Lecture 3: Designing simulations

Last time

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

How would you study the importance of the normality assumption?

Simulating data

To start, simulate data for which the normality assumption holds:

```
n <- 100 # sample size
beta0 <- 0.5 # intercept
beta1 <- 1 # slope

x <- runif(n, min=0, max=1)
noise <- rnorm(n, mean=0, sd=1)
y <- beta0 + beta1*x + noise
```

- `runif(n, min=0, ,max=1)` samples X_i uniformly between 0 and 1
- `rnorm(n, mean=0, sd=1)` samples $\varepsilon_i \sim N(0, 1)$

Fit a model

```
n <- 100 # sample size
beta0 <- 0.5 # intercept
beta1 <- 1 # slope

x <- runif(n, min=0, max=1)
noise <- rnorm(n, mean=0, sd=1)
y <- beta0 + beta1*x + noise

lm_mod <- lm(y ~ x)
lm_mod
```

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
     0.2836       1.4302
```

Calculate confidence interval

```
lm_mod <- lm(y ~ x)

ci <- confint(lm_mod, "x", level = 0.95)
ci
```

```
      2.5 %   97.5 %
x 0.6883911 2.172003
```

- **Question:** How can we check whether the confidence interval contains the true β_1 ?

Calculate confidence interval

```r
lm_mod <- lm(y ~ x)

ci <- confint(lm_mod, "x", level = 0.95)
ci
```

```
      2.5 %   97.5 %
x 0.6883911 2.172003
```

- **Question:** How can we check whether the confidence interval contains the true β_1 ?

```r
ci[1] < 1 & ci[2] > 1
```

```
[1] TRUE
```

Repeat!

```
nsim <- 1000
n <- 100 # sample size
beta0 <- 0.5 # intercept
beta1 <- 1 # slope
results <- rep(NA, nsim)

for(i in 1:nsim){
  x <- runif(n, min=0, max=1)
  noise <- rnorm(n, mean=0, sd=1)
  y <- beta0 + beta1*x + noise

  lm_mod <- lm(y ~ x)
  ci <- confint(lm_mod, "x", level = 0.95)

  results[i] <- ci[1] < 1 & ci[2] > 1
}
mean(results)
```

- What fraction of the time should the confidence interval contain β_1 ?

Repeat!

```r
nsim <- 1000
n <- 100 # sample size
beta0 <- 0.5 # intercept
beta1 <- 1 # slope
results <- rep(NA, nsim)

for(i in 1:nsim){
  x <- runif(n, min=0, max=1)
  noise <- rnorm(n, mean=0, sd=1)
  y <- beta0 + beta1*x + noise

  lm_mod <- lm(y ~ x)
  ci <- confint(lm_mod, "x", level = 0.95)

  results[i] <- ci[1] < 1 & ci[2] > 1
}
mean(results)
```

```
[1] 0.952
```

- What should we do next?

Class activity

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

That is, how important is the assumption that $\varepsilon_i \sim N(0, \sigma^2)$?

Continue simulation from last time, but experiment with different values of n and different distributions for the noise term.

https://sta279-f23.github.io/class_activities/ca_lecture_3.html

Class activity

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

How does confidence interval coverage change when you change the distribution of ε_i?

Class activity

```r
nsim <- 1000
n <- 100 # sample size
beta0 <- 0.5 # intercept
beta1 <- 1 # slope
results <- rep(NA, nsim)

for(i in 1:nsim){
  x <- runif(n, min=0, max=1)
  noise <- rchisq(n, 1)
  y <- beta0 + beta1*x + noise

  lm_mod <- lm(y ~ x)
  ci <- confint(lm_mod, "x", level = 0.95)

  results[i] <- ci[1] < 1 & ci[2] > 1
}
mean(results)
```

```
[1] 0.963
```

ADEMP: A useful framework for simulation studies

- **Aims:** Why are we doing the study?
- **Data generation:** How are the data simulated?
- **Estimand/target:** What are we estimating for each simulated dataset?
- **Methods:** What methods are we using for model fitting, estimation, etc?
- **Performance measures:** How do we measure performance of our chosen methods?

ADEMP

For the normal errors simulation study:

- **Aims:**
- **Data generation:**
- **Estimand/target:**
- **Methods:**
- **Performance measures:**