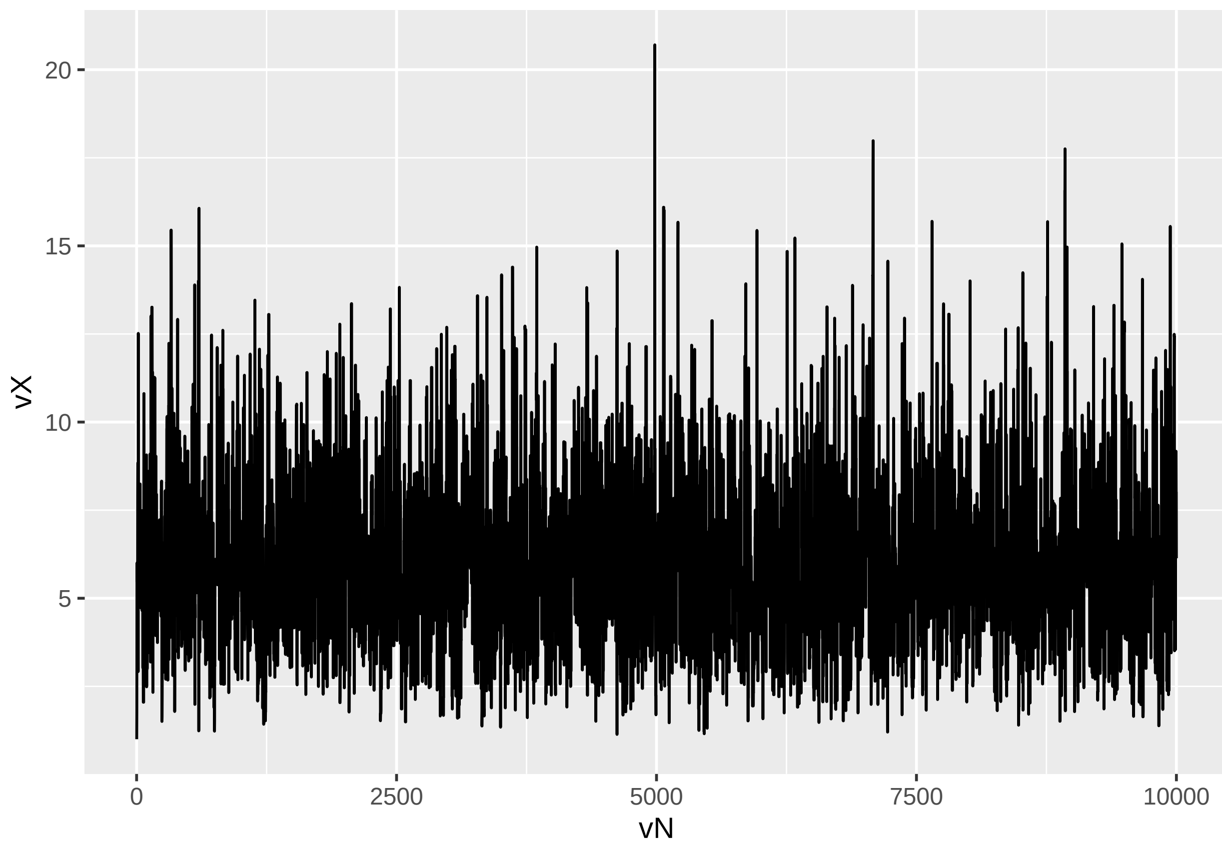
Lab 4

Group 4

1.1

```r
f_x <- function(x) {
  return(ifelse(x > 0, x^5 * exp(1)^(-x), NA))
}

mh_lnorm <- function(nstep, X0, props) {
  vN <- 1:nstep
  vX <- rep(X0, nstep)
  for(i in 2:nstep) {
    X <- vX[i-1]
    Y <- rlnorm(1, meanlog = log(X), sdlog = props)
    u <- runif(1)
    a <- min(c(1, (f_x(Y) * dlnorm(X, meanlog = log(Y), sdlog = props)) /
                 (f_x(X) * dlnorm(Y, meanlog = log(X), sdlog = props))))
    if(u <= a) {
      vX[i] <- Y
    }
    else {
      vX[i] <- X
    }
  }
  return(data.frame(vX, vN))
}

res_lnorm <- mh_lnorm(10000, 1, 1)

ggplot(res_lnorm, aes(x = vN, y = vX)) +
  geom_line()
```

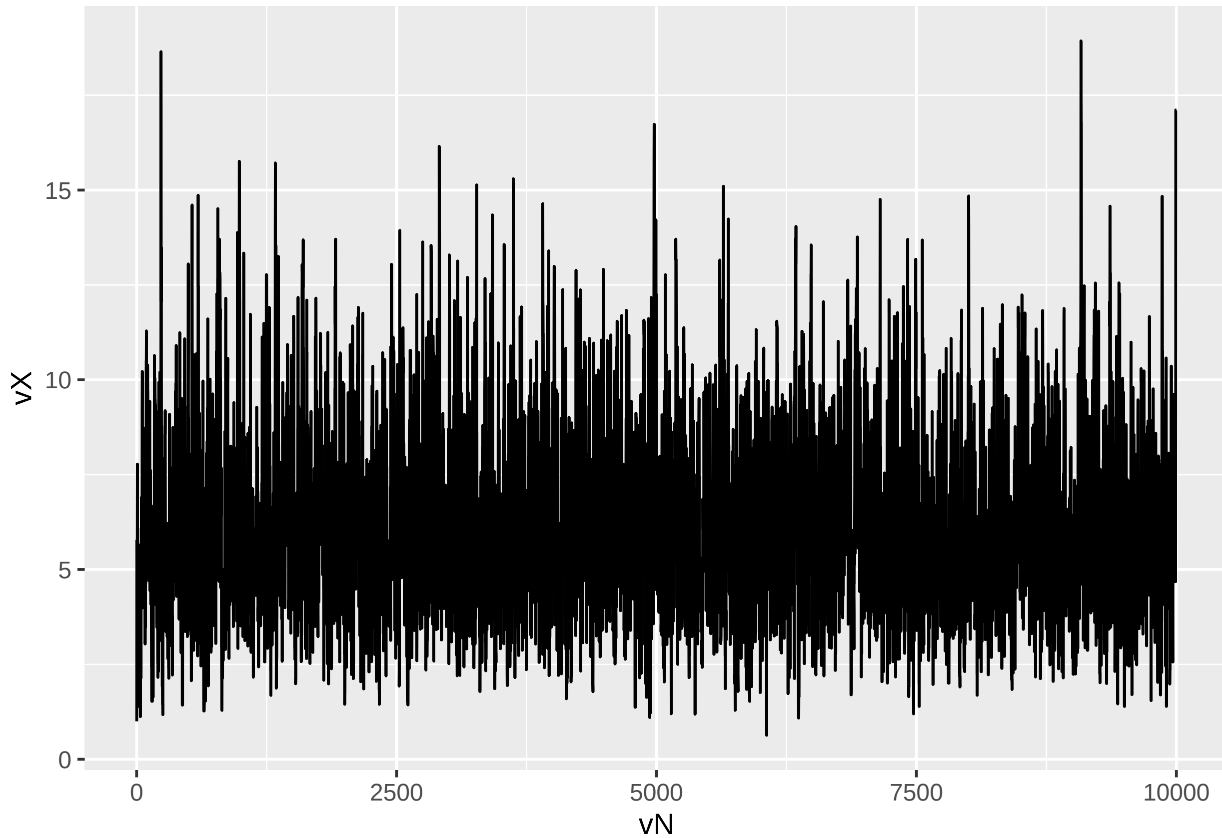


Neither the mean nor the variance appear to change over the iterations which indicates that the chain has converged to a stationary distribution. There does not seem to be any clear burn-in period.

1.2

```
mh_chisq <- function(nstep, X0) {
  vN <- 1:nstep
  vX <- rep(X0, nstep)
  for(i in 2:nstep) {
    X <- vX[i-1]
    Y <- rchisq(1, df = floor(X+1))
    u <- runif(1)
    a <- min(c(1, (f_x(Y) * dchisq(X, df = floor(Y+1))) / (f_x(X) * dchisq(Y, df = floor(X+1)))))
    if(u <= a) {
      vX[i] <- Y
    }
    else {
      vX[i] <- X
    }
  }
  return(data.frame(vX, vN))
}

res_chisq <- mh_chisq(10000, 1)
```

```r
ggplot(res_chisq, aes(x = vN, y = vX)) +
  geom_line()
```



The chain appear to almost immediately converge to a stationary distribution as neither the mean nor the variance seem to change over the iterations.

1.3

```r
library(coda)
chains <- mcmc.list()

for(i in 1:10) {
  chains[[i]] <- mcmc(mh_chisq(1000, i)[,1])
}

gelman.diag(chains)
```

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]       1.01       1.02
```

The value of the Gelman-Rubin factor $\sqrt{R}$ is close to 1 and ≤ 1.2 which indicates that the chain has achieved convergence.

1.4

The integral is equal to the expected value of X (i.e. the mean):

$$\int_0^\infty xf(x)\ dx = E(X)$$

Mean of the samples from step 1:

```
mean(res_lnorm[,1])
```

```
## [1] 6.037219
```

The mean of the samples from step 2:

```
mean(res_chisq[,1])
```

```
## [1] 6.067646
```

1.5

The expected value of a gamma distribution is:

$$E(X) = \frac{\alpha}{\beta} = \frac{6}{1} = 6$$

This is very close to the mean of the samples from step 6, which is further proof that the chain has converged and that the chi-square distribution gives a good approximation of the target distribution.

The expected value of a gamma distribution with parameters $\alpha = 6$ and $\beta = 1$ is very close to the mean of the samples from step 1 and 2.

2.1

Likelihood:

$$p(\vec{Y}|\vec{\mu}) = \prod_{i=2}^{n}(2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \mu_i)^2\right)$$
$$= (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu_i)^2\right)$$

Prior:

$$p(\mu_1) = 1$$

$$p(\mu_{i+1}|\mu_i) = N(\mu_i, 0.2), i = 1, ..., n-1$$

$$p(\vec{\mu}) = p(\mu_1)p(\mu_2|\mu_1)p(\mu_3|\mu_2)...p(\mu_n|\mu_{n-1}) =$$
$$= \prod_{i=1}^{n-1}(2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}(\mu_{i+1} - \mu_i)^2\right)$$

$$= (2\pi\sigma^2)^{-\frac{n-1}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n-1}(\mu_{i+1} - \mu_i)^2\right)$$

2.2

Posterior:

$$p(\vec{\mu}|\vec{Y}) \propto p(\vec{Y}|\vec{\mu}) \cdot p(\vec{\mu})$$

For $(\mu_1|\vec{\mu}_{-1}, \vec{Y})$:

$$p(\mu_1|\vec{\mu}_{-1}, \vec{Y}) \propto \exp\left(-\frac{1}{2\sigma^2}\left((y_1-\mu_1)^2 + (\mu_2-\mu_1)^2\right)\right)$$

$$\propto \exp\left(-\frac{\left(\mu_1 - \frac{y_1+\mu_2}{2}\right)^2}{\frac{2\sigma^2}{2}}\right)$$

$$\propto \exp\left(-\frac{1}{\sigma^2}\left(\mu_1 - \frac{y_1+\mu_2}{2}\right)^2\right)$$

$$\propto N\left(\frac{y_1+\mu_2}{2}, \frac{\sigma^2}{2}\right)$$

For $(\mu_n|\vec{\mu}_{-n}, \vec{Y})$:

$$p(\mu_n|\vec{\mu}_{-n}, \vec{Y}) \propto \exp\left(-\frac{1}{2\sigma^2}\left((y_n-\mu_n)^2 + (\mu_n-\mu_{n-1})^2\right)\right)$$

$$\propto \exp\left(-\frac{\left(\mu_n - \frac{y_n+\mu_{n-1}}{2}\right)^2}{\frac{2\sigma^2}{2}}\right)$$

$$\propto \exp\left(\frac{1}{\sigma^2}\left(\mu_n - \frac{y_n+\mu_{n-1}}{2}\right)^2\right)$$

$$\propto N\left(\frac{y_n+\mu_{n-1}}{2}, \frac{\sigma^2}{2}\right)$$

For $(\mu_i|\vec{\mu}_{-i}, \vec{Y})$:

$$p(\mu_i|\vec{\mu}_{-i}, \vec{Y}) \propto \exp\left(-\frac{1}{2\sigma^2}\left((\mu_i-\mu_{i-1})^2 + (y_i-\mu_i)^2 + (\mu_{i+1}-\mu_i)^2\right)\right)$$

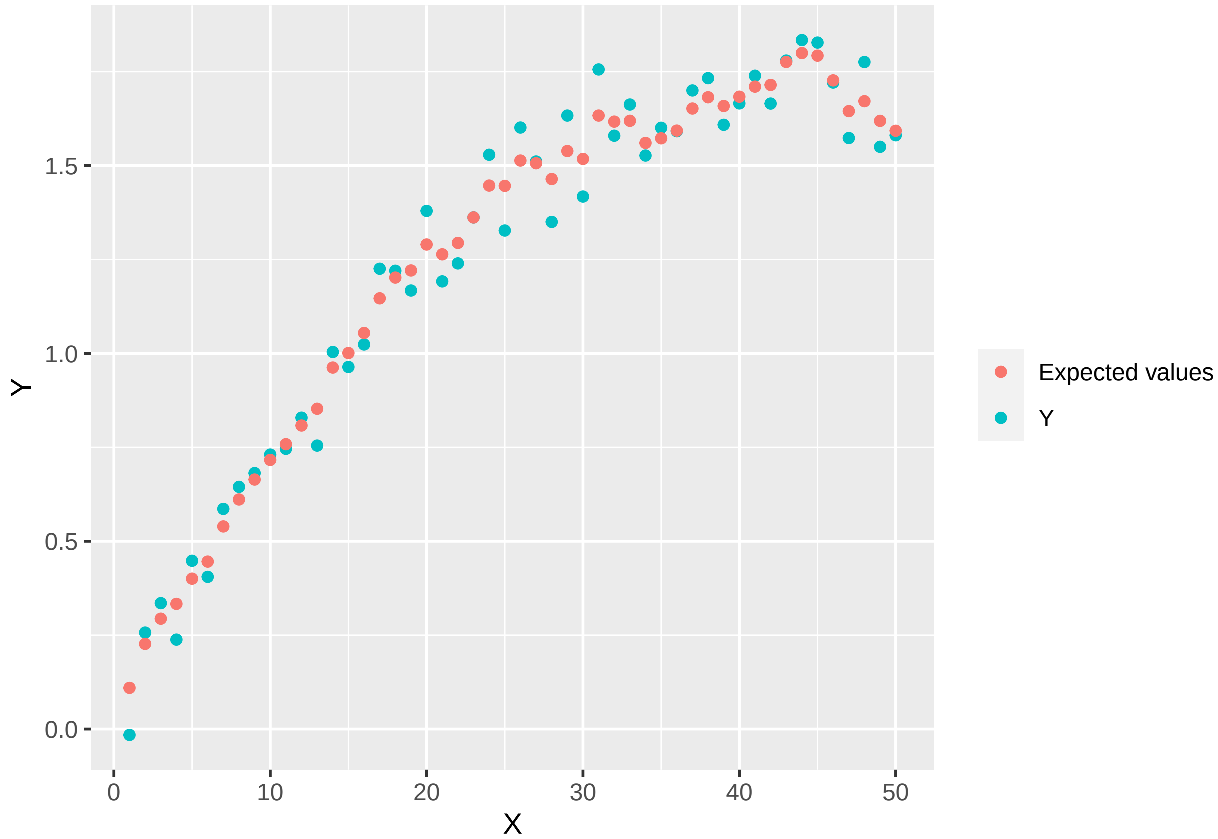
$$\propto \exp\left(-\frac{\left(\mu_i - \frac{\mu_{i-1}+y_i+\mu_{i+1}}{3}\right)^2}{\frac{2\sigma^2}{3}}\right)$$

$$\propto \exp\left(-\frac{3}{2\sigma^2}\left(\mu_i - \frac{\mu_{i-1}+y_i+\mu_{i+1}}{3}\right)^2\right)$$

$$\propto N\left(\frac{\mu_{i-1}+y_i+\mu_{i+1}}{3}, \frac{\sigma^2}{3}\right)$$

2.3

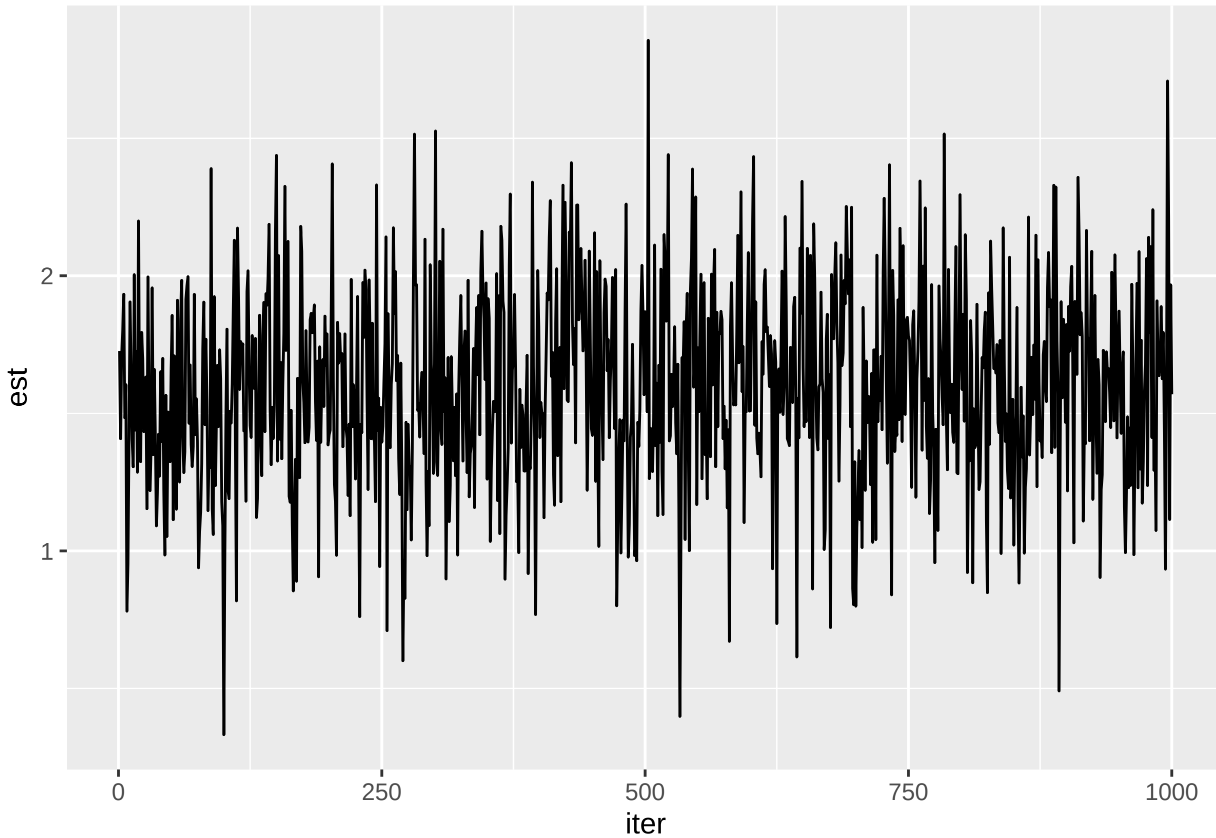
```r
load("chemical.RData")

df <- data.frame(X, Y)

gibbs_sampler <- function(data, n) {
  d <- nrow(data)
  mX <- matrix(0, nrow = n, ncol = d)
  for(i in 1:n) {
    for(j in 1:d) {
      if(j == 1) {
        mX[i,j] <- rnorm(n = 1,
                         mean = (data[1,2] + mX[i,2])/2,
                         sd = sqrt(0.2/2))
      }
      else if(j > 1 && j < d) {
        mX[i,j] <- rnorm(n = 1,
                         mean = (mX[i,(j-1)] + data[j,2] + mX[i,(j+1)])/3,
                         sd = sqrt(0.2/3))
      }
      else if(j == d) {
        mX[i,j] <- rnorm(n = 1,
                         mean = (data[d,2] + mX[i,(d-1)])/2,
                         sd = sqrt(0.2/2))
      }
    }
    if(i < n) {
      mX[i+1,] <- mX[i,]
    }
  }
  return(mX)
}

samples <- gibbs_sampler(n = 1000, data = df)

m_samples <- colMeans(samples)

df$`Expected values` <- colMeans(samples)
df_long <- pivot_longer(df, cols = c(2,3))

ggplot(df_long, aes(x = X, y = value, color = name)) +
  geom_point() +
  ylab("Y") +
  theme(legend.title=element_blank())
```



Much of the noise appear to have been removed, and the expected value of $\vec{\mu}$ seem to catch the true underlying dependence between Y and X very well.

2.4

```
df2 <- data.frame(est = samples[,50])
df2$iter <- 1:nrow(df2)

ggplot(df2, aes(x = iter, y = est)) +
  geom_line()
```



The mean and the variance does not seem to change over the iterations, which indicates that the chain as good as immediately converges to a stationary distribution without any apparent burn-in period.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
library(tidyverse)
set.seed(12345)
setwd("C:/Users/Simon/Desktop/Statistik/Computational statistics/Labs/lab4/komplettering")
f_x <- function(x) {
  return(ifelse(x > 0, x^5 * exp(1)^(-x), NA))
}

mh_lnorm <- function(nstep, X0, props) {
  vN <- 1:nstep
  vX <- rep(X0, nstep)
  for(i in 2:nstep) {
    X <- vX[i-1]
    Y <- rlnorm(1, meanlog = log(X), sdlog = props)
    u <- runif(1)
    a <- min(c(1, (f_x(Y) * dlnorm(X, meanlog = log(Y), sdlog = props)) /
                 (f_x(X) * dlnorm(Y, meanlog = log(X), sdlog = props))))
    if(u <= a) {
```

```
      vX[i] <- Y
    }
    else {
      vX[i] <- X
    }
  }
  return(data.frame(vX, vN))
}

res_lnorm <- mh_lnorm(10000, 1, 1)

ggplot(res_lnorm, aes(x = vN, y = vX)) +
  geom_line()
mh_chisq <- function(nstep, X0) {
  vN <- 1:nstep
  vX <- rep(X0, nstep)
  for(i in 2:nstep) {
    X <- vX[i-1]
    Y <- rchisq(1, df = floor(X+1))
    u <- runif(1)
    a <- min(c(1, (f_x(Y) * dchisq(X, df = floor(Y+1))) / (f_x(X) * dchisq(Y, df = floor(X+1)))))
    if(u <= a) {
      vX[i] <- Y
    }
    else {
      vX[i] <- X
    }
  }
  return(data.frame(vX, vN))
}

res_chisq <- mh_chisq(10000, 1)

ggplot(res_chisq, aes(x = vN, y = vX)) +
  geom_line()
library(coda)
chains <- mcmc.list()

for(i in 1:10) {
  chains[[i]] <- mcmc(mh_chisq(1000, i)[,1])
}

gelman.diag(chains)
mean(res_lnorm[,1])

mean(res_chisq[,1])
load("chemical.RData")

df <- data.frame(X, Y)

gibbs_sampler <- function(data, n) {
  d <- nrow(data)
  mX <- matrix(0, nrow = n, ncol = d)
```

```
  for(i in 1:n) {
    for(j in 1:d) {
      if(j == 1) {
        mX[i,j] <- rnorm(n = 1,
                         mean = (data[1,2] + mX[i,2])/2,
                         sd = sqrt(0.2/2))
      }
      else if(j > 1 && j < d) {
        mX[i,j] <- rnorm(n = 1,
                         mean = (mX[i,(j-1)] + data[j,2] + mX[i,(j+1)])/3,
                         sd = sqrt(0.2/3))
      }
      else if(j == d) {
        mX[i,j] <- rnorm(n = 1,
                         mean = (data[d,2] + mX[i,(d-1)])/2,
                         sd = sqrt(0.2/2))
      }
    }
    if(i < n) {
      mX[i+1,] <- mX[i,]
    }
  }
  return(mX)
}

samples <- gibbs_sampler(n = 1000, data = df)

m_samples <- colMeans(samples)

df$`Expected values` <- colMeans(samples)
df_long <- pivot_longer(df, cols = c(2,3))

ggplot(df_long, aes(x = X, y = value, color = name)) +
  geom_point() +
  ylab("Y") +
  theme(legend.title=element_blank())
df2 <- data.frame(est = samples[,50])
df2$iter <- 1:nrow(df2)

ggplot(df2, aes(x = iter, y = est)) +
  geom_line()
```