

Assignment 1

590049bm and

11/3/2021

Assignment 1

At the beginning of the code, set the random seed to 810 using `set.seed()`. Failure to do so will be penalised.

```
set.seed(810)
```

- Simulate 100,000 observations from the DGP

```
x <- rnorm(100000, mean = 6, sd = sqrt(3))
alpha <- 5
e <- rnorm(100000, 0, 1)

y <- alpha + 0.3*x + 0.1*x^2 + e

dataset <- data.frame(x = x, x_sq = x^2, y = y)
```

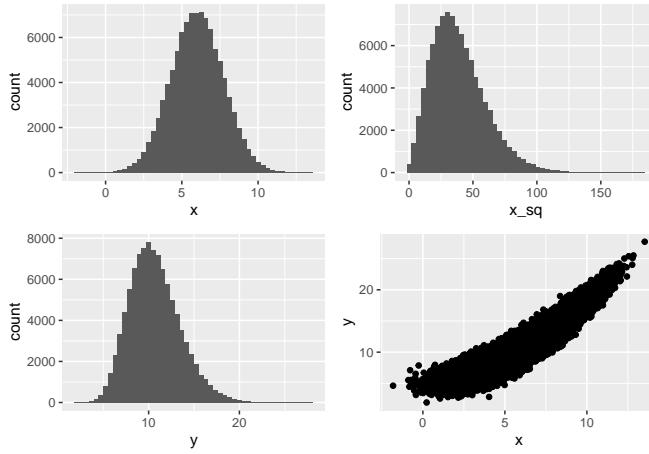
```
p1 <- dataset %>%
  ggplot(aes(x = x)) + geom_histogram(bins = 50)

p2 <- dataset %>%
  ggplot(aes(x = x_sq)) + geom_histogram(bins = 50)

p3 <- dataset %>%
  ggplot(aes(x = y)) + geom_histogram(bins = 50)

p4 <- dataset %>%
  ggplot(aes(x = x, y = y)) + geom_point()

cowplot::plot_grid(p1, p2, p3, p4, nrow = 2, ncol = 2)
```



- b. Break the data into 1000 datasets of 100 observations sequentially (i.e. dataset 1 comprises observations 1-100 from your simulation, dataset 2 comprises observations 101-200 and so on

```
hundred <- dataset %>%
  group_by((row_number()-1) %% (n()/1000)) %>%
  nest %>% pull(data)
```

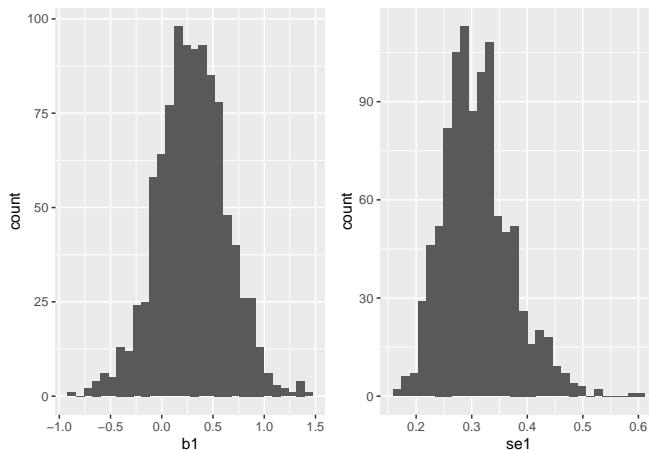
```
beta1 <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x + x_sq) %>%
  .$coefficients %>%
  .[2])

se1 <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x + x_sq) %>%
  summary() %>%
  .$coefficients %>%
  .[2,2])

p1_een <- data.frame(b1 = beta1, se1 = se1) %>%
  ggplot(aes(x = b1)) + geom_histogram()

p2_een <- data.frame(b1 = beta1, se1 = se1) %>%
  ggplot(aes(x = se1)) + geom_histogram()

cowplot::plot_grid(p1_een, p2_een)
```



```

beta_omv <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x) %>%
  .$coefficients %>%
  .[2])

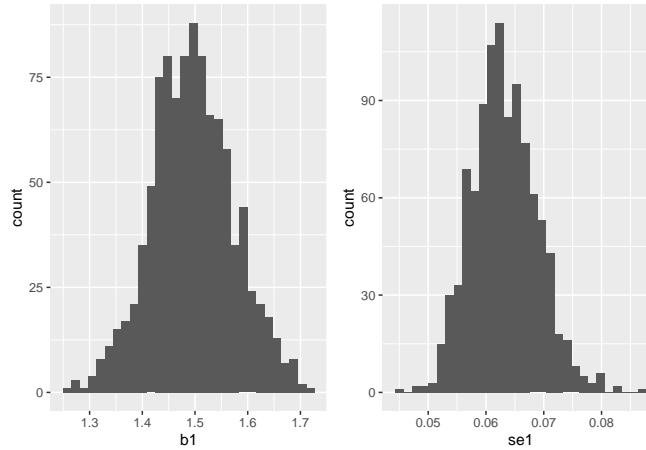
se_omv <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x) %>%
  summary() %>%
  .$coefficients %>%
  .[2, 2])

p1_omv <- data.frame(b1 = beta_omv, se1 = se_omv) %>%
  ggplot(aes(x = b1)) + geom_histogram()

p2_omv <- data.frame(b1 = beta_omv, se1 = se_omv) %>%
  ggplot(aes(x = se1)) + geom_histogram()

cowplot::plot_grid(p1_omv, p2_omv)

```



We see that the coefficients from the fully specified model (without omitted variable biased) are normally distributed around the true coefficient value from the DGP, whereas the coefficient in the wrongly specified model is noisy distributed around a wrong value.

- c. Imagine we had instead generated $X_i = c$ for all x_i and tried to perform our simulation above. This would fail because we would violate one of the necessary conditions for computing the least squares estimator. Which one, and how?

$$\sum_{i=1}^N (x_i - \bar{x})^2 \geq 0$$

is not met. If we have no finite sum of squares, we have $\text{Var}X = 0$, and hence, we cannot estimate the OLS estimator.

- d. Simulate another dataset as in a), but with 1,000,000 observations. Repeat part b), but now using 1000 observations per model. Only fit the equation $Y_i = \alpha + \beta_1 X_i + \beta_2 X_i^2 + \epsilon$ this time. Plot the histograms of the estimate of β_1 and $se(\beta_1)$ and compare them to the estimates from the same model in (b). What happens to the distribution of the coefficient estimates and standard errors when we increase the sample size per model?

```

x2 <- rnorm(1000000, mean = 6, sd = sqrt(3))
alpha2 <- 5
e2 <- rnorm(1000000, 0, 1)

y2 <- alpha2 + 0.3*x2 + 0.1*x2^2 + e2

dataset2 <- data.frame(x = x2, x_sq = x2^2, y = y2)

thousand <- dataset %>%
  group_by((row_number()-1) %% (n()/1000)) %>%
  nest %>% pull(data)

beta_ef <- map_dbl(thousand, ~ lm(data = .x, formula = y ~ x + x_sq) %>%
  .$coefficients %>%
  .[2])

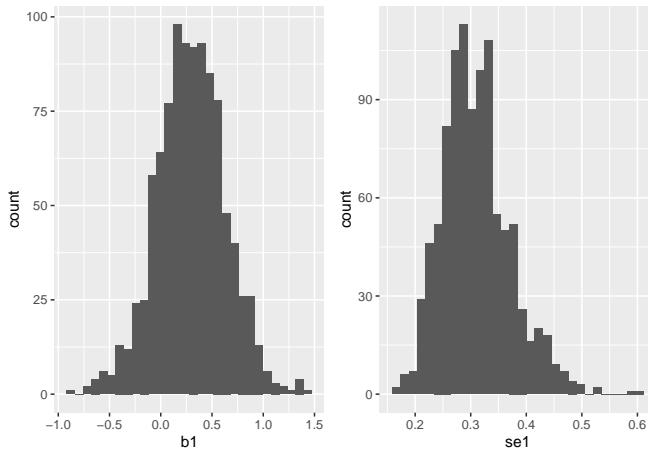
se_ef <- map_dbl(thousand, ~ lm(data = .x, formula = y ~ x + x_sq) %>%
  summary() %>%
  .$coefficients %>%
  .[2,2])

p1_ef <- data.frame(b1 = beta_ef, se1 = se_ef) %>%
  ggplot(aes(x = b1)) + geom_histogram()

p2_ef <- data.frame(b1 = beta_ef, se1 = se_ef) %>%
  ggplot(aes(x = se1)) + geom_histogram()

cowplot::plot_grid(p1_ef, p2_ef)

```



- e. Now, generate a new variable C_i that is correlated with X_i . Do this by creating a vector of observations drawn from $N(1,2)$ and adding them to X_i . Add C_i the dataset from the previous questions (i.e with 100,000 observations total).

```

c <- rnorm(100000, mean = 1, sd = sqrt(2)) + x
dataset <- data.frame(x = x, x_sq = x^2, y = y, c = c)

hundred <- dataset %>%
  group_by((row_number()-1) %% (n()/1000)) %>%
  nest %>% pull(data)

```

```

beta_twee <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x + x_sq + c) %>%
  .$coefficients %>%
  .[2])

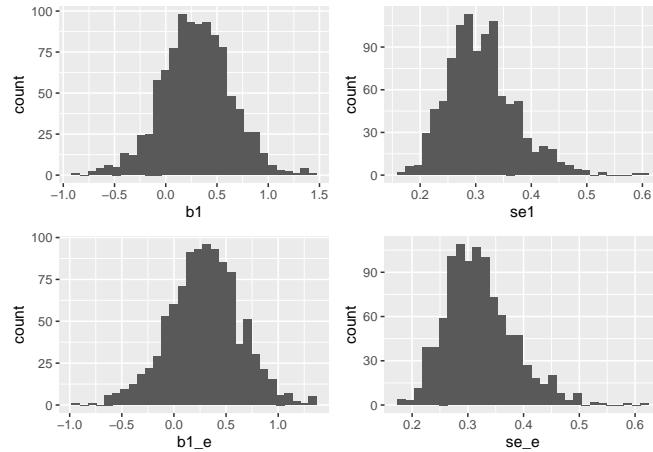
se_twee <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x + x_sq + c) %>%
  summary() %>%
  .$coefficients %>%
  .[2,2])

p1_twee <- data.frame(b1_e = beta_twee,
  se_e = se_twee) %>%
  ggplot(aes(x = b1_e)) + geom_histogram()

p2_twee <- data.frame(b1_e = beta_twee,
  se_e = se_twee) %>%
  ggplot(aes(x = se_e)) + geom_histogram()

cowplot::plot_grid(p1_een, p2_een,
  p1_twee, p2_twee,
  nrow = 2, ncol = 2)

```



Re-run the regressions as $Y_i = \alpha + \beta_1 X_i + \beta_2 X_i^2 + \beta_3 C_i + \epsilon_i$ and plot the distribution of these coefficient estimates and standard errors for β_1 next to the coefficient estimates from the second part. What happens to the coefficient estimates, and why?

The coefficients are not impacted, because there is no omitted variable bias. There is no relationship between Y and C , hence, the OLS coefficient for β_1 is still unbiased. We do pay a small penalty in terms of efficiency for adding an unnecessary variable, but with $N = 1000$, this is negligible.