Math Statistics

Shisham Adhikari

4/12/2020

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.2
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ggplot2)
library(knitr)
library(mvtnorm)
set.seed(1999) # set seed for reproducibility
```

Warm-up

Using matrix notation and the hat matrix, finding $Var(\hat{Y})$:

$$\hat{Y} = X(X'X)^{-1}X'Y = HY$$

$Var(\hat{Y}) = X(X'X)^{-1}X'Var(Y)X(X'X)^{-1}X' = \sigma^2$

MLR Simulator

The core of this assignment is the creation of a MLR simulator that you will use to investigate the properties of the method. The model under investigation is the following.

$$Y = X\beta + \epsilon$$

Where Y and ϵ are vectors of length n, X is an $n \times 3$ design matrix (the first column is just ones) and β is a vector of length 3.

```r
# set params
B0 <- 1
B1 <- 2
B2 <- 3
B <- c(B0, B1, B2)
sigma <- 0.1
# complete specification
n <- 100
mean <- c(1,-1)
```

```
cov <- matrix(c(1,0,0,1),
              byrow = TRUE,
              ncol = 2)
X <- cbind(rep(1,n),rmvnorm(n,mean,cov))
# simulate ys (this part inside a for loop)
epsilon <- rnorm(n,0,sd=sigma)
y <- X%*%B + epsilon
C <- solve(t(X)%*%X)
```

Part I. Sampling distributions

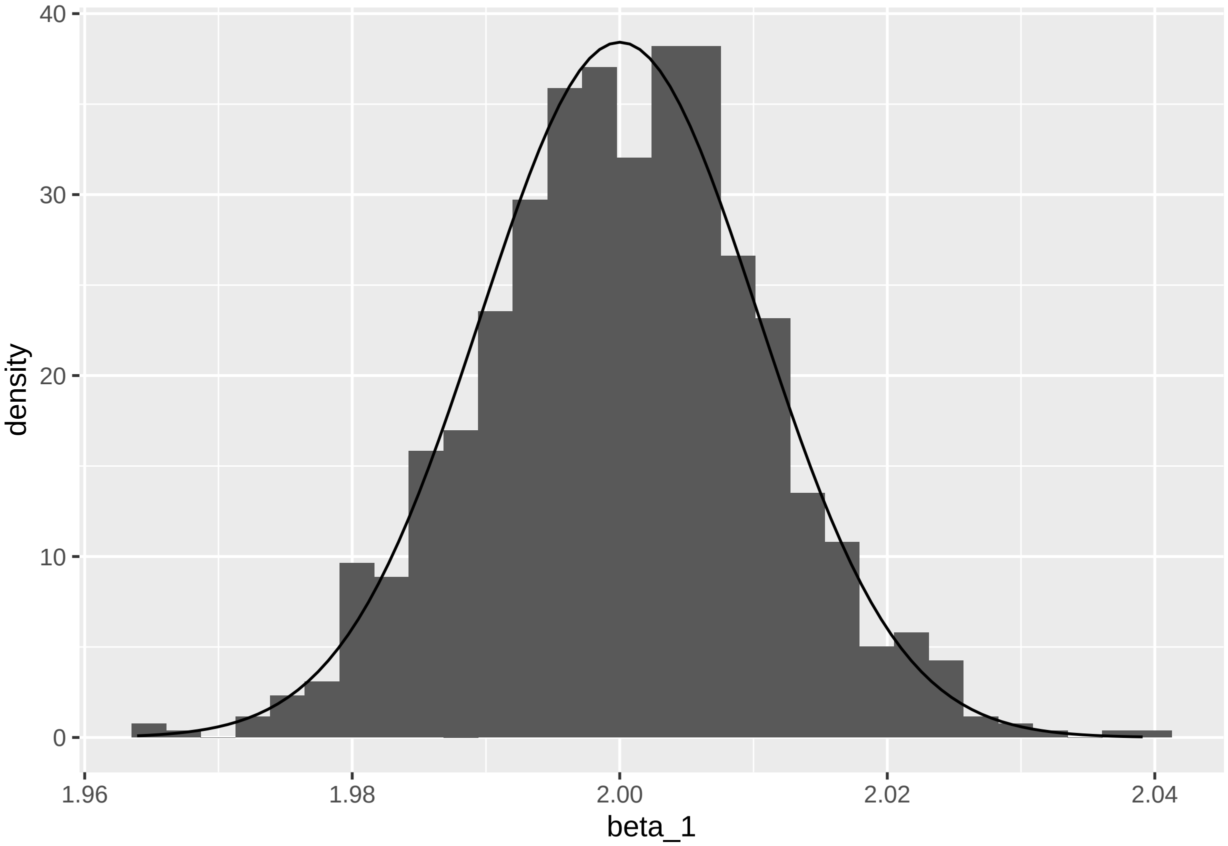
Using our simulator to create an MC approximation of the true sampling distribution of the estimates of β_1, $E(Y_s)$=X_sB, and Y_s corresponding to a fixed new observation x_s.

```
x_s <- c(1,1,1)
it <- 1000
beta_1 <- rep(NA, it)
beta_0 <- rep(NA, it)
beta_2<- rep(NA, it)
exp_y <- rep(NA, it)
y_s <- rep(NA, it)
for(i in 1:it){
  epsilon <- rnorm(n,mean=0,sd=sigma)
  y <- X %*% B + epsilon
  df <- data.frame(y,X)
  fit <- lm(y ~ X2 + X3, data = df)
  #beta1
  beta_1[i] <- coef(fit)[2]
   beta_0[i] <- coef(fit)[1]
   beta_2[i] <- coef(fit)[3]
  #E[y_s]
  exp_y[i] <- coef(fit) %*% x_s
  #Y_s
  y_s[i] <- x_s %*% B + epsilon
}
```

Now to compare how these empirical distributions compare to their analytical form in terms of center, shape, and spread, we draw both of the distributions in the same graph:

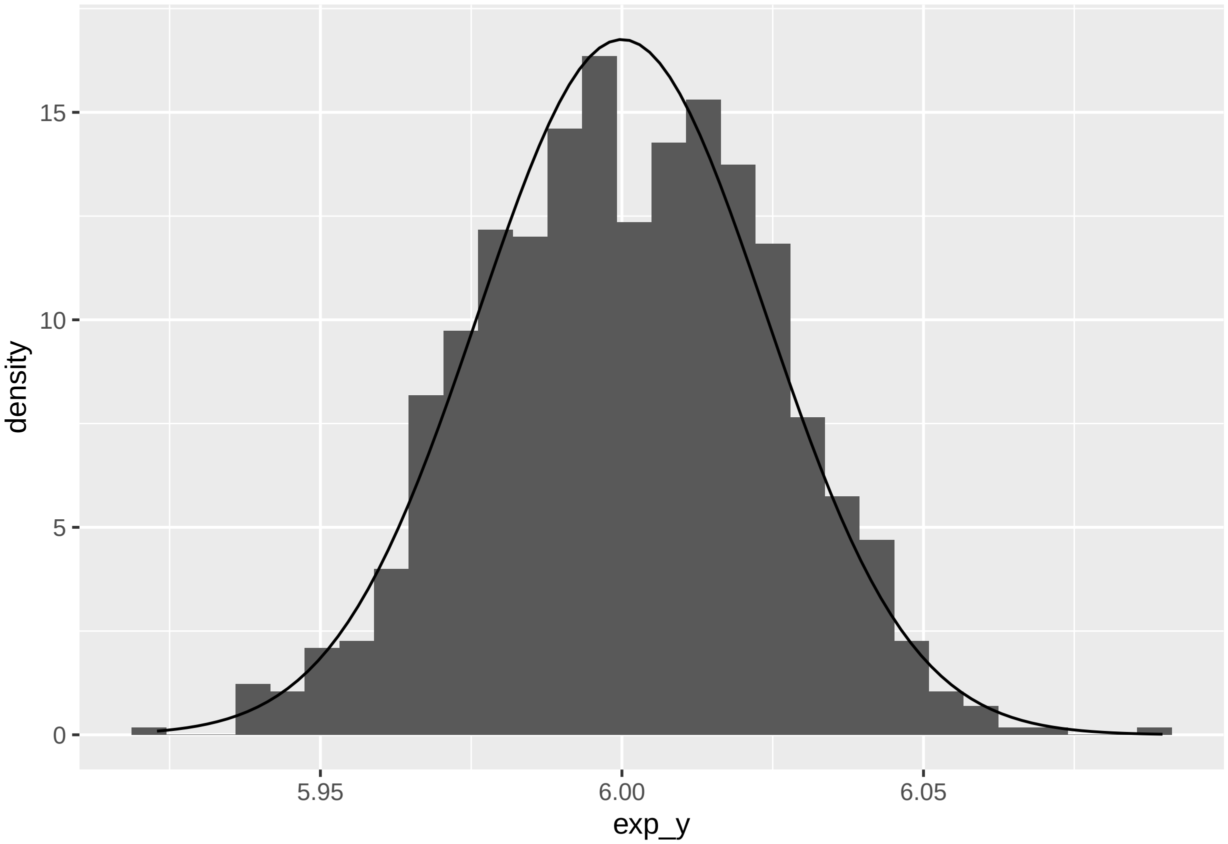
```
beta_1 <- data.frame(beta_1)
ggplot(beta_1, aes(x=beta_1)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=B1, sd=sqrt(sigma^2*C[2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



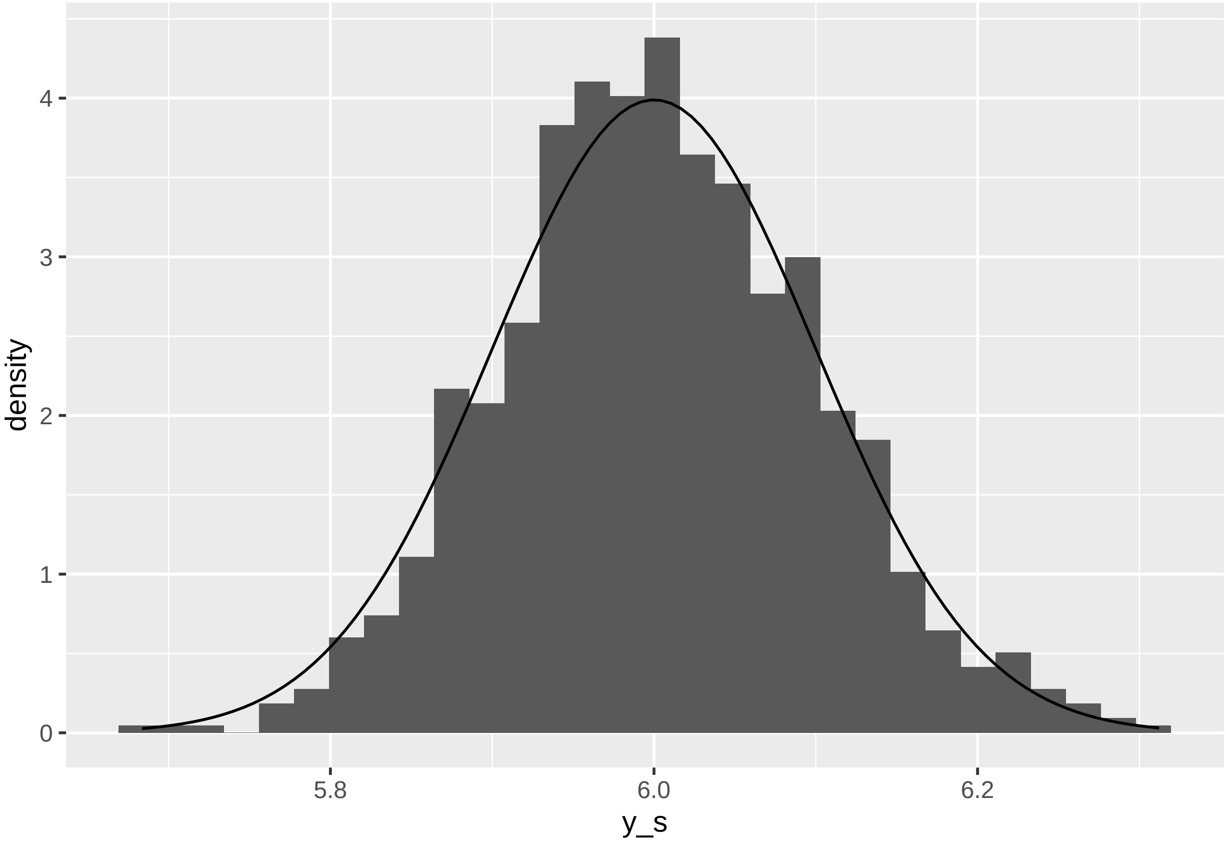
```
mean = x_s%*%B
sd = sqrt(sigma^2 * t(x_s) %*% solve(t(X)%*%X) %*% x_s)
exp_yx <- data.frame(exp_y)
ggplot(exp_yx, aes(x=exp_y)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=mean, sd=sd))
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```r
mean = x_s%*%B
yx_xs <- data.frame(y_s)
sd1 = sqrt(sigma^2)
ggplot(yx_xs, aes(x=y_s)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=mean, sd=sd1))
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



For each three of the estimates, we see that the distribution we got from our model is very close to their analytical forms in terms of their center, spread, and mean.

Part II. A different model

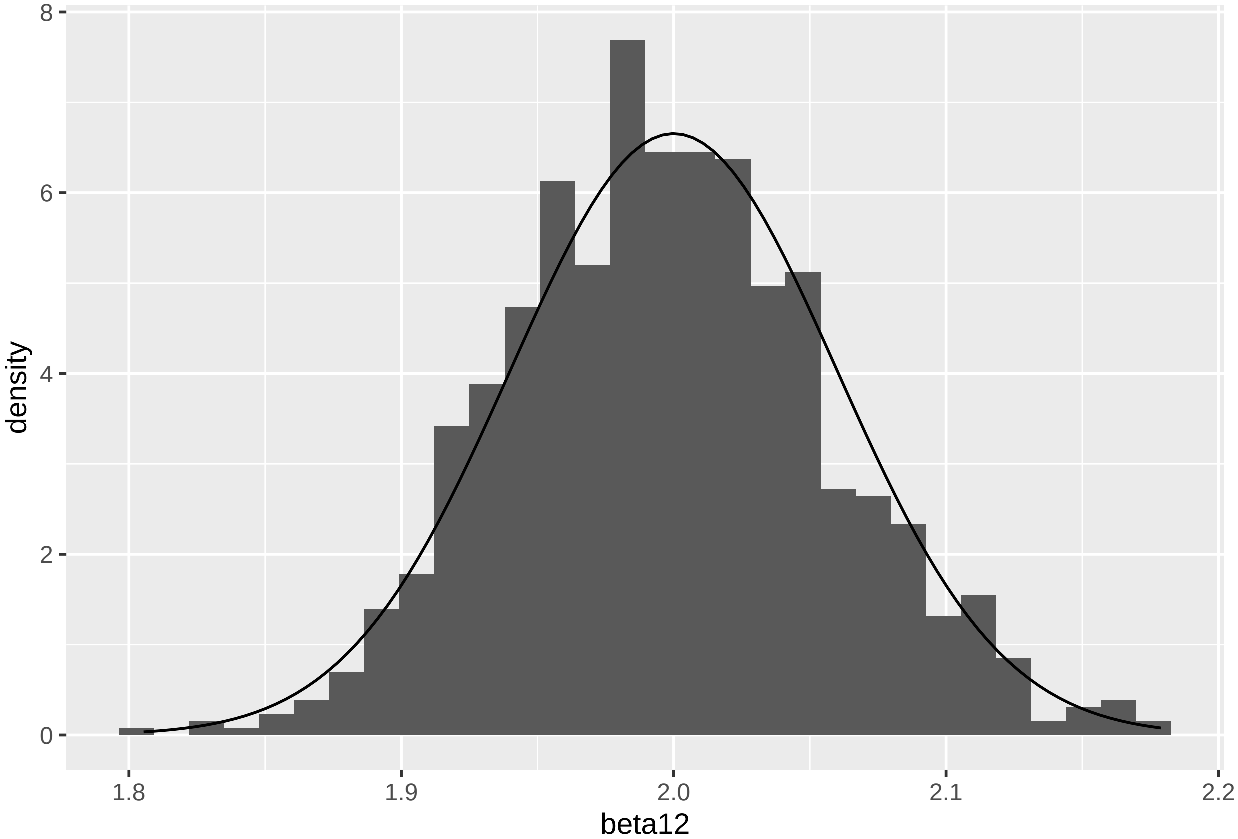
We introduced two variations on the model: 1. Changed the marginal distribution of the ϵ to uniform distribution with minimum=-1 and maximum=1 so that it still should be centered at 0 and the variance is 4/3.

```
beta12 <- rep(NA, it)
expy12 <- rep(NA, it)
y12 <- rep(NA, it)
for(i in 1:it){
  epsilon2 <- runif(n, min=-1, max=1)
  y2 <- X %*% B + epsilon2
  df2 <- data.frame(y2,X)
  fit2 <- lm(y2 ~ X2 + X3, data = df2)
  #beta1
   beta12[i] <- coef(fit2)[2]
  #E[y_s]
  expy12[i] <- coef(fit2) %*% x_s
  #Y_s
  y12[i] <- x_s %*% B + epsilon2
}
```

Comparing our empirical distribution with the analytical form:

```
beta12 <- data.frame(beta12)
ggplot(beta12, aes(x=beta12)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=B1, sd=sqrt((1/3)*C[2,
```

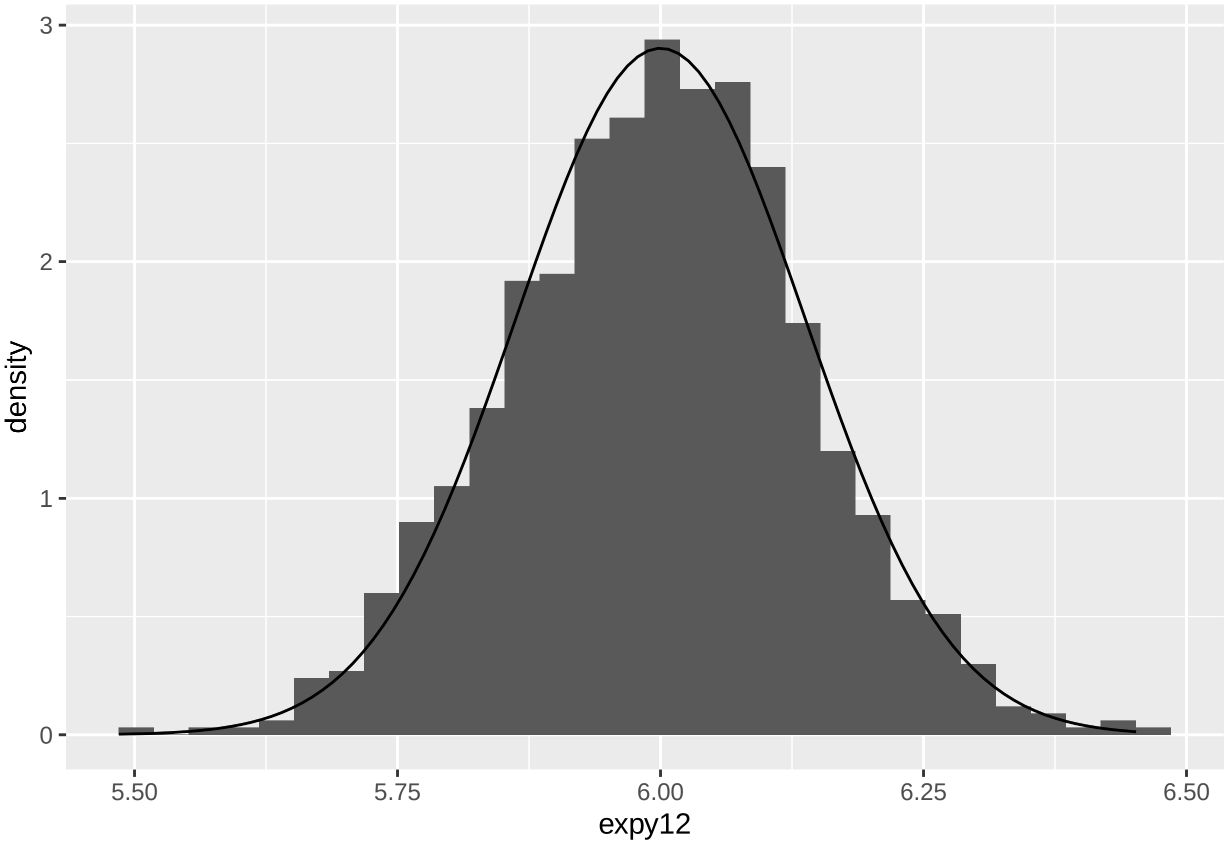
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We see that the distribution of the estimates of β_1 under this model doesn't change as much from the prio model and also the empirical distribution is close enough to the analytical form.

```
mean = x_s%*%B
sd = sqrt((1/3) * t(x_s) %*% solve(t(X)%*%X) %*% x_s)
expy12 <- data.frame(expy12)
ggplot(expy12, aes(x=expy12)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=mean, sd=sd))
```

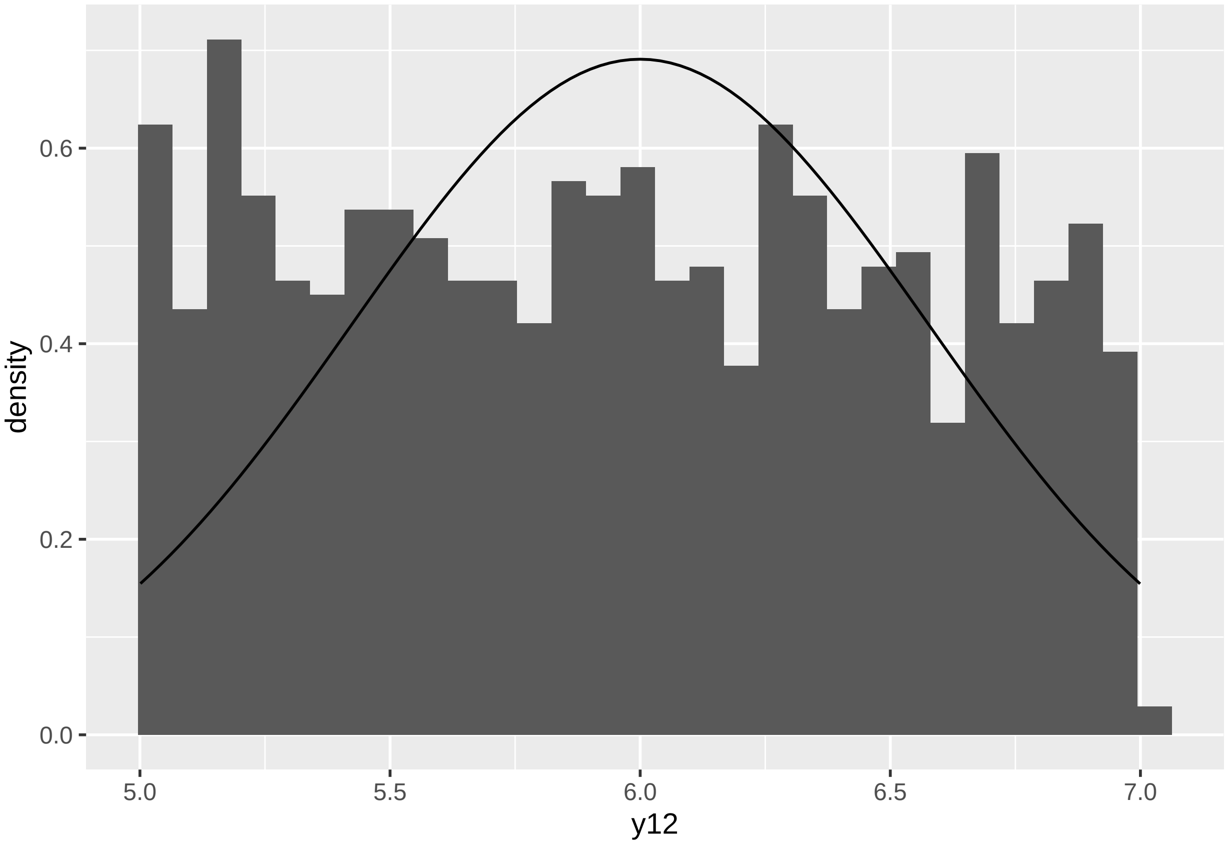
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We see that the distribution of the estimates of $E(Y_s)$=X__sB under this model resembles the prior model and also is close to the analytical form.

```
y12 <- data.frame(y12)
sd12 = sqrt(1/3)
ggplot(y12, aes(x=y12)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=mean, sd=sd12))
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



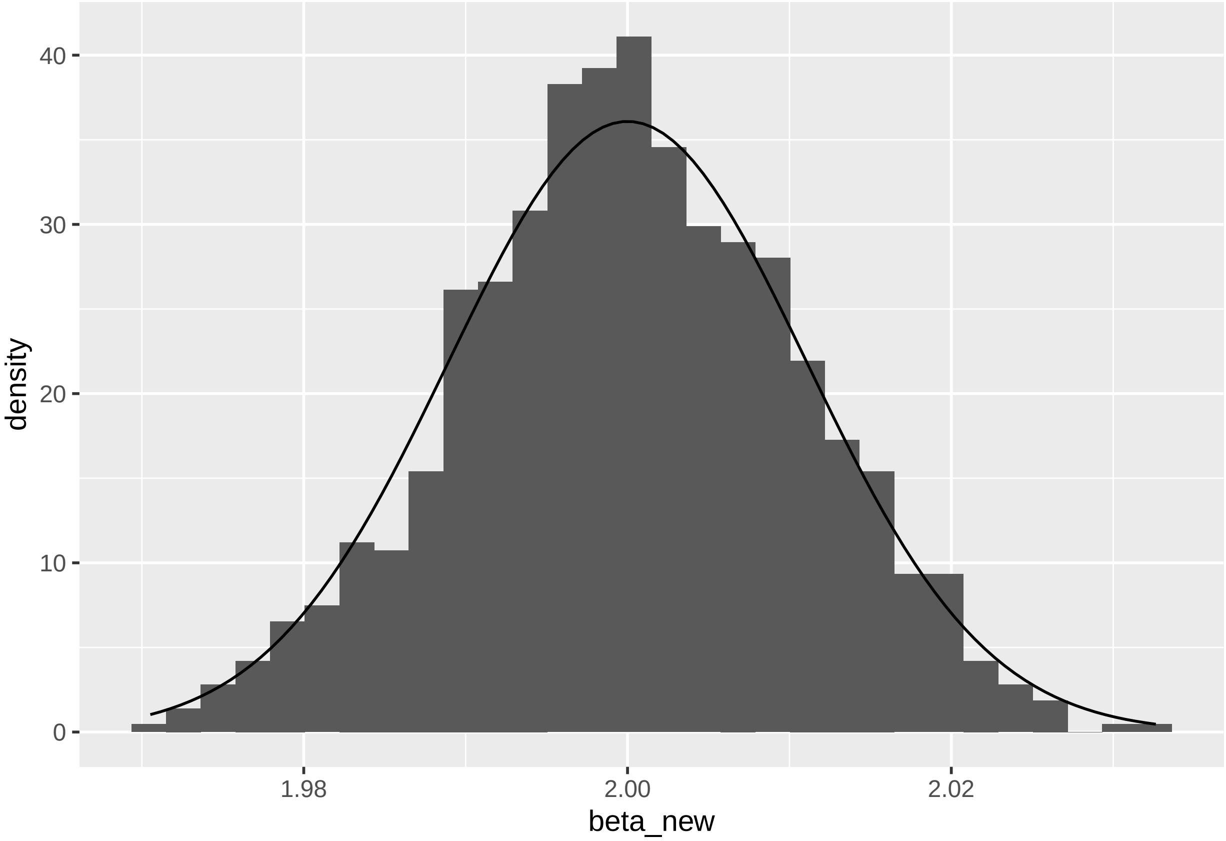
The above graph for the approximation of the distribution of the estimates of Y_s corresponding to a fixed new observation x_s is what we would expect for a uniform distribution epsilons. We see that the empirical distribution has more variance than the analytical form but still the approximation is close enough.

2. Introduced non-zero covariance into the joint distribution of the X using `rvmnorm()`:

```
mean1 <- c(1,-2)
cov1 <- matrix(c(1,0.5,0.5,1),
            byrow = TRUE,
            ncol = 2)
X1 <- cbind(rep(1,n),rmvnorm(n,mean1,cov1))
C1 <- solve(t(X1)%*%X1)
beta_new <- rep(NA, it)
exp_ynew <- rep(NA, it)
y_snew <- rep(NA, it)
for(i in 1:it){
epsilon1 <- rnorm(n,mean=0,sd=sigma)
y1 <- X1 %*% B + epsilon1
xy <- data.frame(y1,X1)
fitt <- lm(y1 ~ X2 + X3, data = xy)
#beta1
beta_new[i] <- coef(fitt)[2]
 #E[y_s]
exp_ynew[i] <- coef(fitt) %*% x_s
y_snew[i] <- x_s %*% B + epsilon1
}
```

```
beta_new <- data.frame(beta_new)
ggplot(beta_new, aes(x=beta_new)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=B1, sd=sqrt(sigma^2*C1
```

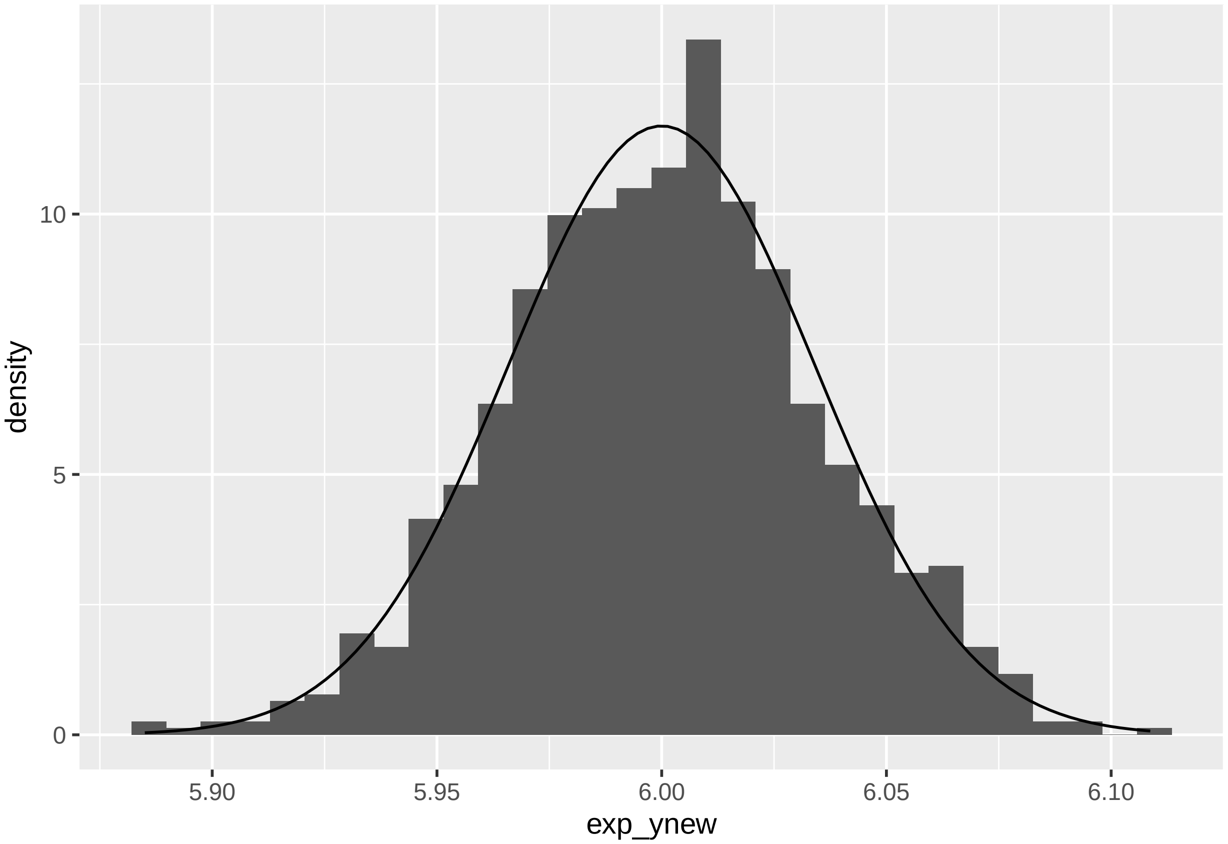
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



We see that our MC approximation of the true sampling distribution of the estimates of β_1 is still close to the analytical form under this model.

```
mean_new = x_s%*%B
sd_new = sqrt(sigma^2 * t(x_s) %*% solve(t(X1)%*%X1) %*% x_s)
exp_ynew <- data.frame(exp_ynew)
ggplot(exp_ynew, aes(x=exp_ynew)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=mean_new, sd=sd_new))
```

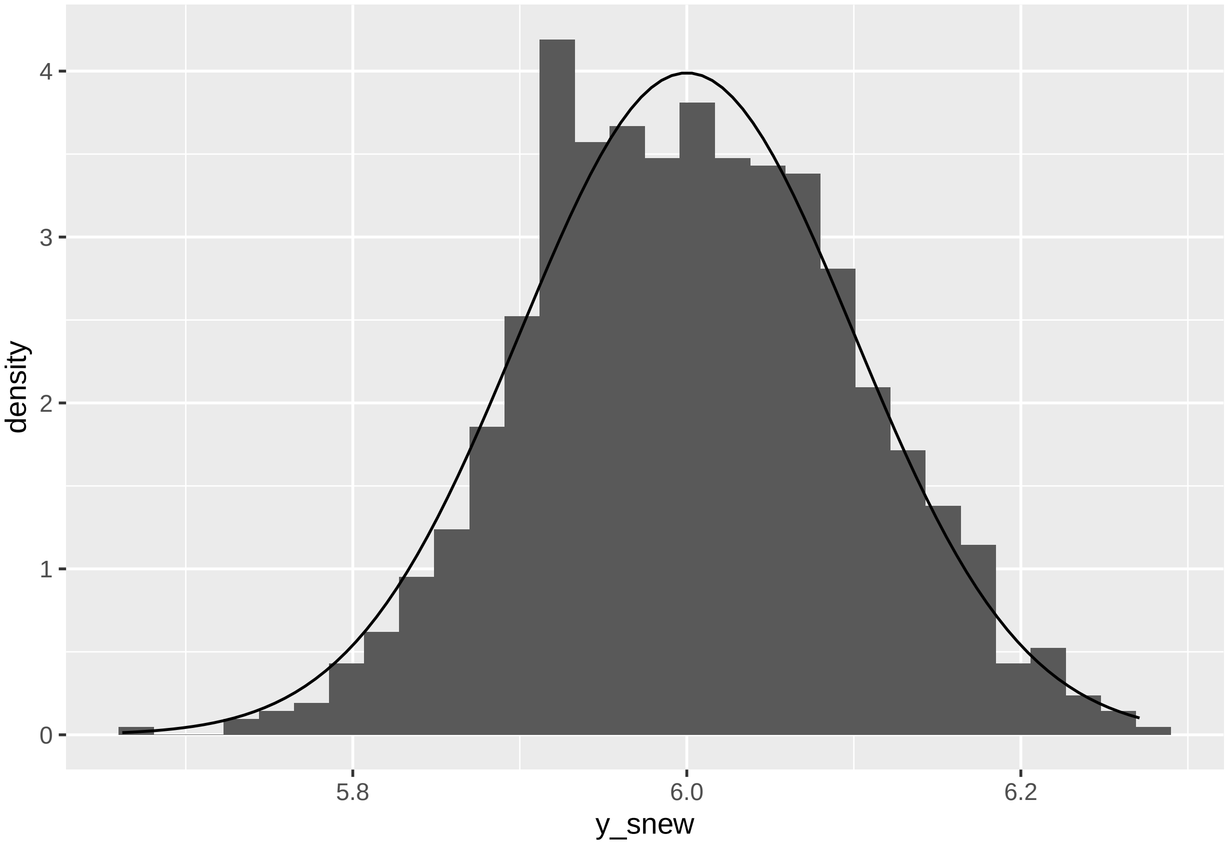
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



We see that our MC approximation of the true sampling distribution of the estimates of $E(Y_s)$=X_sB is still close to the analytical form in terms of spread, shape, and center.

```
mean11 = x_s%*%B
y_snew <- data.frame(y_snew)
sd11 = sqrt(sigma^2)
ggplot(y_snew, aes(x=y_snew)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=mean11, sd=sd11))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

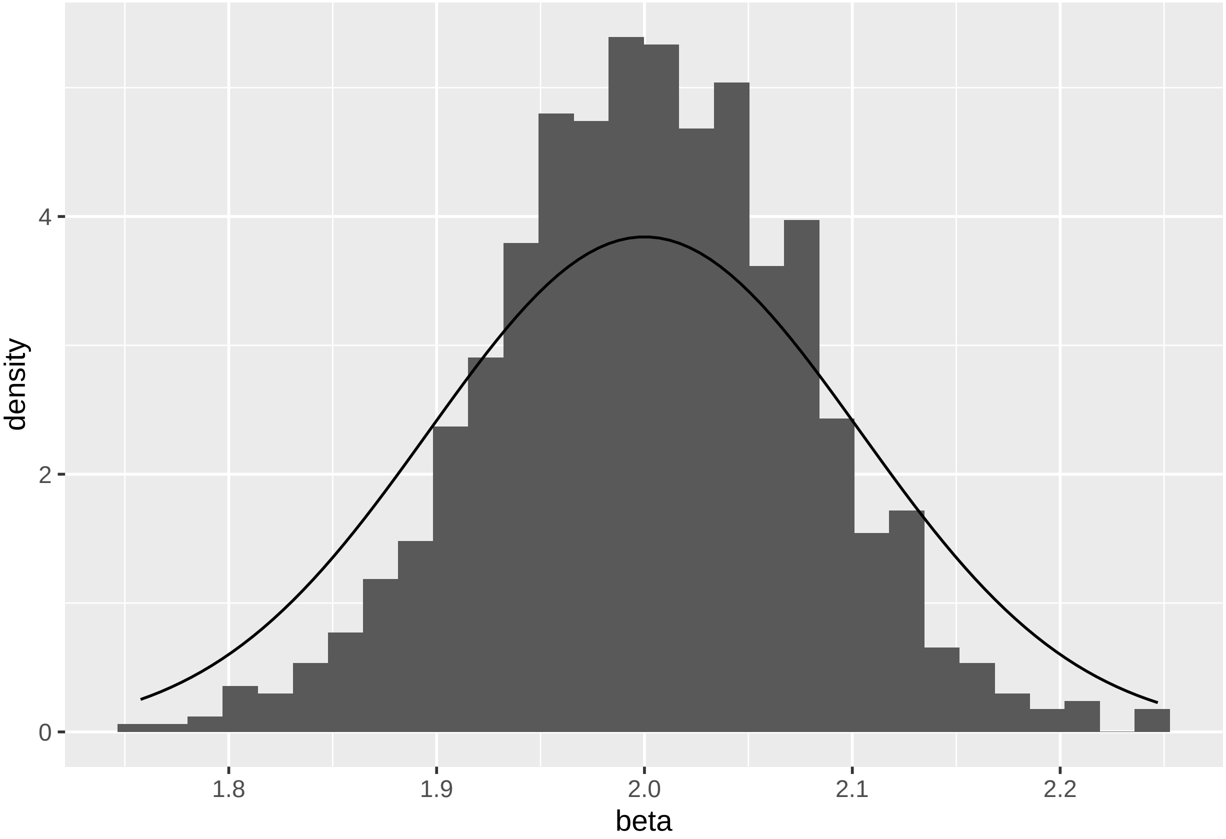


Again the graph depicts that the MC approximation of the true sampling distribution of the estimates of Y_s corresponding to a fixed new observation x_s is still close to the analytical form in terms of spread, shape, and center. Overall we see that introducing non-zero covariance into the join distribution of the X doesn't change our approximation of the estimates significantly.

3. Introduced non-zero covariance into the joint distribution of the ϵ:

```
error.cov.mat <- matrix(rep(.5,n*n),n)
  diag(error.cov.mat) <- 1
beta <- rep(NA, it)
y22 <- rep(NA, it)
exp_y2 <- rep(NA, it)
for(i in 1:it){
  getCorrErrors <- function(){rmvnorm(1, sigma = error.cov.mat) %>% as.numeric}
  y2 <- X %*% B + getCorrErrors()
  df2 <- data.frame(y2,X)
  fit2 <- lm(y2 ~ X2 + X3, data = df2)
  #beta1
  beta[i] <- coef(fit2)[2]
  #E[y_s]
  exp_y2[i] <- coef(fit2) %*% x_s
  #Y_s
  y22[i] <- x_s %*% B + getCorrErrors()
}
```

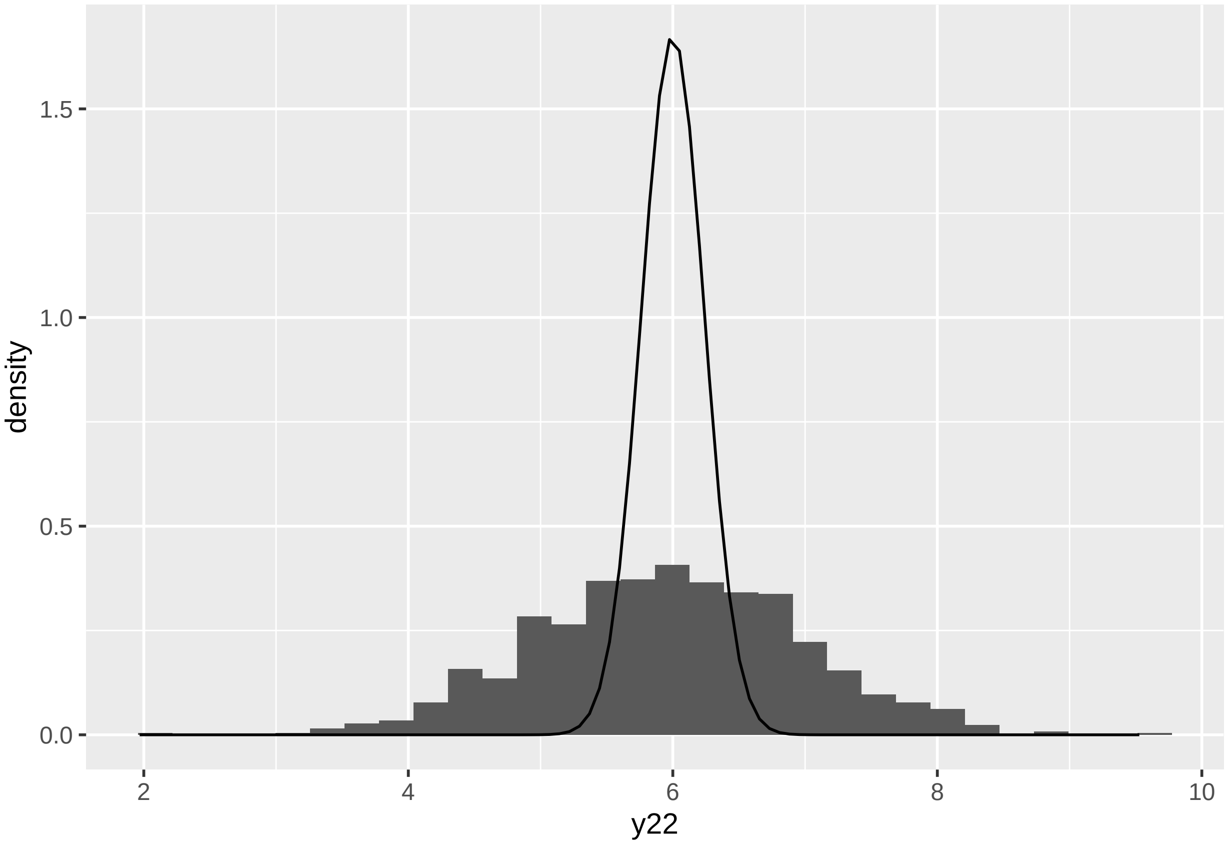
```
beta2 <- data.frame(beta)
ggplot(beta2, aes(x=beta)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=B1, sd=sqrt(1*C[2,2]))
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Under this model, we see that our approximation of the distribution of the estimates of beta_1 has lower variance than the actual analytical form. This makes sense intuitively as we are restricting our betas by using epsilon with non-zero covariance.

```
mean2 = x_s%*%B
sd2 = sqrt(1* t(x_s) %*% solve(t(X)%*%X) %*% x_s)
y22 <- data.frame(y22)
ggplot(y22, aes(x=y22)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=mean2, sd=sd2))
```

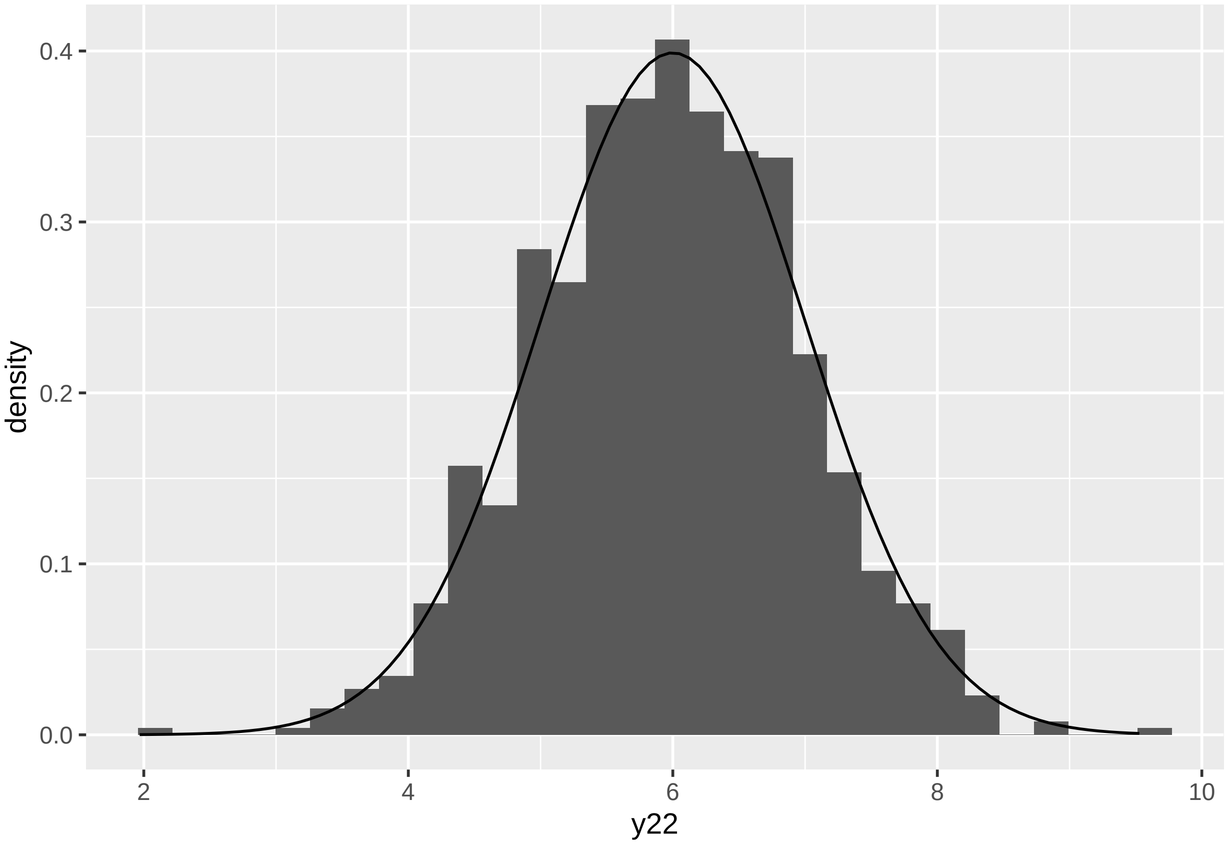
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We see that our approximation of the distribution of the estimates of $E(Y_s)$=X_sB has much higher variance than the actual analytical form. This makes sense intuitively as our errors are autocorrelated, this would make our prediction off by greater degree than before.

```
sd22 = sqrt(1^2)
ggplot(y22, aes(x=y22)) +
  geom_histogram(aes(y=..density..)) + stat_function(fun = dnorm, args=list(mean=mean2, sd=sd22))
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



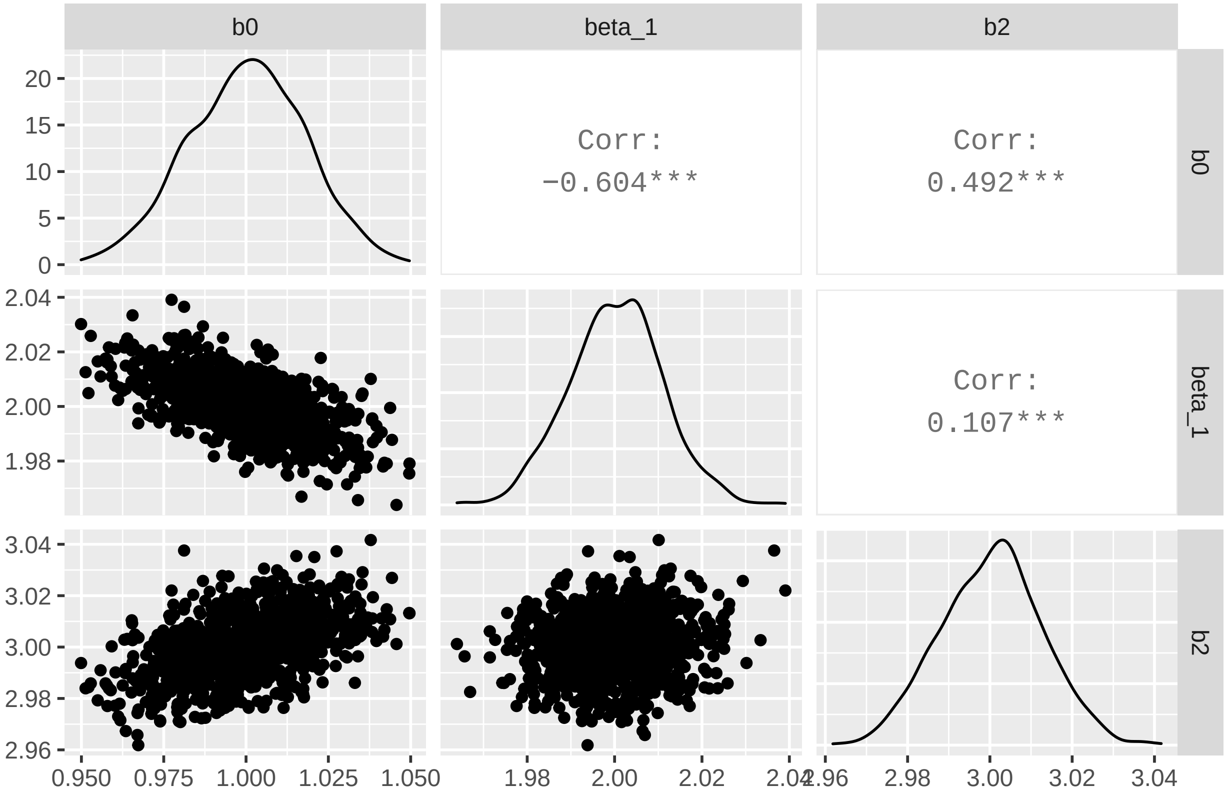
Unlike other two estimates, we observe that our approximation of the distribution of the estimates Y_s corresponding to a fixed new observation x_s is very close to the analytical form.

Part III. Variance/Covariance

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
coef <- data.frame(b0=beta_0, b1=beta_1, b2=beta_2)
ggpairs(coef, columns = 1:ncol(coef), title = "Scatter matrix for the three simulated regression coeffi
```


Scatter matrix for the three simulated regression coefficients

Description: Above scatterplot matrix shows us the pairwise relationships between the three simulated regression coefficients in the original model (from Part I). The distribution of our estimates makes sense as estimate of b__0 is centered at 1 which is our original B0, b__1 is centered near 2 which is our original value of B1,and b__2 is centered near 3 which is what we assigned our b__2 to be. All of them have almost symmetrical normal distribution.Based on the correlation parts of the matrix, we see that the beta__0 and beta__1 have a high negative correlation, the beta__0 and beta__2 have a high positive correlation, and finally the beta__1 and beta__2 have a very small positive correlation. Now comparing the empirical covariance matrix to the analytical form that we derived in class, we get:

```
#empirical covariance matrix
emcovmatrix <- cov(coef)
#theoritical covariance matrix
thcovmatrix <- sigma^2 * solve(t(X)%*%X)
emcovmatrix
```

```
##                     b0        beta_1           b2
## b0      0.0003127277 -1.144253e-04 1.066600e-04
## beta_1 -0.0001144253  1.145874e-04 1.404316e-05
## b2      0.0001066600  1.404316e-05 1.502954e-04
```

```
thcovmatrix
```

```
##               [,1]          [,2]         [,3]
## [1,]  0.0002943681 -1.030239e-04 1.012681e-04
## [2,] -0.0001030239  1.078145e-04 1.429034e-05
## [3,]  0.0001012681  1.429034e-05 1.395833e-04
```

We see that the values in the empirical covariance matrix is very close to the values in the analytical form. This is what we would expect or want our model to give.