Exercises: Week 1

Prof. Conlon

Due: 2/1/21

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

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

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

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

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

```
sample <- generate_sample(n_obs, beta, x1_var, x2_var, e_var, e_type)
```

The function should take the following arguments:

- n__obs: number of observations in the sample
- 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?)

```
# Assume centered means for simplicity
gen_n_samples <- function(reps = 1e3, ...){
 samples <- replicate(reps,
              generate_sample(n_obs, beta, x1_var, x2_var, e_var, e_type),
```

```r
               simplify = FALSE)
 return(samples)
}
```

```r
hundred_samples <- gen_n_samples(100)
hundred_samples_unif <- gen_n_samples(100, e_type = "uniform")

thousand_samples <- gen_n_samples()
# thousand_samples_unif  <- gen_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.

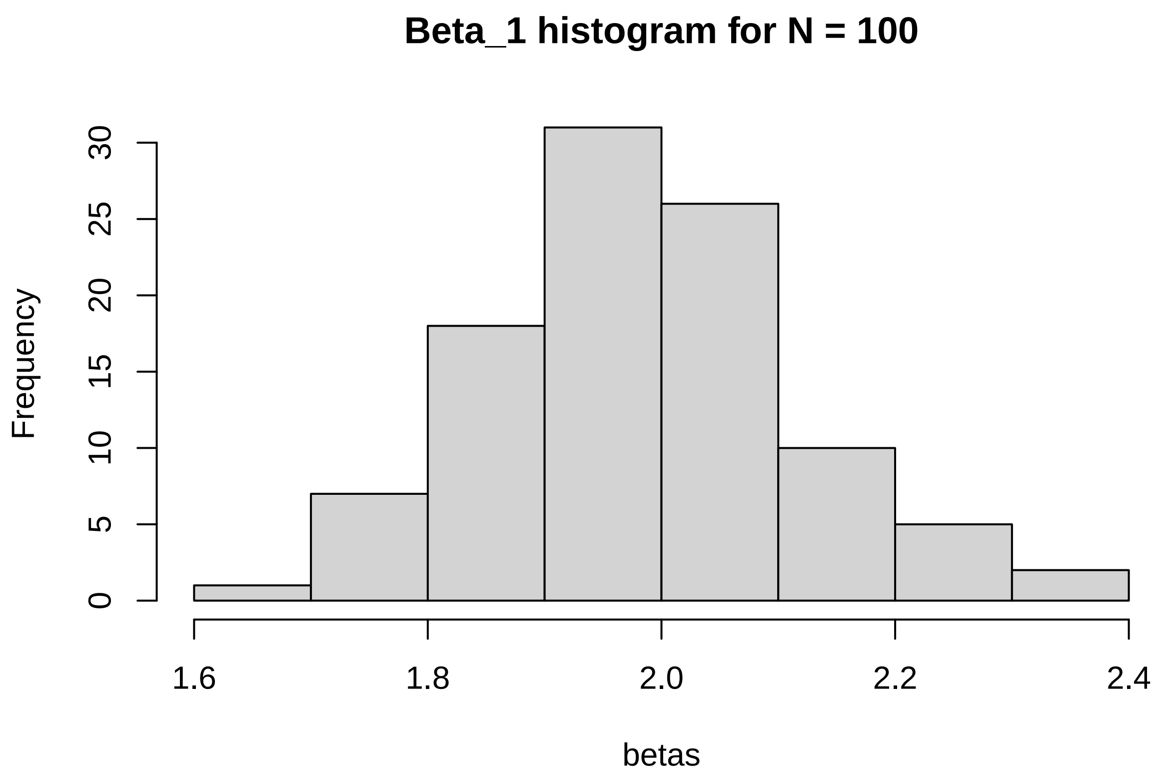
```r
reg_out <- function(sample, a = rep(0,3)) {
  est <- lm(y ~ x1 + x2, data = sample)
  est_out <- tidy(est) %>%
    mutate(custom_t = (est$coefficients - a) / sqrt(diag(vcov(est))),
           r2 = summary(est)$r.squared,
           mse = mean(est$residuals^2))
  return(est_out)
}
```

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

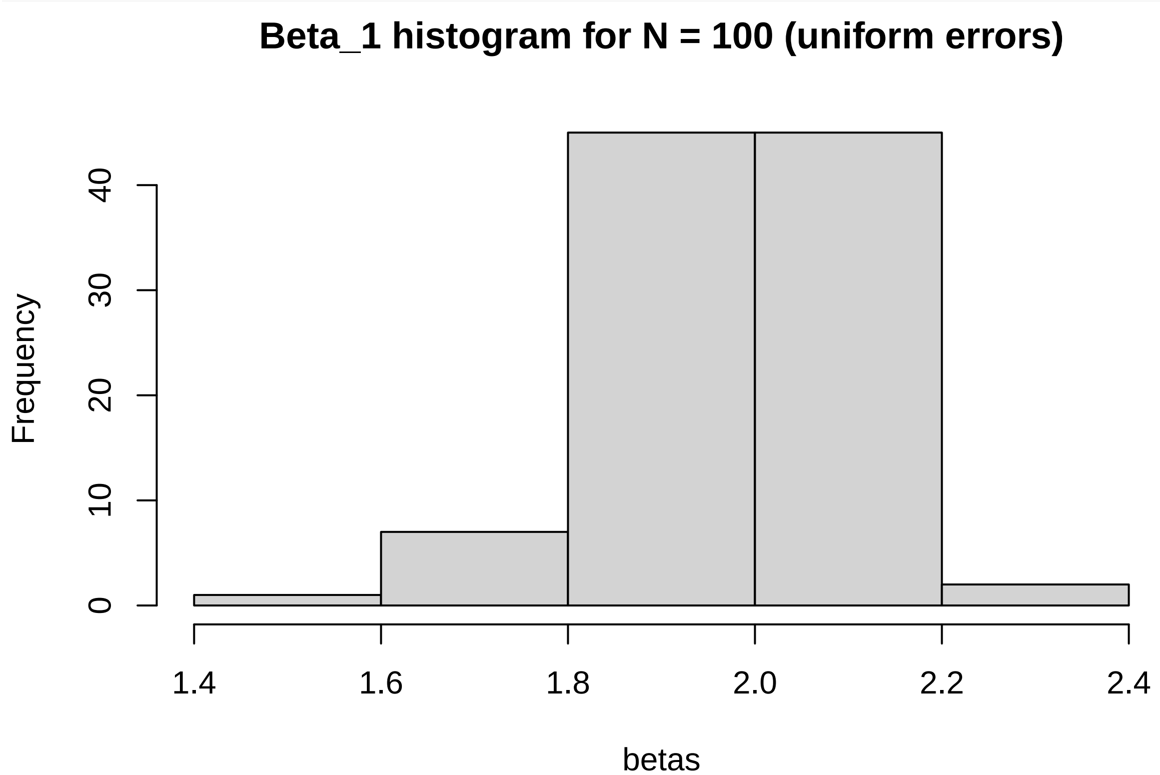
```
## # A tibble: 3 x 8
##   term        estimate std.error statistic  p.value custom_t    r2   mse
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <dbl>
## 1 (Intercept)     1.08    0.0605      17.8 4.88e-62    17.8  0.845  3.65
## 2 x1              1.84    0.122       15.0 3.09e-46     6.85 0.845  3.65
## 3 x2              2.95    0.0409      72.1 0.          23.2  0.845  3.65
```

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.

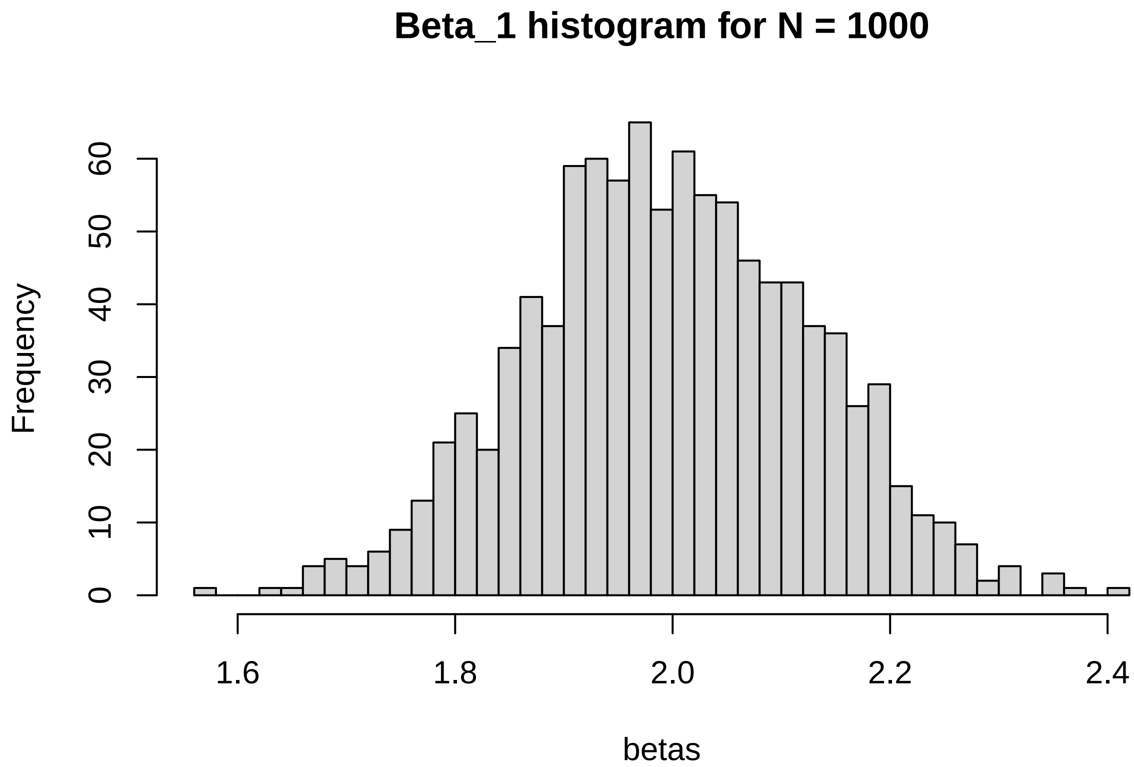
```r
get_beta <- function(x){
  tmp <- reg_out(x) %>%
    pull(estimate) %>%
    nth(2) # beta1
}

betas <- sapply(hundred_samples, get_beta)
hist(betas, main = "Beta_1 histogram for N = 100", breaks = 5)
```




```
betas <- sapply(hundred_samples_unif, get_beta)
hist(betas, main = "Beta_1 histogram for N = 100 (uniform errors)", breaks = 5)
```




```
betas <- sapply(thousand_samples, get_beta)
hist(betas, main = "Beta_1 histogram for N = 1000", breaks = 50)
```



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.

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?