MATH 392 Problem Set 8

Sam D. Olson

Time Spent: 10(ish) Hours

Warm-up

Using matrix notation and the hat matrix, find $Var(\hat{Y})$.

$Var(\hat{Y}) = Var(X\beta + \epsilon) = Var(X\beta) + Var(\epsilon)$

Note: $\hat{Y} = HY$, where X denotes the hat matrix:

$H = X(X'X)^{-1}X'$

Thus, we have:

$Var(\hat{Y}) = Var(HY)$

Taking advantage of a linear algebra identity, we can say:

$Var(\hat{Y}) = Var(HY) = HVar(Y)H'$

Substituting the definition of the hat matrix gives us:

$Var(\hat{Y}) = HVar(Y)H' = X(X'X)^{-1}X'Var(Y)(X(X'X)^{-1}X')'$

Note: $Var(Y) = \sigma^2 I$, giving us:

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

As σ^2 is a constant, we can pull it out. Giving us:

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

Noting the properties of symmetry and idempotence, we may say:

$Var(\hat{Y}) = \sigma^2 HH' = \sigma^2 H^2 = \sigma^2 H$

Thus we conclude:

$Var(\hat{Y}) = \sigma^2 H$

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.

You're welcome to select any values that you like for the parameters and any distribution that you like for the x's (you may be want to check out the `rmvnorm()` function in the `mvtnorm` package). The core bit of code should look something like:

```r
library(mvtnorm)
# set params
B0 <- 4
B1 <- 2
B2 <- 1
B <- c(B0, B1, B2)
p <- 3
# Example rmvnorm code
# sigma <- matrix(c(1,0,0,0,2,0,0,0,3), 3, 3)
# mu <- c(0,2,4)
# X <- rmvnorm(100, mean = mu, sigma = sigma)

# complete specification
sigma <- .5
n <- 100
x_0 <- rep(1, n)
x_1 <- rnorm(n, mean = 2, sd = 1)
x_2 <- rnorm(n, mean = 4, sd = .5)
X <- cbind(x_0, x_1, x_2)

# simulate ys (this part inside a for loop)
epsilon <- rnorm(n, mean = 0, sd = sigma)

y <- (X %*% B) + epsilon
```

Part I. Sampling distributions

Use your simulator to create an MC approximation of the true sampling distribution of the estimates of β_1, $E(Y_s)$, and Y_s corresponding to a fixed new observation x_s. How do these empirical distributions compare to their analytical form in terms of center, shape, and spread?

Empirical Notes Regarding the analytical form of the sampling distributions, with an initial specification of normality, we have:

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

$\hat{\beta} \sim N(\beta, \sigma^2(X'X)^{-1})$

$\beta_1 \sim N(\beta_1, \sigma^2 C_{11})$ where $C = (X'X)^{-1}$

(2): $E(Y_s) = E(\beta_0 + \beta_1 x_s + \beta_2 x_s + \epsilon_s) = Y_s$

$E(Y_s) \sim N(Y_s, Var(E(\beta_0 + \beta_1 x_s + \beta_2 x_s + \epsilon_s)))$

Taking advantage of independence, we can evaluate each expected value piecewise and separate elements within the equation, giving us:

$Var(E(\beta_0 + \beta_1 x_s + \beta_2 x_s + \epsilon_s)) = Var(E(\beta_0) + E(\beta_1 x_s) + E(\beta_2 x_s) + E(\epsilon_s)) = Var(\beta_0 + \beta_1 x_s + \beta_2 x_s + 0) = \sigma^2(X'X)^{-1})$

Thus, we're left with:

$E(Y_s) \sim N(Y_s, \sigma^2(X'X)^{-1})$

Alternative Note:

$\hat{y}(x_s) = x_s'\hat{\beta} = \hat{\beta}_0 + \hat{\beta}_1 x_{s1} + \hat{\beta}_2 x_{s2}$

(3): $Y_s = \beta_0 + \beta_1 x_s + \beta_2 x_s + \epsilon_s$

$Y_s \sim N(\beta_0 + \beta_1 x_s + \beta_2 x_s + \epsilon_s, Var(\beta_0 + \beta_1 x_s + \beta_2 x_s + \epsilon_s))$

Taking advantage of (1), and independence, we simplify this to:

$Y_s \sim N(\beta_0 + \beta_1 x_s + \beta_2 x_s + \epsilon_s, \sigma^2 (X'X)^{-1} + \sigma^2 I)$, where I denotes the identity matrix.

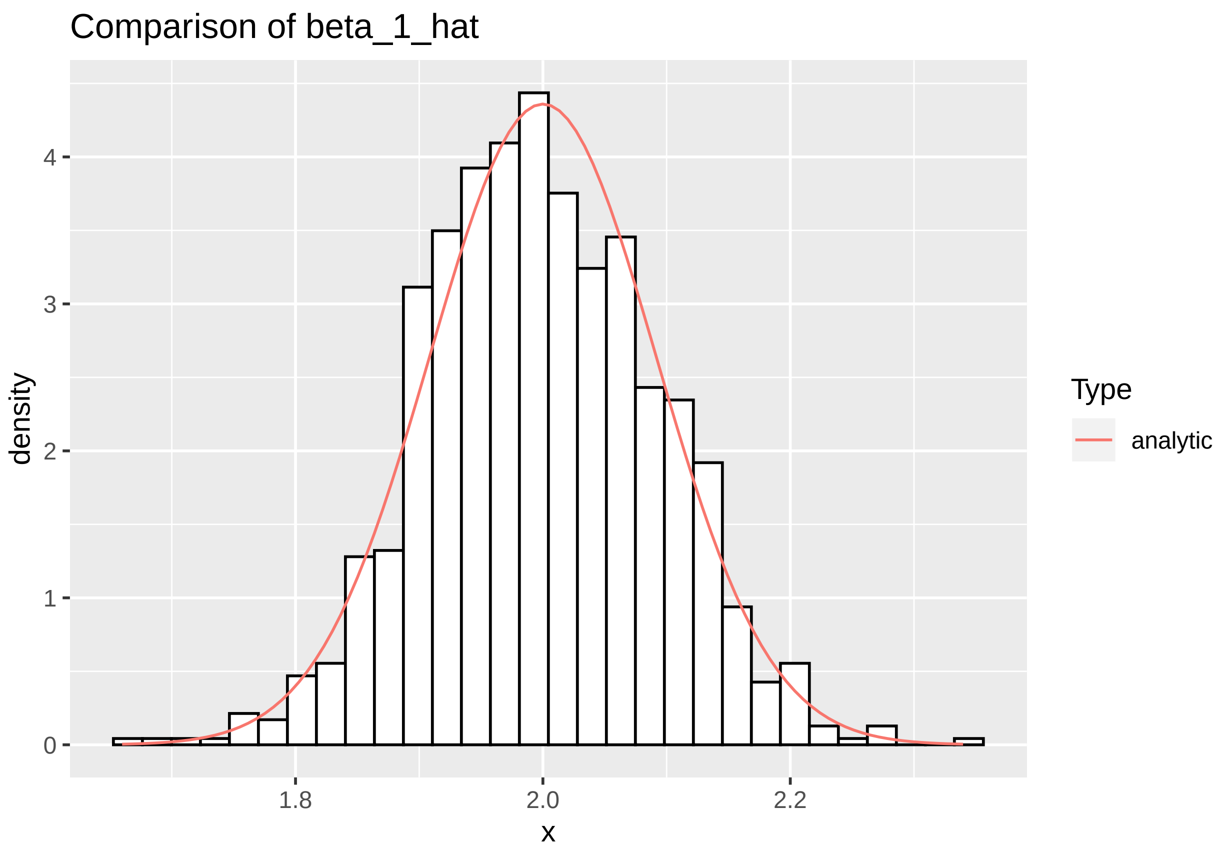
Commentary

The empirical distributions all closely align with their analytical distributions, albeit with some "lumpiness" in center and shape as a result of the simulation. However, it is clear the distributions align closely in the below plots.

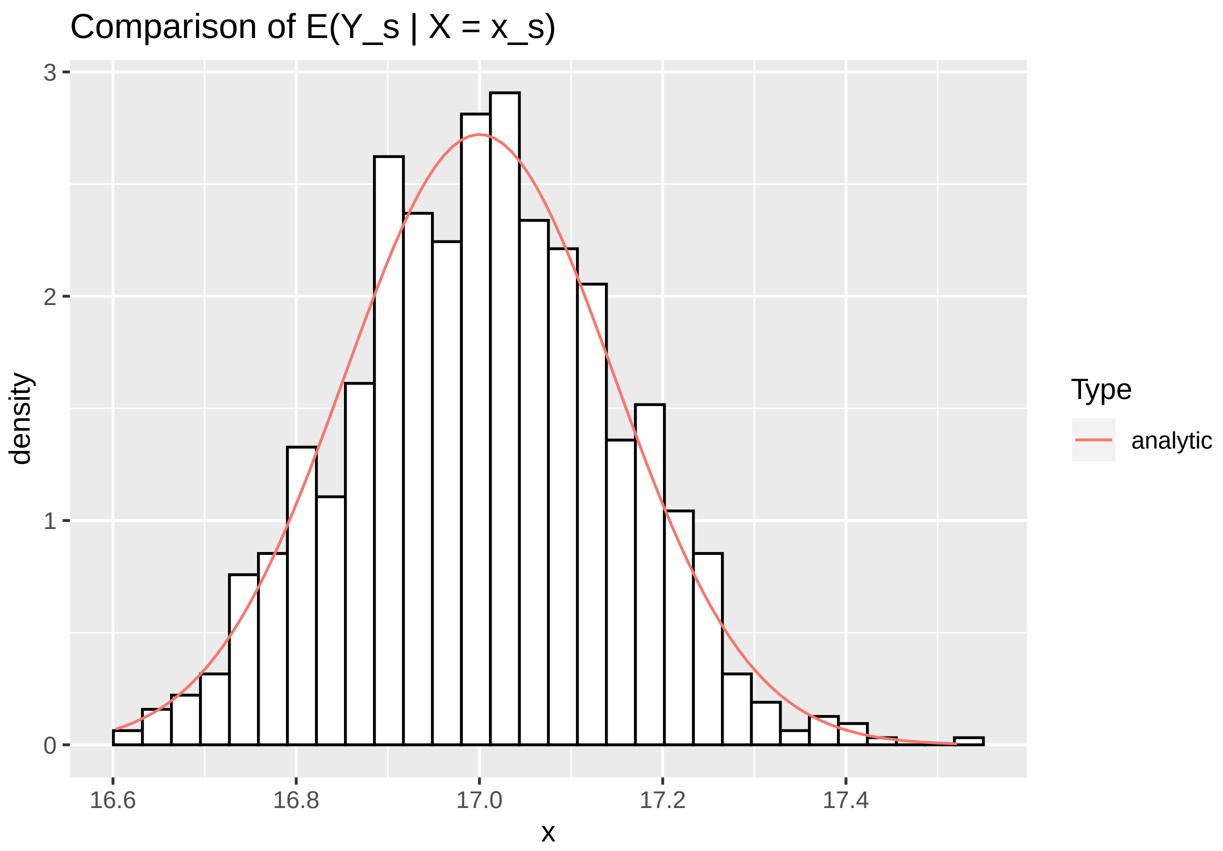
Code

```
# set params
B0 <- 1
B1 <- 2
B2 <- 4
B <- c(B0, B1, B2)
sigma <- 1
# complete specification
n <- 100
X <- rmvnorm(n, c(1,2,4),matrix(c(0,0,0,0,2,0,0,0,1), byrow = TRUE, nrow = 3))
# simulation
x_s <- c(1,2,3) # Given a fixed new observation
B1_samp <- rep(NA, 1000)
Y_s_hat_samp <- rep(NA, 1000)
Y_s_samp <- rep(NA, 1000)
for(i in 1:1000){
  epsilon <- rnorm(n, mean = 0, sd = sigma)
  Y <- X %*% B + epsilon
  B_hat <- solve(t(X)%*%X)%*%t(X)%*%Y
  B1_samp[i] <- B_hat[2]
  Y_s_hat_samp[i] <- t(x_s)%*%B_hat
  Y_s_samp[i] <- t(x_s)%*%B + rnorm(1,mean=0,sd=sigma)
}
# Find the analytical form
B1_hat_mean <- B1
B1_hat_var <- sigma^2*solve(t(X)%*%X)[2,2]
Y_s_hat_mean <- t(x_s)%*%B
Y_s_hat_var <- t(x_s)%*%solve(t(X)%*%X)%*%x_s
Y_s_mean <- t(x_s)%*%B
Y_s_var <- sigma^2
```

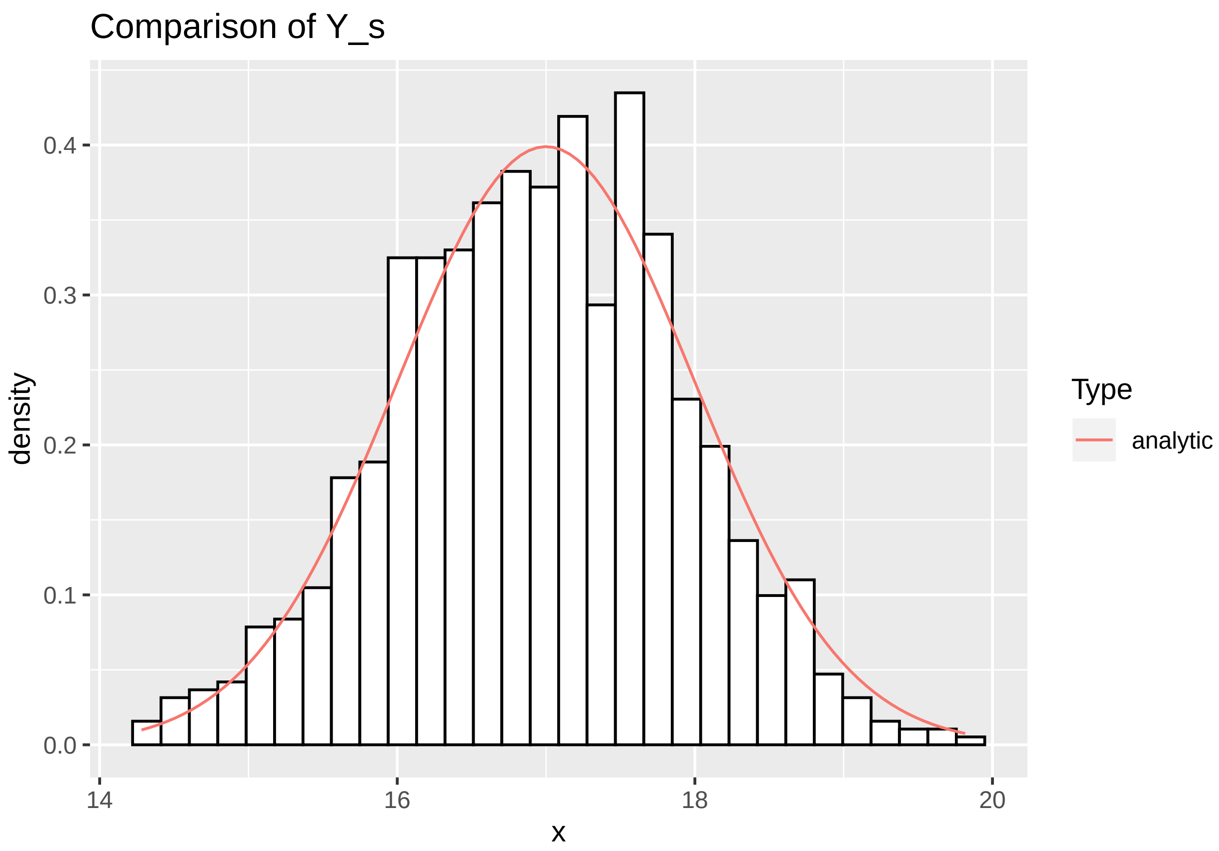
```
ggplot(data.frame(x = B1_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = B1_hat_mean, sd = sqrt(B1_hat_var)), aes(color = "analy
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of beta_1_hat")
```



```
ggplot(data.frame(x = Y_s_hat_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = Y_s_hat_mean, sd = sqrt(Y_s_hat_var)), aes(color = "anal
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of E(Y_s | X = x_s)")
```



```
ggplot(data.frame(x = Y_s_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = Y_s_mean, sd = sqrt(Y_s_var)), aes(color = "analytic"))
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of Y_s")
```



Part II. A different model

Consider two variations on the model:

1. Change the marginal distribution of the ϵ (though it still should be centered at 0).

Commentary

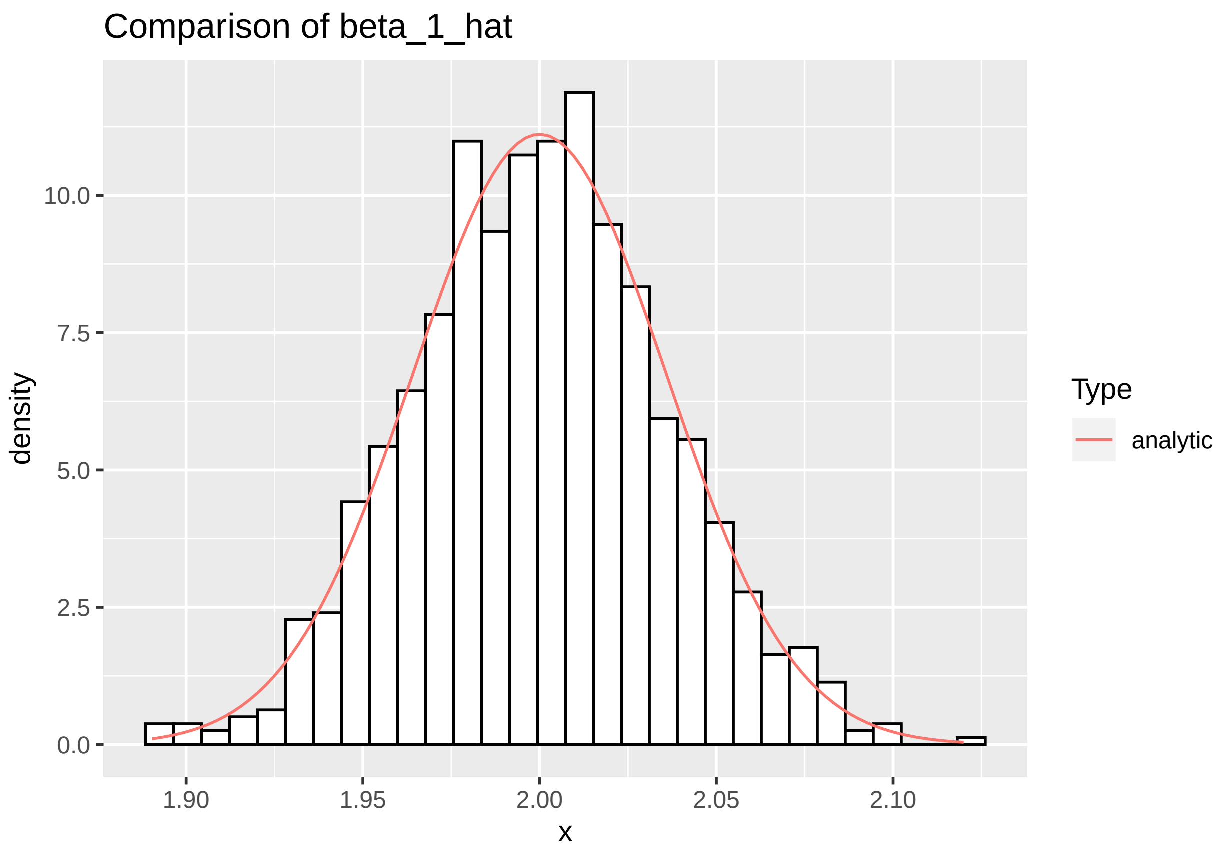
Changing the marginal distribution of the ϵ impacts the distributions of the variables of interest(β_1, Y_S, and $E(Y_s)$. However, using the plots below, $E(Y_s)$ appears to be the most impacted by a decrease in the variance of ϵ, specifically the spread of values. Most values of $E(Y_s)$ are centered near the median (center) of the distribution. The other statistics, β_1, Y_S, appear as before, closely aligning to the analytical form.

Code

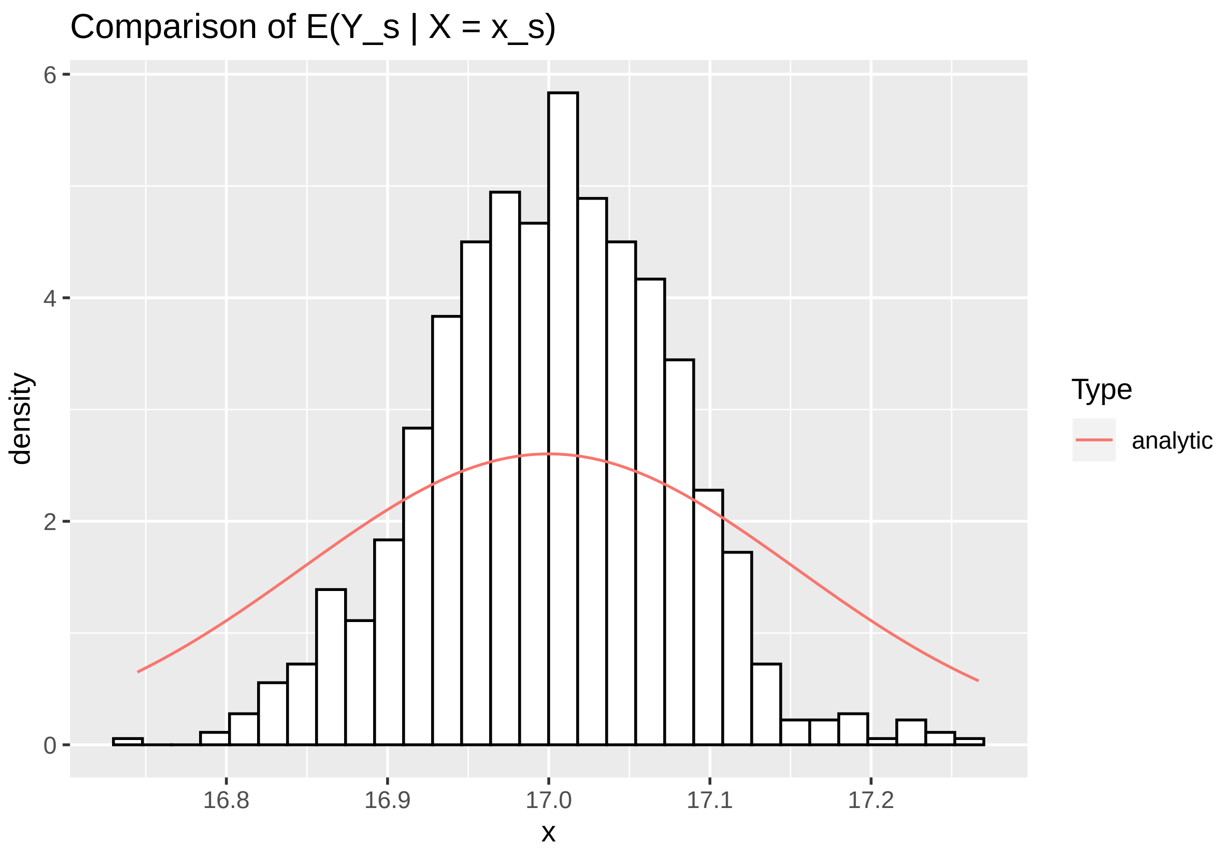
```
# set params
B0 <- 1
B1 <- 2
B2 <- 4
B <- c(B0, B1, B2)
sigma <- .5
# complete specification
```

```
n <- 100
X <- rmvnorm(n, c(1,2,4),matrix(c(0,0,0,0,2,0,0,0,1), byrow = TRUE, nrow = 3))
# simulation
x_s <- c(1,2,3) # Given a fixed new observation
B1_samp <- rep(NA, 1000)
Y_s_hat_samp <- rep(NA, 1000)
Y_s_samp <- rep(NA, 1000)
for(i in 1:1000){
  epsilon <- rnorm(n, mean = 0, sd = sigma)
  Y <- X %*% B + epsilon
  B_hat <- solve(t(X)%*%X)%*%t(X)%*%Y
  B1_samp[i] <- B_hat[2]
  Y_s_hat_samp[i] <- t(x_s)%*%B_hat
  Y_s_samp[i] <- t(x_s)%*%B + rnorm(1,mean=0,sd=sigma)
}
# Find the analytical form
B1_hat_mean <- B1
B1_hat_var <- sigma^2*solve(t(X)%*%X)[2,2]
Y_s_hat_mean <- t(x_s)%*%B
Y_s_hat_var <- t(x_s)%*%solve(t(X)%*%X)%*%x_s
Y_s_mean <- t(x_s)%*%B
Y_s_var <- sigma^2
```

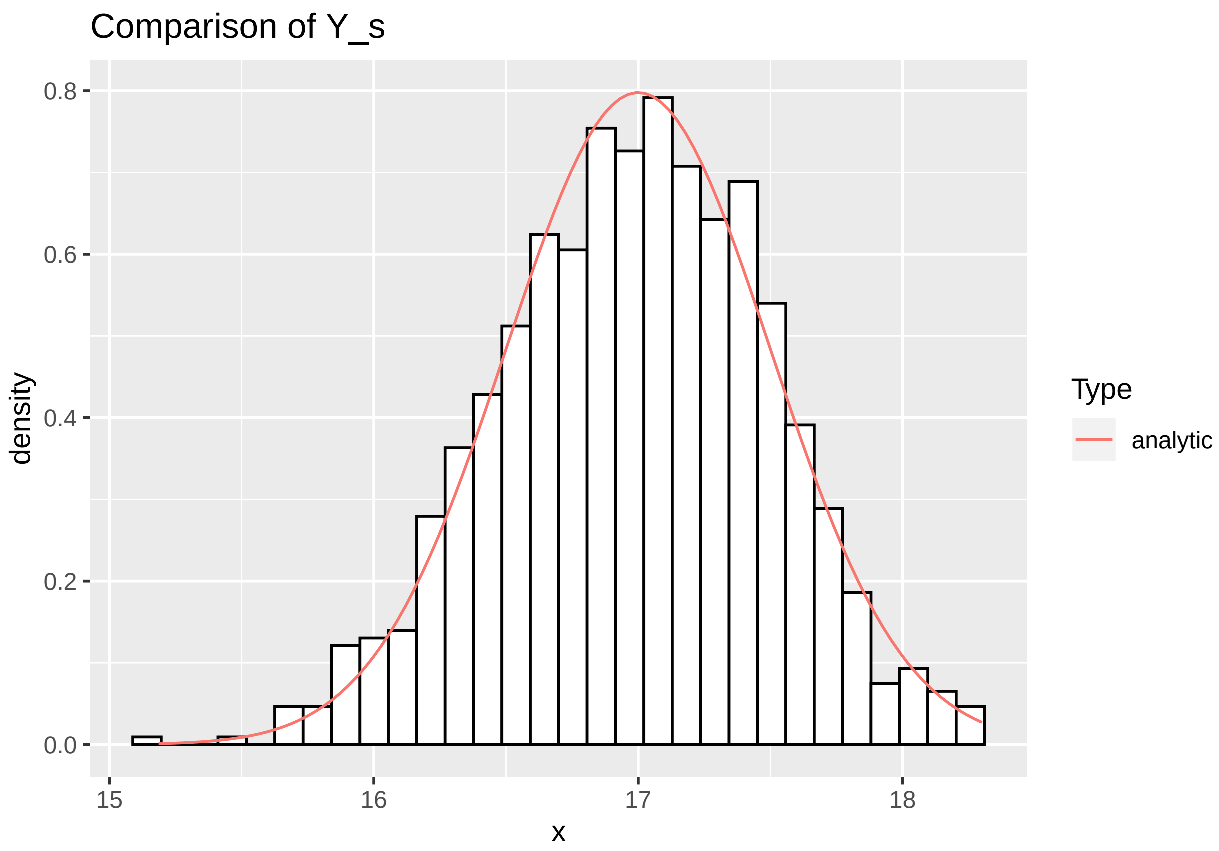
```
ggplot(data.frame(x = B1_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = B1_hat_mean, sd = sqrt(B1_hat_var)), aes(color = "analy
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of beta_1_hat")
```



```
ggplot(data.frame(x = Y_s_hat_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = Y_s_hat_mean, sd = sqrt(Y_s_hat_var)), aes(color = "anal
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of E(Y_s | X = x_s)")
```



```r
ggplot(data.frame(x = Y_s_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = Y_s_mean, sd = sqrt(Y_s_var)), aes(color = "analytic"))
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of Y_s")
```



2. Introduce non-zero covariance into the joint distribution of the X (`rvmnorm()` is helpful here).

Commentary

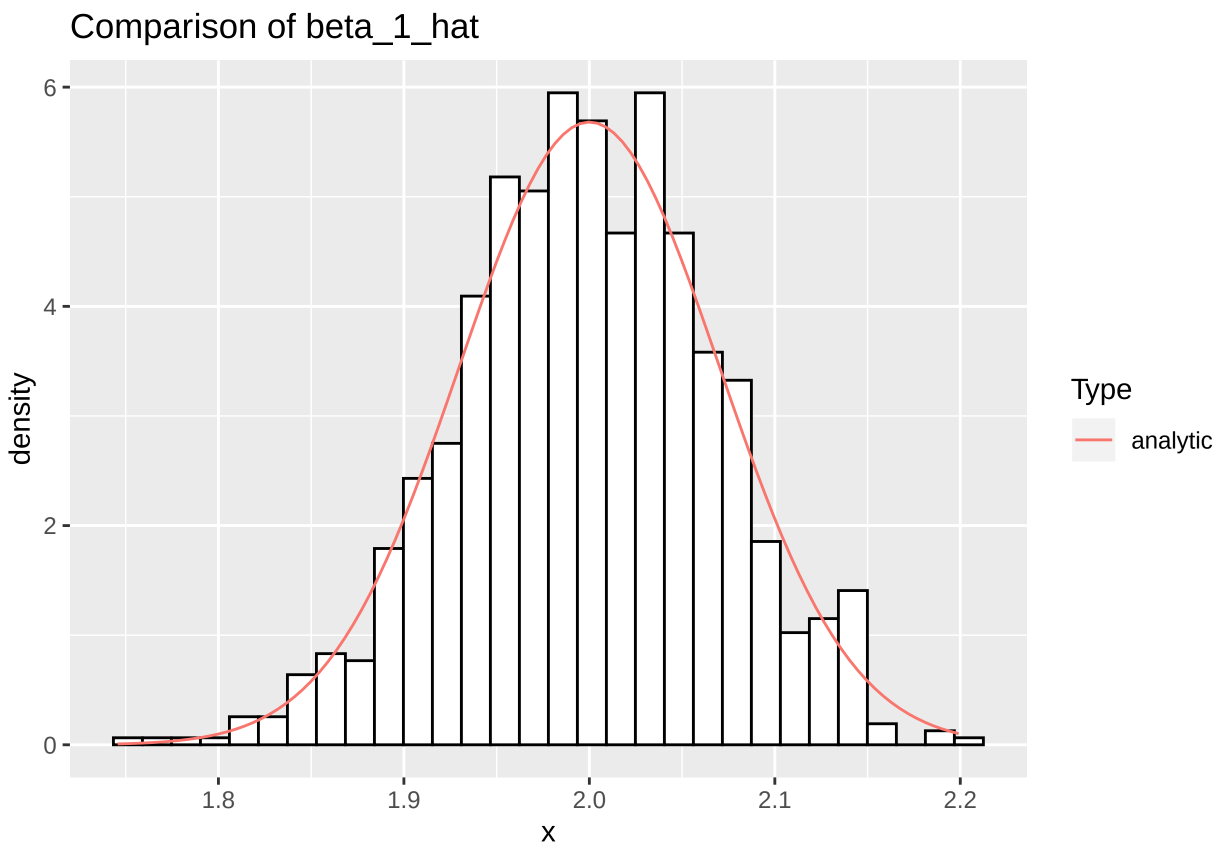
Introducing non-zero covariance into the joint distribution of the X impacts the distribution of all variables of interest, β_1, Y_s, and $E(Y_s)$. Notably for β_1, the distribution almost appears to be a mixture distribution, with two distinct points with the highest density—these points being to the left and right of the initial median of the analytical form. This feature of the distribution appears for Y_s and $E(Y_s)$ as well, albeit less noticeably.

Code

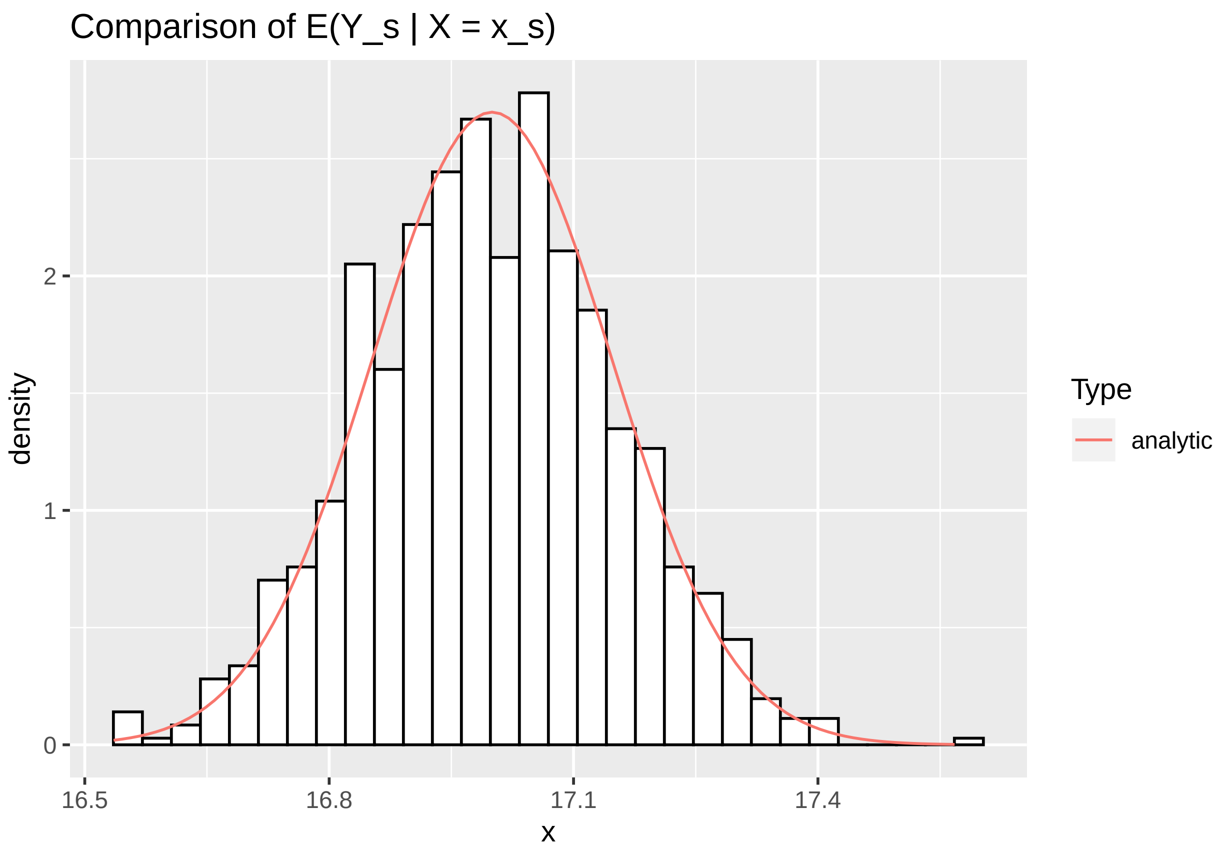
```
# set params
B0 <- 1
B1 <- 2
B2 <- 4
B <- c(B0, B1, B2)
sigma <- 1
# complete specification
n <- 100
X <- rmvnorm(n, c(1,2,4),matrix(c(1,.25,.5,.25,2,.25,.5,.25,1), byrow = TRUE, nrow = 3))
x_0 <- rep(1, n)
X[,1] <- x_0
```

```r
# simulation
x_s <- c(1,2,3) # Given a fixed new observation
B1_samp <- rep(NA, 1000)
Y_s_hat_samp <- rep(NA, 1000)
Y_s_samp <- rep(NA, 1000)
for(i in 1:1000){
  epsilon <- rnorm(n, mean = 0, sd = sigma)
  Y <- X %*% B + epsilon
  B_hat <- solve(t(X)%*%X)%*%t(X)%*%Y
  B1_samp[i] <- B_hat[2]
  Y_s_hat_samp[i] <- t(x_s)%*%B_hat
  Y_s_samp[i] <- t(x_s)%*%B + rnorm(1,mean=0,sd=sigma)
}
# Find the analytical form
B1_hat_mean <- B1
B1_hat_var <- sigma^2*solve(t(X)%*%X)[2,2]
Y_s_hat_mean <- t(x_s)%*%B
Y_s_hat_var <- t(x_s)%*%solve(t(X)%*%X)%*%x_s
Y_s_mean <- t(x_s)%*%B
Y_s_var <- sigma^2
```

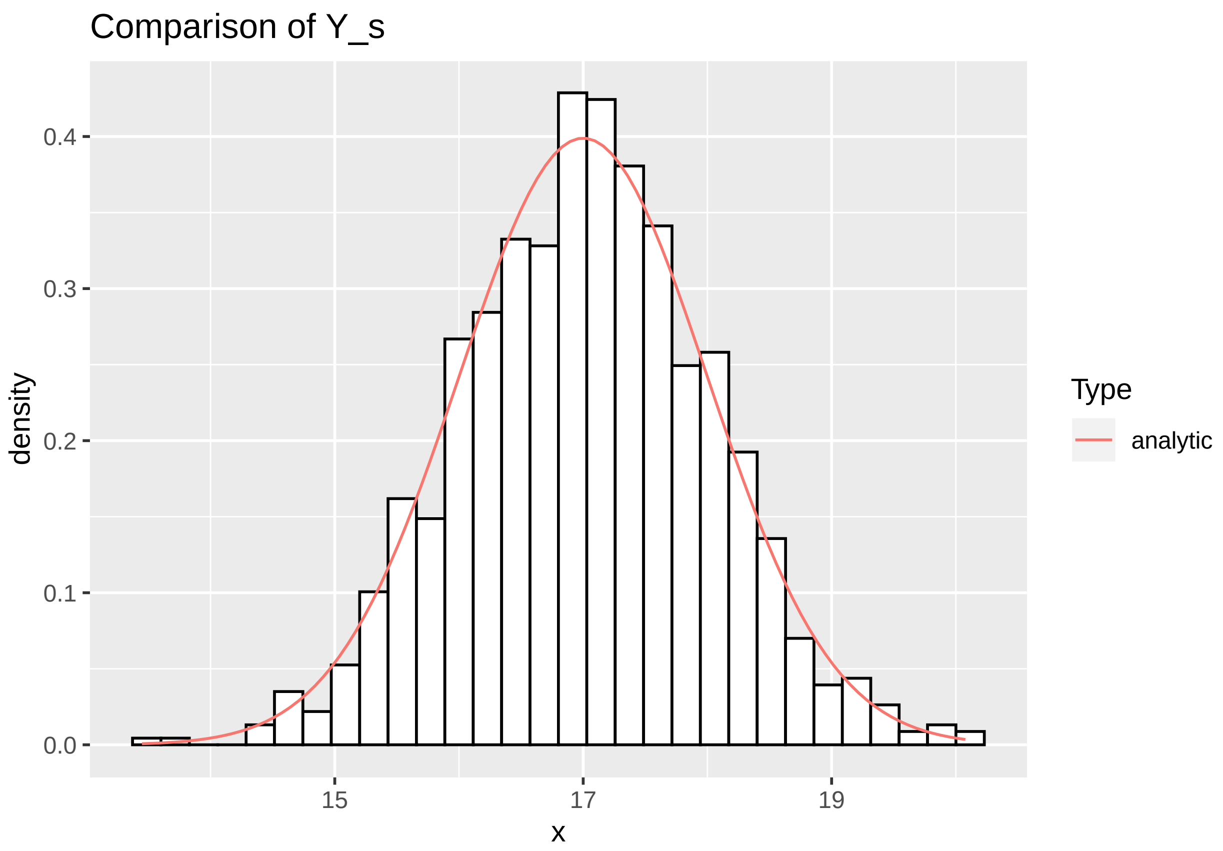
```r
ggplot(data.frame(x = B1_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = B1_hat_mean, sd = sqrt(B1_hat_var)), aes(color = "analy
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of beta_1_hat")
```



```
ggplot(data.frame(x = Y_s_hat_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = Y_s_hat_mean, sd = sqrt(Y_s_hat_var)), aes(color = "anal
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of E(Y_s | X = x_s)")
```



```
ggplot(data.frame(x = Y_s_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = Y_s_mean, sd = sqrt(Y_s_var)), aes(color = "analytic"))
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of Y_s")
```



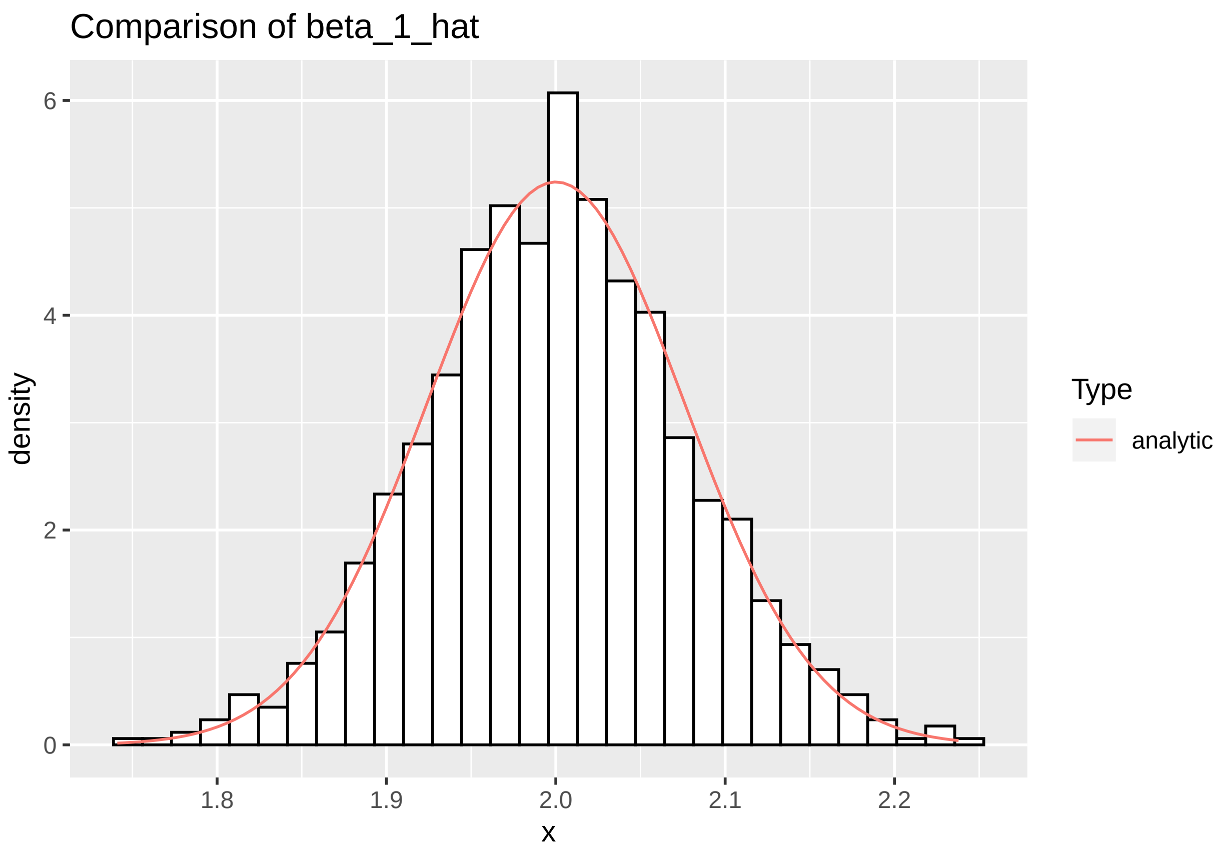
3. Introduce non-zero covariance into the joint distribution of the ϵ.

Commentary

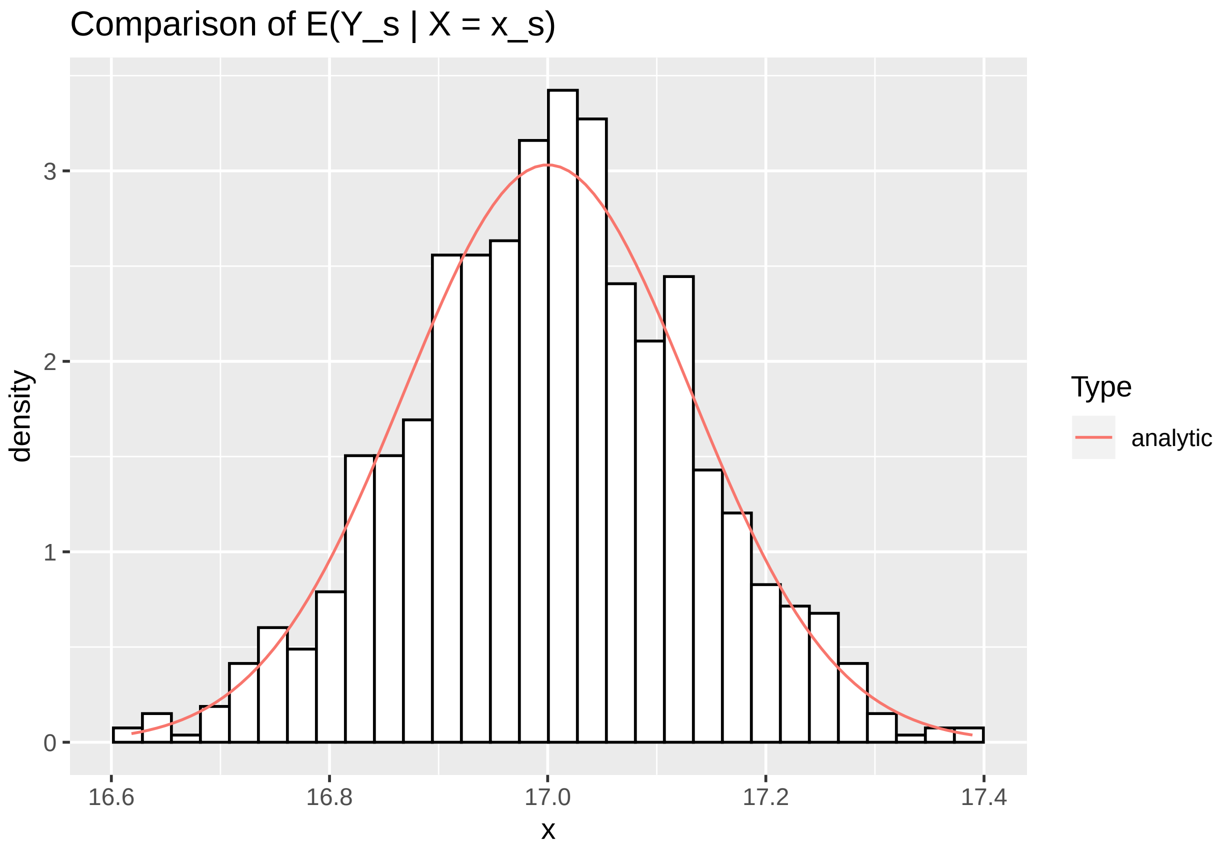
Introducing non-zero covariance appears to have a marginal impact on the sampling distributions of β_1, Y_s, and $E(Y_s)$. However, though the shape, spread, and centers of these statistics closely mirror their analytical forms, there are some slight differences for $E(Y_s)$. It is important to note these differences may be exacerbated by increasing the observations, number of simulations, or the magnitude of covariance (including signage). ### Code

```
# set params
B0 <- 1
B1 <- 2
B2 <- 4
B <- c(B0, B1, B2)
sigma <- 1
# complete specification
n <- 100
X <- rmvnorm(n, c(1,2,4),matrix(c(0,0,0,0,2,0,0,0,1), byrow = TRUE, nrow = 3))
# simulation
x_s <- c(1,2,3) # Given a fixed new observation
B1_samp <- rep(NA, 1000)
Y_s_hat_samp <- rep(NA, 1000)
Y_s_samp <- rep(NA, 1000)
for(i in 1:1000){
  epsilon <- rmvnorm(n, c(0,0,0),matrix(c(1,.5,.75,.5,2,.5,.75,.5,3), byrow = TRUE, nrow = 3))[,1]
  Y <- X %*% B + epsilon
  B_hat <- solve(t(X)%*%X)%*%t(X)%*%Y
  B1_samp[i] <- B_hat[2]
  Y_s_hat_samp[i] <- t(x_s)%*%B_hat
  Y_s_samp[i] <- t(x_s)%*%B + rnorm(1,mean=0,sd=sigma)
}
# Find the analytical form
B1_hat_mean <- B1
B1_hat_var <- sigma^2*solve(t(X)%*%X)[2,2]
Y_s_hat_mean <- t(x_s)%*%B
Y_s_hat_var <- t(x_s)%*%solve(t(X)%*%X)%*%x_s
Y_s_mean <- t(x_s)%*%B
Y_s_var <- sigma^2
```

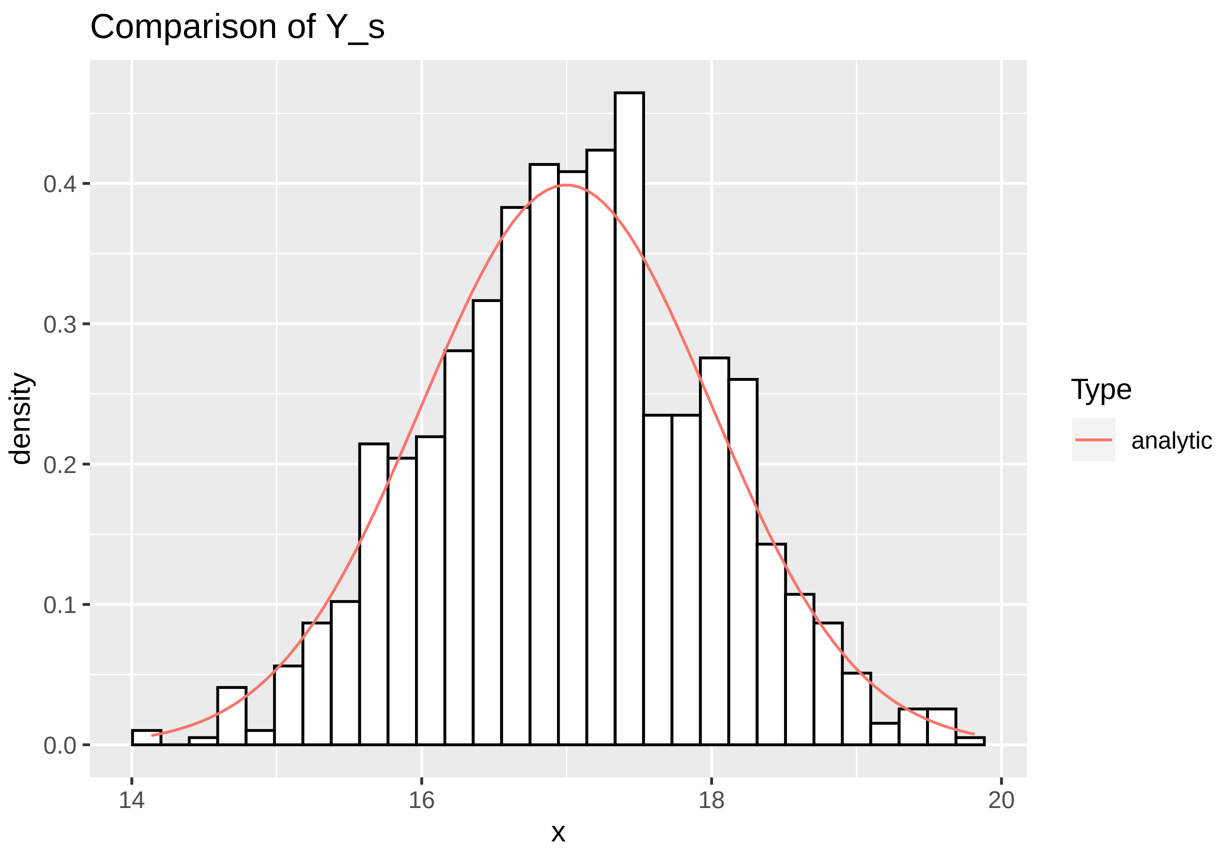
```
ggplot(data.frame(x = B1_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = B1_hat_mean, sd = sqrt(B1_hat_var)), aes(color = "analy
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of beta_1_hat")
```



```
ggplot(data.frame(x = Y_s_hat_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = Y_s_hat_mean, sd = sqrt(Y_s_hat_var)), aes(color = "anal
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of E(Y_s | X = x_s)")
```



```
ggplot(data.frame(x = Y_s_samp),aes(x)) +
  geom_histogram(aes(color = "simulation", y = ..density..), colour="black", fill="white") +
  stat_function(fun = dnorm, args = list(mean = Y_s_mean, sd = sqrt(Y_s_var)), aes(color = "analytic"))
  guides(color=guide_legend(title="Type")) +
  ggtitle("Comparison of Y_s")
```



Does the inference change for the statistics investigated in the previous exercise? For this part, you should be able to recycle much of your code from the previous part.

Part III. Variance/Covariance

Generate a scatterplot matrix involving all pairwise relationships between the three simulated regression coefficients in the original model. To be clear, this involves generating a good number fitted betas from your simulator and plotting them. Based on a visual assessment of these plots, please characterize the joint distribution of these statistics in terms of center, shape, spread, and correlation. Compare the empirical covariance matrix to the analytical form that we derived in class.

Empirical Note

The covariance matrix C is given as follows:

$$C = \begin{pmatrix} Cov(\beta_0, \beta_0) & Cov(\beta_0, \beta_1) & Cov(\beta_0, \beta_2) \\ Cov(\beta_1, \beta_0) & Cov(\beta_1, \beta_1) & Cov(\beta_1, \beta_2) \\ Cov(\beta_2, \beta_0) & Cov(\beta_2, \beta_1) & Cov(\beta_2, \beta_2) \end{pmatrix}$$

Where $Cov(\beta_0, \beta_0) = Var(\beta_0)$, $Cov(\beta_1, \beta_1) = Var(\beta_1)$, and $Cov(\beta_2, \beta_2) = Var(\beta_2)$

Additionally, when considering covariances, it is helpful to note:

$Cov(\hat{\beta}|X) = \sigma^2(X'X)^{-1}$

Commentary

As the non-diagonal entries of the covariance matrix are identical (symmetric about the diagonal), e.g. $Cov(\beta_1, \beta_0) = Cov(\beta_0, \beta_1)$, there are three distributions of note:

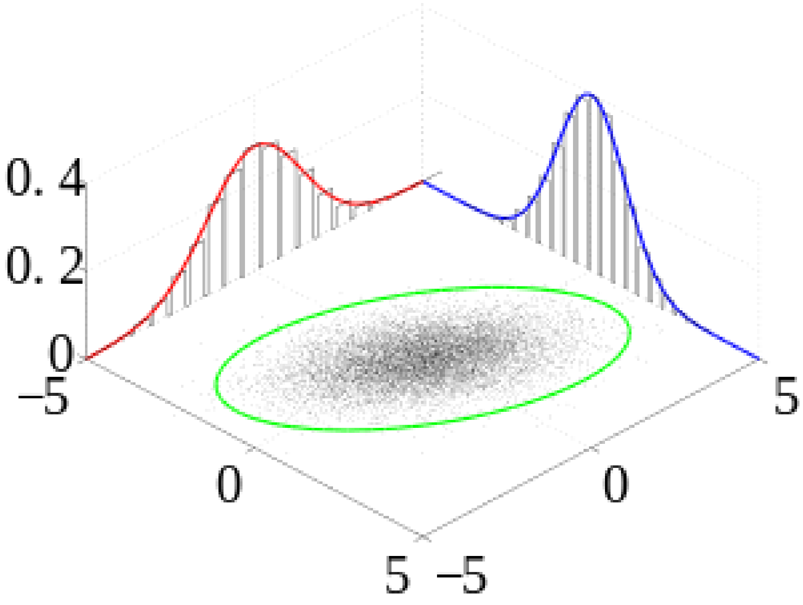


Figure 1: Joint Probability Distribution

(1): $Cov(\beta_1, \beta_0)$

Similar to the image attached above, the joint distribution of β_1 and β_0 indicate that both distributions appear normally distributed with little to no correlation between each other.

(2): $Cov(\beta_2, \beta_1)$

Similar to (1) and the image attached above, the joint distribution of β_1 and β_2 indicate that both distributions appear normally distributed with little to no correlation between each other.

(3): $Cov(\beta_2, \beta_0)$

Interestingly, we get a very different picture of the distribution of β_0 and β_2 compared to (1) and (2). While the joint distributions of (1) and (2) were circular, the joint distribution of (3) is elongated, making a “football-like” shape. Additionally, β_0 and β_2 appear negatively correlated.

(4):

In reviewing (1) through (3), it is apparent that each of the covariances are negative. This is similar to the empirical results we obtained in class of the non-diagonal entries having a negative value (though that value was $-\bar{x}$) for the case when p=2.

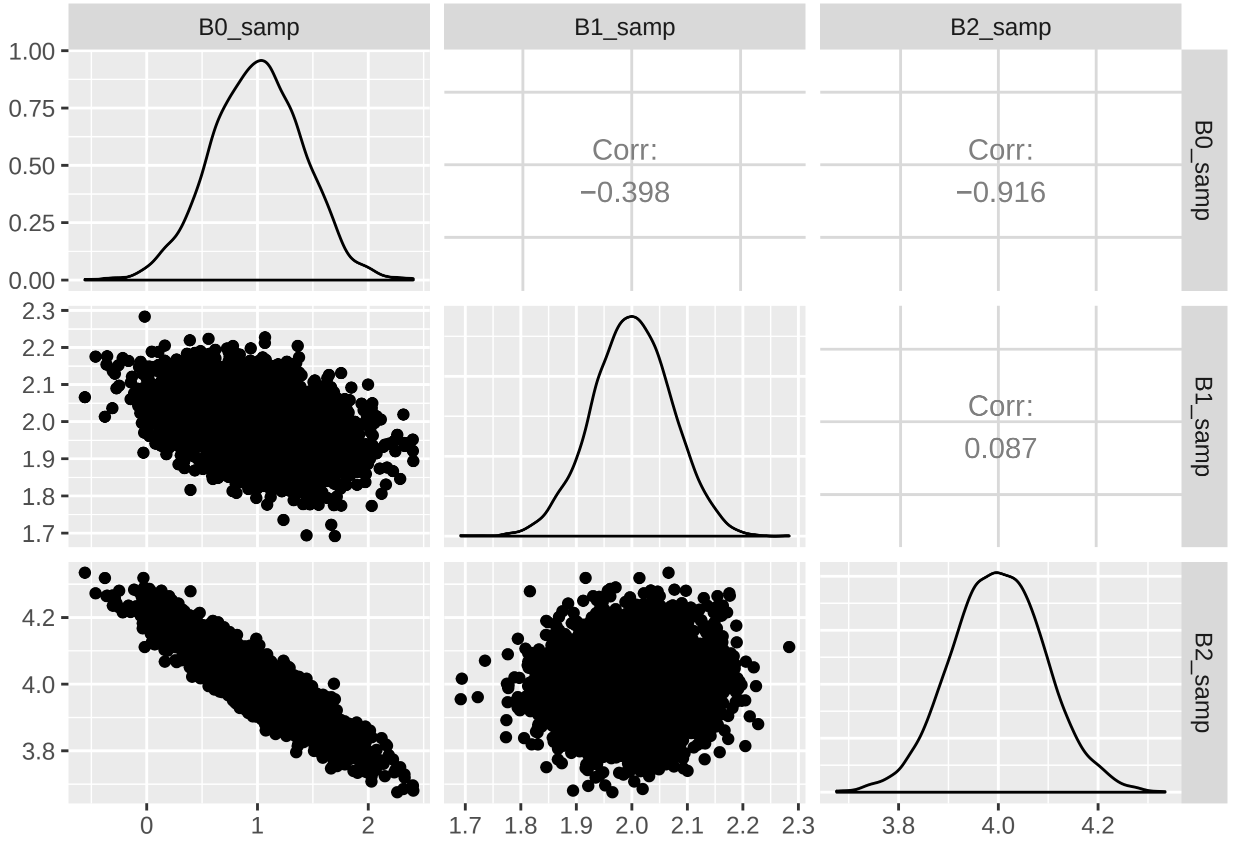
Code

```
# set params
B0 <- 1
B1 <- 2
B2 <- 4
B <- c(B0, B1, B2)
```

```
sigma <- 1
# complete specification
n <- 100
X <- rmvnorm(n, c(1,2,4),matrix(c(0,0,0,0,2,0,0,0,1), byrow = TRUE, nrow = 3))
# simulation
x_s <- c(1,2,3) # Given a fixed new observation
B0_samp <- rep(NA, 5000)
B1_samp <- rep(NA, 5000)
B2_samp <- rep(NA, 5000)
for(i in 1:5000){
  epsilon <- rnorm(n, mean = 0, sd = sigma)
  Y <- X %*% B + epsilon
  B_hat <- solve(t(X)%*%X)%*%t(X)%*%Y
  B0_samp[i] <- B_hat[1]
  B1_samp[i] <- B_hat[2]
  B2_samp[i] <- B_hat[3]
}
# Find the analytical form
B0_hat_mean <- B0
B0_hat_var <- sigma^2*solve(t(X)%*%X)[1,1]
B1_hat_mean <- B1
B1_hat_var <- sigma^2*solve(t(X)%*%X)[2,2]
B2_hat_mean <- B2
B2_hat_var <- sigma^2*solve(t(X)%*%X)[3,3]

# Dataset of Betas
Betas <- cbind(B0_samp,B1_samp,B2_samp)
Betas <- data.frame(Betas)
```

```
library(GGally)
ggpairs(Betas)
```


B0_samp
B1_samp
B2_samp
B0_samp
B1_samp
B2_samp
1.00
0.75
0.50
0.25
0.00
Corr:
−0.398
Corr:
−0.916
2.3
2.2
2.1
2.0
1.9
1.8
1.7
Corr:
0.087
4.2
4.0
3.8
0
1
2
1.7
1.8
1.9
2.0
2.1
2.2
2.3
3.8
4.0
4.2