STAT-S631

Assignment 3

John Koo

Question 1

Show that $\hat{\beta}_0$ and $\hat{\beta}_1$ can be written as a linear combination of $y_1, ..., y_n$

$\hat{\beta}_1 = \frac{\text{SXY}}{\text{SXX}}$, and $\text{SXY} = \sum_i (x_i - \bar{x}) y_i$. SXX does not depend on any of the y_is. Therefore:

$$\begin{aligned}\hat{\beta}_1 &= \frac{\text{SXY}}{\text{SXX}} \\ &= \sum_i \frac{(x_i - \bar{x})}{\text{SXX}} y_i\end{aligned}$$

Which is a linear combination of y_is.

On the other hand, $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$, and $\frac{1}{n}\bar{y} = \sum_i y_i$, so:

$$\begin{aligned}\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\ &= \sum_i \frac{y_i}{n} - \sum_i \frac{(x_i - \bar{x})\bar{x}}{\text{SXX}} y_i \\ &= \sum_i \left(\frac{1}{n} - \frac{(x_i - \bar{x})\bar{x}}{\text{SXX}} \right) y_i\end{aligned}$$

Which is a linear combination of y_i.

Question 2

[From ALR 2.2]

Part 1

Points above the line $y = x$ represent cities for which rice prices were higher in 2009 than in 2003. Similarly, points below the line $y = x$ represent cities for which rice prices were lower in 2009 than in 2003.

Part 2

```
ubs.df <- alr4::UBSprices
ubs.df %>%
  dp$mutate(city = rownames(.)) %>%
  dp$mutate(rice.price.diff = rice2009 - rice2003) %>%
  dp$filter(rice.price.diff == max(rice.price.diff)) %>%
  .$city
```

```
[1] "Vilnius"
```

Part 3

No, there is a $\hat{\beta}_0$ term.

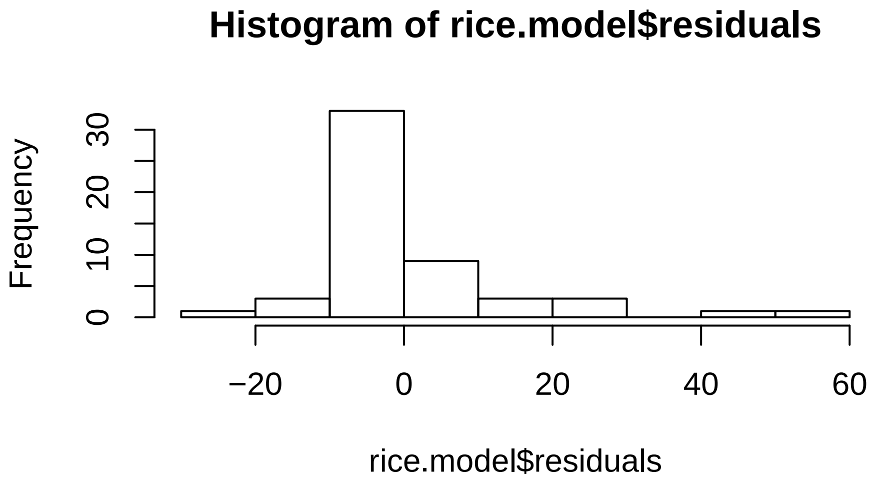
If rice prices in 2009 were lower than rice prices in 2003 according to the model, then the inequality $\hat{\beta}_0+\hat{\beta}_1 x < x$ would have to be true. Then (assuming $\hat{\beta}_1 < 1$ as in the question), the solution of this inequality is $x > \frac{\hat{\beta}_0}{1-\hat{\beta}_1}$. In other words, according to the model the statement is true only when rice prices in 2003 were greater than $\frac{\hat{\beta}_0}{1-\hat{\beta}_1}$.

Furthermore, this is an empirical model for the overall trend, so it does not describe the data exactly regardless.

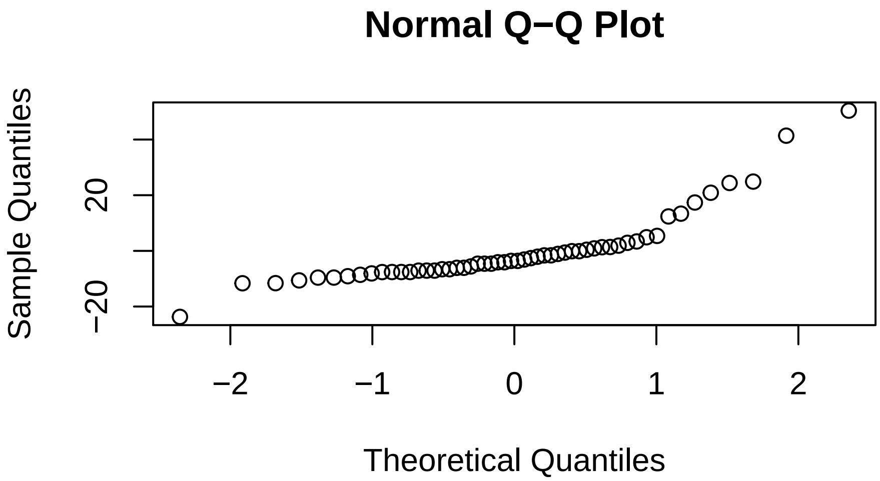
Part 4

We can check if the data fit the OLS assumptions:

```
rice.model <- lm(rice2009 ~ rice2003, data = ubs.df)
hist(rice.model$residuals)
```

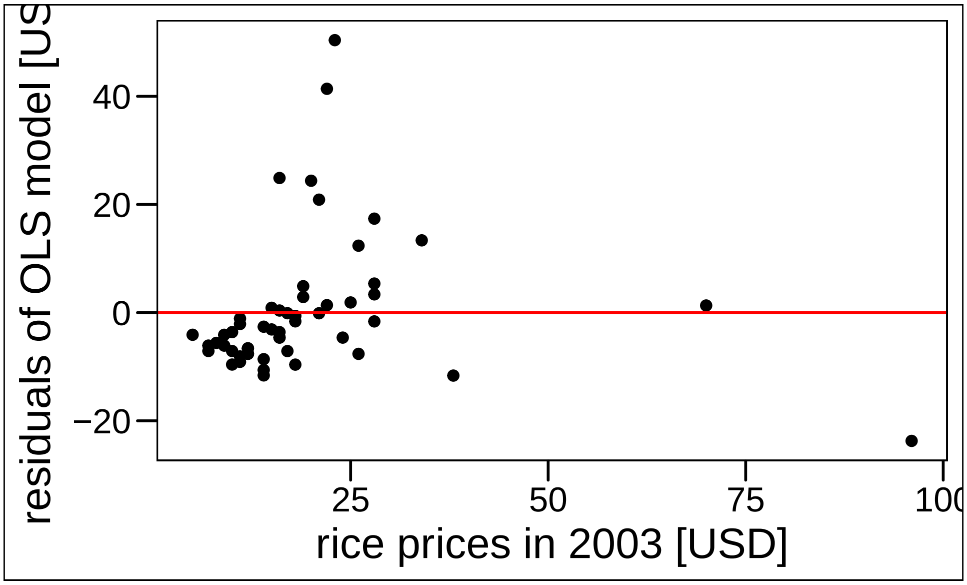


```
qqnorm(rice.model$residuals)
```



```
ubs.df %>%
  dp$mutate(resids = rice.model$residuals) %>%
  ggplot() +
```

```r
  geom_point(aes(x = rice2003, y = resids)) +
  labs(x = 'rice prices in 2003 [USD]',
       y = 'residuals of OLS model [USD]') +
  geom_hline(yintercept = 0, colour = 'red')
```



Here we can see that the residuals are not i.i.d. normal conditioned on our input variable. In particular, the distribution appears skewed to the right based on the histogram, and the plot of the residuals vs the input variable is not a null plot.

Question 3

Simulation: Assume the simple linear regression model:

$$y_i = \beta_0 + \beta_1 x_i + e_i, i = 1, ..., n$$

where e_i $N(0, \sigma^2)$ for $i = 1, ..., n$.

Let's set $\beta_0 = 10$, $\beta_1 = -2.5$, and $n = 30$

```r
b0 <- 10
b1 <- -2.5
n <- 30
```

Part a

Set $\sigma = 100$ and $x_i = i$ for $i = 1, ..., n$.

```r
s <- 100
x <- seq(n)
```

Part b

Your simulation will have 10,000 iterations. Before you start your iterations, set a random seed using your birthday date (MMDD) and report the seed with your responses. For each iteration, obtain and store your linear regression parameter estimates: $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\sigma}^2$. (Include syntax. DO NOT include output)

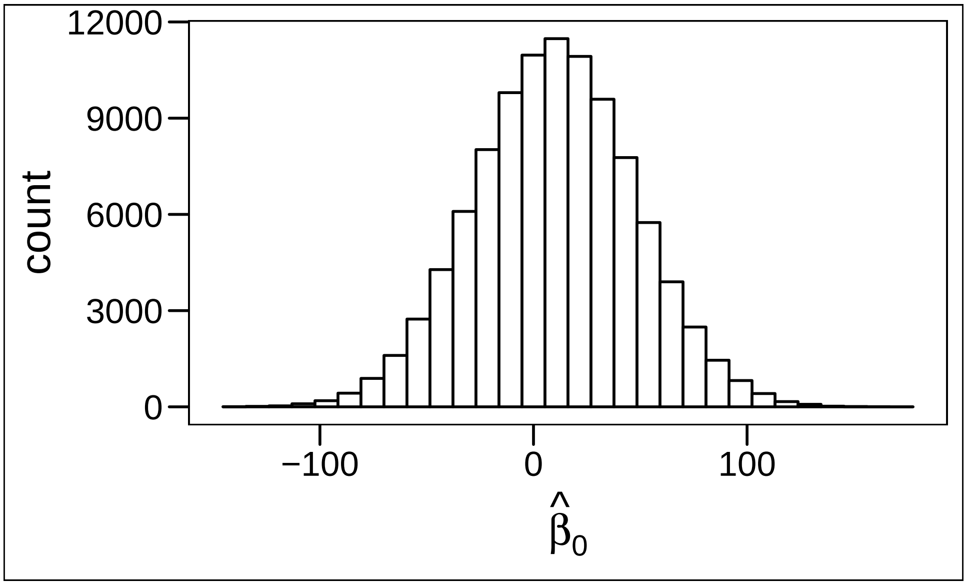
```
iter <- 1e5
set.seed(0825)
estimates.df <- foreach(i = seq(iter), .combine = dp$bind_rows) %dopar% {
  y <- b0 + b1 * x + rnorm(length(x), sd = s)
  temp.model <- lm(y ~ x)
  dplyr::data_frame(b0.hat = coef(temp.model)['(Intercept)'],
                    b1.hat = coef(temp.model)['x'],
                    sigma.hat = sigma(temp.model))
}
```

Part c

Obtain and present three histograms, one for each $\hat{\beta}_0$'s, $\hat{\beta}_1$'s, and $\hat{\sigma}^2$'s. Briefly describe the main characteristics of these histograms (shape of the estimated distributions). (Include syntax and output)

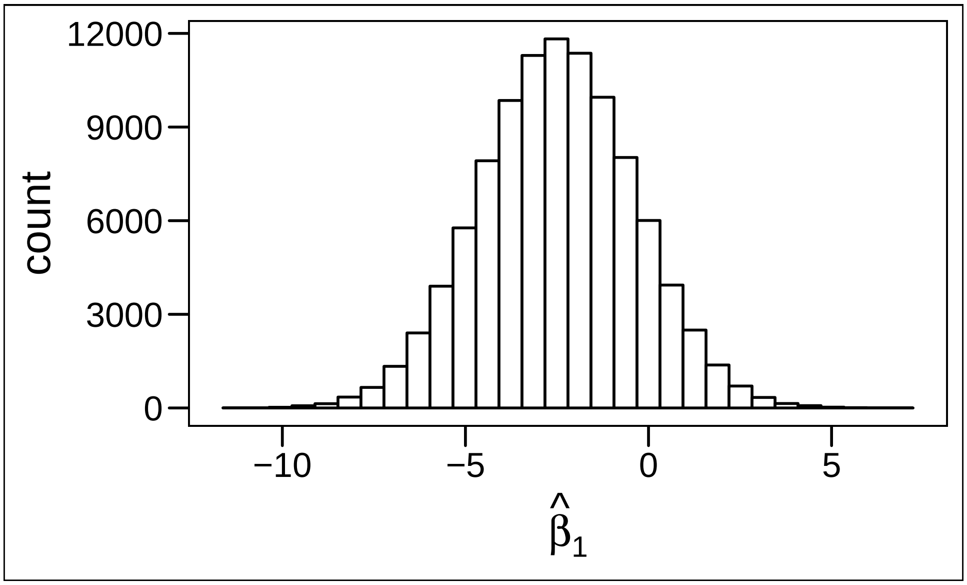
```
ggplot(estimates.df) +
  geom_histogram(aes(x = b0.hat),
                 fill = 'white', colour = 'black') +
  labs(x = expression(hat(beta)[0]))
```

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



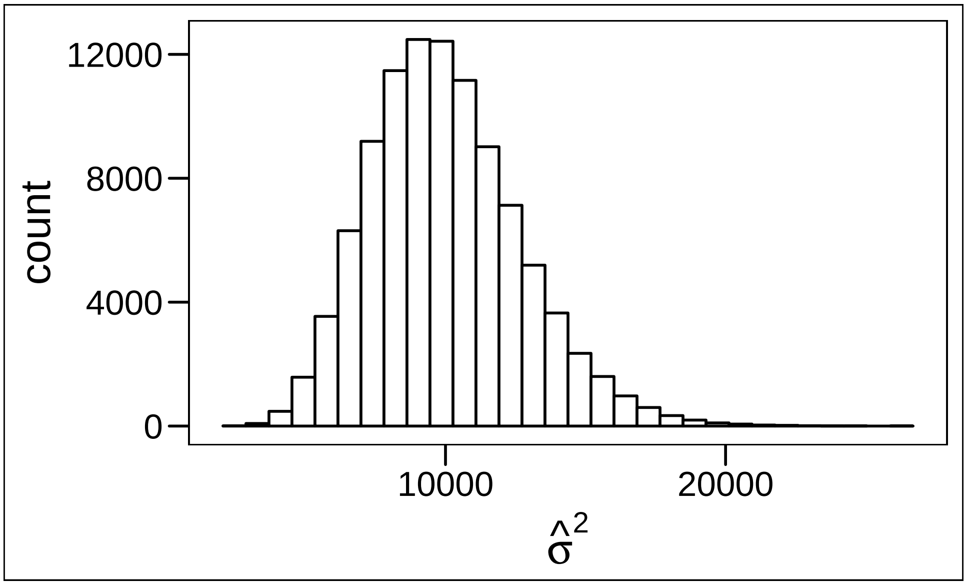
```
ggplot(estimates.df) +
  geom_histogram(aes(x = b1.hat),
                 fill = 'white', colour = 'black') +
  labs(x = expression(hat(beta)[1]))
```

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



```
ggplot(estimates.df) +
  geom_histogram(aes(x = sigma.hat ** 2),
                 fill = 'white', colour = 'black') +
  labs(x = expression(hat(sigma)^2))
```

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



$\hat{\beta}_0$ and $\hat{\beta}_1$ appear to be normally distributed with means at around the true values. The histogram of $\hat{\sigma}^2$ is skewed to the right, similar to that of an χ^2-distribution.

Part d

Find the averages of $\hat{\beta}_0$'s, $\hat{\beta}_1$'s, and $\hat{\sigma}^2$'s. How do they compare with the true parameters? Briefly explain. (Include syntax and output)

```
estimates.df %>%
  dp$mutate(sigma2.hat = sigma.hat ** 2) %>%
  dp$select(-sigma.hat) %>%
  dp$summarise_all(mean)
```

```
# A tibble: 1 x 3
    b0.hat    b1.hat sigma2.hat
     <dbl>     <dbl>      <dbl>
1 9.879461 -2.494049   9997.601
```

They are close to the true parameters, $(10, -2.5, 10000)$. We can say that the expected relative error is on the order of $\frac{1}{\sqrt{\text{iterations}}} = .01$, which appears consistent with the results.

Part e

Find the (sample) variance of $\hat{\beta}_0$'s and $\hat{\beta}_1$'s. How do they compare with the true variances? Briefly explain. (Include syntax and output)

```
estimates.df %>%
  dp$select(-sigma.hat) %>%
  dp$summarise_all(var)
```

```
# A tibble: 1 x 2
    b0.hat   b1.hat
     <dbl>    <dbl>
1 1404.679 4.456576
```

```
cross.prod.sum <- function(x, y = NULL) {
  # sum of cross product (e.g., SXX, SYY, SXY)
  if (is.null(y)) y <- x
  sum((x - mean(x)) * (y - mean(y)))
}

# var(beta1.hat | x)
s ** 2 / cross.prod.sum(x)
```

```
[1] 4.449388
```

```
# var(beta0.hat | x)
s ** 2 * (1 / n + mean(x) ** 2 / cross.prod.sum(x))
```

```
[1] 1402.299
```

The values are very close to the true values.

Part f

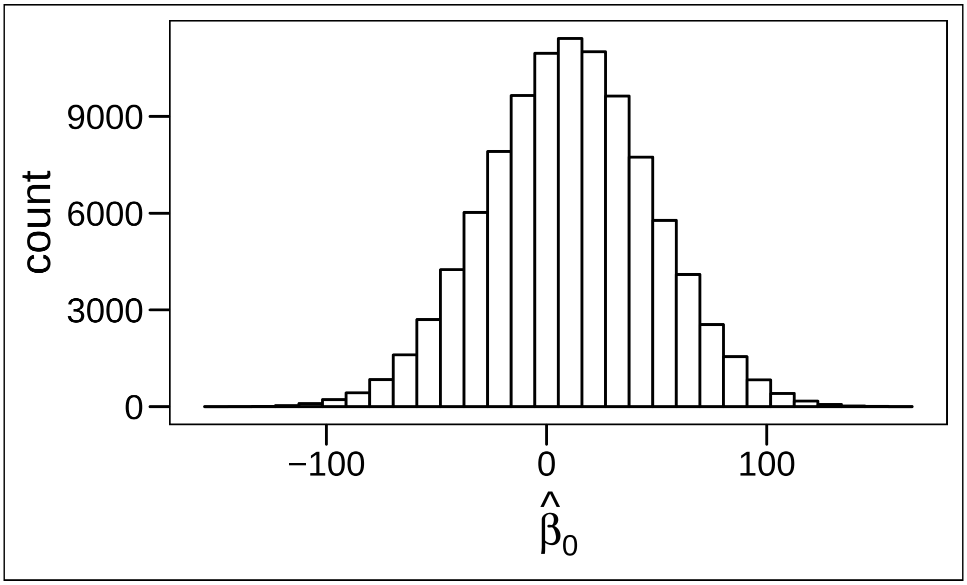
Now set $\sigma = 100$ and $x_i = 100i$ for $i = 1, ..., n$. Repeat parts b), d), and e). How does the (sample) variance of $\hat{\beta}_0$'s and $\hat{\beta}_1$'s compare with your previous results (in part e))? Briefly explain why.

```
# redefine the variables
s <- 100
x <- 100 * seq(n)

# run the simulation
set.seed(0825)
estimates.df <- foreach(i = seq(iter), .combine = dp$bind_rows) %dopar% {
  y <- b0 + b1 * x + rnorm(length(x), sd = s)
  temp.model <- lm(y ~ x)
  dplyr::data_frame(b0.hat = coef(temp.model)['(Intercept)'],
                    b1.hat = coef(temp.model)['x'],
                    sigma.hat = sigma(temp.model))
}
```

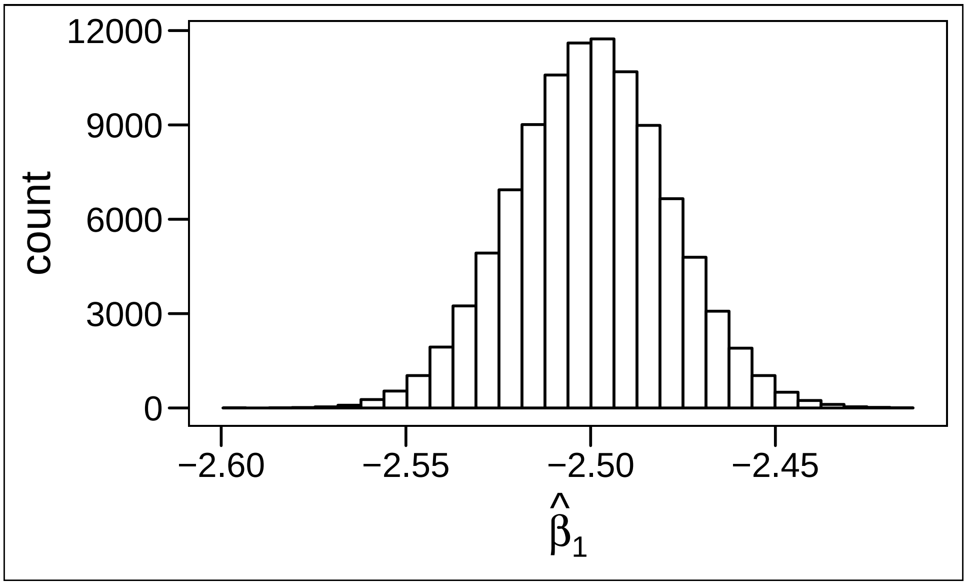
```
# histograms
ggplot(estimates.df) +
  geom_histogram(aes(x = b0.hat),
                 fill = 'white', colour = 'black') +
  labs(x = expression(hat(beta)[0]))
```

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



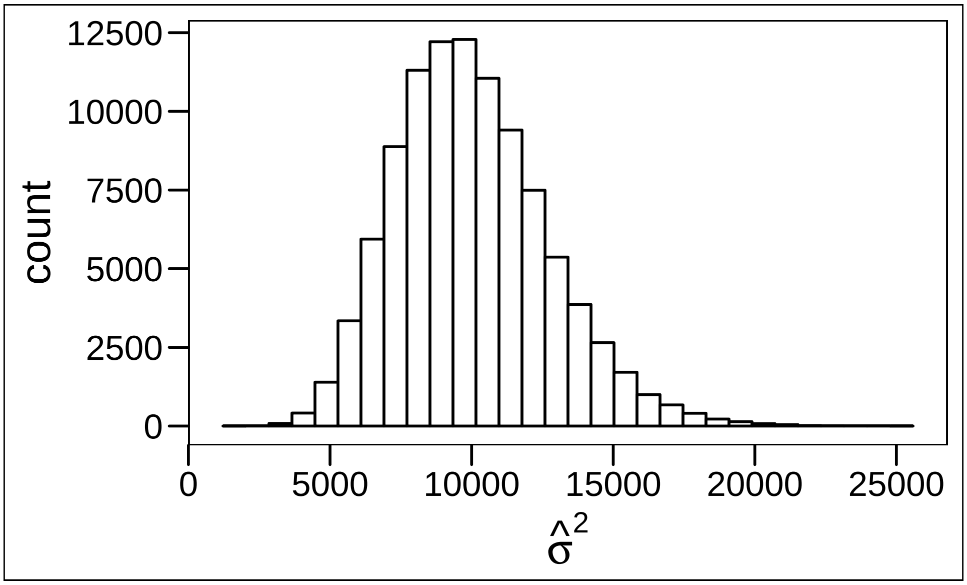
```
ggplot(estimates.df) +
  geom_histogram(aes(x = b1.hat),
                 fill = 'white', colour = 'black') +
  labs(x = expression(hat(beta)[1]))
```

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



```
ggplot(estimates.df) +
  geom_histogram(aes(x = sigma.hat ** 2),
                 fill = 'white', colour = 'black') +
  labs(x = expression(hat(sigma)^2))
```

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



```
# mean of estimates
estimates.df %>%
  dp$mutate(sigma2.hat = sigma.hat ** 2) %>%
  dp$select(-sigma.hat) %>%
  dp$summarise_all(mean)
```

```
# A tibble: 1 x 3
    b0.hat    b1.hat sigma2.hat
     <dbl>     <dbl>      <dbl>
1 10.23124 -2.500078   10002.58
```

```
# variance of estimates
estimates.df %>%
  dp$select(-sigma.hat) %>%
  dp$summarise_all(var)
```

```
# A tibble: 1 x 2
    b0.hat       b1.hat
     <dbl>        <dbl>
1 1398.697 0.0004453688
```

```
# var(beta1.hat | x)
s ** 2 / cross.prod.sum(x)
```

```
[1] 0.0004449388
```

```
# var(beta0.hat | x)
s ** 2 * (1 / n + mean(x) ** 2 / cross.prod.sum(x))
```

```
[1] 1402.299
```

The means of the estimators do not change much, in support of the fact that they are unbiased.

The variance of $\hat{\beta}_0$ is also fairly close to the original value. Looking at the true value, we can see that there is a $\frac{\bar{x}^2}{\text{SXX}}$ term, and SXX $\sim x^2$ and $\bar{x}^2 \sim x^2$, so the change in the xs cancel each other out.

The variance of $\hat{\beta}_1$ is much smaller, by a factor of 10^4. Looking at the true value of the variance, we can see that it's inversely proportional to SXX, so it's inversely proportional to x^2. Since the scale of x changed by a factor of 100, the scale of the variance changed by a factor of 10000.