theta_2 + n\bar{X}_1 - n\rho\bar{X}_2}{n+1-\rho^2} \\ &= \frac{\sum_i X_{i1} + \rho(n\theta_1 - \sum_i X_{i2})}{n+1-\rho^2} \end{aligned}$$

Thus, the conditional density $p(\theta_1 | X_1, \dots, X_n, \rho, \theta_2) = N\left(\frac{\sum_i X_{i1} + \rho(n\theta_1 - \sum_i X_{i2})}{n+1-\rho^2}, \frac{1-\rho^2}{n+1-\rho^2}\right)$.

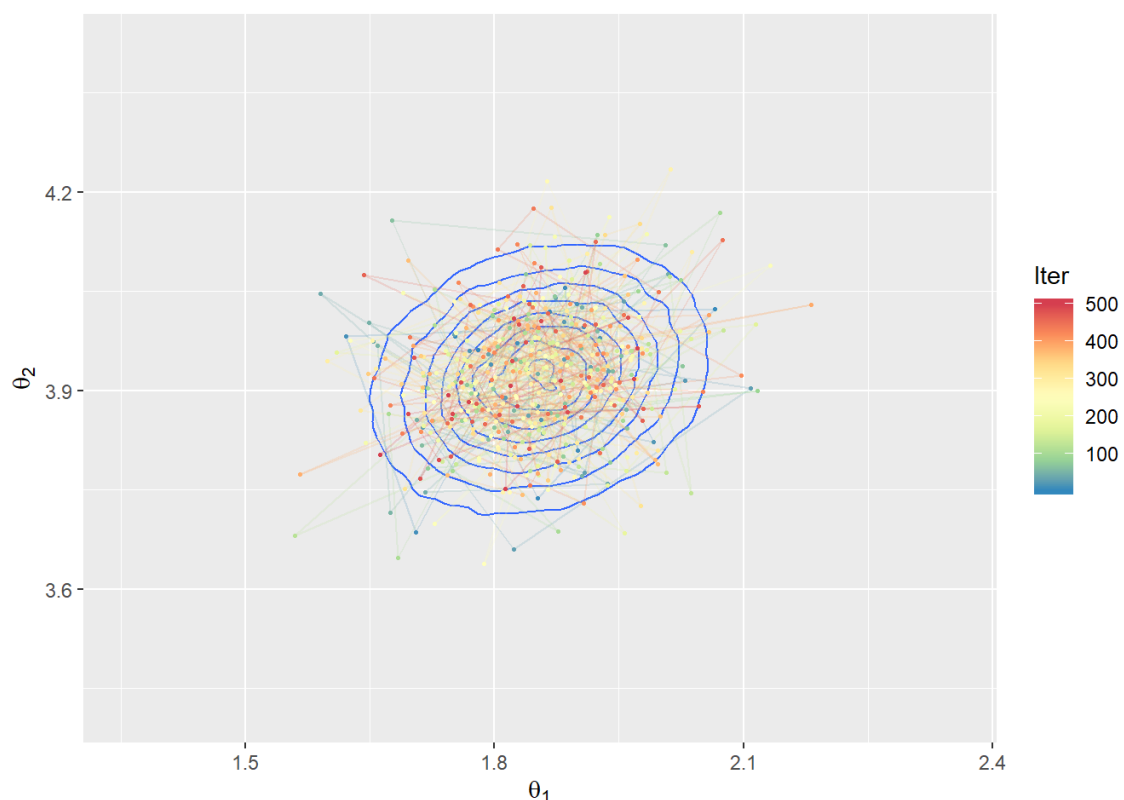
Exercise 2: What is the conditional density $p(\theta_2 | X_1, \dots, X_n, \rho, \theta_1)$?

Logic is exactly the same as in exercise 1. Therefore, $p(\theta_2 | X_1, \dots, X_n, \rho, \theta_1) = \frac{\sum_i X_{i2} + \rho(n\theta_1 - \sum_i X_{i1})}{n+1-\rho^2}, \frac{1-\rho^2}{n+1-\rho^2}$

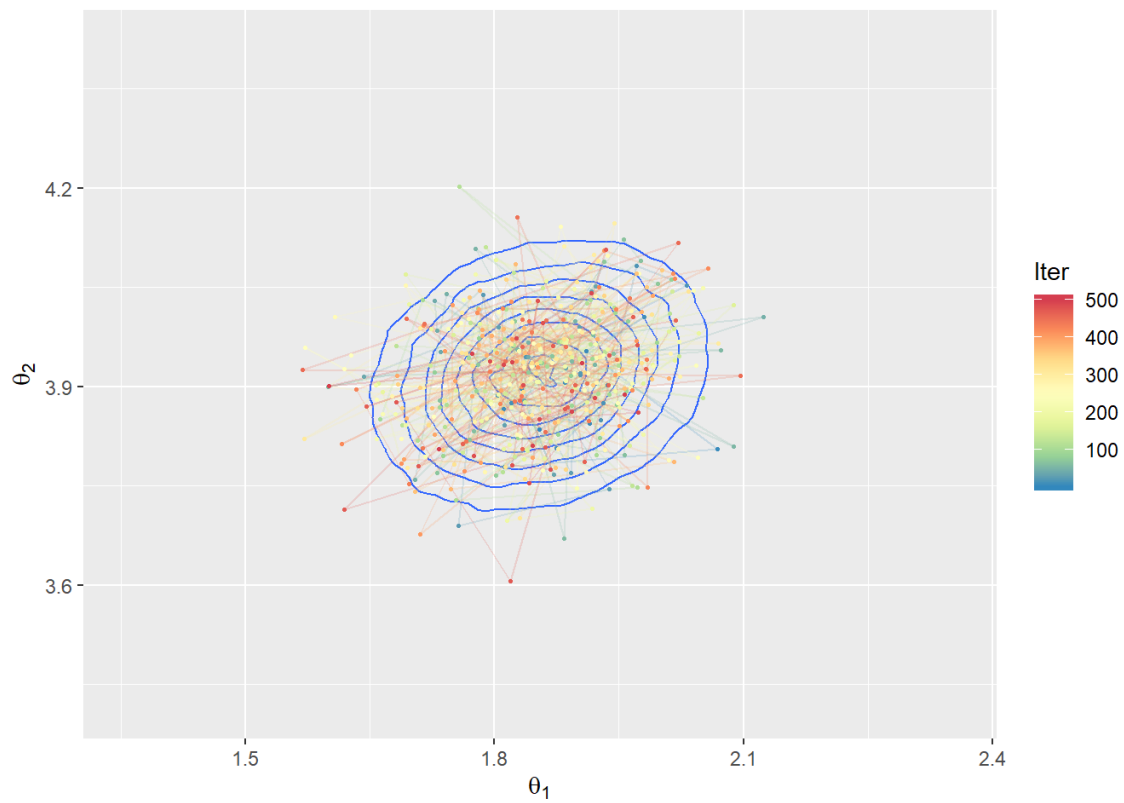
Exercise 3: Specifically, write a function called `normal_gibbs_sampler` that takes as arguments (1) the number of samples desired (S) (2) an $n \times 2$ matrix of data values (X) (3) the given correlation parameter (rho).

```
normal_gibbs_sampler <- function(S, X, rho){
  theta_1 <- rep(0, S)
  theta_2 <- rep(0, S)
  n <- nrow(X)
  for(s in 2:S){
    theta_1[s] <- rnorm(1,
                        mean = (sum(X[,1]) + rho*(n*theta_2[s-1] - sum(X[,2]))) /
                        (n + 1 - rho^2),
                        sd = sqrt((1 - rho^2)/(n + 1 - rho^2)))
    theta_2[s] <- rnorm(1,
                        mean = (sum(X[,2]) + rho*(n*theta_1[s] - sum(X[,1]))) /
                        (n + 1 - rho^2),
                        sd = sqrt((1 - rho^2)/(n + 1 - rho^2)))
  }
  return(cbind(theta_1, theta_2))
}
```

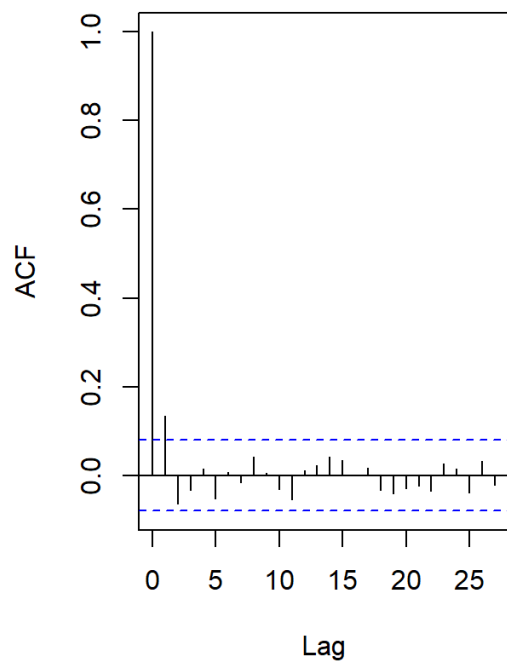
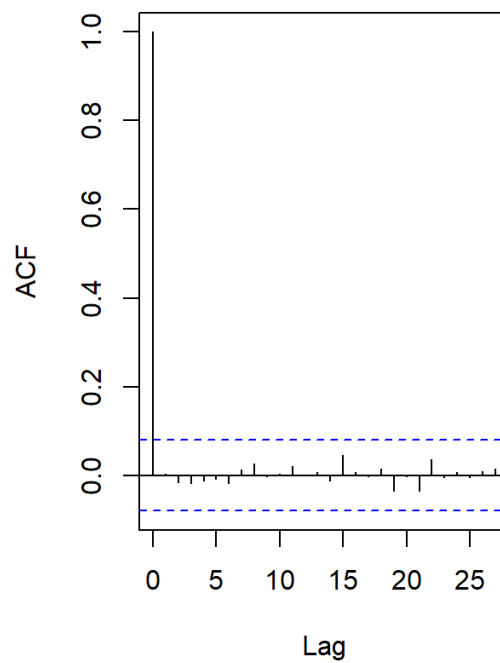
```
# rho = 0.2: Generate samples from posterior distribution (normal gibbs samps)
n <- 100
rho <- 0.2
X <- MASS::mvrnorm(n = n, mu = c(2, 4), Sigma = matrix(c(1, rho, rho, 1), nrow = 2))
Sigma_post <- matrix(((1-rho^2)/((n+1-rho^2)^2 - (n^2)*(rho^2)))*c(n+1-rho^2, n*rho, n*rho, n+1-rho^2), nrow = 2)
mu_post <- n*Sigma_post%*%matrix(c(1/(1-rho^2), -rho/(1-rho^2),
                                   -rho/(1-rho^2), 1/(1-rho^2)),
                                   nrow = 2)%*%colMeans(X)
norm_gibbs_samps <- normal_gibbs_sampler(600, X, rho)
#
true_post <- MASS::mvrnorm(n = 100000,
                           mu = mu_post,
                           Sigma = Sigma_post)
data.frame(norm_gibbs_samps) %>%
  magrittr::set_colnames(c("theta_1", "theta_2")) %>%
  dplyr::mutate(iter = 1:n()) %>%
  dplyr::filter(iter > 100) %>%
  dplyr::mutate(iter = 1:n()) %>%
  ggplot2::ggplot() +
  geom_density2d(data = data.frame(true_post) %>%
                 magrittr::set_colnames(c("true_1", "true_2")),
                 aes(x = true_1, y = true_2)) +
  geom_path(aes(x = theta_1, y = theta_2, colour = iter), alpha = 0.2, size = 0.5) +
  geom_point(aes(x = theta_1, y = theta_2, colour = iter), size = 0.5) +
  scale_color_distiller(palette = "Spectral", name = "Iter") +
  labs(x = expression(theta[1]), y = expression(theta[2])) +
  xlim(c(mu_post[1] - 0.5, mu_post[1] + 0.5)) +
  ylim(c(mu_post[2] - 0.5, mu_post[2] + 0.5))
```



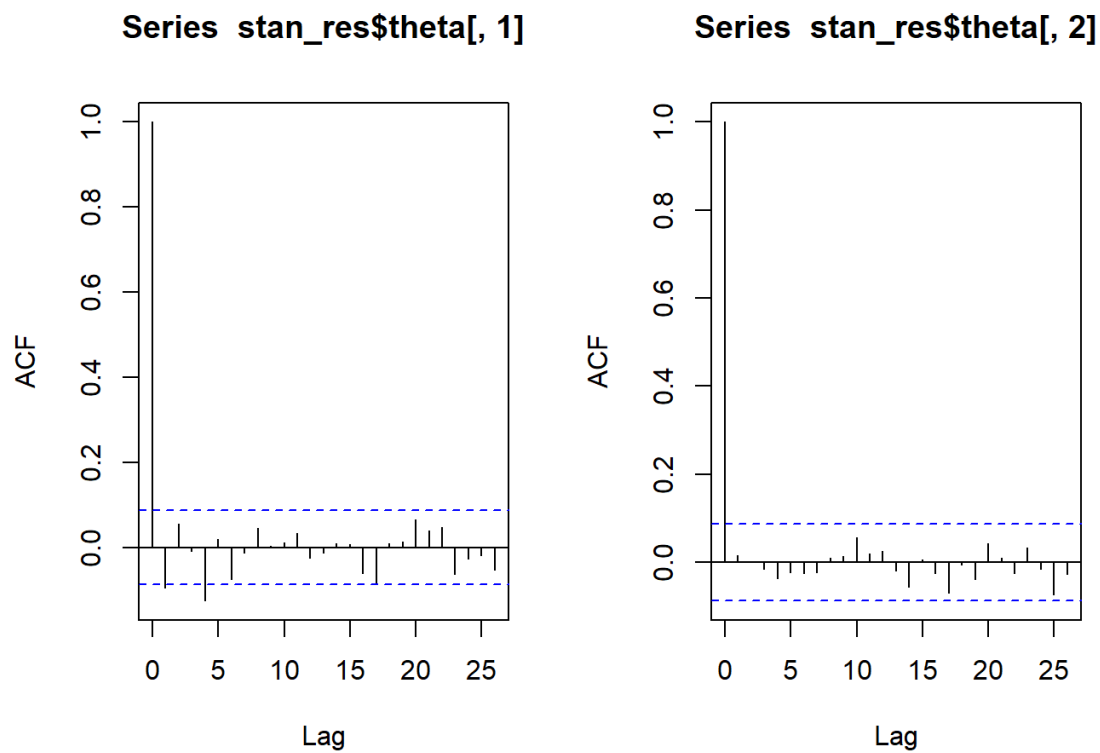
```
# stan_res
stan_res <- rstan::stan("lab-07-hmc_norm_example.stan", data = list(X = X,
                                                                    N = nrow(X),
                                                                    Sigma = matrix(c(1, rho, rho, 1), nrow = 2)),
                      chains = 1, iter = 600, warmup = 100, verbose = F, refresh = 0) %>%
  rstan::extract()
#
data.frame(stan_res$theta) %>%
  magrittr::set_colnames(c("theta_1", "theta_2")) %>%
  dplyr::mutate(iter = 1:n()) %>%
  ggplot2::ggplot() +
  geom_density2d(data = data.frame(true_post) %>%
    magrittr::set_colnames(c("true_1", "true_2")),
    aes(x = true_1, y = true_2)) +
  geom_path(aes(x = theta_1, y = theta_2, colour = iter), alpha = 0.2, size = 0.5) +
  geom_point(aes(x = theta_1, y = theta_2, colour = iter), size = 0.5) +
  scale_color_distiller(palette = "Spectral", name = "Iter") +
  labs(x = expression(theta[1]), y = expression(theta[2])) +
  xlim(c(mu_post[1] - 0.5, mu_post[1] + 0.5)) +
  ylim(c(mu_post[2] - 0.5, mu_post[2] + 0.5))
```



```
#  
par(mfrow = c(1,2))  
acf(norm_gibbs_samps[,1])  
acf(norm_gibbs_samps[,2])
```

Series norm_gibbs_samps[, 1]**Series norm_gibbs_samps[, 2]**

```
#  
par(mfrow = c(1,2))  
acf(stan_res$theta[,1])  
acf(stan_res$theta[,2])
```



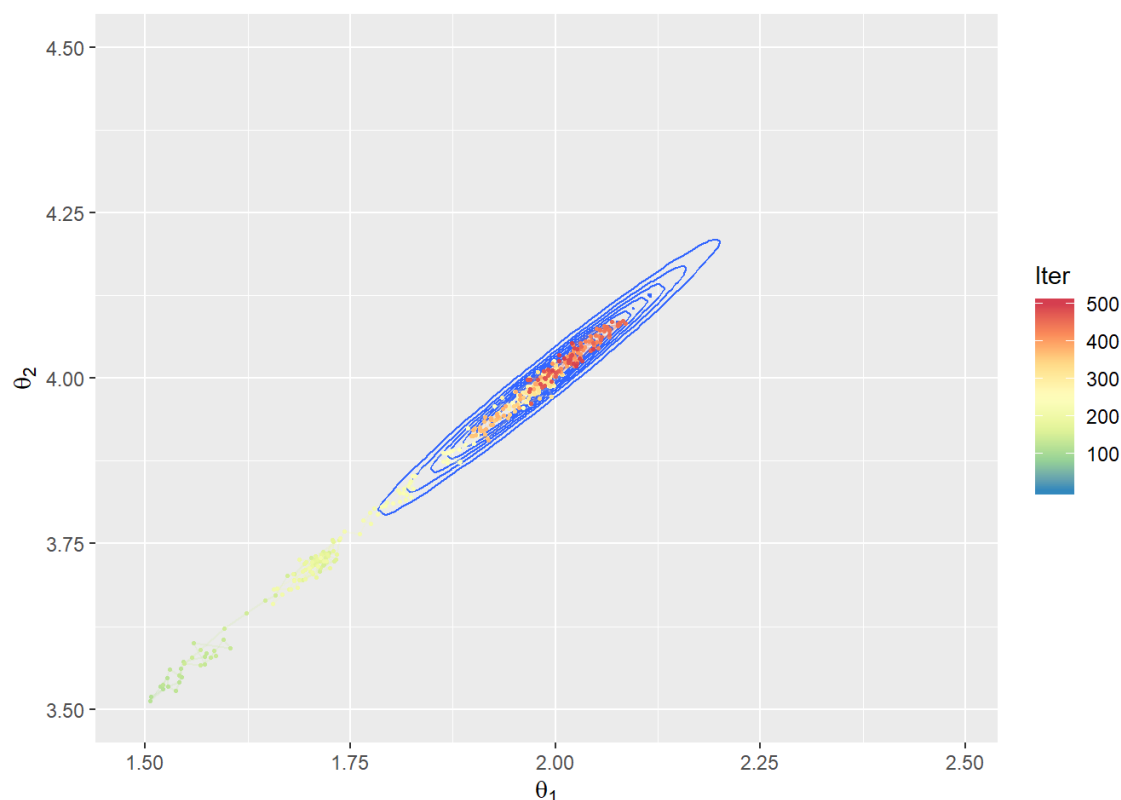
```

# rho = 0.995
n <- 100
rho <- 0.995
X <- MASS::mvrnorm(n = n, mu = c(2, 4), Sigma = matrix(c(1, rho, rho, 1), nrow = 2))
Sigma_post <- matrix((((1-rho^2)/((n+1-rho^2)^2 - (n^2)*(rho^2))))*c(n+1-rho^2, n*rho, n*rho, n+1-rho^2), nrow = 2)
mu_post <- n*Sigma_post%%matrix(c(1/(1-rho^2), -rho/(1-rho^2),
                                -rho/(1-rho^2), 1/(1-rho^2)),
                                nrow = 2)%%colMeans(X)
norm_gibbs_samps <- normal_gibbs_sampler(600, X, rho)
#
true_post <- MASS::mvrnorm(n = 100000,
                           mu = n*Sigma_post%%(matrix(c(1/(1-rho^2), -rho/(1-rho^2),
                                                           -rho/(1-rho^2), 1/(1-rho^2)),
                                                           nrow = 2)%%colMeans(X)),
                           Sigma = Sigma_post)
#
data.frame(norm_gibbs_samps) %>%
  magrittr::set_colnames(c("theta_1", "theta_2")) %>%
  dplyr::mutate(iter = 1:n()) %>%
  dplyr::filter(iter > 100) %>%
  dplyr::mutate(iter = 1:n()) %>%
  ggplot2::ggplot() +
  geom_density2d(data = data.frame(true_post) %>%
                 magrittr::set_colnames(c("true_1", "true_2")),
                 aes(x = true_1, y = true_2)) +
  geom_path(aes(x = theta_1, y = theta_2, colour = iter), alpha = 0.2, size = 0.5) +
  geom_point(aes(x = theta_1, y = theta_2, colour = iter), size = 0.5) +
  scale_color_distiller(palette = "Spectral", name = "Iter") +
  labs(x = expression(theta[1]), y = expression(theta[2])) +
  xlim(c(mu_post[1] - 0.5, mu_post[1] + 0.5)) +
  ylim(c(mu_post[2] - 0.5, mu_post[2] + 0.5))

```

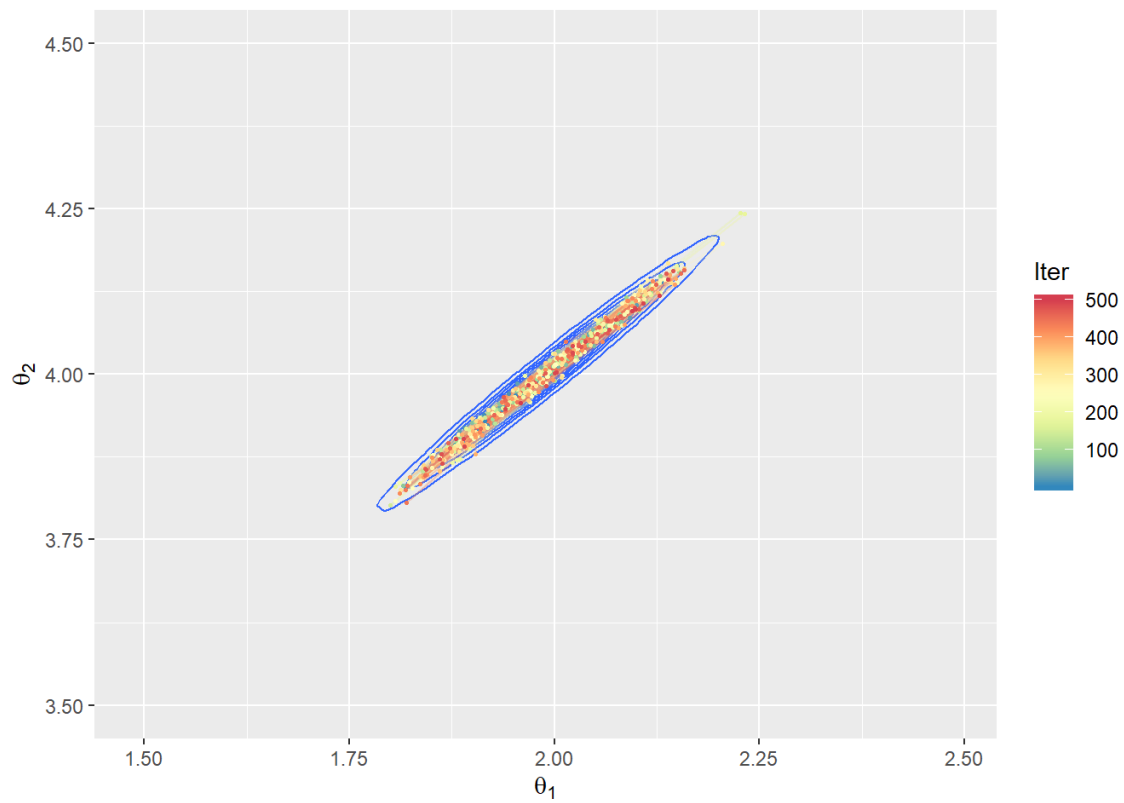
```
## Warning: Removed 114 rows containing missing values (geom_path).
```

```
## Warning: Removed 114 rows containing missing values (geom_point).
```

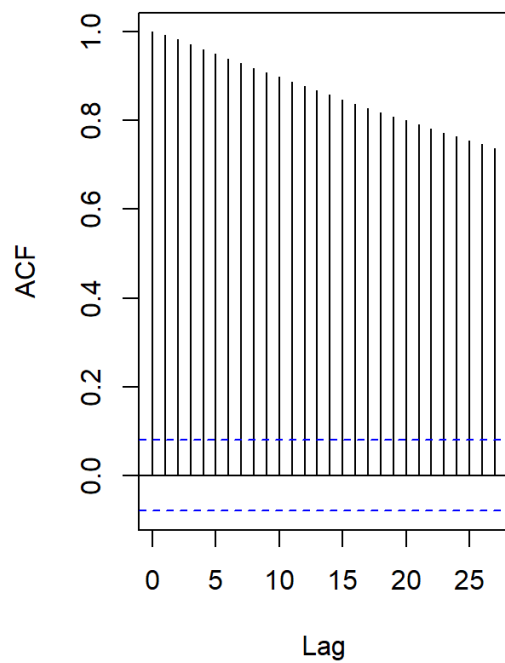
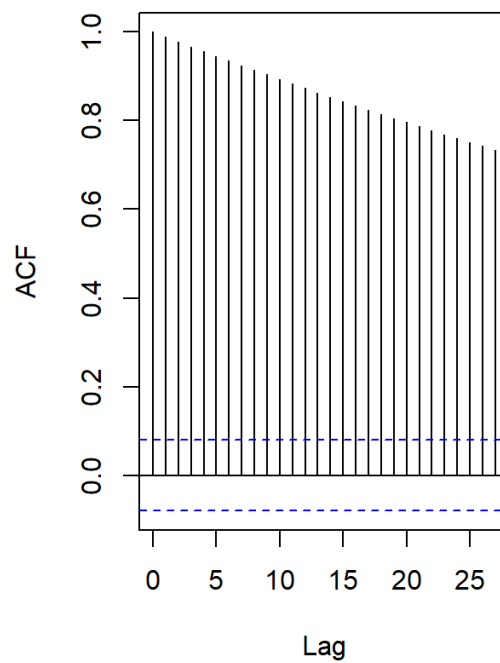


```
#
stan_res <- rstan::stan("lab-07-hmc_norm_example.stan", data = list(X = X,
                                                                    N = nrow(X),
                                                                    Sigma = matrix(c(1, rho, rho, 1), nrow = 2)),
                      chains = 1, iter = 600, warmup = 100, verbose = F, refresh = 0) %>%
  rstan::extract()
data.frame(stan_res$theta) %>%
  magrittr::set_colnames(c("theta_1", "theta_2")) %>%
  dplyr::mutate(iter = 1:n()) %>%
  ggplot2::ggplot() +
  geom_density2d(data = data.frame(true_post) %>%
    magrittr::set_colnames(c("true_1", "true_2")),
    aes(x = true_1, y = true_2)) +
  geom_path(aes(x = theta_1, y = theta_2, colour = iter), alpha = 0.2, size = 0.5) +
  geom_point(aes(x = theta_1, y = theta_2, colour = iter), size = 0.5) +
  scale_color_distiller(palette = "Spectral", name = "Iter") +
  labs(x = expression(theta[1]), y = expression(theta[2])) +
  xlim(c(mu_post[1] - 0.5, mu_post[1] + 0.5)) +
  ylim(c(mu_post[2] - 0.5, mu_post[2] + 0.5))
```

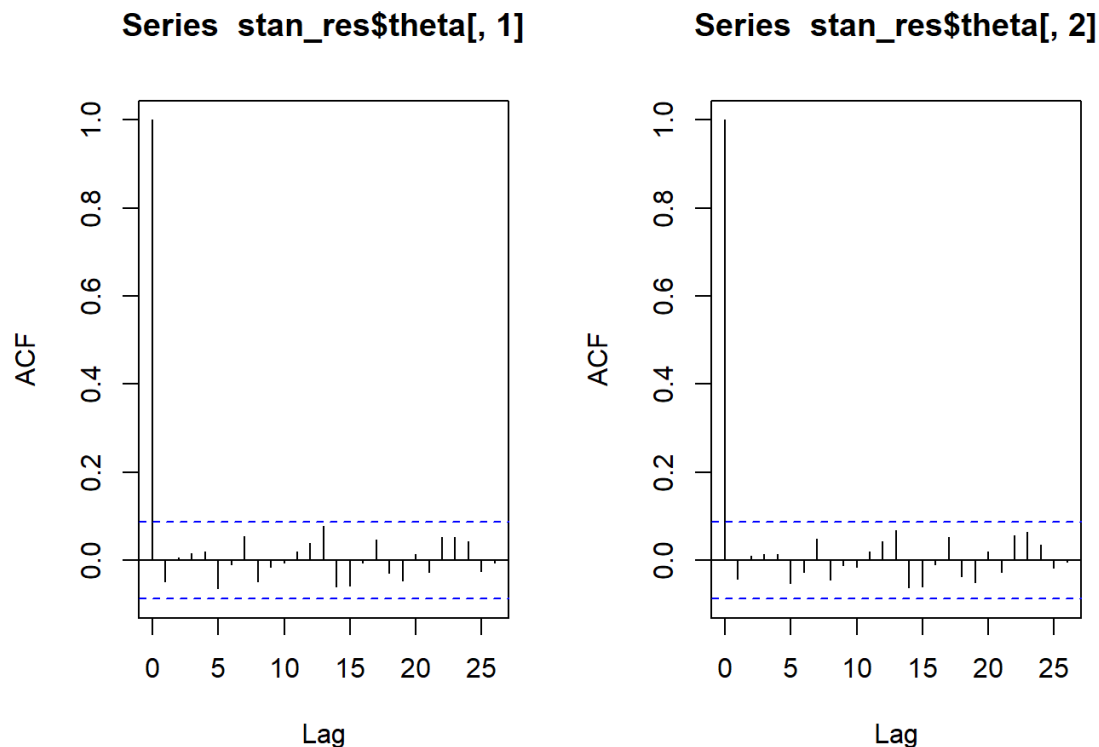
```
## Warning: Removed 1 rows containing missing values (geom_point).
```



```
#  
par(mfrow = c(1,2))  
acf(norm_gibbs_samps[,1])  
acf(norm_gibbs_samps[,2])
```

Series norm_gibbs_samps[, 1]**Series norm_gibbs_samps[, 2]**


```
#
par(mfrow = c(1,2))
acf(stan_res$theta[,1])
acf(stan_res$theta[,2])
```



Exercise 4: How do the results of the Gibbs sampler differ from those obtained from HMC?

When target density is nicely behaved on R^2 (when two variables have low correlation), Gibbs sampler and HMC'S results look pretty similar and they all perform well. However, when two variables have really high correlation ($\rho = 0.995$), Gibbs sampler has a much worse autocorrelation plot (autocorrelation still remains after 25th lag), which means slow mixing due to the inability of the sampler to move around the space well. In addition, the path of sampling shows that if initial point was chosen poorly, it took Gibbs many iterations to reach the high density region and also took many effort for it to get out of the neighborhood of peaked regions. Whereas HMC performs well on high correlation situation.

Exercise 5: Why do the samples from the Gibbs sampler exhibit this behavior?

Because Gibbs sampling alters only one variable at a time, it can only move vertically or horizontally on the Cartesian plane. It'll be unable to reach regions of high posterior probability that lie diagonally from one another.