Lecture 8: Putting it all together

Question

Consider the simple linear regression model

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

Assumptions:

- $\varepsilon_i \sim N(0, \sigma^2)$ (*normality*, *constant variance*, and *zero mean*)
- Shape (linearity)
- Independence
- Randomness

Question

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

Assumptions:

- $\varepsilon_i \sim N(0, \sigma^2)$ (*normality*, *constant variance*, and *zero mean*)
- Shape (linearity)
- Independence
- Randomness

Question: How do we assess importance of the normality assumption?

Simulation study plan

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

Implementation

```
assess_coverage <- function(n, nsim, beta0, beta1, noise_dist){
  results <- rep(NA, nsim)

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

    lm_mod <- lm(y ~ x)
    ci <- confint(lm_mod, "x", level = 0.95)
    results[i] <- ci[1] < beta1 & ci[2] > beta1
  }
  return(mean(results))
}
```

Implementation

Iterating over different distributions

```r
set.seed(45)

noise_dists <- list(rnorm, rexp,
                    function(m) {return(rchisq(m, df=1))})
ci_coverage <- rep(NA, length(noise_dists))

for(i in 1:length(noise_dists)){
  ci_coverage[i] <- assess_coverage(n = 100, nsim = 1000,
                                    beta0 = 0.5, beta1 = 1,
                                    noise_dist = noise_dists[[i]])
}
ci_coverage
```

```
[1] 0.949 0.960 0.946
```

R tools and topics (so far)

- Vectors and indexing; creating vectors
- Random sampling (from vectors and from distributions)
- Iteration (`for` loops and `while` loops)
- Functions; function defaults, anonymous functions, function scope
- Lists

What’s next

- Introduction to Python
- File management, version control, and GitHub
- Advanced data wrangling

Class activity

https://sta279-f23.github.io/class_activities/ca_lecture_8.html