Problem Set 8 – MATH392

Theodore Dounias

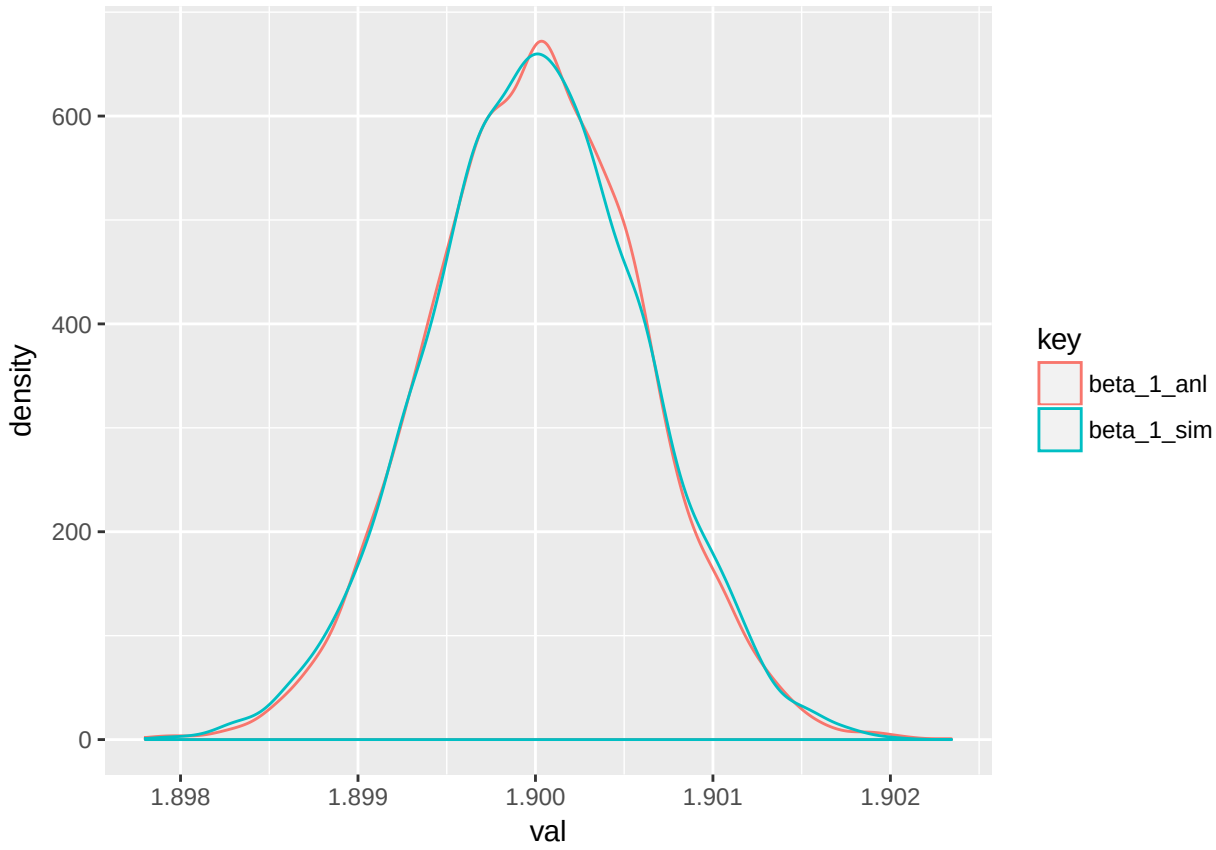
3/31/2018

Simulator

```
# set params
b0 <- 2.5
b1 <- 1.9
b2 <- 8.1
sigma <- .1

beta<- matrix(c(b0, b1, b2), nrow = 3, ncol = 1)

# complete specification
n <- 1000
epsilon <- matrix(rnorm(n, 0, sigma), nrow = n, ncol = 1)
x <-matrix(c(rep(1, n), x_1 <- rexp(n, .2), x_2 <- rexp(n, .1)), nrow = n, ncol = 3)

# simulate ys
y_simulate <- function(x){
  y <- x%*%beta + epsilon
}

y_true <- x%*%beta
```

Sampling Distribution

```
#Beta_1

it <- 5000
beta_1_sim <- rep(0, it)
for(i in 1:it){
  epsilon <- matrix(rnorm(n, 0, sigma), nrow = n, ncol = 1)
  y <- y_simulate(x)

  lm1 <- lm(y ~ x_1 + x_2)

  beta_1_sim[i] <- as.numeric(coef(lm1)[2])
}

beta_1_anl <- rmvnorm(it, mean = beta, sigma = sigma^2* solve(t(x) %*% x))[,2]

df <- data.frame(beta_1_sim, beta_1_anl)

df <- gather(df, key, val)

ggplot(df, aes(col = key, x = val), col = "red", "blue") +
  geom_density()
```



The results for values of beta are almost identical between the MC approximation simulation and the analytical distribution.

To find the analytical distribution, we need the variance of $E(Y_s), Y_s$, which we can find in the following way, since H is idempotent:

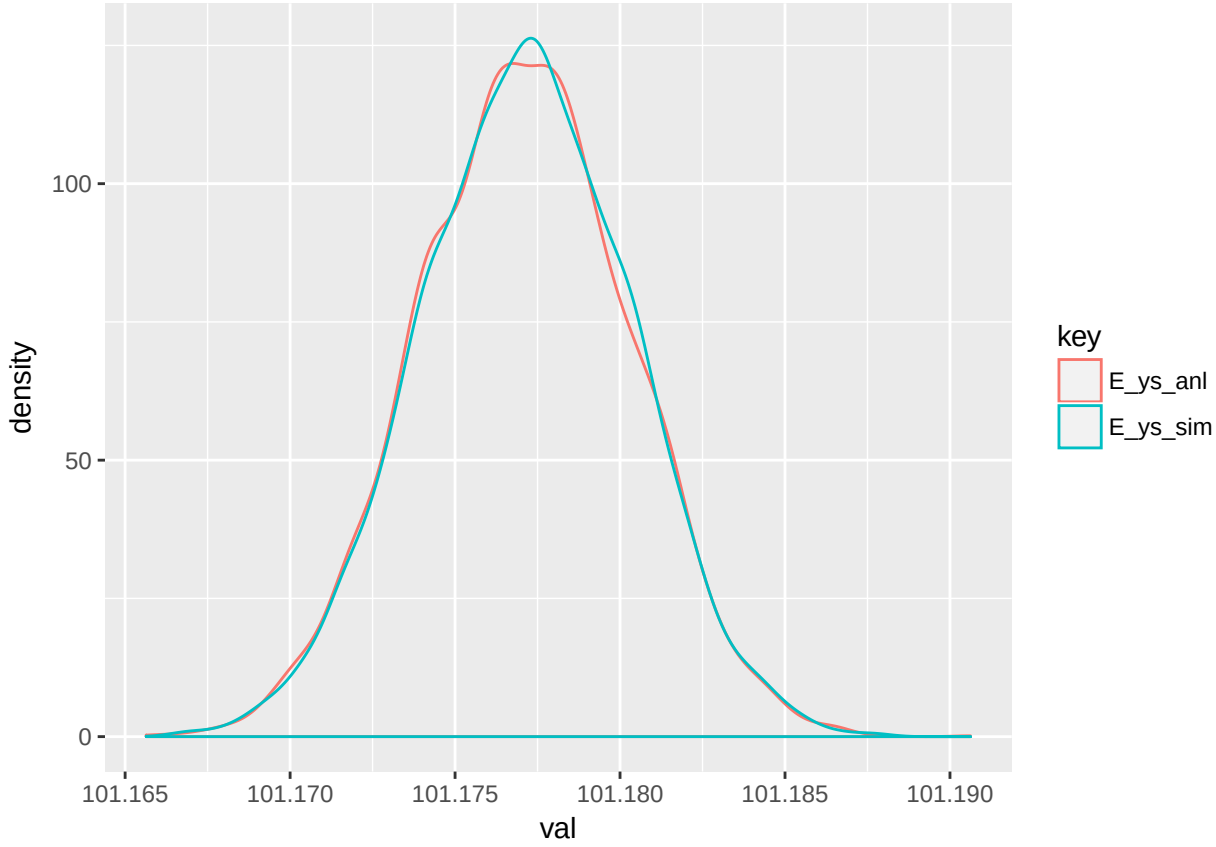
$$Var(E(Y_s)|X = x_s) = Var(XB) = Var(HY) = HVar(Y)H^T = \sigma^2 HH^T = \sigma^2 H$$

For the variance of Y_s, The calculation is similar, with the addition of an error term:

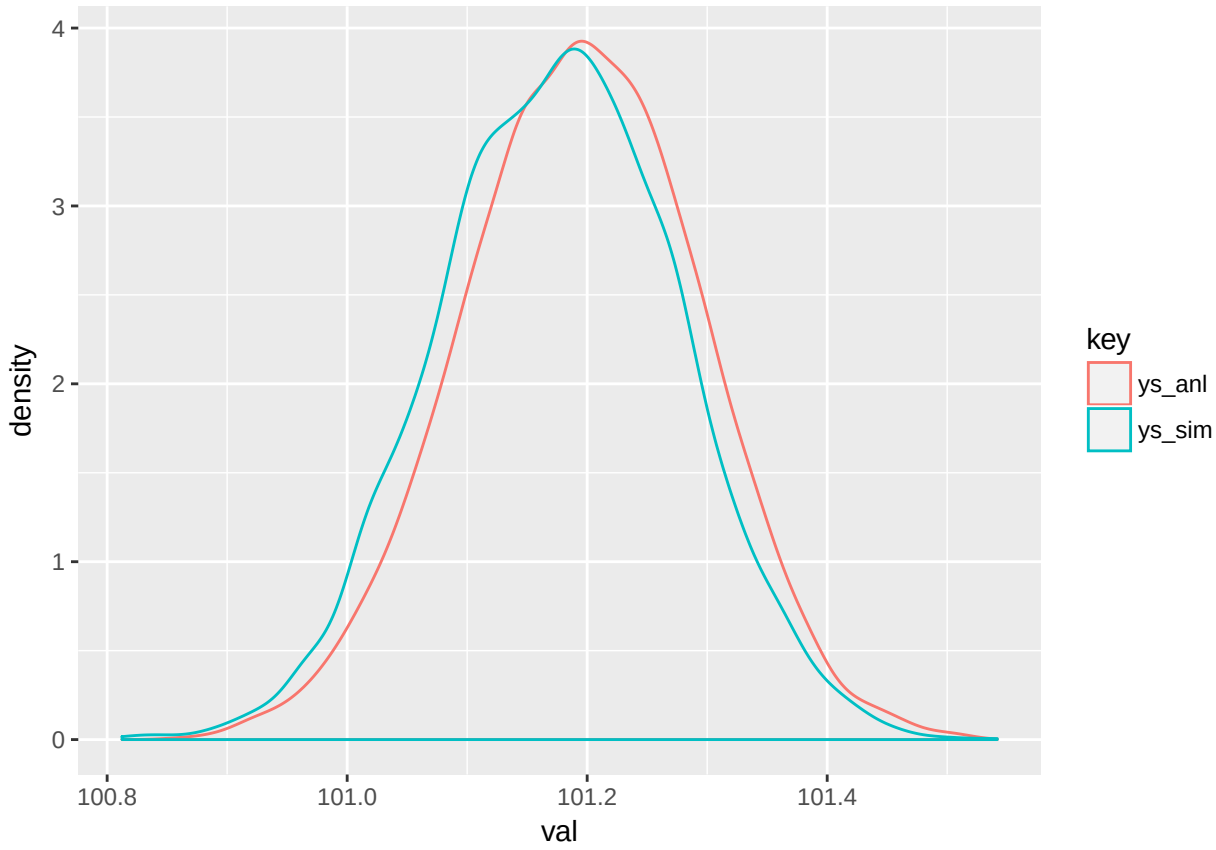
$$Var(Y_s|X = x_s) = Var(XB+\hat{e}) =_{indep} Var(HY)+Var(\hat{e}) = HVar(Y)H^T+\sigma^2(I_{nxn}-H) = \sigma^2(H+I_{nxn}-H) = \sigma^2 I_{nxn}$$

```
#E(Ys)

E_ys_sim <- rep(0, it)
ys_sim <- rep(0, it)

xs <- x[12,]
for(i in 1:it){
  epsilon <- matrix(rnorm(n, 0, sigma), nrow = n, ncol = 1)
  y <- y_simulate(x)
  lm2 <- lm(y ~ x_1 + x_2)
  coef <- as.matrix(coef(lm2))

  E_ys_sim[i] <- xs %*% coef
  ys_sim[i] <- xs %*% coef + epsilon[12]
}
```

```
H <- x %*% solve(t(x) %*% x) %*% t(x)

E_ys_anl <- rmvnorm(it, x%*%beta, sigma^2 * H)[,12]

ys_anl <- rmvnorm(it, x%*%beta + epsilon, sigma^2 * diag(1000))[,12]

df <- data.frame(E_ys_sim, E_ys_anl)

df <- gather(df, key, val)

ggplot(df, aes(col = key, x = val), col = "red", "blue") +
  geom_density()
```




```
df <- data.frame(ys_sim, ys_anl)

df <- gather(df, key, val)

ggplot(df, aes(col = key, x = val), col = "red", "blue") +
  geom_density()
```



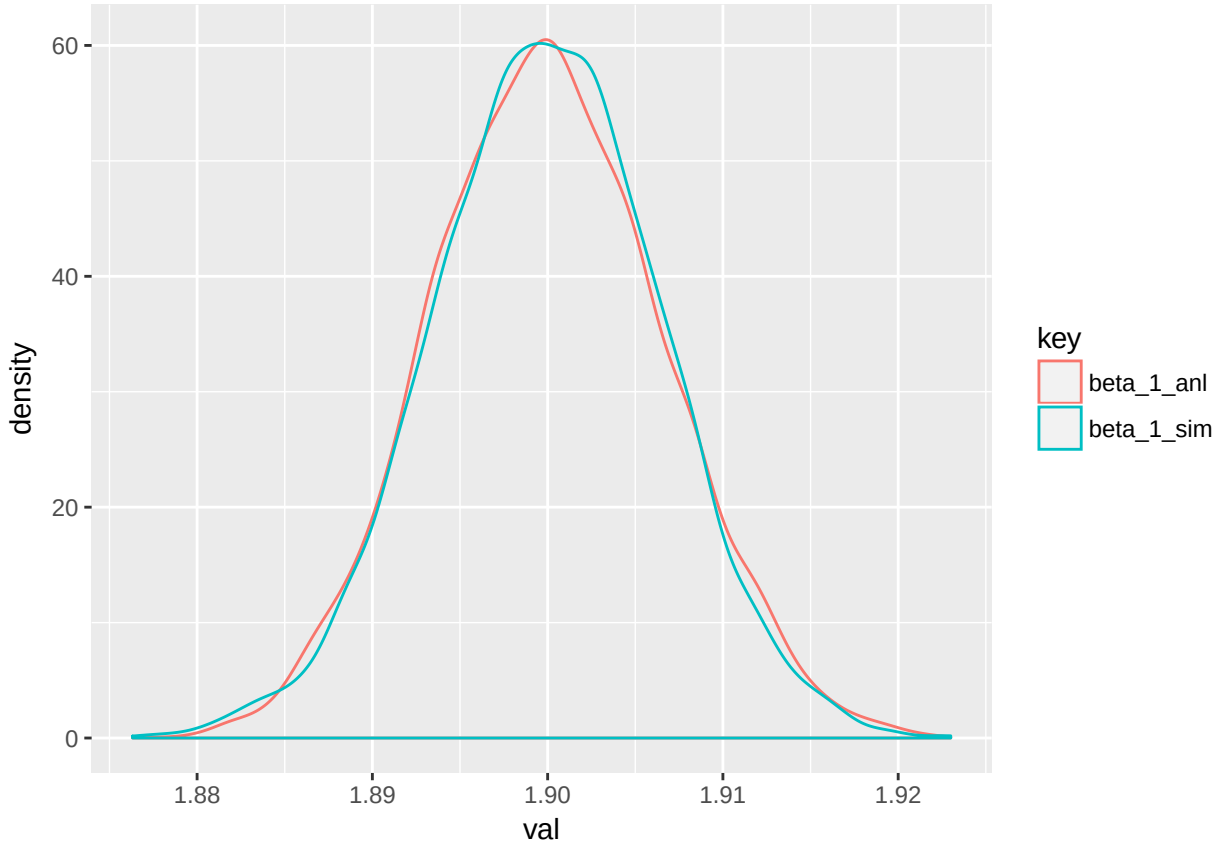
The means of our estimated fitted value have the same analytical and MC approximated distributions. The fitted values y_s themselves have the same shape and spread, but not the same center; possibly due to the existence of error rates.

A Different Model

1. If we give an alternate distribution of the X vector, no result should be different. There will be different betas, but the conclusions should still stand, as the only thing that we actually shift are the values that y takes, not anything substantial in our process. Indeed, if I repeat the code:

```
xalt <-matrix(c(rep(1, n), x_1alt <- rgamma(n, 2, 3), x_2alt <- rgamma(n, 1, 3)), nrow = n, ncol = 3)

#Beta_1
it <- 5000
beta_1_sim <- rep(0, it)
for(i in 1:it){
  epsilon <- matrix(rnorm(n, 0, sigma), nrow = n, ncol = 1)
  y <- y_simulate(xalt)

  lm1 <- lm(y ~ x_1alt + x_2alt)

  beta_1_sim[i] <- as.numeric(coef(lm1)[2])
}

beta_1_anl <- rmvnorm(it, mean = beta, sigma = sigma^2* solve(t(xalt) %*% xalt))[,2]

df <- data.frame(beta_1_sim, beta_1_anl)
```

```
df <- gather(df, key, val)

ggplot(df, aes(col = key, x = val), col = "red", "blue") +
  geom_density()
```

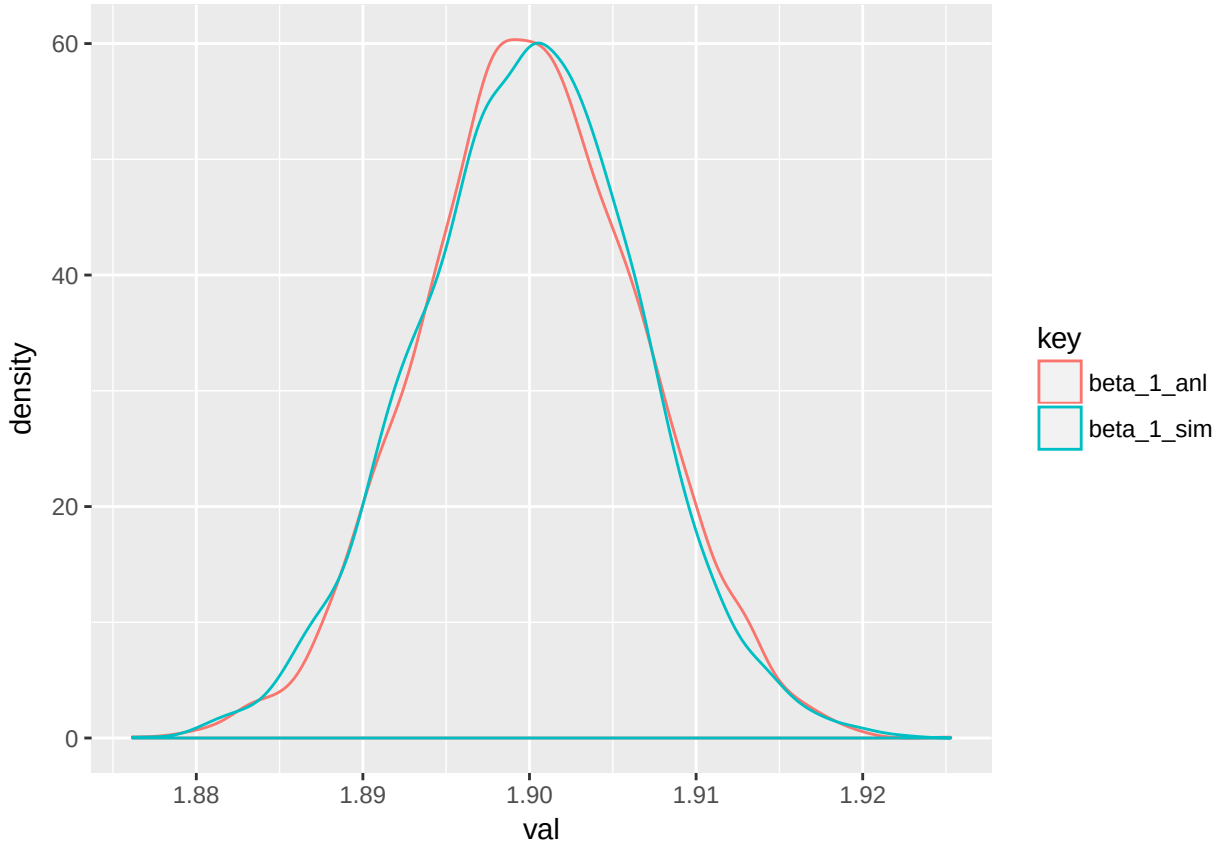


I ran this for the second and third test statistic an reached similar conslusions; these are ommitted for the sake of brevity.

2. The form of inference would be affected for all statistics, since the variance of the error terms plays a role in calculating the three analytical distributions. However, none of the results should be affected, since the distribution of error terms is still centered at 0. I, again, repeat the first inference:

```
#Beta_1
it <- 5000
beta_1_sim <- rep(0, it)
for(i in 1:it){
  epsilon <- matrix(rt(n, 14), nrow = n, ncol = 1)
  y <- y_simulate(x)

  lm1 <- lm(y ~ x_1 + x_2)

  beta_1_sim[i] <- as.numeric(coef(lm1)[2])
}

beta_1_anl <- rmvnorm(it, mean = beta, sigma = (14/12)* solve(t(x) %*% x))[,2]

df <- data.frame(beta_1_sim, beta_1_anl)
```

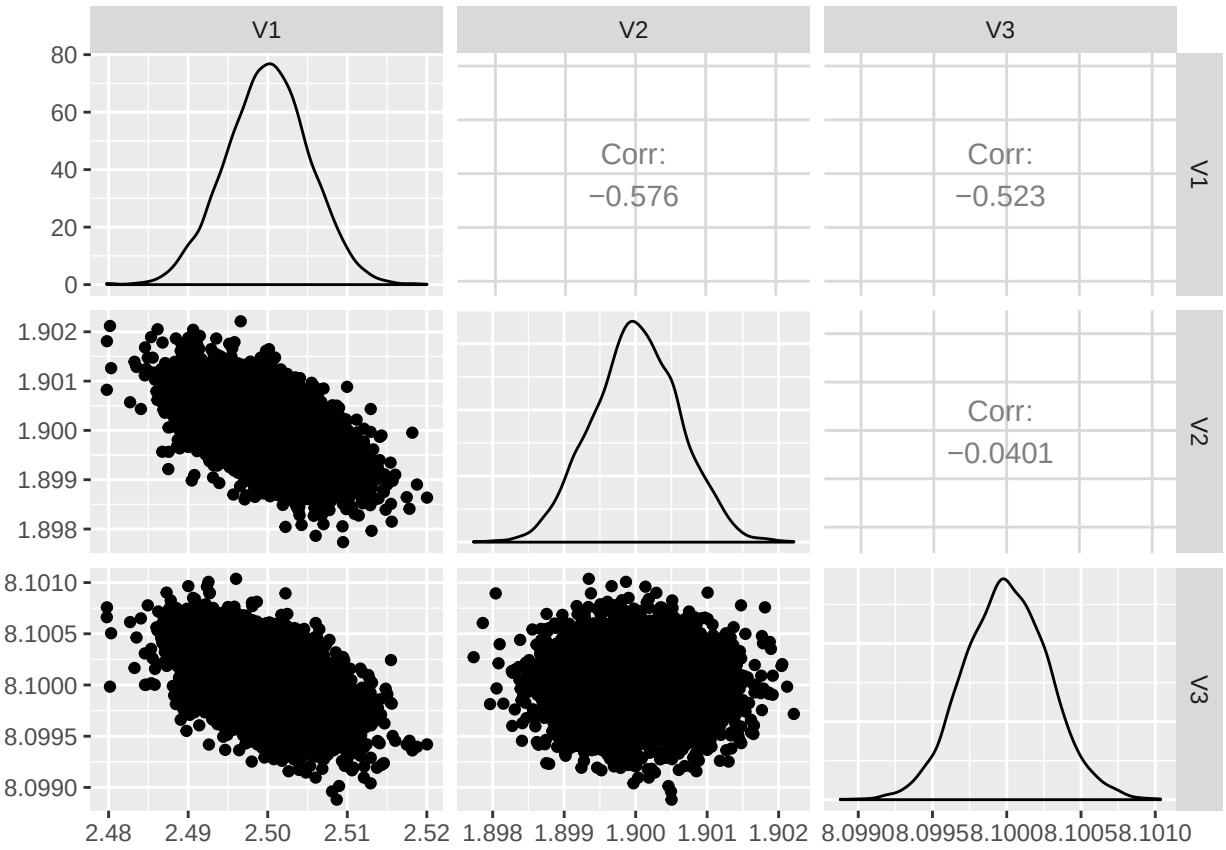
```
df <- gather(df, key, val)

ggplot(df, aes(col = key, x = val), col = "red", "blue") +
  geom_density()
```



As expected, the results are unchanged when using Student's t-distribution instead of a normal distribution for the error terms.

Variance/Covariance

```
beta_sim <- matrix(rep(0, 3*it), nrow = it, ncol = 3)
for(i in 1:it){
  epsilon <- matrix(rnorm(n, 0, sigma), nrow = n, ncol = 1)
  y <- y_simulate(x)

  lm1 <- lm(y ~ x_1 + x_2)

  beta_sim[i, 1] <- as.numeric(coef(lm1)[1])
  beta_sim[i, 2] <- as.numeric(coef(lm1)[2])
  beta_sim[i, 3] <- as.numeric(coef(lm1)[3])
}

ggpairs(as.data.frame(beta_sim))
```



The centers of each individual beta are 2.5, 1.9, and 8.1 respectvelly, which are also the values set for each of them. The shape of the joints between 1-3 and 1-2 appear to have some linearity, while 2-3 seem independent. The values for correlation are written in the plot above, from which we can also easily infer a similar value for covariance, due to the relation between the statistics. 1-3, 1-2 are negativelly correlated.