

## The sample mean and its properties

Suppose we have a sample of size  $n$

$$X_1, X_2, \dots, X_n$$

from a population that we are studying.

Depending on the situation, we may be willing to assume that the  $X_i$  are identically distributed, implying that they have a common mean  $\mu$  and variance  $\sigma^2$ . That is,

$$EX_i = \mu \quad \text{var}X_i = \sigma^2$$

As a further assumption, we may be willing to assume that the  $X_i$  are independent.

The sample mean (“average”)

$$\bar{X} = (X_1 + \cdots + X_n)/n = \sum_i X_i/n$$

is a random variable with its own distribution, called the sampling distribution.

The expected value of  $\bar{X}$  is

$$E\bar{X} = \mu$$

and the variance of  $\bar{X}$  is

$$\text{var}\bar{X} = \sigma^2/n$$

## Learning about the sampling distribution through simulation

We can study the sampling behavior of  $\bar{X}$  by simulating many data sets and calculating the  $\bar{X}$  value for each set.

The following program simulates `nrep` data sets, each containing `nsamp` independent, identically distributed (iid) values. For this simulation, the values are simulated from a normal (Gaussian) distribution. The population mean and population standard deviation of the data values are specified by the variables `pop_mean` and `pop_sd`.

```
nsamp = 20      ## The number of data values in each data set
nrep = 1000    ## The number of data sets to generate
pop_mean = 0   ## The population mean of each data value
pop_var = 1    ## The population variance of each data value

## Generate a nrep x nsamp array of standard normal draws.
D = rnorm(nrep*nsamp, mean=pop_mean, sd=sqrt(pop_var)) ## **
X = array(D, c(nrep,nsamp))

## Get the mean of each row of X.
Y = apply(X, 1, mean)

## Calculate the sample variance of Y.
V = var(Y)
```

To visualize the results of the simulation, we can generate histograms for the raw data (blue) and sample means (red). They are plotted together to show how they relate to each other.

```
## Generate a histogram of the raw data.
h1 = hist(D[1:nrep])

## Generate a histogram of the sample means.
h2 = hist(Y)

## Plot the raw data histogram in blue.
plot(h1, col='blue', ylim=c(0,max(h1$counts, h2$counts)),
      xlab='', main='')

## Overplot the sample means histogram in red.
lines(h2, col='red')

## Add a legend to the plot.
legend(x='topright', legend=c('Raw data', 'Averages'),
       col=c('blue', 'red'), lty=c(1,1))
```

## Questions to ask yourself

- Compare  $V$  and `pop_var`. Ensure that what you see is compatible with the fact that  $\text{var}\bar{X} = \sigma^2/n$ .
- Vary the values of `nsamp` and `pop_var` to check that the value of  $V$  changes as expected.
- Confirm that changing `pop_mean` and `nrep` has no systematic effect on the result of the program (as long as `nrep` is not too small).
- Make sure you understand how the spread of the histograms relates to `pop_var` and `nsamp`.

## Does the distribution of the data matter?

Change one line in the previous simulation (the line with the `**` comment) to one of the following two lines. This will use data from a different distribution in the simulation.

```
## Generate data with a standard uniform distribution.  
D = runif(nrep*nsamp)  
  
## Generate data with a standard exponential distribution.  
D = rexp(nrep*nsamp)
```

Reconsider each of the questions on the previous slide.

## **A closer look at the sample size**

Now suppose we want to look more systematically at the effect of changing the sample size.

We can loop over a range of sample sizes and carry out the simulation study separately for each sample size.

```
NSamp = seq(10,100,10) ## The sample sizes to consider
nrep = 1000           ## The number of data sets to generate

## A place to store the results.
V = NULL

## Vary the sample size over 10, 20, ..., 100.
for (k in 1:length(NSamp)) {

  ## The sample size to use in this iteration.
  nsamp = NSamp[k]

  ## Generate a 1000xnsamp array of standard normal draws.
  D = rnorm(nrep*nsamp)
  X = array(D, c(nrep,nsamp))

  ## Get the mean of each row of X.
  Y = apply(X, 1, mean)

  ## Calculate the sample variance of Y.
```

```
V[k] = var(Y)  
}
```

When the simulation is finished,  $V$  and  $NSamp$  will have the same length. The value of  $V[k]$  will be the variance of  $\bar{X}$  when the sample size is  $NSamp[k]$ .

The following code produces a plot that summarizes the results of the simulation.

```
## Plot the simulation results.
plot(NSamp, V, t='b', xlab='Sample size', ylab='Variance of Xbar')

## Overplot the exact results.
lines(NSamp, 1/NSamp, t='b', col='red')

## Add a legend.
legend(x='topright', legend=c('Simulation', 'Theory'),
       col=c('black', 'red'), lty=c(1,1))
```

## The tradeoff between sample size and variance

Suppose we have two instruments for measuring a quantity of interest. Let  $X$  denote a measurement from the first instrument, and let  $Y$  denote a measurement from the second instrument.

Assume both instruments are calibrated correctly, so

$$EX = EY = \mu,$$

where  $\mu$  is the exact value being measured.

Suppose the first instrument is more precise, so that

$$\text{var}(X) = \sigma_X^2 < \text{var}(Y) = \sigma_Y^2.$$

## The tradeoff between sample size and variance (continued)

Our goal is to estimate  $\mu$ . Suppose we have the choice of using the first instrument with a sample size of  $n_X$ , or the second instrument with a sample size of  $n_Y$ . Since

$$E\bar{X} = EX_i = \mu = EY_i = E\bar{Y},$$

either instrument can be used to form an unbiased average. The variances will be

$$\text{var}\bar{X} = \sigma_X^2/n_X \quad \text{var}\bar{Y} = \sigma_Y^2/n_Y.$$

Therefore if

$$\sigma_X^2/\sigma_Y^2 = n_X/n_Y,$$

the two averages are equally precise. Let's check this with a simulation.

```
nrep = 1000

## The second instrument has twice the variance, but we get to
## use twice the sample size.
nx = 10
vx = 1
ny = 20
vy = 2

## Generate the data for the first instrument.
D = rnorm(nrep*nx, sd=sqrt(vx))
X = array(D, c(nrep,nx))
MX = apply(X, 1, mean)

## Generate the data for the first instrument.
D = rnorm(nrep*ny, sd=sqrt(vy))
Y = array(D, c(nrep,ny))
MY = apply(Y, 1, mean)
```

We can compare the averages of the first instrument to those of the second instrument using box plots.

```
## Concatenate the means into a single vector.  
M = c(MX, MY)  
  
## A group id vector.  
G = c(array(1, 1000), array(2, 1000))  
  
## Generate side by side boxplots.  
boxplot(M ~ G, names=c('First instrument', 'Second instrument'))
```

## Exceptional cases

The *Cauchy distribution* has no mean and infinite variance. In this case, averaging doesn't improve the precision.

```
V = NULL

## Sample sizes.
NSamp = c(10,20,40,80,160)

## Loop over a sequence of sample sizes.
for (k in 1:length(NSamp))
{
  ## The sample size for this iteration.
  r = NSamp[k]

  ## Generate nrep data sets containing r values each.
  X = array(rcauchy(r*1000), c(1000,r))

  ## The sample mean of each data set.
  Y = apply(X, 1, mean)

  V[k] = var(Y)
}
```

## Theoretical properties of the expected value

If  $c$  is a constant and  $X$  and  $Y$  are random variables, the expected value has the following properties:

$$E(X + c) = (EX) + c$$

$$E(c \cdot X) = c \cdot EX$$

$$E(X + Y) = EX + EY$$

If  $X$  and  $Y$  are uncorrelated then

$$E(X \cdot Y) = EX \cdot EY.$$

## Theoretical properties of the sample mean

Suppose  $X_1, \dots, X_n$  is a sequence of numbers, and let  $Y_i = X_i + c$  and  $Z_i = cX_i$ .  
Then

$$\bar{Y} = \bar{X} + c$$

$$\bar{Z} = c\bar{X}$$

Suppose  $U_i$  and  $V_i$  are sequences of numbers and  $W_i = U_i + V_i$ , then

$$\bar{W} = \bar{U} + \bar{V}.$$

## Theoretical properties of the variance

$\text{var}(X + c) = \text{var}(X)$  when  $c$  is a constant.

$\text{var}(X) = 0$  when  $X$  is constant.

$\text{var}(c \cdot X) = c^2 \text{var}(X)$  when  $c$  is a constant.

$\text{var}(X + Y) = \text{var}(X) + \text{var}(Y)$  when  $X$  and  $Y$  are uncorrelated.

## Theoretical properties of the standard deviation

The population standard deviation is defined as  $SD(X) = \sqrt{\text{var}(X)}$ .

$SD(X + c) = SD(X)$  when  $c$  is a constant.

$SD(X) = 0$  when  $X$  is constant.

$SD(c \cdot X) = |c|SD(X)$  when  $c$  is a constant.

$SD(X + Y) = \sqrt{SD(X)^2 + SD(Y)^2}$  when  $X$  and  $Y$  are uncorrelated.

We can explore some of these properties using simulations.

```
## X and Y are uncorrelated.  
X = rexp(1e4)  
Y = rexp(1e4)  
S = X + Y    ## compare mean(X)+mean(Y) to mean(S)  
Z = X*Y      ## compare mean(X)*mean(Y) to mean(Z)  
  
## U and V are correlated because they both include A.  
A = rexp(1e4)  
U = A + rexp(1e4)  
V = A + rexp(1e4)  
T = U + V    ## compare mean(U)+mean(V) to mean(T)  
W = U*V      ## compare mean(U)*mean(V) to mean(W)
```

## Functions of random variables

Suppose  $X$  is a random variable and we define a new random variable  $Y = f(X)$ , where  $f(x)$  is a mathematical function. How does the mean of  $X$  relate to the mean of  $Y$ ?

As a crude approximation

$$Ef(X) \approx f(EX).$$

The approximation is exact when  $f$  is linear, i.e.  $f(X) = a + bX$  for constants  $a$  and  $b$ . In other cases it can be moderately or substantially incorrect.

We can check this approximation using simulation.

```
## Generate uniform data on (0,1). The expected value is 1/2.  
X = runif(1e4)  
  
## Simulate the exact result for the log function.  
L1 = mean(log(X))  
  
## The mathematical approximation for the log function.  
L2 = log(1/2)  
  
## Simulate the exact result for the squaring function.  
S1 = mean(X^2)  
  
## The mathematical approximation for the squaring function.  
S2 = (1/2)^2
```

Can we say something more exact? The answer is yes in certain cases.

If  $f(x)$  is a concave function ( $f''$  is always negative, e.g. log or square-root), then

$$Ef(X) \leq f(EX).$$

If  $f(x)$  is a convex function ( $f''$  is always positive, e.g.  $\exp(x)$  or  $x^2$ ), then

$$Ef(X) \geq f(EX).$$

What will be the sign of  $E1-E2$  after executing the following program?

```
## 1000 standard exponential draws
X = rexp(1000)

## Estimate E log(X)
E1 = mean(log(X))

## Estimate log(EX)
E2 = log(mean(X))
```

What will be the sign of  $E1 - E2$  after executing the following program?

```
## 1000 standard normal draws
X = rnorm(1000)

## Estimate E X^2
E1 = mean(X^2)

## Estimate (EX)^2
E2 = mean(X)^2
```

Note that many functions are neither convex nor concave (e.g.  $f(x) = x^3$ ), so these results cannot always be applied.

Can we say anything about  $\text{var}f(X)$ ? The answer is yes, but only approximately.

By Taylor's theorem, we can pick a point  $X_0$  and write

$$f(X) \approx f(X_0) + (X - X_0)f'(X_0),$$

with the approximation holding better when  $X$  is close to  $X_0$ . Taking the variance of both sides of the approximation gives us

$$\text{var}(f(X)) \approx \text{var}(X) \cdot f'(X_0)^2.$$

We can choose any value for  $X_0$ , but the approximation tends to hold well when  $X_0$  is set to  $EX$ .

Here is a simulation to assess this approximation. Note that the variance of a uniform random variable on  $(0,1)$  is  $1/12$ .

```
## Generate uniform data on (0,1). The expected value is 1/2.  
X = runif(1e4)  
  
## Simulate the exact result for the log transform.  
L1 = var(log(X))  
  
## The mathematical approximation for the log function.  
L2 = 1/3  
  
## Simulate the exact result for the squaring transform.  
S1 = var(X^2)  
  
## The mathematical approximation for the squaring function.  
S2 = 1/12
```

## Estimating the variance

The sample variance

$$\hat{\sigma}^2 = \sum_i (X_i - \bar{X})^2 / (n - 1)$$

is the standard way to estimate the population variance from data.

The following simulation demonstrates that  $\hat{\sigma}^2$  is unbiased. Make sure you understand how this program differs from the simulations given previously.

```
nsamp = 20      ## The number of data values in each data set
nrep = 1000    ## The number of data sets to generate
pop_mean = 0   ## The population mean of each data value
pop_var = 1    ## The population variance of each data value

## Generate a nrep x nsamp array of standard normal draws.
D = rnorm(nrep*nsamp, mean=pop_mean, sd=sqrt(pop_var))
X = array(D, c(nrep,nsamp))

## Get the variance of each row of X.
Y = apply(X, 1, var)

## Calculate the sample mean of Y.
V = mean(Y)
```

We might also be interested in the variance of  $\hat{\sigma}^2$ , which reflects our ability to precisely estimate the population variance  $\sigma^2$ .

It is important to understand that the  $\sigma^2/n$  formula does not apply to  $\hat{\sigma}^2$ .

However  $\hat{\sigma}^2$  does belong to a broad class of estimators for which the sampling variance is approximately cut in half every time the sample size doubles.

```
nrep = 1e4
V = NULL
NSamp = c(20,40,80,160)

for (k in 1:length(NSamp)) {

  nsamp = NSamp[k]

  ## Generate a nrep x nsamp array of standard normal draws.
  D = rnorm(nrep*nsamp)
  X = array(D, c(nrep,nsamp))

  ## Get the variance of each row of X.
  Y = apply(X, 1, var)

  ## Calculate the sample variance of Y.
  V[k] = var(Y)
}
```