STAT 403 HW5

Chongyi Xu

May 2, 2018

Question 1

(a) First generate $n = 500$ data points such that $X_1, \cdots, X_n \sim$N(0,1) and $Y_1, \cdots, Y_n$ from

$$P(Y_i = 1|X_i) = \frac{e^{1+2X_i}}{1 + e^{1+2X_i}}$$

Fit the logistic regression, what are the fitted parameter $\hat{\beta}_0$ and $\hat{\beta}_1$?

```
n <- 500
set.seed(403)
x <- rnorm(n, mean=0, sd=1)
p <- exp(1+2*x)/(1 + exp(1+2*x))
y <- rbinom(n,1,p)

model.fit <- glm(y~x, family=binomial)
model.fit
```

```
##
## Call:  glm(formula = y ~ x, family = binomial)
##
## Coefficients:
## (Intercept)            x
##      0.8375       1.9080
##
## Degrees of Freedom: 499 Total (i.e. Null);  498 Residual
## Null Deviance:       656.8
## Residual Deviance: 450.6     AIC: 454.6
```

```
print(paste('The fitted parameter beta_0 is ', model.fit$coefficients[1]))
```

```
## [1] "The fitted parameter beta_0 is  0.837467947867638"
```

```
print(paste('The fitted parameter beta_1 is ', model.fit$coefficients[2]))
```

```
## [1] "The fitted parameter beta_1 is  1.90797191397814"
```

(b) Using Monte Carlo simulation to repeat the above procedure $N = 2000$ times. Use two histograms to show the distribution of them.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

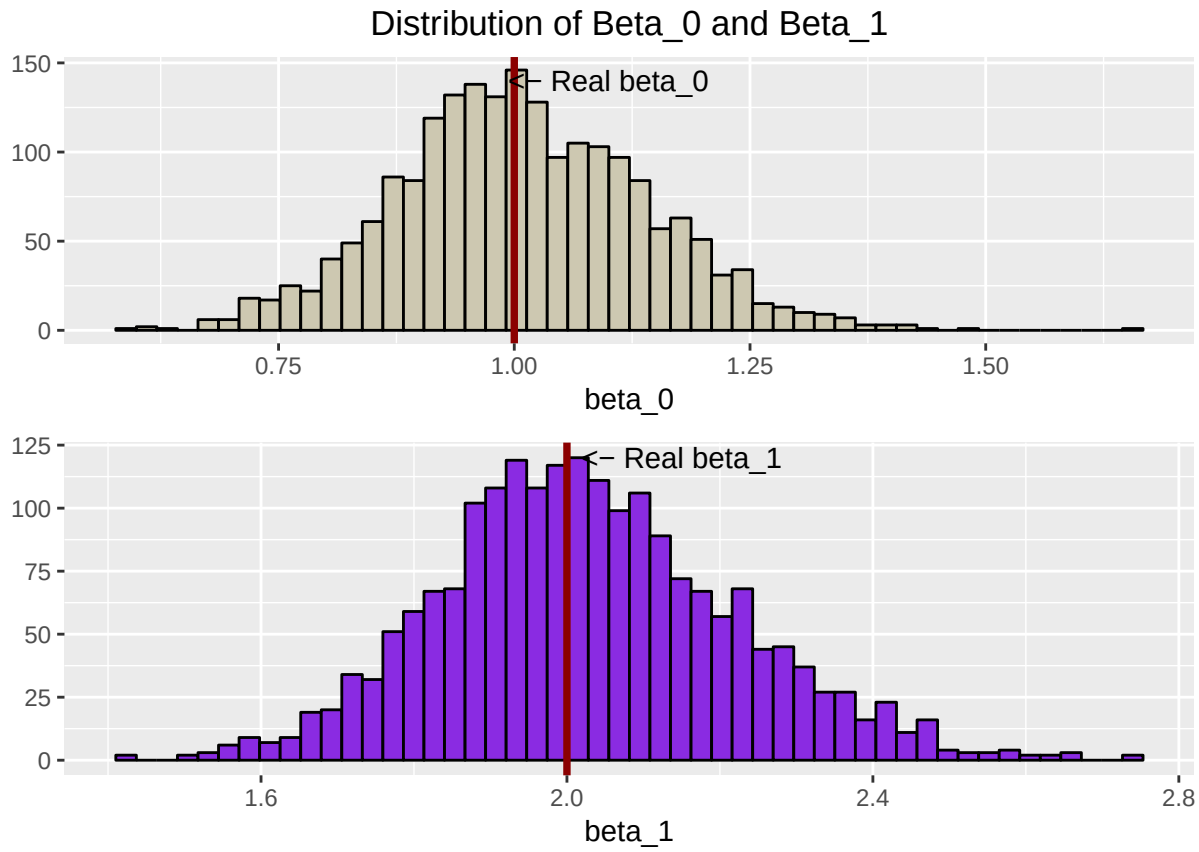
```
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```
## Warning: package 'gridExtra' was built under R version 3.4.4
```

```
N <- 2000
b0 <- rep(0, N)
b1 <- rep(0, N)
```

```r
n <- 500
set.seed(403)
for (i in 1:N) {
  x <- rnorm(n, mean=0, sd=1)
  p <- exp(1+2*x)/(1 + exp(1+2*x))
  model.fit <- glm(rbinom(n,1,p)~x, family=binomial)
  b0[i] <- model.fit$coefficients[1]
  b1[i] <- model.fit$coefficients[2]
}

p1 <- ggplot() + geom_histogram(aes(b0),
                           bins=50,
                           color="black",
                           fill="cornsilk3") +
  geom_vline(xintercept=1,
             color='darkred',
             size=1.3) +
  annotate('text', x=1.1,
           y=140, label='<- Real beta_0') +
  xlab('beta_0') + ylab('') +
  ggtitle('Distribution of Beta_0 and Beta_1') +
  theme(plot.title = element_text(hjust = 0.5))

p2 <- ggplot() + geom_histogram(aes(b1),
                           bins=50,
                           color="black",
                           fill="blueviolet") +
  geom_vline(xintercept=2,
             color='darkred',
             size=1.3) +
  annotate('text', x=2.15,
           y=120, label='<- Real beta_1') +
  xlab('beta_1') + ylab('')

grid.arrange(p1, p2, nrow=2)
```



(c) Juding from the previous two histograms, do $\hat{\beta}_0$ and $\hat{\beta}_1$ follow roughly a Normal distribution? Why or why not?

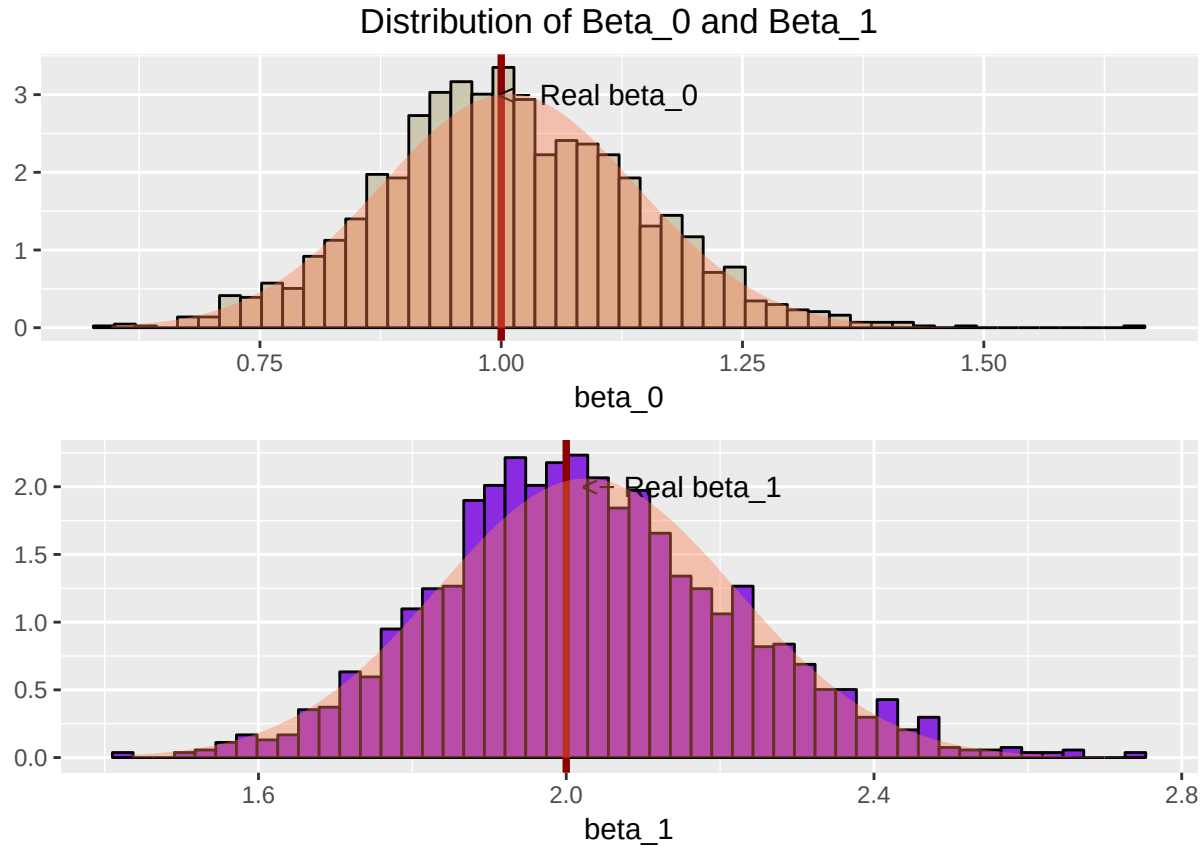
Yes they do. We can simply fit a normal distribution to the histogram.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.4.4
```

```
b0.fit <- fitdistr(b0, densfun='normal')
b1.fit <- fitdistr(b1, densfun='normal')

b0.x <- seq(from=min(b0), to=max(b0), length.out=length(b0))
b1.x <- seq(from=min(b1), to=max(b1), length.out=length(b1))

b0.d <- dnorm(b0.x, b0.fit$estimate[1], b0.fit$estimate[2])
b1.d <- dnorm(b1.x, b1.fit$estimate[1], b1.fit$estimate[2])

p1 <- ggplot() + geom_histogram(aes(b0, y=..density..),
                          bins=50,
                          color="black",
                          fill="cornsilk3") +
  geom_vline(xintercept=1,                                          color='darkred',
             size=1.3) +
  annotate('text', x=1.1,
           y=3, label='<- Real beta_0') +
  geom_area(aes(b0.x, b0.d),
            fill='coral', alpha=0.4) +
```

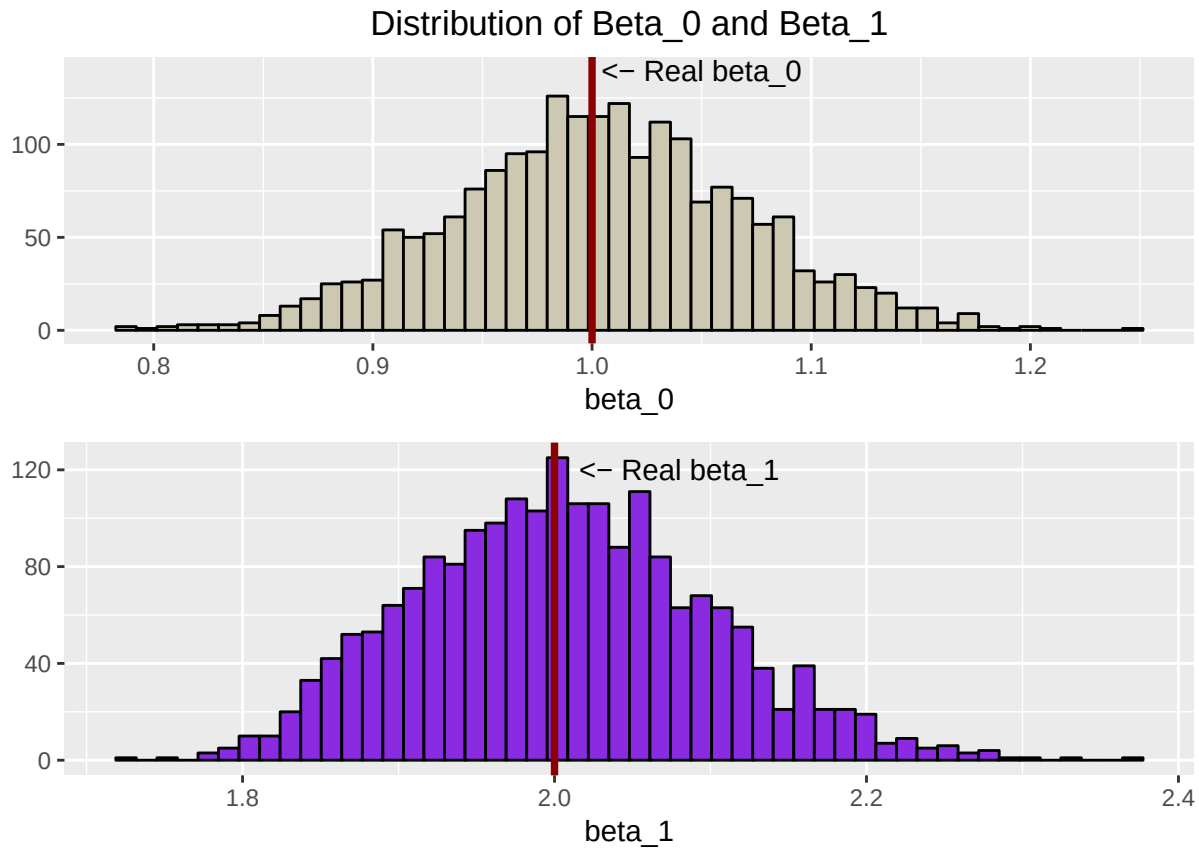
```
  xlab('beta_0') + ylab('') +
  ggtitle('Distribution of Beta_0 and Beta_1') +
  theme(plot.title = element_text(hjust = 0.5))

p2 <- ggplot() + geom_histogram(aes(b1, y=..density..),
                           bins=50,
                           color="black",
                           fill="blueviolet") +
  geom_vline(xintercept=2,
             color='darkred',
             size=1.3) +
  annotate('text', x=2.15,
           y=2, label='<- Real beta_1') +
  geom_area(aes(b1.x, b1.d),
            fill='coral',
            alpha=0.4) +
  xlab('beta_1') + ylab('')

grid.arrange(p1, p2, nrow=2)
```



From the previous plot, we can see that both fit the normal curve pretty well. Therefore, we can conclude that both follow roughly normal distribution.

(d) Now increase the sample size to $n = 2000$, repeat the procedure in question (b) and plot the histograms. Does the histogram concentrate more around the tru parameter values?

```r
N <- 2000
b0 <- rep(0, N)
b1 <- rep(0, N)
n <- 2000
set.seed(403)
for (i in 1:N) {
  x <- rnorm(n, mean=0, sd=1)
  p <- exp(1+2*x)/(1 + exp(1+2*x))
  model.fit <- glm(rbinom(n,1,p)~x, family=binomial)
  b0[i] <- model.fit$coefficients[1]
  b1[i] <- model.fit$coefficients[2]
}

p1 <- ggplot() + geom_histogram(aes(b0),
                           bins=50,
                           color="black",
                           fill="cornsilk3") +
  geom_vline(xintercept=1,
             color='darkred',
             size=1.3) +
  annotate('text', x=1.05,
           y=140, label='<- Real beta_0') +
  xlab('beta_0') + ylab('') +
  ggtitle('Distribution of Beta_0 and Beta_1') +
  theme(plot.title = element_text(hjust = 0.5))

p2 <- ggplot() + geom_histogram(aes(b1),
                           bins=50,
                           color="black",
                           fill="blueviolet") +
  geom_vline(xintercept=2,                                          color='darkred',
             size=1.3) +
  annotate('text', x=2.08,
           y=120, label='<- Real beta_1') +
  xlab('beta_1') + ylab('')

grid.arrange(p1, p2, nrow=2)
```

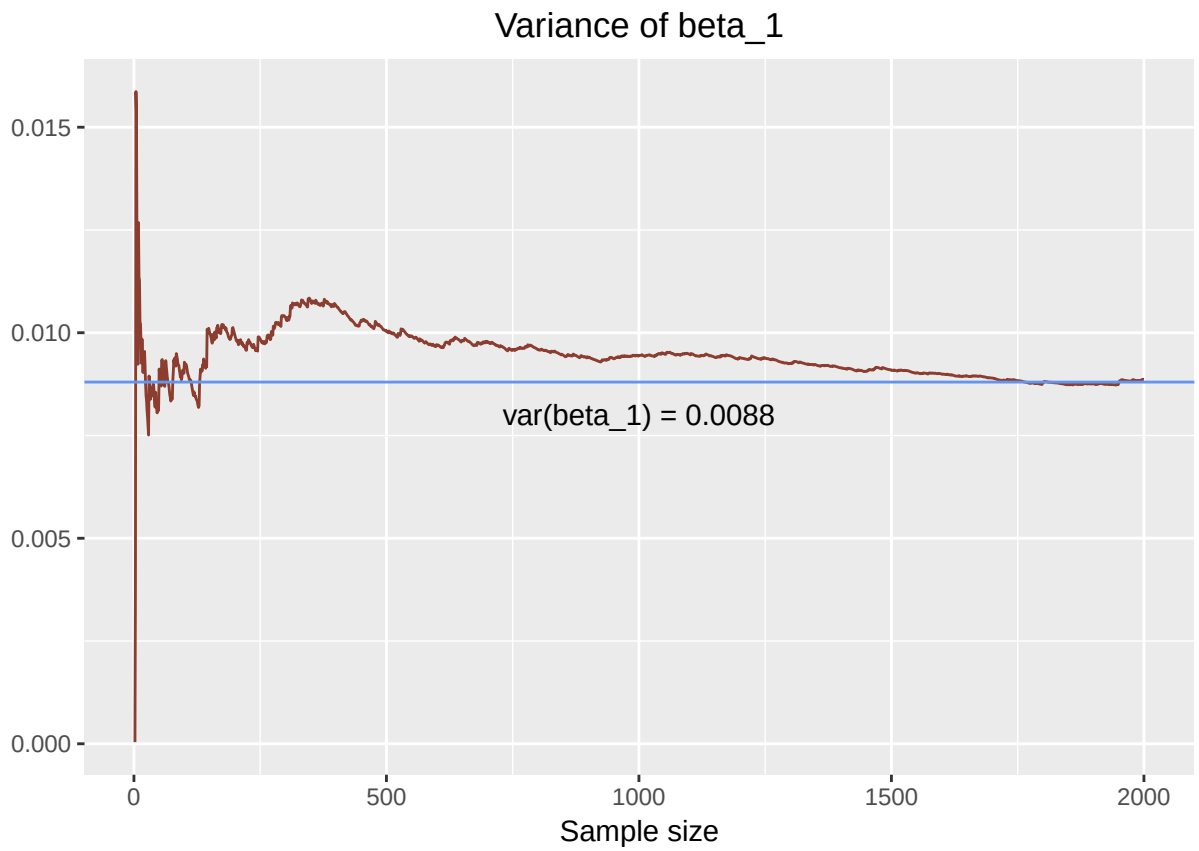


And comparing to the plot in part (b), the histogram does concentrate more around the true value.

(e) Now we focus on the slope parameter β_1. Use Monte Carlo simulations and a plot to illustrate the convergence of $\hat{\beta}_1$ towards β_1.

```
N <- 2000
b1 <- rep(0, N)
bi.var <- rep(0, N)
n <- 2000
set.seed(403)
for (i in 1:N) {
  x <- rnorm(n, mean=0, sd=1)
  p <- exp(1+2*x)/(1 + exp(1+2*x))
  model.fit <- glm(rbinom(n,1,p)~x, family=binomial)
  b1[i] <- model.fit$coefficients[2]
  if (i >= 2) {
    bi.var[i] <- var(b1[1:i])
  }
}
```

```
ggplot() + geom_line(aes(x=2:N, y=bi.var[-1]), color='coral4') + geom_hline(yintercept=0.0088, color='c
  ylab('') + annotate('text', x=1000, y=0.008, label='var(beta_1) = 0.0088') +
  ggtitle('Variance of beta_1') +
  theme(plot.title = element_text(hjust = 0.5))
```



From the plot, we can see that the variance of $\hat{\beta}_1$ converges as sample size increases.

Question 2

(a) If a random variable X has CDF $F(x) = f(x) = \frac{e^x}{1+e^x}$, what is the PDF p(x)? What are the mean and median of this random varaible?

The PDF of the random variable is just the derivative of the CDF $f(x) = \frac{e^x}{1+e^x}$

$$\begin{aligned}\frac{d}{dx}\frac{e^x}{1+e^x} &= \frac{e^x}{1+e^x} - \frac{e^{2x}}{(1+e^x)^2} \\ &= \frac{e^x(e^x+1) - e^{2x}}{(1+e^x)^2} \\ &= \frac{e^x}{(1+e^x)^2}\end{aligned}$$

Using a simulation to find the mean and median of this random variable

```
f <- function(x) {
  return(exp(x)/(1+exp(x))^2)
}
sample_size <- 1000
x <- seq(from=-10, to=-10, length.out=sample_size)
print(paste('The mean of the random variable is ', mean(f(x))))
```

```
## [1] "The mean of the random variable is  4.53958077359517e-05"
```

```
print(paste('The median of the random variable is ', mean(f(x))))
```

```
## [1] "The median of the random variable is  4.53958077359517e-05"
```

The result tells out that both the mean and the median of the random variable are approximately 0.

(b) Write down a procedure to generate X.

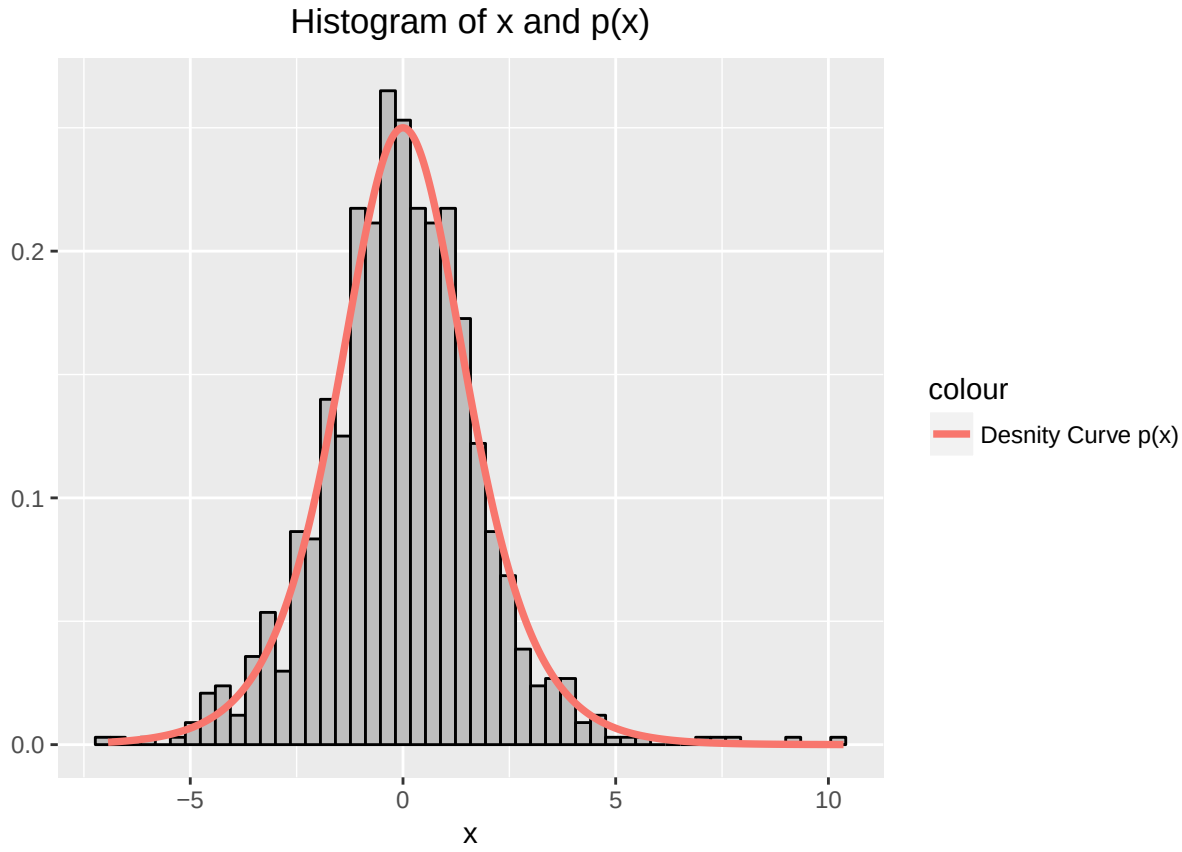
Rejection sampling could help us generate X from the CDF. Denote the density function as $p(x) = \frac{d}{dx}F(x)$

- We choose a number $M \geq \sup_x \frac{p(x)}{q(x)}$ where $q(x)$ is a proposal density function, in this case, we choose it to be the double exponential distribution function
- We generate a random number Y from our $q(x)$ and another random number U from Uni[0,1].
- If $U < \frac{p(Y)}{M \cdot q(Y)}$, we set $X = Y$. Otherwise, go back to the previous step to draw another pair of Y and U.

(c) Use Monte Carlo Simulation to generate at least $n = 10000$ points from the CDF $F(x)$.

```
set.seed(403)
n <- 50000
M <- 2
U <- runif(N, min=0, max=1)
Y <- rexp(N, rate=1)*(rbinom(N,1,0.5)*2-1)

Q <- dexp(abs(Y)) / 2
X <- Y[U < f(Y)/(M*Q)]
```

```
xx <- seq(from=min(X), to=max(X), length.out=length(X))
ggplot() + geom_histogram(aes(X, y=..density..),
                          bins=50,
                          color='black',
                          fill='grey') +
           geom_line(aes(xx, f(xx),
                 color='Desnity Curve p(x)'),
             size=1.3) + xlab('x') + ylab('') + ggtitle('Histogram of x and p(x)') +
  theme(plot.title = element_text(hjust = 0.5))
```



Comparing the density curve, we can see that the density curve fitted perfectly to the histogram.