05 Sampling Distribution of Statistics

The Sample Mean and Its Properties

- Suppose we have a sample of size n

$$X_1, X_2, \ldots, X_n$$

  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 μ and variance σ^2.
- That is,

$$EX_i = \mu \qquad \text{and} \qquad \operatorname{Var} X_i = \sigma^2$$

  for $i = 1, \ldots, n$.
- As a further assumption, we may be willing to assume that the $X_1, \ldots, X_n$ are independent with each other.

The Sample Mean and Its Properties

- The sample mean ("average")

$$\bar{X} = \frac{X_1 + X_2 + \ldots + X_n}{n} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

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

$$\mathrm{Var}\,\bar{X} = \frac{\sigma^2}{n}$$

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`.

```
set.seed(1234)
nsamp <- 20    ## The number of samples in each data set
nrep <- 1000   ## The number of data sets to generate
pop_mean <- 0  ## The population mean
pop_var <- 1   ## The population variance

## 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)
## Compare the theoretical and simulation means.
c(pop_mean, mean(Y))
## Compare the theoretical and simulation variances.
c(pop_var/nsamp, var(Y))
```

Example: Sampling Distribution

- To visualize the simulation results, 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(X[ ,sample(1:nsamp,1)])
## Generate a histogram of the sample means.
h2 <- hist(Y)

plot(h1, col="blue", xlab="", main="",
             ylim=c(0, max(h1$counts, h2$counts)))
lines(h2, col="red")

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

Questions to Ask Yourself

- Compare `V` and `pop_var`. Ensure that what you see is compatible with the fact that

$$\operatorname{Var} \bar{X} = \frac{\sigma^2}{n}.$$

- 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`.

```r
set.seed(1111)
nsamp <- 100
nrep <- 1000
pop_mean <- 0
pop_var <- 1

D <- rnorm(nrep*nsamp, mean=pop_mean, sd=sqrt(pop_var))
X <- array(D, c(nrep, nsamp))
Y <- apply(X, 1, mean)
c(pop_var/nsamp, var(Y))

h1 <- hist(X[ ,sample(1:nsamp,1)])
h2 <- hist(Y)
plot(h1, col="blue", xlab="", main="",
             ylim=c(0, max(h1$counts, h2$counts)))
lines(h2, col="red")
```

Exercise

- Generate random values

$$X_1, X_2, \ldots, X_n$$

from a standard normal distribution, and let us denote

$$Y_i = I\left(X_i \geq 0\right)$$

Consider 3 different sample sizes such as $n = 10$, 50 and 100.

1. Compute $E(Y)$ and $\mathrm{Var}(Y)$ for each sample size.
2. Compute a sample mean and a sample variance of Y for each sample size.

Central Limit Theorem

- Does the distribution of the population matter?
- If X has the $N(\mu, \sigma)$ distribution, then

$$\bar{X} \sim N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

- Central Limit Theprem (CLT)
  If X has any distribution with a mean μ and a standard deviation σ, and n is large enough, then

$$\bar{X} \sim N\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

Central Limit Theorem

- Change one line in the previous simulation to one of the following two lines. This will use data from a different distribution in the simulation.

```
## Generate from a standard uniform distribution.
D <- runif(nrep*nsamp, min=0, max=1)

## Generate from a standard exponential distribution.
D <- rexp(nrep*nsamp, rate=1)
```

- Compute the sample mean and the sample variance. Confirm that the results are still

$$E\bar{X} = \mu \qquad \text{and} \qquad \operatorname{Var}\bar{X} = \frac{\sigma^2}{n}$$

The Effect of 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.

```
set.seed(1111)
## The sample sizes to be considered
NSamp <- seq(10, 100, 10)

## The number of data sets to generate
nrep <- 1000

## 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 nrep * nsamp array of standard
    ## normal random 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)
}
```

The Effect of the Sample Size

- 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.

```
M <- cbind(1/NSamp, V)
rownames(M) <- NSamp
colnames(M) <- c("Theory", "Simulation")
M

plot(NSamp, V, t="b", xlab="Sample size",
        ylab="Variance of Xbar")
lines(NSamp, 1/NSamp, t="b", col="red")
legend("topright", c("Simulation","Theory"),
             col=c("black","red"), lty=1)
```

The Trade-off between Sample Size and Variance

- Suppose that we have two populations for measuring a quantity of interest. Let X denote a measurement from the first population, and let Y denote a measurement from the second population.
- Assume that both populations have the same mean, so

$$EX = EY = \mu$$

- Suppose that the variance of the first population is smaller than that of the second population, so that

$$\mathrm{Var}(X) = \sigma_X^2 < \mathrm{Var}(Y) = \sigma_Y^2$$

The Tradeoff between Sample Size and Variance

- Our goal is to estimate μ. Suppose we collects samples from the first population with a sample size of n_X, and from the second population with a sample size of n_Y. Since

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

either population can be used to form an average.

- The variances will be

$$\mathrm{Var}\,\bar{X} = \frac{\sigma_X^2}{n_X} \qquad\qquad \mathrm{Var}\,\bar{Y} = \frac{\sigma_Y^2}{n_Y}$$

Therefore, if

$$\frac{\sigma_X^2}{\sigma_Y^2} = \frac{n_X}{n_Y},$$

two variances are the same.

- Let's check this with a simulation.

Example

- $\sigma_Y^2 = 2\sigma_X^2$ and $n_Y = 2n_X$

```
set.seed(12345)
nrep <- 1000
nx <- 10; vx <- 1
ny <- 20; vy <- 2

D <- rnorm(nrep*nx, sd = sqrt(vx))
X <- array(D, c(nrep, nx))
MX <- apply(X, 1, mean)

D <- rnorm(nrep*ny, sd = sqrt(vy))
Y <- array(D, c(nrep, ny))
MY <- apply(Y, 1, mean)

c(var(MX), var(MY))
```

Example

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

```
## Concatenate the means into a single vector.
M <- c(MX, MY)

## A group id vector.
G <- rep(1:2, each=1000)

## Generate side by side boxplots.
boxplot(M ~ G, names=c('First Pop', 'Second Pop'),
        col=c("cyan", "pink"), boxwex=0.4, xlab="")
```

Exceptional Cases for Sample Size and Variance

- The Cauchy distribution has no mean and infinite variance.

```
set.seed(123)
V <- NULL
NSamp <- c(10, 20, 40, 80, 160)

for (k in 1:length(NSamp))
{
    r <- NSamp[k]
    X <- matrix(rcauchy(1000*r), 1000, r)
    Y <- apply(X, 1, mean)
    V[k] <- var(Y)
}
cbind("Simulation"=V, "Theory"=1/NSamp)
```

Exercise

- Suppose that

$$X_i \sim U(0,1) \quad \text{and} \quad Y_i \sim U(0,1)$$

  Let us define

$$T_i = \cos(2\pi X_i)\sqrt{-2\log Y_i}$$

  Fix the sample size as $n = 100$.

  1. Compute
$$E(\bar{T}_n) = E\left(\frac{1}{n}\sum_{i=1}^{n} T_i\right)$$
  2. Compute $\mathrm{Var}\left(\bar{T}_n\right)$

Estimating the Variance

- The sample variance

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

  is the standard way to estimate the population variance from data.
- The following simulation demonstrates that $\hat{\sigma}^2$ is unbiased. That is,

$$E(\hat{\sigma}^2) = \sigma^2$$

- Make sure you understand how this program differs from the simulations given previously.

```r
set.seed(1234)
nsamp <- 30      ## The number of samples in each data set
nrep <- 1000     ## The number of data sets to generate
pop_mean <- 0    ## The population mean
pop_var <- 1     ## The population variance

## Generate a nrep * 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)

c(pop_var, V)
```

```
set.seed(12345)
var2 <- function(x) (length(x)-1)*var(x)/length(x)

Nsamp <- c(3, 5, 10, 20 ,30, 50, 100)
out <- matrix(0, length(Nsamp), 2)
for (i in 1:length(Nsamp)) {
    X <- matrix(rnorm(nrep*Nsamp[i]), nrep, Nsamp[i])
    V1 <- apply(X, 1, var)
    V2 <- apply(X, 1, var2)
    out[i,] <- c(mean(V1), mean(V2))
}
colnames(out) <- c("1/(n-1)", "1/n")
rownames(out) <- Nsamp
out
```

Variance of the Sample Variance

- We might also be interested in the variance of $\hat{\sigma}^2$, which reflects our ability to precisely estimate the population variance σ^2.
- It is important to understand that the σ^2/n formula does not apply to σ^2, i.e.,

$$\text{Var}(\hat{\sigma}^2) \neq \text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

- 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.
- When $X \sim N(\mu, \sigma)$ and $\hat{\sigma}^2$ is a sample variance,

$$\text{Var}(\hat{\sigma}^2) = \frac{2\sigma^4}{n-1}$$

```
set.seed(54321)
nrep <- 1e4
V <- NULL
mu <- 0
sig <- 2
NSamp <- c(10, 20, 40, 80, 160, 300)

for (k in 1:length(NSamp)) {
    nsamp <- NSamp[k]
    D <- rnorm(nrep*nsamp, mu, sig)
    X <- matrix(D, nrep, nsamp)
    Y <- apply(X, 1, var)
    V[k] <- var(Y)
}
out <- cbind(V, 2*sig^4/(NSamp-1))
colnames(out) <- c("Simulation", "Theory")
rownames(out) <- NSamp
out
```

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.

Functions of Random Variables

- We can check this approximation using simulation.

```
set.seed(1234)
X <- runif(1e4)

## Consider the log function (concave)
c(mean(log(X)), log(mean(X)))

## Consider the square root function (concave)
c(mean(sqrt(X)), sqrt(mean(X)))

## Consider the squaring function (convex)
c(mean(X^2), mean(X)^2)

## Consider the exponential function (convex)
c(mean(exp(X)), exp(mean(X)))
```

Jensen's Inequality

- If $f(x)$ is a concave function (f'' is always negative), then

$$Ef(x) \leq f(EX)$$

  - For example, $\log$ or square-root

- If $f(x)$ is a convex function (f'' is always positive), then

$$Ef(x) \geq f(EX)$$

  - For example, $\exp(x)$ or x^2

Example

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

```
set.seed(123456)
D01 <- D02 <- D03 <- NULL

for (i in 1:1000) {
    X <- rnorm(1000, sd = 3)
    ## neither convex or concave
    D01[i] <- mean(X^3) - mean(X)^3
    ## convex
    D02[i] <- mean(X^2) - mean(X)^2
    ## concave
    D03[i] <- mean(-X^2) - (-mean(X)^2)
}
matplot(cbind(D01, D02, D03), type="l", col=c(1,2,4),
        ylab="")
```

Exercise

- Suppose that X_i follows a standard uniform distribution for $i = 1, \ldots, n$ and $n = 10^5$. Compare between

$$\mathrm{Var}(f(X)) \qquad \text{and} \qquad f(\mathrm{Var}(X))$$

when $f(X)$ is either convex or concave function.

1. $f(X) = \log(X)$
2. $f(X) = \sqrt{X}$
3. $f(X) = e^X$
4. $f(X) = X^2$

Sampling Distributions of Test Statistics

- When $X_i \sim N(\mu, \sigma)$ for $i = 1, 2, \ldots, n$,

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1)$$

  and

$$T = \frac{\bar{X} - \mu}{\hat{\sigma}/\sqrt{n}} \sim t_{n-1}$$

  Note that $Z \approx T$ when the sample size n is large enough.

- Additionally,

$$Z^2 = \left(\frac{\bar{X} - \mu}{\sigma/\sqrt{n}}\right)^2 \sim \chi^2_1$$

Example

```
set.seed(123456)
nrep <- 1e5         ## The number of data sets to generate
N <- 30             ## Sample size
mu <- 0; sig <- 1   ## mean and variance

D <- rnorm(nrep * N, mu, sig )
X <- array(D, c(nrep, N))
Xbar <- apply(X, 1, mean)
sighat <- apply(X, 1, sd)

Z <- (Xbar - mu)/(sig/sqrt(N))
T <- (Xbar - mu)/(sighat/sqrt(N))
C <- Z^2
```

```
par(mfrow=c(2,3))
## Sampling distribution of Z
hist(Z, nclass=100)
## Sampling distribution of T with df=n-1
hist(T, nclass=100)
## Sampling distribution of C with df=1
hist(C, nclass=100)

## Standard normal distribution
hist(rnorm(nrep), nclass=100, main="Normal")
## T distribution with df=n-1
hist(rt(nrep, N-1), nclass=100, main="T")
## Chi-square distribution with df=1
hist(rchisq(nrep,1), nclass=100, main="Chi-square")
```

```
## Significance level alpha
alpha <- c(0.1, 0.05, 0.01)

## Simulation and theoretical quantiles of
## standard normal distribution
quantile(Z, 1-alpha/2)
qnorm(1-alpha/2)

## Simulation and theoretical quantiles of
## T distribution with df=n-1
quantile(T, 1-alpha/2)
qt(1-alpha/2, N-1)

## Simulation and theoretical quantiles of
## Chi-square distribution with df=1
quantile(C, 1-alpha)
qchisq(1-alpha, 1)
```

```r
## Theoretical and simulation p-values of
## standard normal distribution
q <- qnorm(alpha/2)
pnorm(q)
c(mean(Z < q[1]), mean(Z < q[2]), mean(Z < q[3]))

## Theoretical and simulation p-values of
## T distribution with df=n-1
t <- qt(alpha/2, N-1)
pt(t, N-1)
c(mean(T < t[1]), mean(T < t[2]), mean(T < t[3]))

## Theoretical and simulation p-values of
## Chi-square distribution with df=1
c <- qchisq(alpha, 1)
pchisq(c, 1)
c(mean(C < c[1]), mean(C < c[2]), mean(C < c[3]))
```