

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)

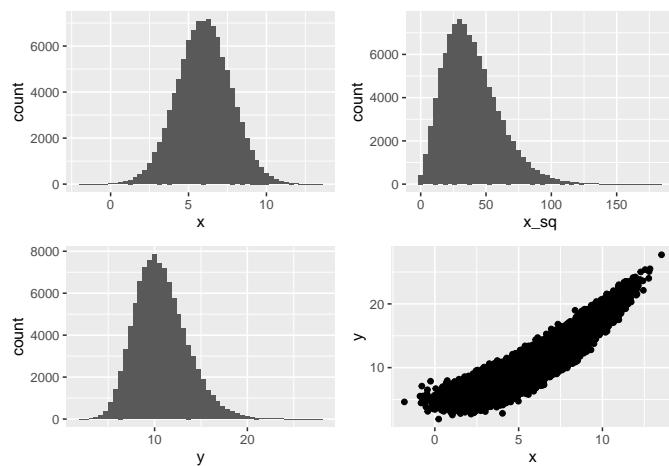
p1 <- dataset %>%
  ggplot(aes(x = x)) + geom_histogram(bins = 50)

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

p3 <- dataset %>%
  ggplot(aes(y = 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)
```



- 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)

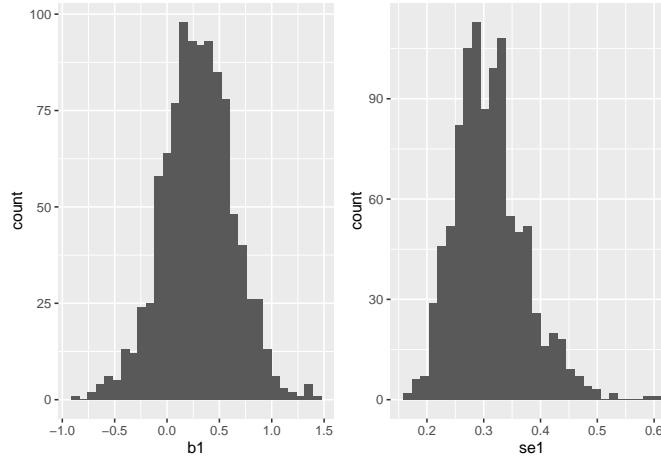
beta1 <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x + x_sq) %>%
  .$coefficients %>%
  .[2])

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)
```



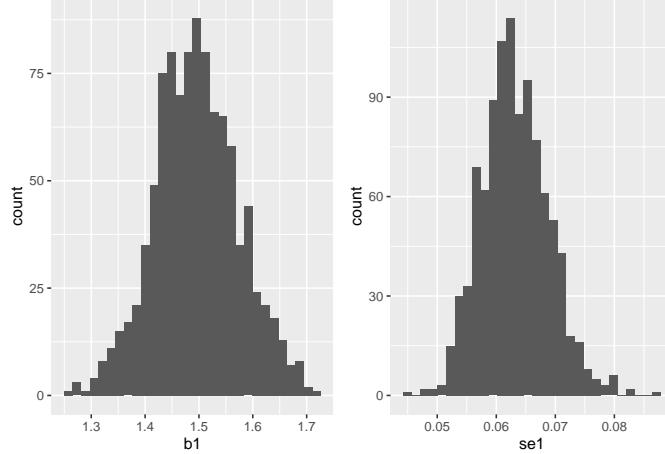
```
beta_omv <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x) %>%
  .$coefficients %>%
  .[2])

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?

We would violate assumption 1:

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

as the sum of the square of the difference between X_i and \bar{X} would be zero if $X_i = c$. $X_i = c$ would also mean that X is not full rank and therefore not invertible. Hence, we could not calculate the least squares 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 $s(\beta_1)$ 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 <- dataset2 %>%
  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])

```

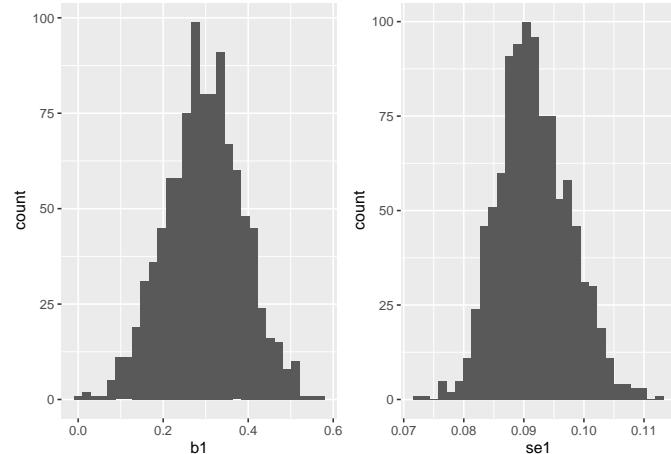
```

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)

```



Comparing this histogram to the histogram from b), we can see that the mean stays approximately the same. However, the spread and therefore the variance of the estimate from d) is smaller than from the estimate of b). This makes sense, as with an increasing number of observations the variance of the estimate decreases. The same can be observed with the standard error of the estimate. The mean stays approximately the same but the variance of the standard error decreases.

- 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])

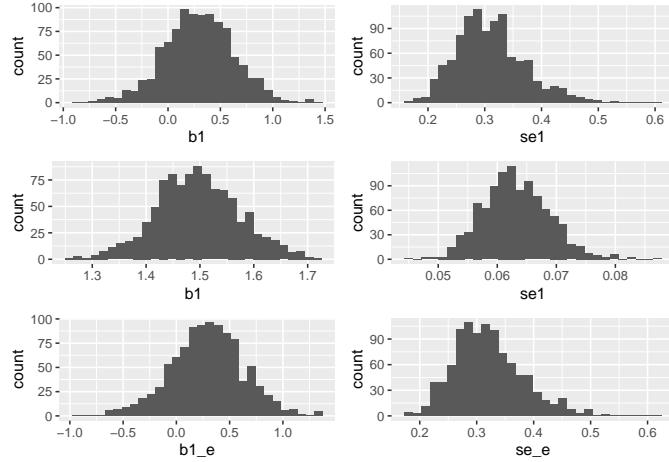
se_twee <- map_dbl(hundred, ~ lm(data = .x, formula = y ~ x + x_sq + c) %>%
  summary() %>%
  .$coefficients %>%
  .[2, 2])

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_omv, p2_omv,
                    p1_twee, p2_twee,
                    nrow = 3, 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?

While the first regression has a different mean, the estimators of regression 2 and 3 have approximately the same mean and distribution. Therefore, the inclusion of c does not add any additional information to our estimation and hence is a redundant variable while including the square of x changes the estimation of beta 1, indicating that it is a non-linear relationship! We also can observe this with the standard errors of the estimate. The standard error of the second and third estimator has approximately the same mean and distribution, indicating that c is a redundant variable.

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.

Question 2

You are a labor economist trying to estimate the gender wage gap within occupations for women with children - that is, the effect of gender on wages given occupational choice. You have access to a dataset containing a set of wages, W_i , a gender dummy D_i , a set of occupational dummies O_{ij} and hours spent on childcare C_i for a sample of men and women with children. Assume that there is a positive covariance between each of your regressors, some covariance between each of the regressors and wages, and that gender at least partially determines occupational choice and hours spent on childcare.

$$W_i = \beta D_i + \gamma O_i + \delta C_i + \epsilon_i$$

- a. Derive the expected value of the least-squares estimator for the coefficient on the gender dummy without controlling for either occupation or hours spent on childcare

We have that $b = (D^T D)^{-1} D^T W$ and

$$\mathbb{E}[b] = \mathbb{E}[(D^T D)^{-1} D^T y] = \mathbb{E}[(D^T D)^{-1} D^T (\beta D + \gamma O + \delta C + \epsilon)]$$

which simplifies to:

$$\beta + \gamma \cdot \mathbb{E}[(D^T D)^{-1} D^T O] + \delta \cdot \mathbb{E}[(D^T D)^{-1} D^T C]$$

The latest expression contains the covariances of D with O and C respectively in the numerator, and the variance of D in the denominator.

The variance of this estimator is:

$$\text{Var}(b) =$$

- b. A friend who has taken an undergraduate econometrics course suggests including both the occupational dummies and hours spent on childcare as control variables, to remove omitted variable bias. Now imagine we control for both of these in our least-squares regression. Derive the coefficient estimate and the variance of the estimator.

Question 3

- a. Derive the log-likelihood function for the service life of n machines.

$$L(\alpha, x) = \prod_{i=1}^N f(x_i) = \alpha^n \exp(\alpha \sum_{i=1}^N x_i)$$

Taking the logs on both sides and applying the corresponding rules gives:

$$\log L(\alpha, x) = \sum_{i=1}^N f(x_i) = n \ln(\alpha) - \alpha \sum_{i=1}^N x_i$$

- b. Derive $\hat{\alpha}$ the maximum likelihood estimator for α . Please check the second-order condition to ensure your result indeed maximizes the log-likelihood function.

Taking the first derivative and setting it equal to zero gives:

$$\alpha_{MLE} = \frac{n}{\sum_{i=1}^N x_i}$$

The second derivative of the log-likelihood function is:

$$\frac{\partial \log L}{\partial \alpha} = -\frac{n}{\alpha^2}$$

which is always negative (given $\alpha \neq 0$) hence, a maximum is attained.

- c. Derive the log-likelihood function (Weibull)

$$L(\beta, \gamma, x) = \prod_{i=1}^N f(x_i) = \prod_{i=1}^N \frac{\beta}{\lambda} \left(\frac{x_i}{\lambda} \right)^{\beta-1} \exp\left\{ -\left(\frac{x_i}{\lambda} \right)^\beta \right\}$$

The log-likelihood is then obtained by taking the log on both sides:

$$n \log \beta - n \log \lambda + (\beta - 1) \sum_{i=1}^N \log x_i - (\beta - 1)n \log \lambda - \sum_{i=1}^N \left(\frac{x_i}{\lambda} \right)^\beta$$

Question 4

- a. Estimate the two following models. Interpret the results of the two models and also compare the results.

```
dataas1 <- readr::read_csv("./DataAS1.csv")
```

	Model 1	Model 2
(Intercept)	3258.239*** (56.012)	3442.130*** (81.213)
age	6.401*** (2.009)	-2.273 (2.308)
smoker	-237.315*** (27.431)	-185.032*** (27.936)
alcohol	-33.934 (97.407)	-17.815 (96.208)
drinks	-12.546 (19.427)	-7.230 (19.187)
unmarried		-244.106*** (28.539)
educ		7.276 (5.593)
N	3000	3000
Adj. R2	0.03	0.06

* p < 0.1, ** p < 0.05, *** p < 0.01

```
model1 <- lm(data = dataas1, birthweight ~ age + smoker + alcohol + drinks)
model2 <- update(model1, . ~ . + unmarried + educ)

modelsummary(list(model1, model2),
            stars = c("*" = 0.10, "**" = 0.05, "***" = 0.01),
            gof_map = gm
            )
```

- b. Test the null hypothesis $H_0: \beta_{unmarried} = \beta_{educ} = 0$. Please include the name of the test, the value of the test statistics, the p-value and the conclusion in your answer.

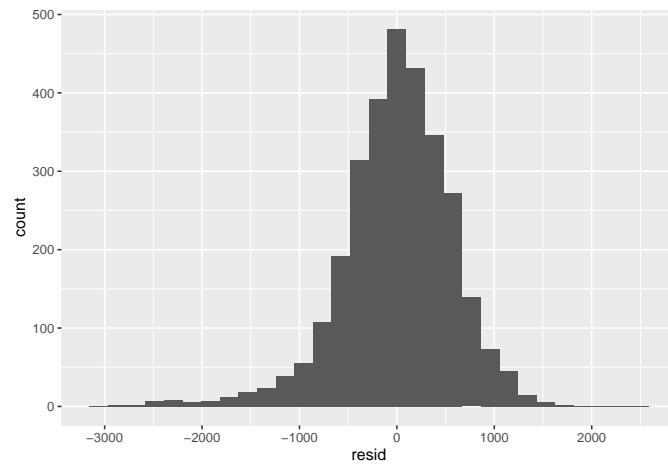
The test is called the F-test. We can execute it using the `car` package:

```
linearHypothesis(model2, c("unmarried=0", "educ=0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## unmarried = 0
## educ = 0
##
## Model 1: restricted model
## Model 2: birthweight ~ age + smoker + alcohol + drinks + unmarried + educ
##
##   Res.Df      RSS Df Sum of Sq    F    Pr(>F)
## 1   2995 1017772843
## 2   2993  991102509  2   26670334 40.27 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- c. Are the residuals of model 2 normally distributed? Provide a histogram for the residuals and also perform a formal test.

```
data.frame(resid = model2$residuals) %>%
  ggplot(aes(x = resid)) + geom_histogram()
```



- d. There could be nonlinear relations between the dependent variable and the independent variables.
Perform a test to investigate the possible non-linear relation. What is your conclusion?

Use the OLS method to estimate a new model (i.e. model 3) whose dependent variable is the log of birthweight and the independent variables are the same as those in model 2.

```
model3 <- lm(data = dataas1, log(birthweight) ~ age + smoker + alcohol +
drinks + unmarried + educ)
```

Then use the maximum likelihood method to estimate model 3 again. Provide a table to show the coefficients of all regressors for two methods. Explain why the coefficients of regressors for the two methods are similar or why they are different.