Simulations, Part 2

Behzad Kianian
April 22, 2019

Overview

Simulations are computer experiments where the truth is known by the researcher, and statistical methods are evaluated in comparison to this truth. Some examples of when you would use simulation:

- You develop a new statistical method, and you want to assess whether it has good statistical properties (bias, coverage, etc.)
- You want to know how a method performs when certain assumptions are violated (e.g. non-normality). Is it robust?

`R` makes it easy to simulate data.

Some of this material is adapted from **Charles DiMaggio**

Simple Simulation

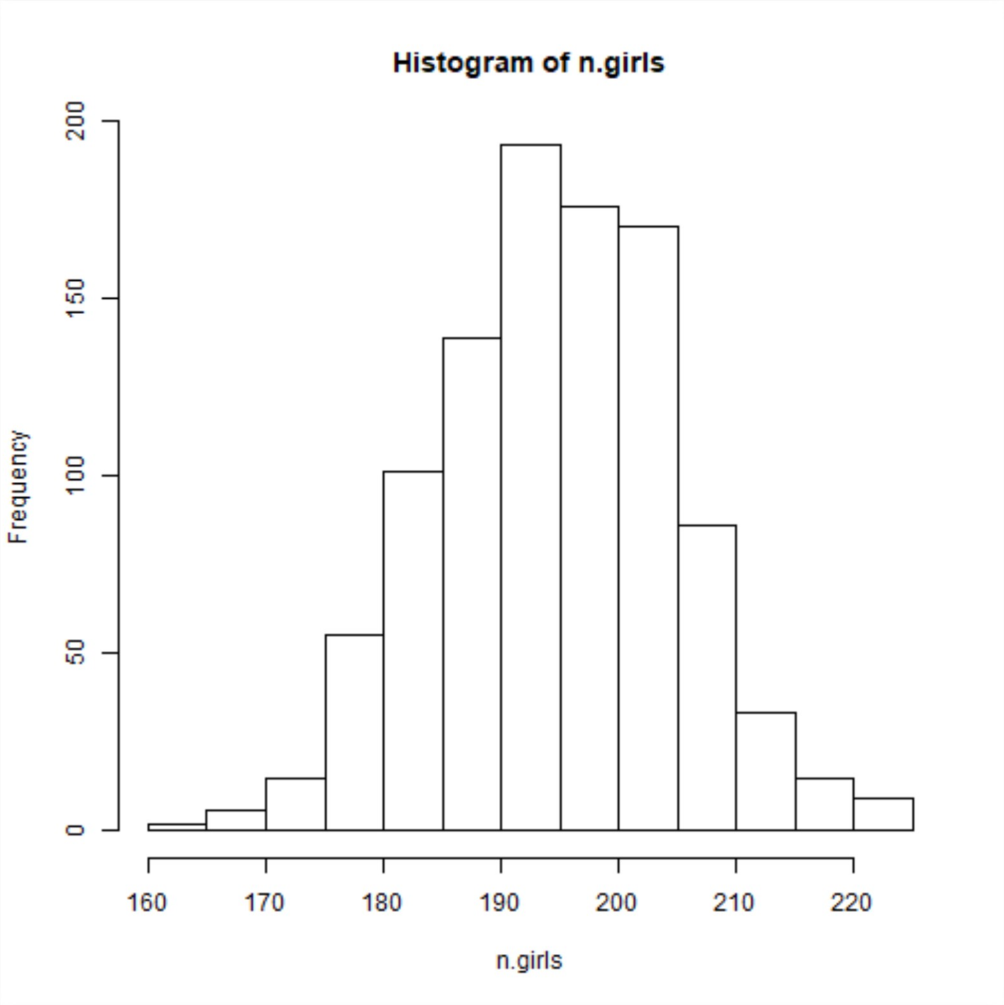
Suppose you have 400 births with a probability of a female birth being 0.488. We can simulate from a binomial distribution to get a prediction:

```
set.seed(111)
n.girls <- rbinom(n = 1, size = 400, prob =
.488)
print(n.girls)
```

```
[1] 199
```

- Simulating once gets us a single prediction - that's not necessarily helpful.
- It is more meaningful to repeat the process many times, as this then gets us a distribution of predictions.
- Try to repeat this process 1000 times and examine the distribution of outcomes.

```r
set.seed(111)
n.sims <- 1000
n.girls <- rbinom(n = n.sims, size = 400,
prob = 0.488)
hist(n.girls, breaks = 20)
```


Histogram of n.girls
Frequency
0
50
100
150
200
160
170
180
190
200
210
220
n.girls

More Complicated Simulation

A more complicated birth example

- $1/125$ chance that we have fraternal twins in a given birth
- The probability of a female birth is 0.488.
- $1/300$ chance that we have identical twins; in this case the probability of *both* being girls is 0.495.

Use simulation again to get a distribution of predictions on the number of female births in a set of 400 births.

Code

```
set.seed(2)
birth.type <- sample(c("fraternal twin", "identical twin", "single birth"),
                     size=400, replace=TRUE,
                     prob=c(1/125, 1/300, 1 - 1/125 - 1/300))
girls <- rep (NA, 400)
for (i in 1:400){
  if (birth.type[i]=="single birth"){
    girls[i] <- rbinom (1, 1, .488)
  } else if (birth.type[i]=="identical twin"){
    # If the twins are girls, we need to be sure to add 2 births
    girls[i] <- 2*rbinom (1, 1, .495)
  } else if (birth.type[i]=="fraternal twin"){
    girls[i] <- rbinom (1, 2, .495)
  }
}
n.girls <- sum(girls)
print(n.girls)
```

```
[1] 210
```

Code

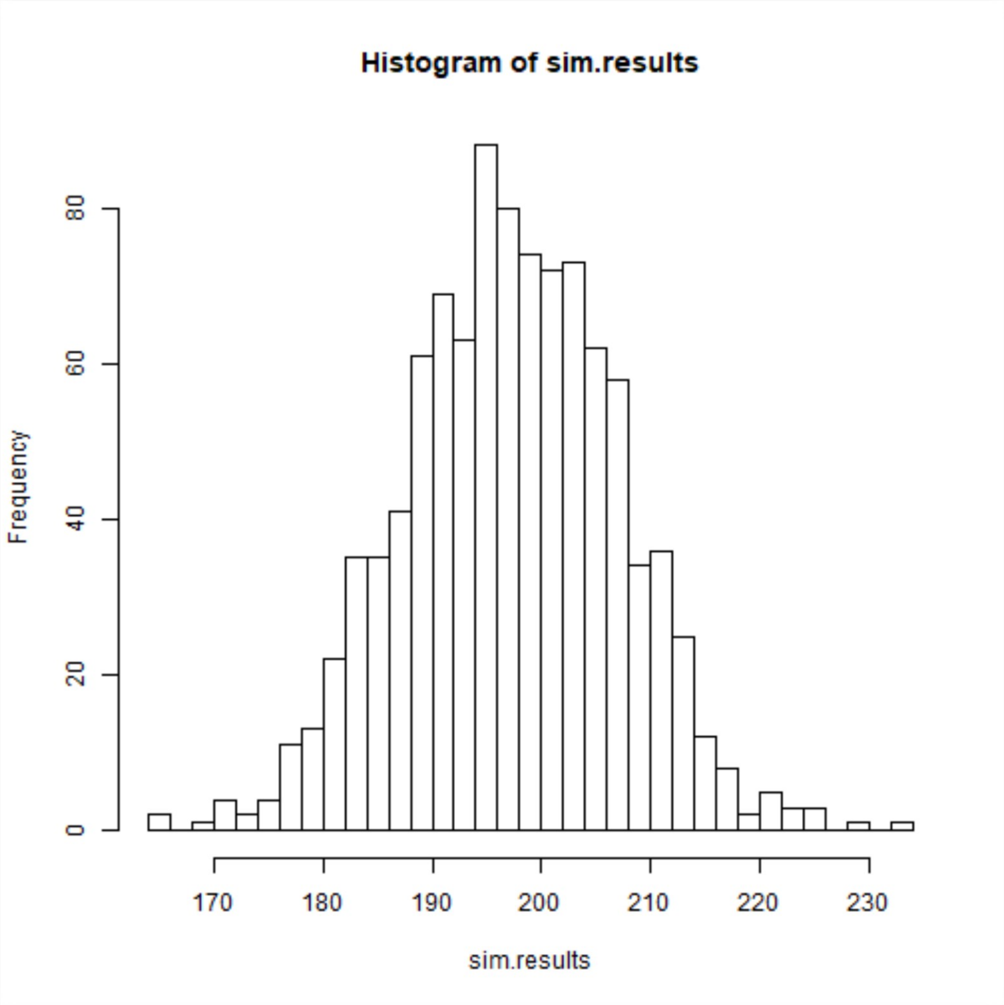
Now wrap this into a function so that we can repeatedly use it:

```
sim.girls <- function(nbirth) {
  birth.type <- sample( c("fraternal twin", "identical twin", "single
birth"),
                        size=nbirth, replace=TRUE,
                        prob=c(1/125, 1/300, 1 - 1/125 - 1/300))
  girls <- rep (NA, nbirth)
  for (i in 1:nbirth){
    if (birth.type[i]=="single birth"){
      girls[i] <- rbinom (1, 1, .488)
    } else if (birth.type[i]=="identical twin"){
      # If the twins are girls, we need to be sure to add 2
      girls[i] <- 2*rbinom (1, 1, .495)
    } else if (birth.type[i]=="fraternal twin"){
      girls[i] <- rbinom (1, 2, .495)
    }
  }
  return(sum(girls))
}

sim.girls(nbirth = 400)
```

```
[1] 203
```

Use the function `replicate` to call the function repeatedly

```r
set.seed(3)
sim.results <- replicate(1000, sim.girls(nbirth = 400))
hist(sim.results, breaks = 40)
```


Histogram of sim.results
Frequency
0
20
40
60
80
170
180
190
200
210
220
230
sim.results

Confidence Intervals for Complex Quantities

New example:

- Consider a survey of 1000 people, with 500 men ($n_m = 500$) and 500 women ($n_w = 500$).
- Suppose that 75% of men support the death penalty ($\hat{p}_m = 0.75$)
- 65% of women support the death penalty ($\hat{p}_w = 0.65$).
- The ratio for men to women's support is 1.15.

How do we get a confidence interval for this ratio?

There are ways to estimate the confidence interval analytically, but this will not always be the case.

One option we have is to simulate data using these true proportions.

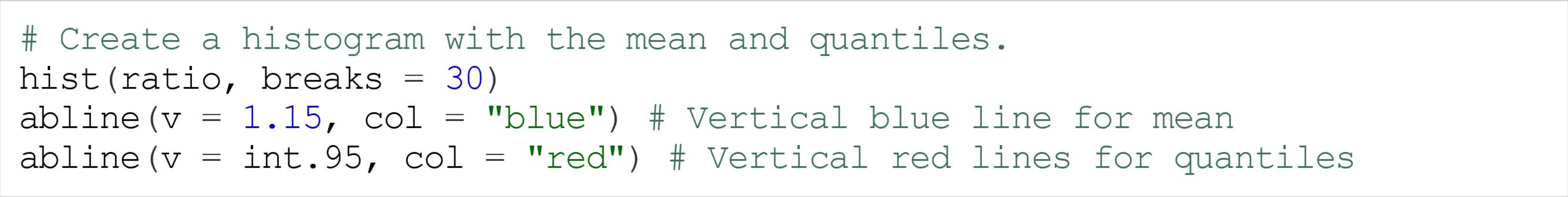
The idea here is that for men and women (with samples of 500 each), we have the estimated proportions for support of the death penalty, and we know the standard error from statistical theory for each fo these, which is given by $\sqrt{p(1-p)/n}$.

Simulation strategy:

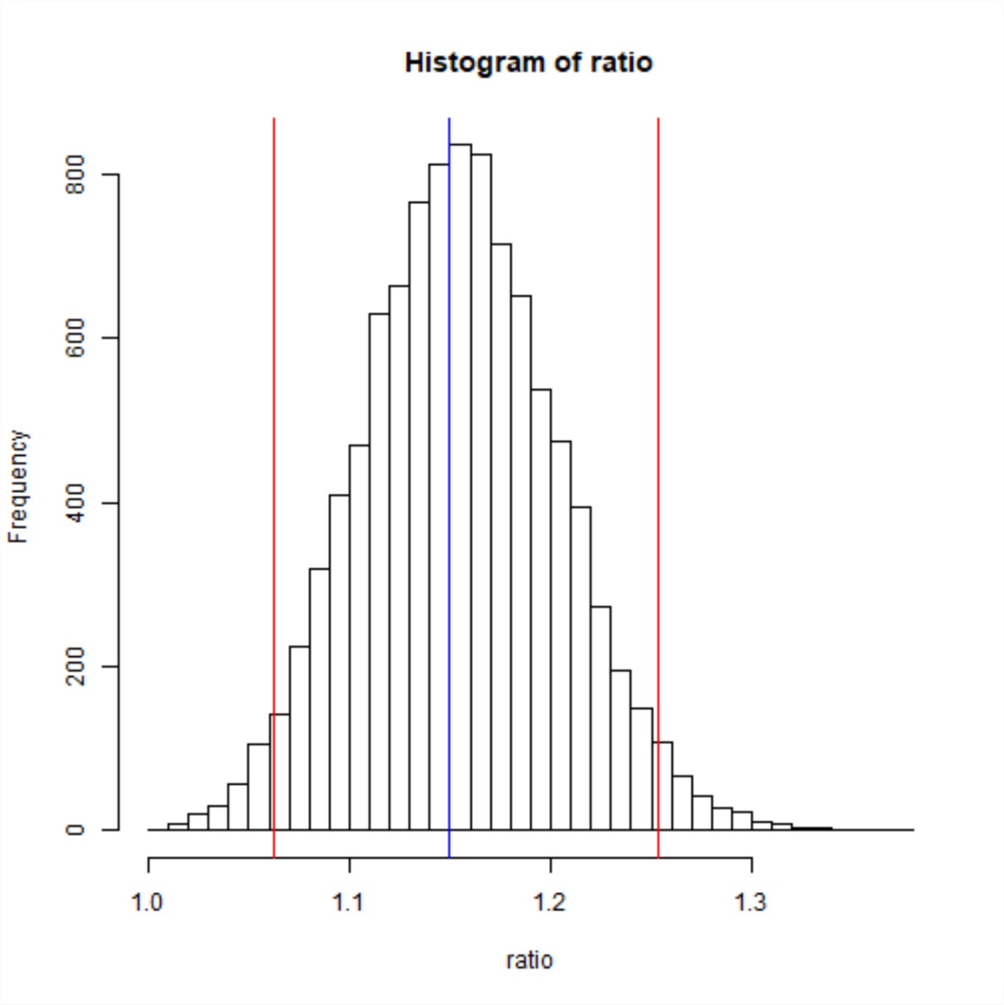
- We simulate from a normal distribution many times for men and women separately: $p \sim N(\hat{p}, \hat{p}(1-\hat{p})/n)$ using `rnorm()`.
- For each simulation, we calculate the ratio.
- Over many simulations, we can observe the distribution of the ratios and see what the confidence interval is using the `quantile()` function.

```r
# Set up the problem parameters
n.men <- 500
p.hat.men <- 0.75
se.men <- sqrt(p.hat.men * (1 - p.hat.men) / n.men)
n.women <- 500
p.hat.women <- 0.65
se.women <- sqrt(p.hat.women * (1 - p.hat.women) / n.women)

# Simulate 10,000 times, men and women separately
n.sims <- 10000
set.seed(4)
p.men <- rnorm(n = n.sims, mean = p.hat.men, sd = se.men)
p.women <- rnorm(n = n.sims, mean = p.hat.women, sd = se.women)

# Ratio of the two vectors
ratio <- p.men / p.women
# Use the quantile() function to find the 2.5% and 97.5% quantiles
print(int.95 <- quantile(ratio, c(0.025, 0.975)))
```

```
    2.5%    97.5% 
1.062003 1.253389 
```



```r
# Create a histogram with the mean and quantiles.
hist(ratio, breaks = 30)
abline(v = 1.15, col = "blue") # Vertical blue line for mean
abline(v = int.95, col = "red") # Vertical red lines for quantiles
```



Linear Regression Example

If we fit a linear regression model, we easily obtain coefficient standard errors and 95% confidence intervals.

Recall that a 95% **confidence interval** tells us that if we repeatedly sampled the population of interest and re-fit the models, the 95% confidence interval would cover the true value about 95% of the time.

How do we know that the estimated 95% confidence intervals are “correct”?

Linear Regression Example, Continued

We have two basic ways to answer this question:

1. Statistical theory.
2. Simulation.

Sometimes theory says one thing for large samples, but it may be difficult to know how performance deviates in smaller samples.

Other times we simply want to “confirm” to ourselves that the estimator we have actually works. This is a core component in methodological research when new estimation methods might be developed.

Linear Regression Example, Continued

Consider a simple example in the context of linear regression ($Y = \beta X + \epsilon$) where we have a small sample ($N = 30$). What is the coverage rate of the 95% confidence intervals for $\hat{\beta}$?

Steps to conduct this simulation:

- Generate random data where the true coefficient (β) is known: In this case we set $\beta = 2$
- Fit a linear model using `lm()`
- Extract the confidence interval using `confint()`
- Determine if the confidence interval contains the true value of β (1 = Yes, 0 = No).
- Repeat this process many times (1000).

```r
NUM.SIMS = 1000
N = 30
true.beta = 2
covered <- rep(NA, NUM.SIMS)
conf.matrix <- matrix(NA, nrow = NUM.SIMS, ncol = 2)
beta.est = rep(NA, NUM.SIMS)
set.seed(5)
for (i in 1:NUM.SIMS) {
  # Generate X, epsilon, and Y based on true.beta value
  X <- rnorm(N) # Standard normal
  epsilon <- rnorm(N) # Standard normal
  Y <- true.beta*X + epsilon # Intercept is 0
  lm.fit <- lm(Y ~ X)
  beta.est[i] <- coef(lm.fit)["X"]
  # Extract the 95% confidence interval
  ci <- confint(lm.fit)["X", ]
  conf.matrix[i, ] <- ci # Save CI it for later plotting
  # Does the true estimate lie in the confidence interval?
  covered[i] <- ifelse(ci[1] <= true.beta & ci[2] >= true.beta, 1, 0)
}

mean(covered)
```

```
[1] 0.951
```

This is one example of how to use simulation. Often times you will want to see how certain estimators work under different circumstances that deviate from the assumptions the theory relies on.

The basic recipe does not change much:

1. Generate data, such that the truth is known to you (the researcher).
2. Estimate using the method of interest (or several methods).
3. Repeat steps 1 and 2 K times
4. Evaluate the performance of the proposed method over the K simulations.