Exercises: Week 1

Econometrics Prof. Conlon

Ulrich Atz

2021-02-02

```
library(tidyverse)
library(broom)
```

1. Let's start by writing a function that generates fake data

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + e_i$$

```
# Reproduce
set.seed(202102)

# Set some default values
n_obs <- 1e3
beta <- 1:3
x1_var <- 0.5
x2_var <- 1.5
e_var <- 2

# Assume centered means for simplicity
generate_sample <- function(n_obs, beta, x1_var, x2_var, e_var, e_type){
  x1 <- rnorm(n_obs, sd = sqrt(x1_var))
  x2 <- rnorm(n_obs, sd = sqrt(x2_var))
  if (e_type == "normal") {e  <- rnorm(n_obs,
                                       sd = sqrt(e_var))}
  if (e_type == "uniform") {e  <- runif(n_obs,
                                        min = -sqrt(e_var*12)/2,
                                        max =  sqrt(e_var*12)/2)}
    y <- beta[1] + beta[2]*x1 + beta[3]*x2 + e
  sample <- tibble(y, x1, x2)
  return(sample)
}
```

I derive the correct uniform lower and upper bounds from the variance formula: [1]

$$Var[X_{uniform}] = E[X^2] - E[X]^2 = \frac{(b-a)^2}{12}$$

```
sample <- generate_sample(n_obs, beta, x1_var, x2_var, e_var, e_type = "normal") # test
```

The function should take the following arguments:

- n_obs: number of observations in the sample

[1] Uniform variance via https://www.statlect.com/probability-distributions/uniform-distribution

- beta : a vector of coefficients
- x1_var: a variance/scale parameter for x1
- x2_var: a variance/scale parameter for x2
- e_var: a variance/scale parameter for e_i
- e_type: a distribution type for the residual (maybe uniform or normal?)

2. Now let's write a function that takes the same arguments and also takes as an argument the number of simulated datasets (say 1000?)

```
sim_n_samples <- function(reps = 1e3, e_type){
 samples <- replicate(reps,
              generate_sample(n_obs, beta, x1_var, x2_var, e_var, e_type),
              simplify = FALSE)
 return(samples)
}
```

```
hundred_samples <- sim_n_samples(100, e_type = "normal")
hundred_samples_unif <- sim_n_samples(100, e_type = "uniform")

thousand_samples <- sim_n_samples(e_type = "normal")
thousand_samples_unif  <- sim_n_samples(e_type = "uniform")
```

3. Let's write a function that takes in a single dataset and runs a regression and calculates the output (let's keep the estimates of $\widehat{\beta}$ and it's standard error, R^2, MSE, and let's evaluate the a t-statistic for the hypothesis that $H_0 : \beta = a$ for some choice of a). It will be helpful to return everything in a data frame.

```
reg_out <- function(sample, a = rep(0,3)) {
  est <- lm(y ~ x1 + x2, data = sample)
  est_out <- tidy(est) %>%
    mutate(custom_t = (est$coefficients - a) / sqrt(diag(vcov(est))),
           r2 = summary(est)$r.squared,
           mse = mean(est$residuals^2))
  return(est_out)
  # split(est_out, est_out$term) // for next time
}
```

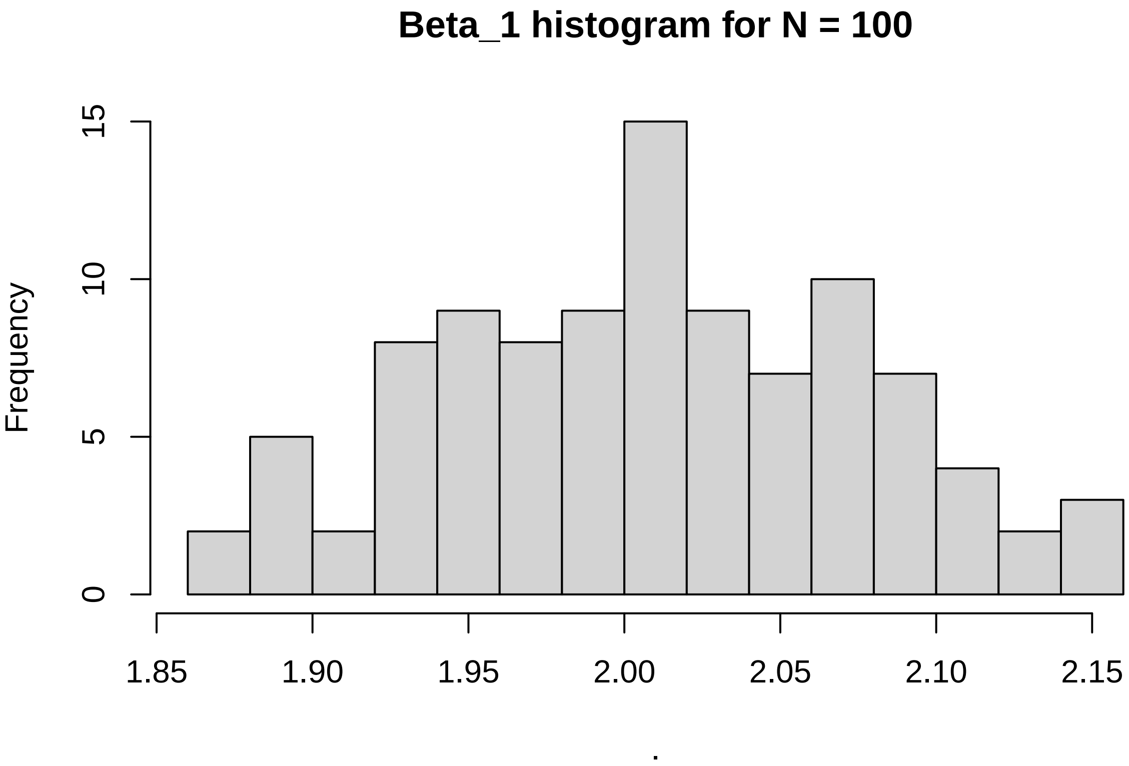
```
reg_out(thousand_samples[[1]], 0:2) # test
```

```
## # A tibble: 3 x 8
##   term        estimate std.error statistic   p.value custom_t    r2   mse
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl> <dbl> <dbl>
## 1 (Intercept)     1.01    0.0458      22.0 1.48e- 87     22.0 0.884  2.09
## 2 x1              2.07    0.0653      31.8 1.35e-153     16.5 0.884  2.09
## 3 x2              3.07    0.0379      81.0 0.            28.2 0.884  2.09
```

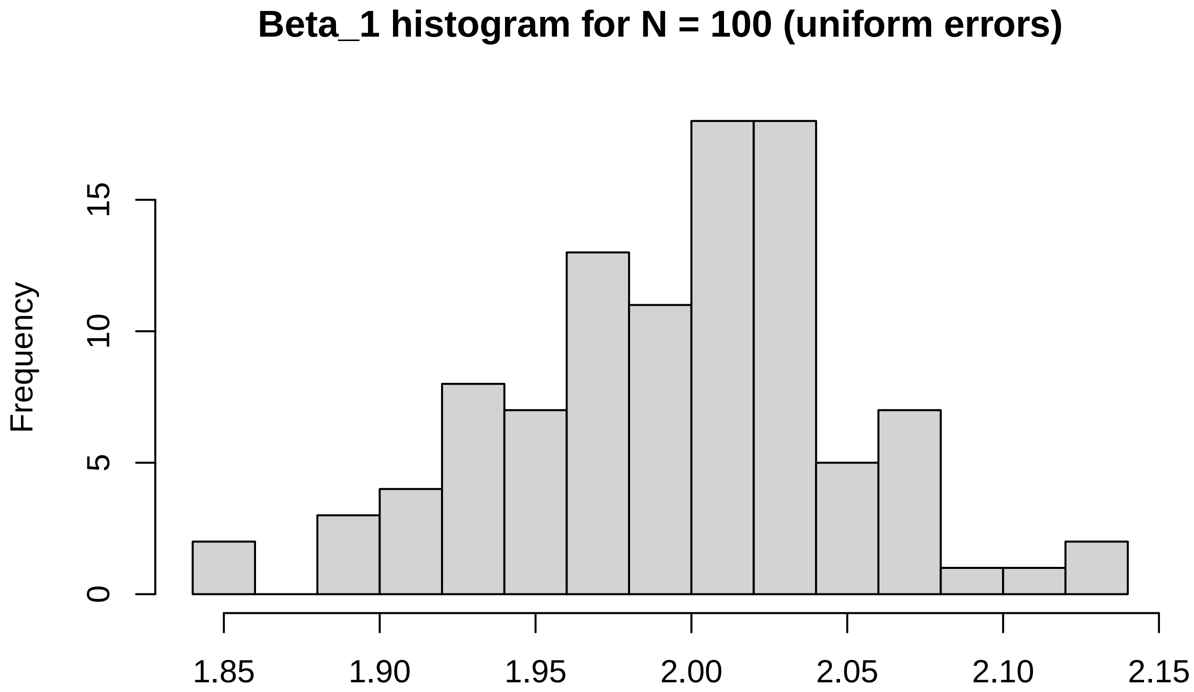
4. Plot the distribution of $\widehat{\beta}_1$ when the sample size is $n = 100$ and see how it compares when e_i is uniform vs. when it is normal across the 1000 samples.

```
get_beta <- function(x){
  tmp <- reg_out(x) %>%
    pull(estimate) %>%
    nth(2) # beta1
}

sapply(hundred_samples, get_beta) %>%
  hist(main = "Beta_1 histogram for N = 100", breaks = 20)
```

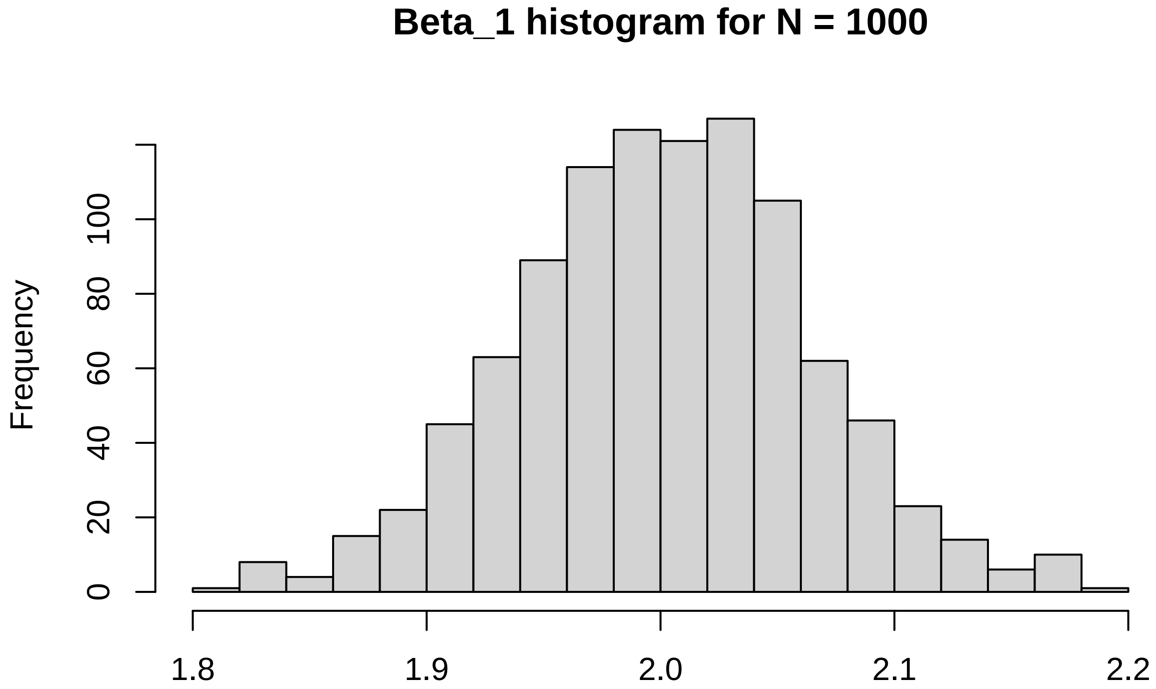


```
sapply(hundred_samples_unif, get_beta) %>%
  hist(main = "Beta_1 histogram for N = 100 (uniform errors)", breaks = 20)
```



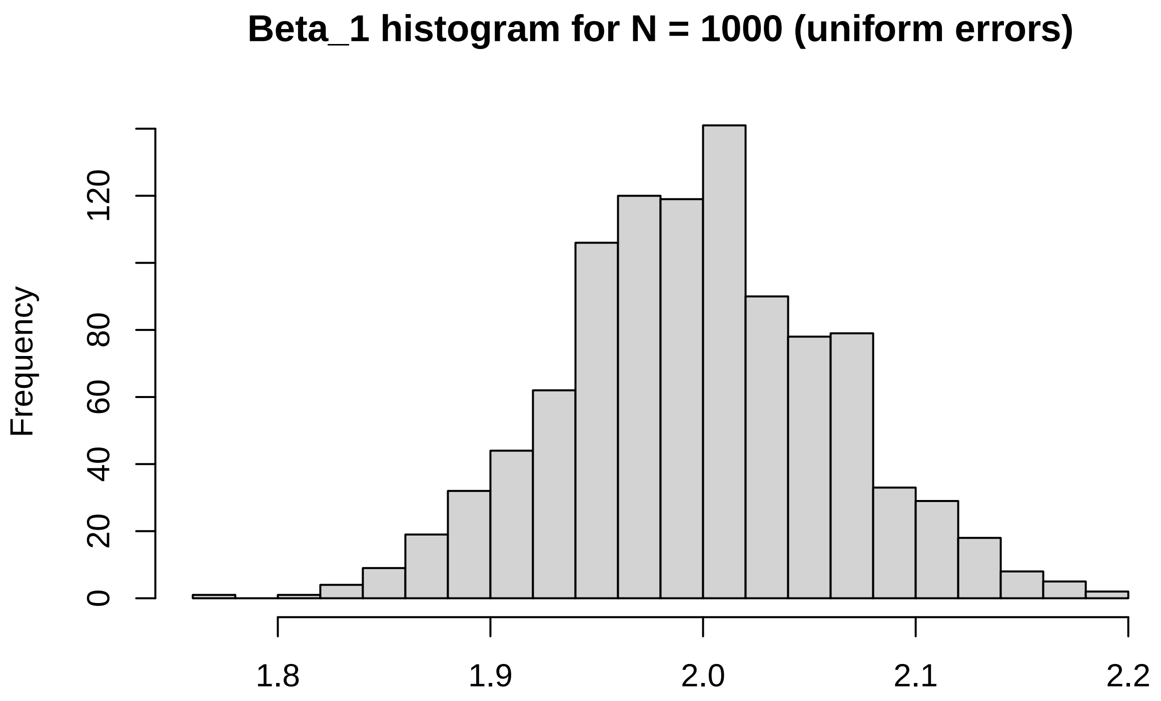
.

```
sapply(thousand_samples, get_beta) %>%
  hist(main = "Beta_1 histogram for N = 1000", breaks = 20)
```



.

```
sapply(thousand_samples_unif, get_beta) %>%
  hist(main = "Beta_1 histogram for N = 1000 (uniform errors)", breaks = 20)
```



.

5. Make a table that shows how $\widehat{\beta}_1$ and computes the mean, the standard deviation, the 5th and 95th percentile, and compare that to the asymptotic standard error under different assumptions about the error distribution.

The asymptotic variance of $\widehat{\beta}_1$ is equal to $(\sigma^2_{error}/n)Q^{-1}$ where Q here is simply $Q = var(x_1)$ because x_1 and x_2 are uncorrelated and have mean zero.

```
# n = 100
# Empirical
empirical_norm <- sapply(hundred_samples, get_beta) %>%
  tibble(beta_1 = .) %>%
  summarize(
    parameter = "empirical (normal)",
    mean = mean(beta_1),
    sd = sd(beta_1),
    q05 = quantile(beta_1, probs = 0.05),
    q95 = quantile(beta_1, probs = 0.95)
  )

empirical_unif <- sapply(hundred_samples_unif, get_beta) %>%
  tibble(beta_1 = .) %>%
  summarize(
    parameter = "empirical (uniform)",
    mean = mean(beta_1),
    sd = sd(beta_1),
    q05 = quantile(beta_1, probs = 0.05),
    q95 = quantile(beta_1, probs = 0.95)
  )
```

```
# Theoretical
theoretical <- tibble(
  parameter = "theoretical",
  mean = beta[2],
  sd = sqrt(e_var / 100 / x1_var),
  q05 = qnorm(0.05, mean = mean, sd = sd),
  q95 = qnorm(0.95, mean = mean, sd = sd)
)

bind_rows(empirical_norm, empirical_unif, theoretical) %>%
  kableExtra::kbl(booktabs = T)
```

parameter	mean	sd	q05	q95
empirical (normal)	2.007981	0.0676545	1.886815	2.109187
empirical (uniform)	1.995776	0.0543084	1.905040	2.068967
theoretical	2.000000	0.2000000	1.671029	2.328971

```
# n = 1000
# Empirical
empirical_norm <- sapply(thousand_samples, get_beta) %>%
  tibble(beta_1 = .) %>%
  summarize(
    parameter = "empirical (normal)",
    mean = mean(beta_1),
    sd = sd(beta_1),
    q05 = quantile(beta_1, probs = 0.05),
    q95 = quantile(beta_1, probs = 0.95)
  )

empirical_unif <- sapply(thousand_samples_unif, get_beta) %>%
  tibble(beta_1 = .) %>%
  summarize(
    parameter = "empirical (uniform)",
    mean = mean(beta_1),
    sd = sd(beta_1),
    q05 = quantile(beta_1, probs = 0.05),
    q95 = quantile(beta_1, probs = 0.95)
  )

# Theoretical
theoretical <- tibble(
  parameter = "theoretical",
  mean = beta[2],
  sd = sqrt(e_var / 1000 / x1_var),
  q05 = qnorm(0.05, mean = mean, sd = sd),
  q95 = qnorm(0.95, mean = mean, sd = sd)
)

bind_rows(empirical_norm, empirical_unif, theoretical) %>%
  kableExtra::kbl(booktabs = T)
```

parameter	mean	sd	q05	q95
empirical (normal)	2.002014	0.0624070	1.899936	2.103996
empirical (uniform)	1.997796	0.0641431	1.892557	2.105205
theoretical	2.000000	0.0632456	1.895970	2.104030

6. How does changing the variance of x_1 and x_2 and e_i affect the results? Can you provide a relative precise quantification?

The standard error of $\widehat{\beta}_1$ is given as above in terms of $\sqrt{(\sigma^2/n)/var(x_1))} = \sqrt{\sigma^2}\sqrt{1/n}\sqrt{1/var(x_1)}$. If x_2 were correlated with x_1 it would also enter this consideration, but here changes in x_2 do not affect $\widehat{\beta}_1$.

Thus, the standard error of $\widehat{\beta}_1$ is inversely related to changes in the variance of x_1 and directly related to changes in the variance of e. Both do not change linearly, but by a function of the power of 2.