

2019R1 Applied Bayesian Methods (STAT6106)

Assignment 4

Yiu Chung WONG 1155017920

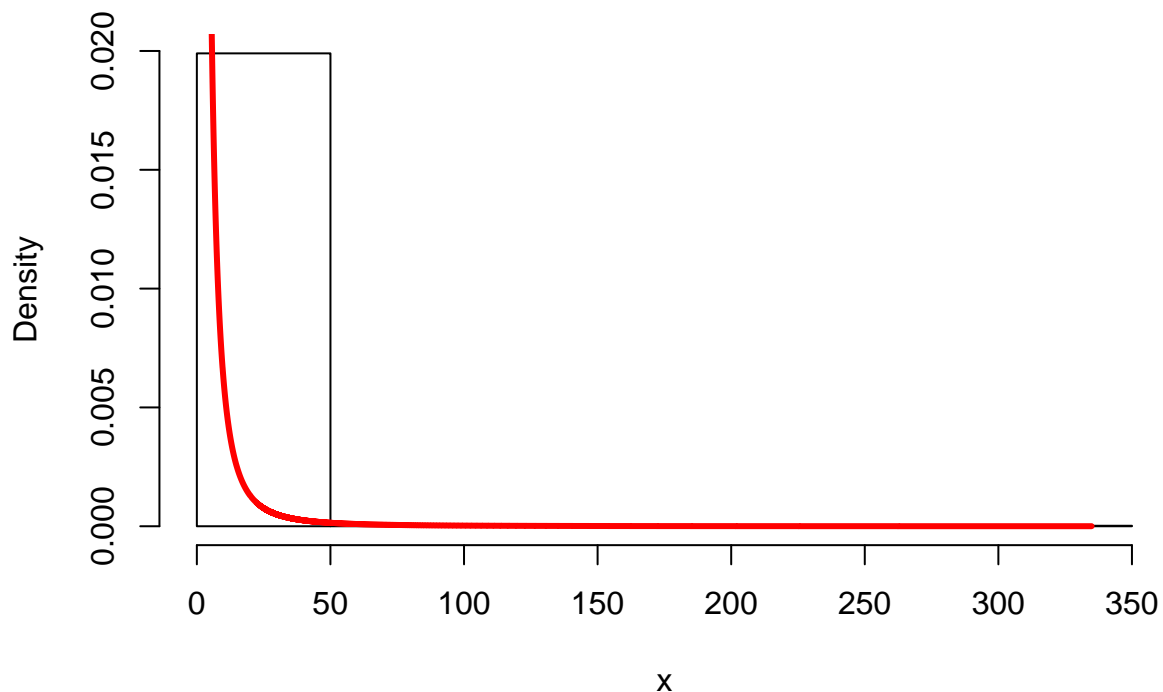
```
set.seed(6106);
```

A.

```
my.rf = function(n, df1, df2)
{
  num <- replicate(sum(rnorm(n = df1)^2) / df1, n = n)
  denom <- replicate(sum(rnorm(n = df2)^2) / df2, n = n)
  return(num / denom)
}

## Test against pdf
df1 <- 2
df2 <- 3
x = my.rf(1000, df1, df2)
hist(x, freq=FALSE, main = "F Distribution Sampling")
t = seq(0, max(x), 0.01)
lines(t, df(t, df1, df2), lwd=3, col='red')
```

F Distribution Sampling



B.

$$\langle F^N \rangle = (b - a) \frac{1}{N} \sum_{i=0}^{N-1} f(X_i).$$

```
monte_carlo <- function(func, n = 10e5, from = 0, to = 1)
{
  estimates <- replicate(n = 10,
    expr = {
      x <- runif(n, min = from, max = 1) * to
      mean(func(x)) * (to - from)
    }
  )
  c(mean(estimates), sd(estimates))
}
```

B1.

```
b1_func <- function(x) exp(-(x^2))
b1 <- monte_carlo(b1_func, to = 4)
b1_average <- b1[1]
b1_sd <- b1[2]
```

- Monte Carlo estimate is 0.8853623 with standard deviation 0.0014345

B2.

```
b2 <- monte_carlo(sin, to = pi)
b2_average <- b2[1]
b2_sd <- b2[2]
```

- Monte Carlo estimate is 2.0003571 with standard deviation 0.001305

C.

```
n <- 10e4
burnin <- n / 2

x <- rep(NA,n)
x[1] <- 1

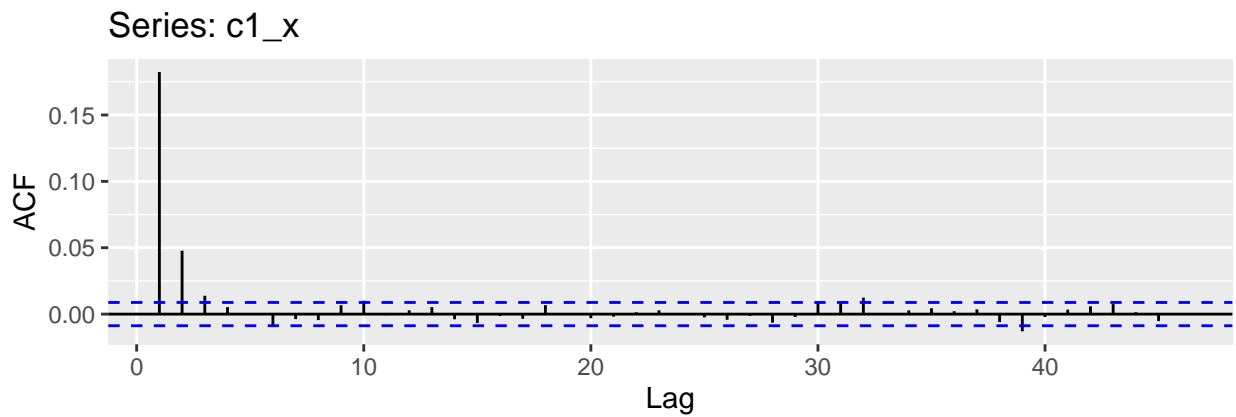
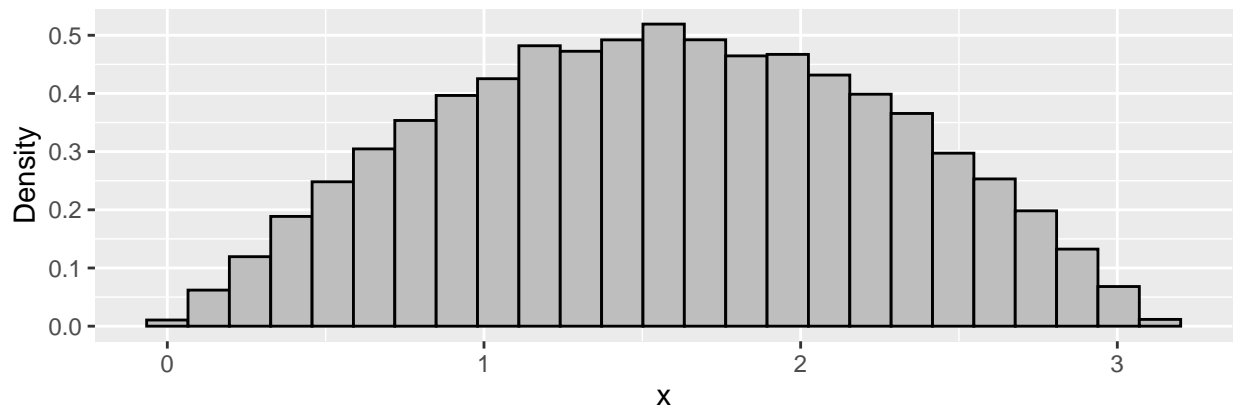
xy <- matrix(data = rep(NA, n), nrow = n, ncol = 2)
xy[1,] <- c(10,10)
```

```
dtnorm <- function(x, mean = 0, sd = 1, lower = -Inf, upper = Inf, log = FALSE) {
  ret <- numeric(length(x))
  ret[x <= lower | x >= upper] <- if (log) {
    -Inf
  } else {
    0
  }
  ret[upper < lower] <- NaN
  ind <- x > lower & x < upper
  if (any(ind)) {
    denom <- pnorm(upper, mean, sd) - pnorm(
      lower, mean,
      sd
    )
    xtmp <- dnorm(x, mean, sd, log)
    if (log) {
      xtmp <- xtmp - log(denom)
    } else {
      xtmp <- xtmp / denom
    }
    ret[x >= lower & x <= upper] <- xtmp[ind]
  }
  ret
}
```

C1.

```
c1_target = function(x) sin(x)
c1_x <- x
for(i in 2:n)
{
  currentx = c1_x[i-1]
  #proposedx = currentx + rnorm(1,mean=0,sd=1)
  proposedx = runif(n = 1, min = 0, max = pi)
  #c <- dnorm(currentx, proposedx, 1) / dnorm(proposedx, currentx, 1)
  A = c1_target(proposedx)/c1_target(currentx)
  c1_x[i] <- ifelse(runif(1) < A, proposedx, currentx)
}
c1_x = c1_x[-(1:burnin)] #remove burnins
c1_hist <- qplot(x = c1_x, y = ..density.., geom="histogram",
  bins = 25, fill=I("grey"), col=I("black"), xlab = "x", ylab = "Density")
```

```
c1_acf <- ggAcf(c1_x)
gridExtra::grid.arrange(grobs = list(c1_hist, c1_acf), nrow=2)
```



```
c1_mean <- mean(c1_x)
c1_var <- var(c1_x)
```

- mean: 1.5730269
- variance: 0.4661384

C2.

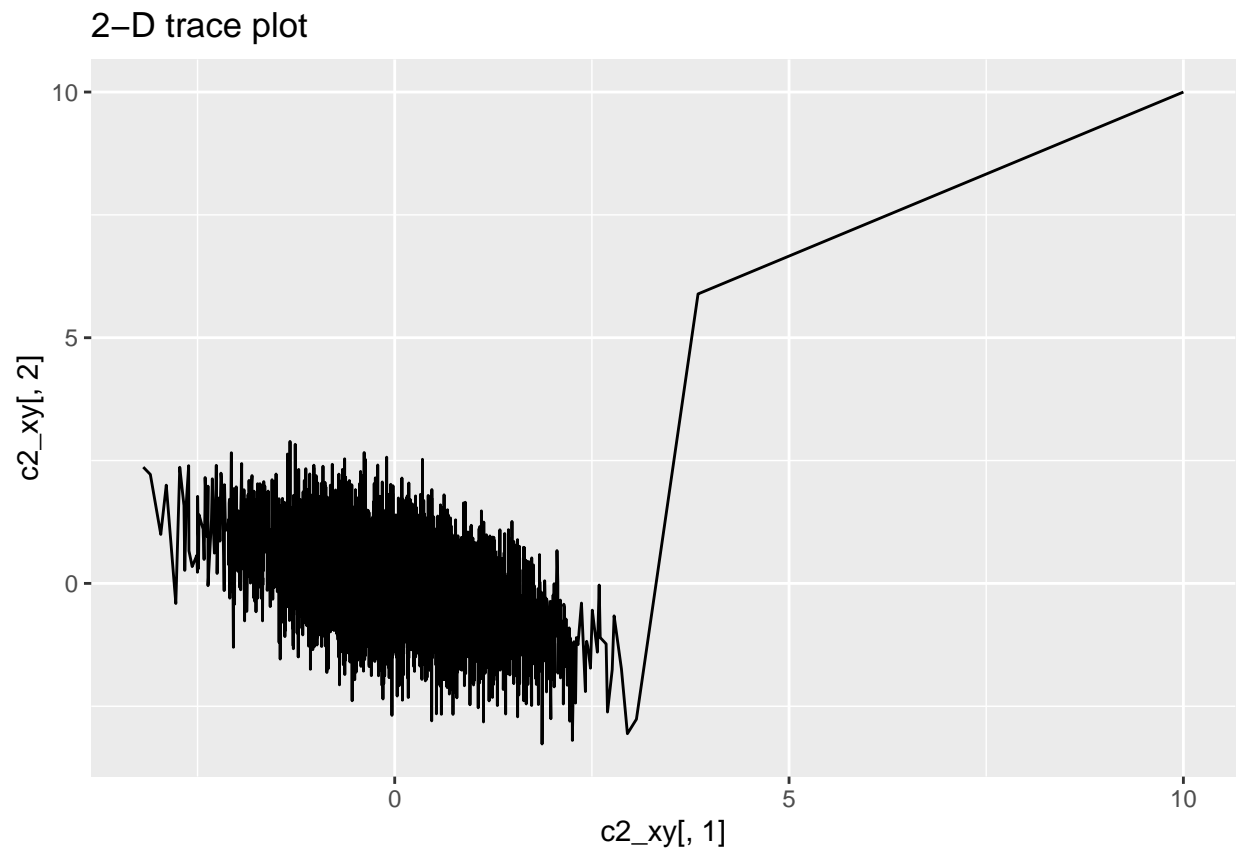
```
c2_target = function(x, y) return(exp(-(x^2+x*y+y^2)))
c2_target_log = function(x, y) return(-(x^2+x*y+y^2))
c2_xy <- xy
sd <- 5

for(i in 2:n)
{
  currentx <- c2_xy[i - 1, 1]
  currenty <- c2_xy[i - 1, 2]
  proposedx <- rnorm(1, currentx, sd)
  proposedy <- rnorm(1, currenty, sd)
  log_A <- c2_target_log(proposedx, proposedy) - c2_target_log(currentx, currenty)
  A <- exp(log_A)
```

```

# c2_xy[i,] <- ifelse(runif(1) < A, c(proposedx, proposedy), c(currentx, currenty))
if(runif(1) < A){
  c2_xy[i,1] <- proposedx
  c2_xy[i,2] <- proposedy
}else {
  c2_xy[i,1] <- currentx
  c2_xy[i,2] <- currenty
}
}
qplot(c2_xy[,1], c2_xy[,2], geom="line", main = "2-D trace plot")

```

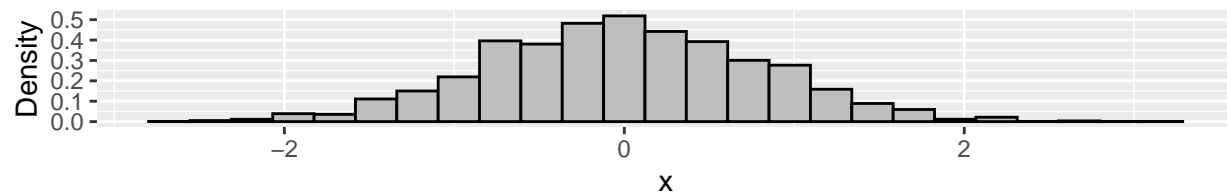


```

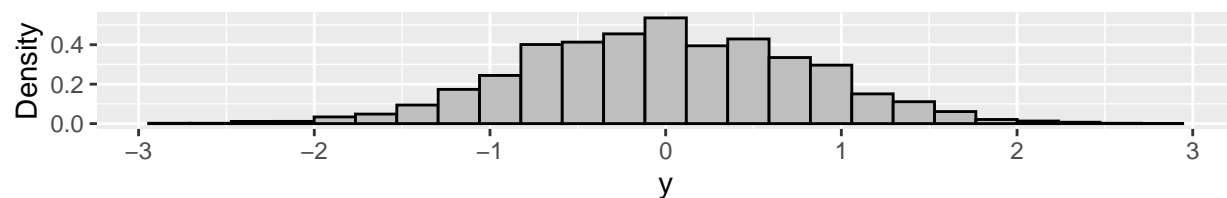
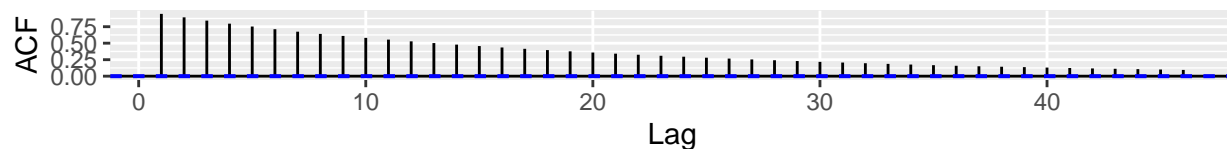
#remove burning
c2_xy <- c2_xy[-(1:burnin),]

#graphs
c2x_hist <- qplot(x = c2_xy[,1], y = ..density.., geom="histogram",
  bins = 25, fill=I("grey"), col=I("black"), xlab = "x", ylab = "Density")
c2x_acf <- ggAcf(c2_xy[,1], main = "x ACF")
c2y_hist <- qplot(x = c2_xy[,2], y = ..density.., geom="histogram",
  bins = 25, fill=I("grey"), col=I("black"), xlab = "y", ylab = "Density")
c2y_acf <- ggAcf(c2_xy[,2], main = "y ACF")
gridExtra::grid.arrange(grobs = list(c2x_hist, c2x_acf, c2y_hist, c2y_acf), nrow=4)

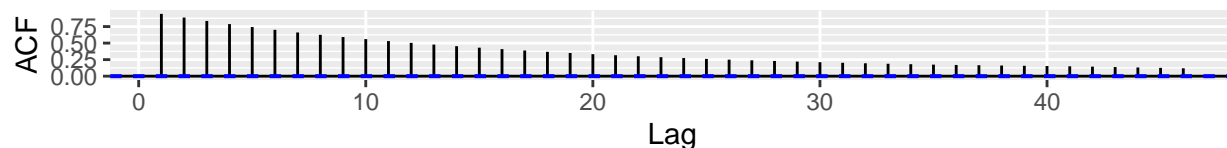
```



x ACF



y ACF



```
# plug x, y samples into target function
c2 <- c2_target(c2_xy[,1], c2_xy[,2])

#summary of target samples
c2_mean <- mean(c2)
c2_var <- var(c2)
```

- mean: 0.5156086
- variance: 0.0814089

C3.

```
c3_target = function(x, y) exp(-(x^3+x*y+y^2))
c3_target_log_x = function(x, y) -(x^3+x*y)
c3_target_log_y = function(x, y) -(x*y+y^2)
c3_xy = xy
sd <- 5

for(i in 2:n)
{
  currentx <- c3_xy[i - 1, 1]
  currenty <- c3_xy[i - 1, 2]

  #MH for x|y[i-1]
```

```

proposedx <- rnorm(1, currentx, sd)

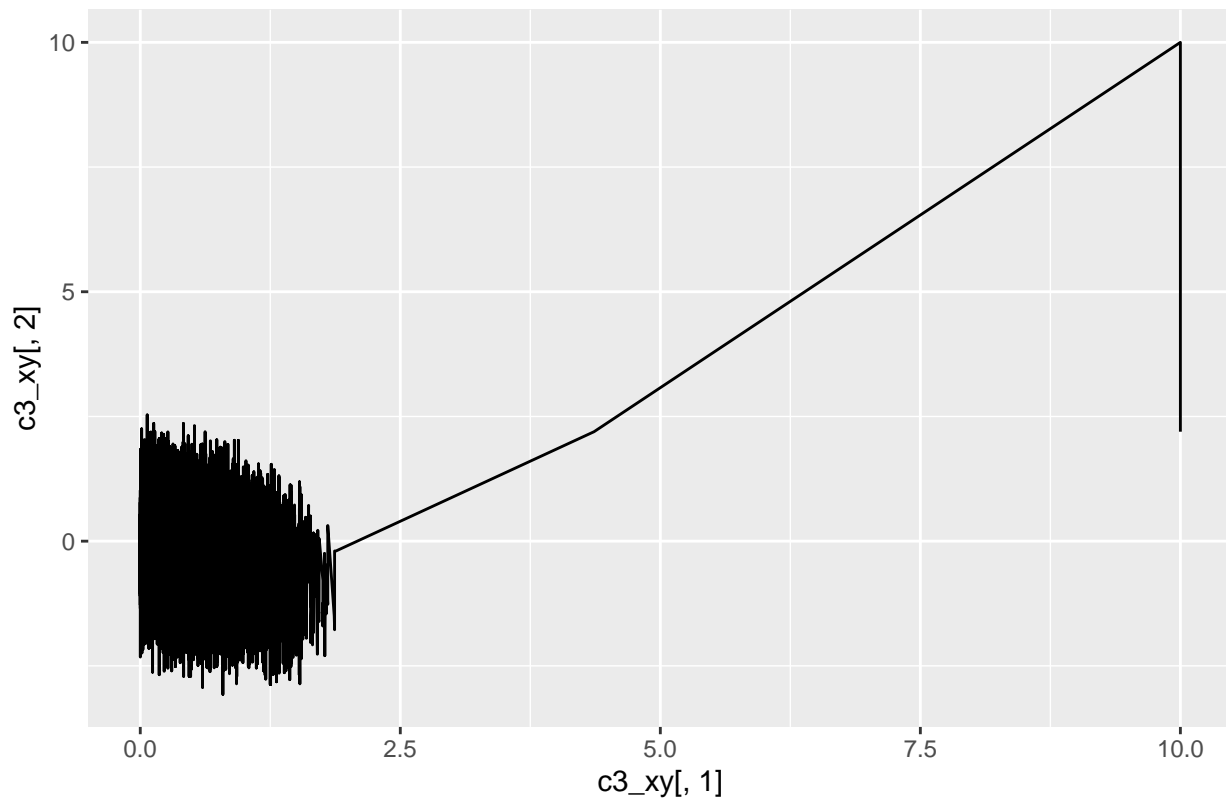
#check if proposed x is outside boundary
if (proposedx <= 0) {
  c3_xy[i,1] <- currentx
} else {
  log_A <- c3_target_log_x(proposedx, currenty) - c3_target_log_x(currentx, currenty)

  #proposal ratio of truncated normal
  c <- dtnorm(x = currentx, mean = proposedx, sd = sd, lower = 0) /
    dtnorm(x = proposedx, mean = currentx, sd = sd, lower = 0)
  A <- exp(log_A) * c
  c3_xy[i,1] <- ifelse(runif(1) < A, proposedx, currentx)
}

#MH for y|x[i]
current_x <- c3_xy[i, 1]
proposedy <- rnorm(1, currenty, sd)
log_A <- c3_target_log_y(current_x, proposedy) - c3_target_log_y(current_x, currenty)
A <- exp(log_A)
c3_xy[i,2] <- ifelse(runif(1) < A, proposedy, currenty)
}
qplot(c3_xy[,1], c3_xy[,2], geom="line", main = "2-D trace plot")

```

2-D trace plot

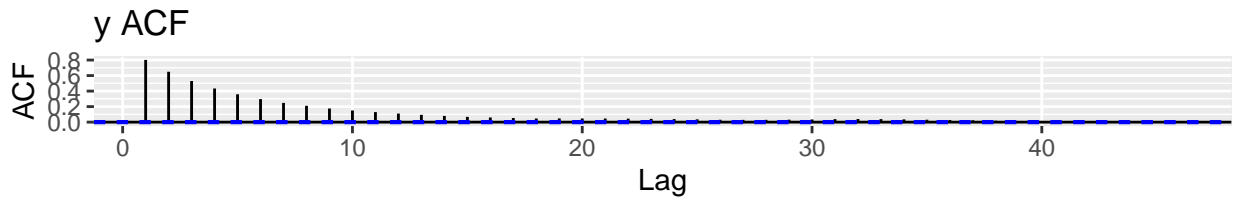
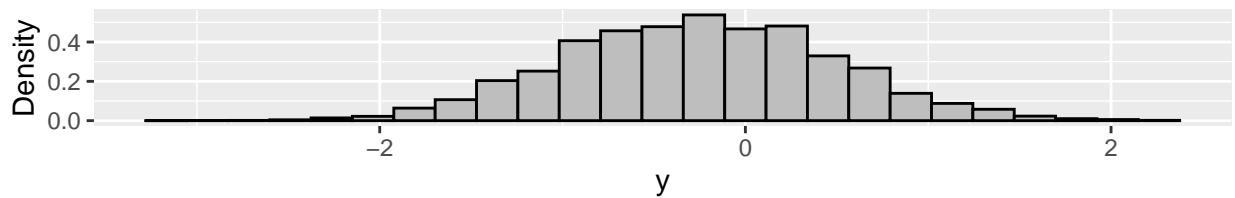
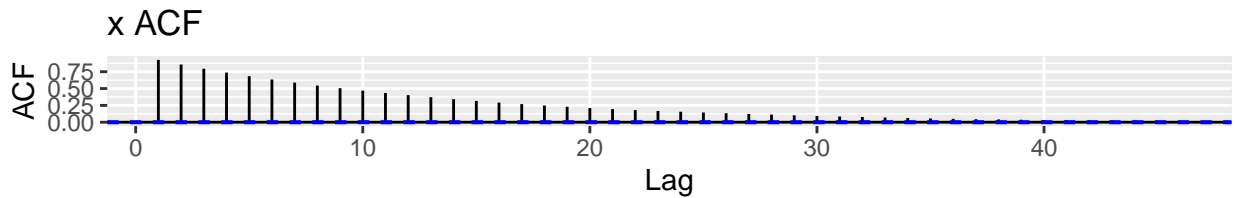
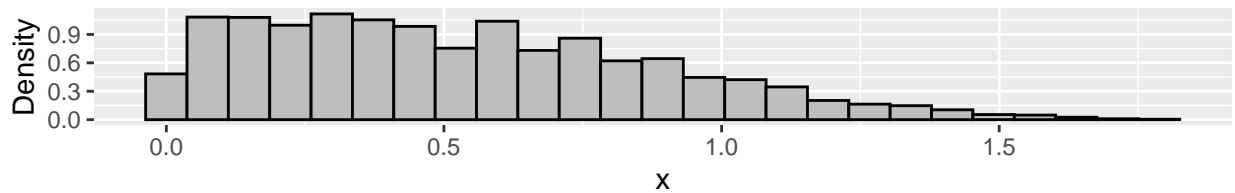


```

#remove burning
c3_xy <- c3_xy[-(1:burnin),]

#graphs
c3x_hist <- qplot(x = c3_xy[,1], y = ..density.., geom="histogram",
                 bins = 25, fill=I("grey"), col=I("black"), xlab = "x", ylab = "Density")
c3x_acf <- ggAcf(c3_xy[,1], main = "x ACF")
c3y_hist <- qplot(x = c3_xy[,2], y = ..density.., geom="histogram",
                 bins = 25, fill=I("grey"), col=I("black"), xlab = "y", ylab = "Density")
c3y_acf <- ggAcf(c3_xy[,2], main = "y ACF")
gridExtra::grid.arrange(grobs = list(c3x_hist, c3x_acf, c3y_hist, c3y_acf), nrow=4)

```



```

# plug x, y samples into target function
c3 <- c3_target(c3_xy[,1], c3_xy[,2])

#summary of target samples
c3_mean <- mean(c3)
c3_var <- var(c3)

```

- mean: 0.5753935
- variance: 0.085473

C4.


```

c4_target = function(x, y) exp(-(x^3+x*y+y^3))
c4_target_log_x = function(x, y) -(x^3+x*y)
c4_target_log_y = function(x, y) -(x*y+y^3)
c4_xy = xy
sd <- 5

for(i in 2:n)
{

  currentx <- c4_xy[i - 1, 1]
  currenty <- c4_xy[i - 1, 2]

  #MH for x|y[i-1]
  proposedx <- rnorm(1, currentx, sd)

  #check if proposed x is outside boundary
  if (proposedx <= 0) {
    c4_xy[i,1] <- currentx
  } else {
    log_A <- c4_target_log_x(proposedx, currenty) - c4_target_log_x(currentx, currenty)

    #proposal ratio of truncated normal
    c <- dtnorm(x = currentx, mean = proposedx, sd = sd, lower = 0) /
      dtnorm(x = proposedx, mean = currentx, sd = sd, lower = 0)
    A <- exp(log_A) * c
    c4_xy[i,1] <- ifelse(runif(1) < A, proposedx, currentx)
  }

  #MH for y|x1[i]
  current_x <- c4_xy[i, 1]
  proposedy <- rnorm(1, currenty, sd)

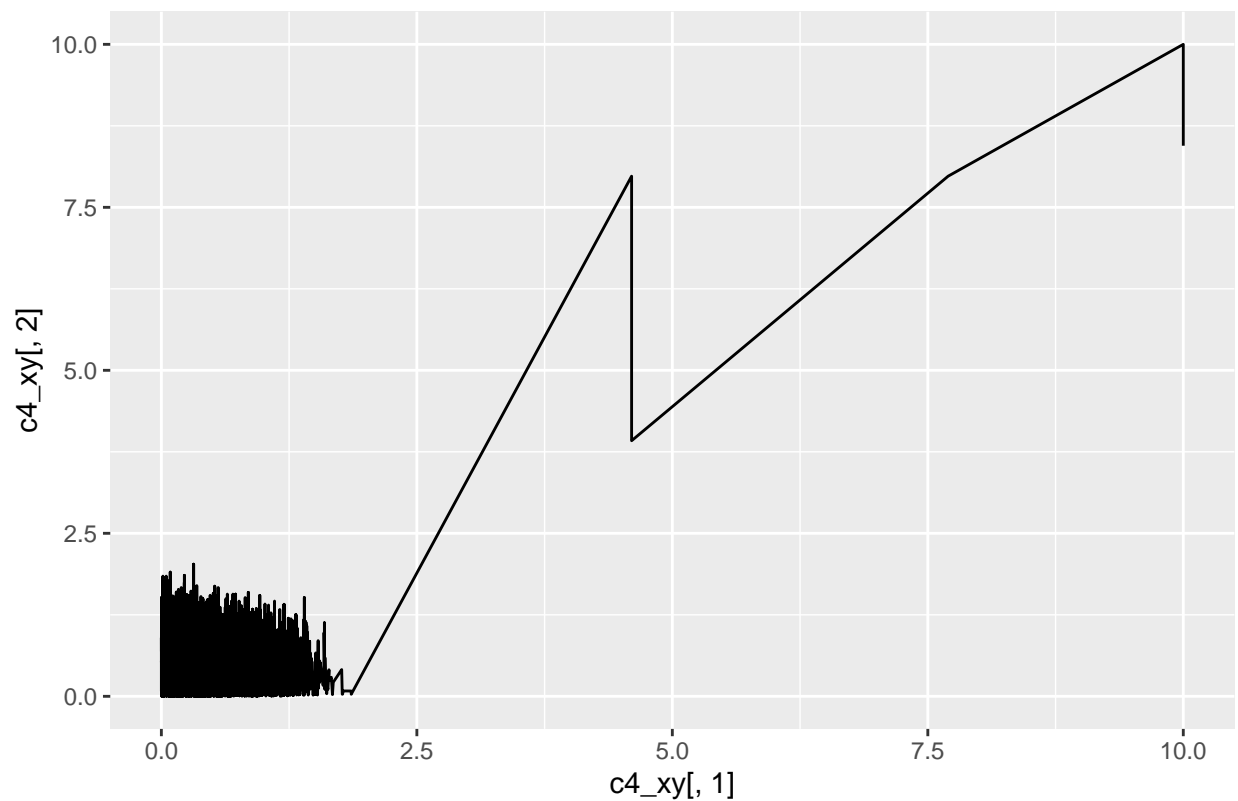
  #check if proposed y is outside boundary
  if (proposedy <= 0) {
    c4_xy[i,2] <- currenty
  } else {
    log_A <- c4_target_log_y(current_x, proposedy) - c4_target_log_y(current_x, currenty)

    #proposal ratio of truncated normal
    c <- dtnorm(x = currenty, mean = proposedy, sd = sd, lower = 0) /
      dtnorm(x = proposedy, mean = currenty, sd = sd, lower = 0)
    A <- exp(log_A) * c
    c4_xy[i,2] <- ifelse(runif(1) < A, proposedy, currenty)
  }
}

qplot(c4_xy[,1], c4_xy[,2], geom="line", main = "2-D trace plot")

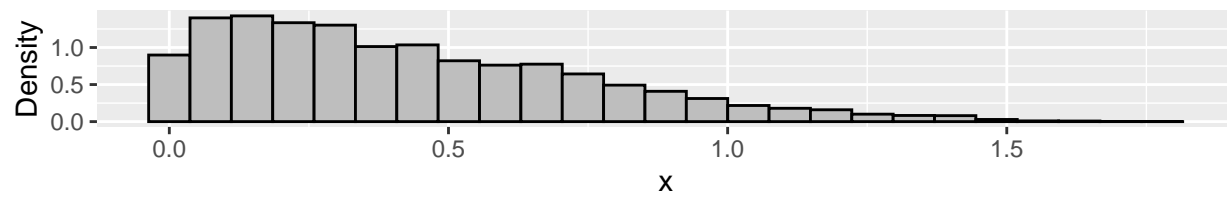
```

2-D trace plot

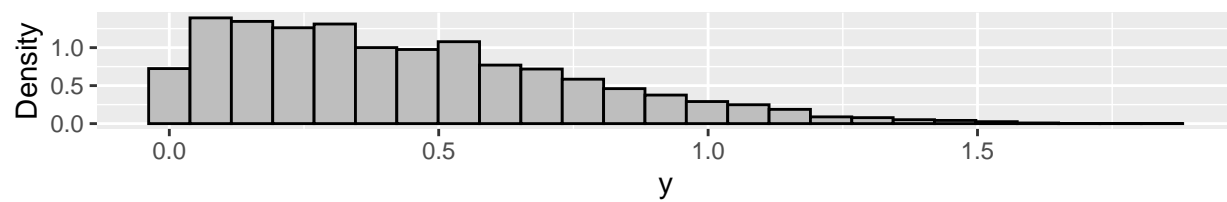
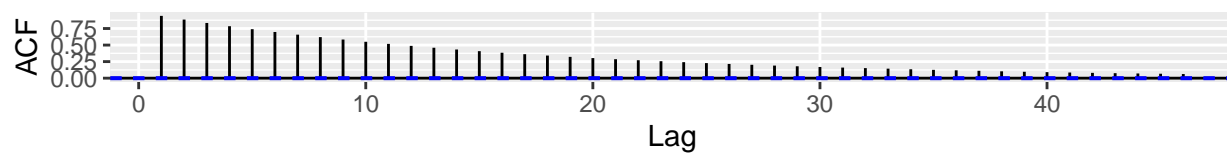


```
#remove burning
c4_xy <- c4_xy[-(1:burnin),]

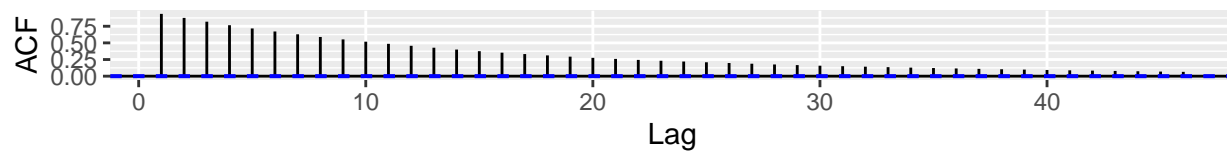
#graphs
c4x_hist <- qplot(x = c4_xy[,1], y =..density.., geom="histogram",
                 bins = 25, fill=I("grey"), col=I("black"), xlab = "x", ylab = "Density")
c4x_acf <- ggAcf(c4_xy[,1], main = "x ACF")
c4y_hist <- qplot(x = c4_xy[,2], y =..density.., geom="histogram",
                 bins = 25, fill=I("grey"), col=I("black"), xlab = "y", ylab = "Density")
c4y_acf <- ggAcf(c4_xy[,2], main = "y ACF")
gridExtra::grid.arrange(grobs = list(c4x_hist, c4x_acf, c4y_hist, c4y_acf), nrow=4)
```



x ACF



y ACF



```
#plug x, y samples into target function
c4 <- c4_target(c4_xy[,1], c4_xy[,2])

#summary of target samples
c4_mean <- mean(c4)
c4_var <- var(c4)
```

- mean: 0.6234716
- variance: 0.0808767