C33 Sampling

David Housman

## Concepts

Suppose we have a known population attribute described by the random variable with pdf , mean , and standard deviation . A *random sample* of the population attribute is the list of independent random variables each having the same distribution as . A *statistic* is a function from random samples. Two important statistics are the sample mean

and the sample variance

We are interested in the distribution of , called the *sampling distribution* of .



An approximation to the sampling distribution can be found via simulation. Before computers became powerful enough to carry out large simulations, many sampling distributions were obtained theoretically using calculus and algebra. Some of these results are listed below.

1. and .
2. and where is the fourth central moment of each . The table provides formulas for for several random variables (mathworld.wolfram.com is a good source for this type of information).

| Random Variable |  |  |  |
| --- | --- | --- | --- |
| Binomial |  |  |  |
| Poisson |  |  |  |
| Geometric |  |  |  |
| NegativeBinomial |  |  |  |
| Uniform |  |  |  |
| Normal |  |  |  |
| LogNormal |  |  |  |
| Exponential |  |  |  |
| Gamma |  |  |  |

1. The limit of the distribution of as approaches is the standard normal distribution.
2. The limit of the distribution of as approaches is the standard normal distribution.
3. If each , then and .
4. If each , then has a Student distribution with degrees of freedom.
5. If each , then .
6. If each , then .

## Cereal Example

A cereal manufacturer packages cereal in boxes with a 12-ounce label weight. Suppose that the actual distribution of weights is normal with mean 12.2 ounces and standard deviation 0.2 ounce. Use both simulation and theoretical approaches to find the probabilities of the following events.

1. A single box is under the labeled weight.

# Simulation approach
nruns = 10000
tibble(x = rnorm(nruns, 12.2, 0.2)) |>
 summarise(prob = sum(x < 12) / nruns)

# A tibble: 1 × 1
 prob
 <dbl>
1 0.156

#Theoretical approach
pnorm(12, 12.2, 0.2)

[1] 0.1586553

1. The mean of a sample of four boxes is under the labeled weight.

# Simulation approach
nruns = 10000
n = 4
tibble(run = rep(1:nruns, each = 4),
 x = rnorm(n\*nruns, 12.2, 0.2)) |>
 group\_by(run) |>
 summarise(mean = mean(x)) |>
 summarise(prob = sum(mean < 12) / nruns)

# A tibble: 1 × 1
 prob
 <dbl>
1 0.0241

#Theoretical approach
pnorm(12, 12.2, 0.2/sqrt(4))

[1] 0.02275013

1. The standard deviation of a sample of four boxes is under 0.2 ounce.

# Simulation approach
nruns = 10000
n = 4
tibble(run = rep(1:nruns, each = 4),
 x = rnorm(n\*nruns, 12.2, 0.2)) |>
 group\_by(run) |>
 summarise(sd = sd(x)) |>
 summarise(prob = sum(sd < 0.2) / nruns)

# A tibble: 1 × 1
 prob
 <dbl>
1 0.604

#Theoretical approach
pgamma(0.2^2, shape = (4-1)/2, scale = 2\*0.2^2/(4-1))

[1] 0.6083748

Comments: Simulation is an all-purpose tool but requires substantial code and with the answers are within with 95% confidence. Theoretical tools provide exact answers (up to the precision of the numerical approximations the integrations) but require that theorems have been proved.

## Widget Lifetime Example

The lifetimes of widgets follow an exponential distribution with a mean of 250 hours.

1. What proportion of the widgets will fail within 120 hours?

# Simulation approach
nruns = 10000
tibble(x = rexp(nruns, rate = 1/250)) |>
 summarise(prob = sum(x < 120) / nruns)

# A tibble: 1 × 1
 prob
 <dbl>
1 0.381

#Theoretical approach
pexp(120, rate = 1/250)

[1] 0.3812166

pgamma(120, shape = 1, scale = 250)

[1] 0.3812166

1. What is the probability that the mean lifetime of a sample of nine widgets is under 120 hours?

# Simulation approach
nruns = 10000
n = 9
tibble(run = rep(1:nruns, each = n),
 x = rexp(n\*nruns, rate = 1/250)) |>
 group\_by(run) |>
 summarise(mean = mean(x)) |>
 summarise(prob = sum(mean < 120) / nruns)

# A tibble: 1 × 1
 prob
 <dbl>
1 0.0331

If is the lifetime of widgets, then , and by a theorem, .

#Theoretical approach
pgamma(120, shape = 9, scale = 250/9)

[1] 0.03249181

The following shows that the sample size must be larger than 9 for the CLT to provide a good approximation.

# CLT approach
pnorm(120, 250, 250/sqrt(9))

[1] 0.05937994

1. What is the probability that in a sample of nine widgets, at least one will fail within 120 hours?

# Simulation approach
nruns = 10000
n = 9
tibble(run = rep(1:nruns, each = n),
 x = rexp(n\*nruns, rate = 1/250)) |>
 group\_by(run) |>
 summarise(min = min(x)) |>
 summarise(prob = sum(min < 120) / nruns)

# A tibble: 1 × 1
 prob
 <dbl>
1 0.987

#Theoretical approach
1 - (1 - pexp(120, rate = 1/250))^9

[1] 0.9867001

## Coin Flips Example

What is the probability that there are fewer than 35 heads in 100 flips of a fair coin?

# Simulation approach 1
nruns = 10000
n = 100
tibble(run = rep(1:nruns, each = n),
 x = sample(0:1, n\*nruns, replace = TRUE)) |>
 group\_by(run) |>
 summarise(heads = sum(x)) |>
 summarise(prob = sum(heads < 35) / nruns)

# A tibble: 1 × 1
 prob
 <dbl>
1 0.001

# Simulation approach 2
nruns = 1000000
tibble(x = rbinom(nruns, 100, 0.5)) |>
 summarise(prob = sum(x < 35) / nruns)

# A tibble: 1 × 1
 prob
 <dbl>
1 0.000895

#Exact amswer
pbinom(34, 100, 0.5)

[1] 0.0008949652

#CLT approximation
pnorm(34.5, 100\*0.5, sqrt(100\*0.5\*(1-0.5)))

[1] 0.0009676032

## Gamma Population Sampling Distributions

Suppose the population attribute has a gamma distribution with shape parameter and scale parameter .

1. Calculate the mean and standard deviation of . Display the pdf of and a normal distribution with the same mean and standard deviation as .

alpha = 1
beta = 1
mu = alpha\*beta; mu

[1] 1

sigma = sqrt(alpha)\*beta; sigma

[1] 1

ggplot() +
 geom\_function(fun = dgamma, args = list(shape = alpha, scale = beta), xlim = c(0, 7), color = "blue") +
 geom\_function(fun = dnorm, args = list(mean = mu, sd = sigma), xlim = c(0, 7), color = "red")



1. For samples of size , simulate sampling distributions for the sample mean, sample variance, and method of moments estimators.

nruns = 10000
n = 8
set.seed(19570830)
samples = tibble(
 run = rep(1:nruns, each = n),
 x = rgamma(n\*nruns, shape = alpha, scale = beta)
)
statistics = samples |>
 group\_by(run) |>
 summarise(
 xbar = mean(x),
 s2 = var(x),
 v = (n-1)/n\*s2,
 alpha.hat = xbar^2 / v,
 beta.hat = v / xbar
 )

1. For the statistic , calculate the theoretical mean and variance and the mean, standard error (standard deviation divided by the square root of the number of simulation runs), and variance of the simulated values. Display the sampling distribution as a density plot with what should be a matching gamma pdf and could be a matching normal pdf if the sample size is sufficiently large. Compare the simulation with theoretical results.

xbar.dist = statistics |>
 summarize(
 tmean = mu,
 mean = mean(xbar),
 se = sd(xbar)/sqrt(nruns),
 tvar = sigma^2 / n,
 var = var(xbar)
 )
xbar.dist

# A tibble: 1 × 5
 tmean mean se tvar var
 <dbl> <dbl> <dbl> <dbl> <dbl>
1 1 0.998 0.00351 0.125 0.123

ggplot(statistics, aes(x = xbar)) +
 geom\_density() +
 geom\_function(fun = dgamma, args = list(shape = n\*alpha, scale = beta/n), color = "blue") +
 geom\_function(fun = dnorm, args = list(mean = mu, sd = sigma/sqrt(n)), color = "red")



The sampling distribution of is the gamma distribution specified in the theorems and is different from but close to a normal distribution (despite being only 8 and the population attribute being far from normal). That the statistic is an unbiased estimator for is confirmed here since the approximated value of is within one standard error of . The approximated value of is also close to the theoretical value.

1. For the statistic , calculate the theoretical mean and variance and the mean, standard error, and variance of the simulated values. Display the sampling distribution as a density plot with what could be a matching normal pdf if the sample size is sufficiently large. Compare the simulation with theoretical results.

s2.dist = statistics |>
 summarize(
 tmean = sigma^2,
 mean = mean(s2),
 se = sd(s2) / sqrt(nruns),
 tvar = 3\*alpha\*(alpha+2)\*beta^4/n - (alpha\*beta^2)^2\*(n-3)/n/(n-1),
 var = var(s2)
 )
s2.dist

# A tibble: 1 × 5
 tmean mean se tvar var
 <dbl> <dbl> <dbl> <dbl> <dbl>
1 1 0.999 0.0101 1.04 1.02

ggplot(statistics, aes(x = s2)) +
 geom\_density() +
 geom\_function(fun = dnorm, args = list(mean = s2.dist$tmean, sd = sqrt(s2.dist$tvar)), color = "red")



The sampling distribution of was not specified by theoretically and is definitely different from a normal distribution. That the statistic is an unbiased estimator for is confirmed here since the approximated value of is within one standard error of . The approximated value of is also close to the theoretical value.

1. For the statistic , calculate the mean, standard error, and standard deviation of the simulated values. Display the sampling distribution as a density plot with a normal pdf with the same mean and standard deviation as the simulated data. Compare the simulation with theoretical results.

beta.hat.dist = statistics |>
 summarize(
 tmean = beta,
 mean = mean(beta.hat),
 se = sd(beta.hat) / sqrt(nruns),
 sd = sd(beta.hat)
 )
beta.hat.dist

# A tibble: 1 × 4
 tmean mean se sd
 <dbl> <dbl> <dbl> <dbl>
1 1 0.778 0.00550 0.550

ggplot(statistics, aes(x = beta.hat)) +
 geom\_density() +
 geom\_function(fun = dnorm, args = list(mean = beta.hat.dist$mean, sd = beta.hat.dist$sd), color = "red")



The statistic is definitely biased as an estimator for and its distribution though unimodal is skewed right and clearly different from a normal distribution.

1. For the statistic , calculate the mean, standard error, and standard deviation of the simulated values. Display the sampling distribution as a density plot with a normal pdf with the same mean and standard deviation as the simulated data. Compare the simulation with theoretical results.

alpha.hat.dist = statistics |>
 summarize(
 tmean = alpha,
 mean = mean(alpha.hat),
 se = sd(alpha.hat) / sqrt(nruns),
 sd = sd(alpha.hat)
 )
alpha.hat.dist

# A tibble: 1 × 4
 tmean mean se sd
 <dbl> <dbl> <dbl> <dbl>
1 1 1.76 0.0120 1.20

ggplot(statistics, aes(x = alpha.hat)) +
 geom\_density() +
 geom\_function(fun = dnorm, args = list(mean = alpha.hat.dist$mean, sd = alpha.hat.dist$sd), color = "red")



The statistic is definitely biased as an estimator for and its distribution though unimodal is skewed right and clearly different from a normal distribution.`

## Proving the Theoretical Results

During our discussion of parameter estimation, we proved the general expressions for , , and . Our goal in this section is to examine result 5 in the Concepts section: If each , then and . We will prove the first half and provide numerical evidence to support the second half.

1. If and are numbers and , then .
* Proof. The cdf of is
* and
* which is the desired pdf.
1. If and are independent continuous random variables with pdfs and , then is a continuous random variable with pdf .
* Proof. Suppose is the cdf for . Then
* and
1. If and are independent, then .
* Proof. Note first that the mean and standard deviation of are correct based on earlier results. By result 1, the pdf of is
* Completing the square on the quadratic expression, we obtain
* Plugging back into our expression for the pdf, we obtain
* $
* which is the desired pdf.
1. If for are independent, then .
* Proof. . By result 1, for . By repeated use of result 3, . By result 1, is the desired distribution.
1. If for are independent, then .

Proof. By result 4, . By result 2, we obtain the desired distribution.

1. Choose a positive integer , a real number , and a positive real number . For samples of size from a population attribute , simulate sampling distributions for the sample variance. Compare the mean, standard deviation, pdf, and cdf from the simulation with the theoretical .

nruns = 10000
n = 7; mu = 29; sigma = 3
set.seed(19570830)
samples = tibble(
 run = rep(1:nruns, each = n),
 x = rnorm(n\*nruns, mean = mu, sd = sigma)
)
statistics = samples |>
 group\_by(run) |>
 summarise(s2 = var(x))
statistics |>
 summarise(
 smean = mean(s2),
 se.smean = sd(s2)/sqrt(nruns),
 tmean = sigma^2,
 ssd = sd(s2),
 tsd = sqrt(3\*sigma^4/n - sigma^4\*(n-3)/n/(n-1))
 )

# A tibble: 1 × 5
 smean se.smean tmean ssd tsd
 <dbl> <dbl> <dbl> <dbl> <dbl>
1 8.97 0.0513 9 5.13 5.20

ggplot(statistics, aes(x = s2)) +
 geom\_density() +
 geom\_function(fun = dgamma, args = list(shape = (n-1)/2, scale = 2\*sigma^2/(n-1)), color = "red")



ggplot(statistics, aes(x = s2)) +
 stat\_ecdf() +
 geom\_function(fun = pgamma, args = list(shape = (n-1)/2, scale = 2\*sigma^2/(n-1)), color = "red")

