Introductory Examples

Example 1 Sampling Computer Chips. *Suppose there are 100 memory chips in a box, of which 90 are "good" and 10 are "bad." We withdraw five of the 100 chips at random to upgrade a computer. What is the probability that all five chips are good?*

Answer is $P(X = 5) = \binom{90}{5} / \binom{100}{5} = 0.5838$. The command in R is

```
> choose(90,5)/choose(100,5)
```

We now simulate $m = 100000$ samples of five chips from the box. First, we regard the ten chips numbered 91 through 100 to be the bad ones.

```
> sample(1:100, 5)
[1] 46 85 68 59 81
> sample(1:100, 5)
[1] 17 43 36 99 84
```

Now try

```
> sample(1:100, 5, rep=T)
```

What do you see?

Beneath the program, we show the result of one run.

```
set.seed(1237)  #this seed for exact result shown
m = 100000 # number of samples to simulate
good = numeric(m) # initialize for use in loop


for (i in 1:m)
{
    pick = sample(1:100, 5)
                    # vector of 5 items from ith box
    good[i] = sum(pick <= 90) # number Good in ith box
}
mean(good == 5)  # approximates P{All Good}
```

```
> mean(good == 5)
[1] 0.58293
```

Example 2 Birthday Matches—Combinatorial Approach. *Suppose there are $n = 25$ people in a room. What is the probability that two or more of them have the same birthday?*

$$P(\text{No Match}) = \frac{(365!)/(365-25)!}{365^{25}} = \prod_{i=0}^{24}(1 - \frac{i}{365}) = 0.4313.$$

So $p = P(\text{At Least One Match}) = 1 - 0.4313 = 0.5687$. In R, $P(\text{No Match})$ can be evaluated as $\mathbf{prod}((365 : (365 - 24))/365)$ or as $\mathbf{prod}(1 - (0 : 24)/365)$.

Intuitively, it seems the probability p of getting a match must increase as the number n of people in the room increases.

```
n = 1:60                          # vector of room sizes
p = numeric(60)                   # initialize vector, all 0s
for (i in n)                      # index values for loop
{
    q = prod(1 - (0:(i-1))/365)   # P(No match) if i people in room
    p[i] = 1 - q                  # changes ith element of p
}
plot(n, p)                        # plot of p against n


p[c(22, 23, 60)]
```

Example 3 *Birthday Matches—Using Simulation.*

```
 > b = sample(1:365, 25, repl=T); b
[1] 74 251 335 104 39 256 193 295 350 41
[11] 100 180 117 205 96 74 142 325 203 308
[21] 325 264 78 83 52
```

In this simulated room of 25 people, there are two birthday matches. The people numbered 1 and 16 were both born on the 74th day of the year, and those numbered 18 and 21

were both born on the 325th day. We would also have said there are two birthday matches if person 18 had been born on the 74th day.

Thus **x = 25 - length(unique(b))** computes $25 - 23 = 2$. This is the number X of birthday matches (redundant birthdays) among the 25 people in the "room" simulated above. In a large poll, we anticipate that the fraction of rooms with no matches will be very nearly $P(\text{No Matches}) = P(X = 0) = 0.4313$,

```
set.seed(1237)
m = 100000; n = 25                  # iterations; people in room
x = numeric(m)                      # vector for numbers of matches
for (i in 1:m)
{
  b = sample(1:365, n, repl=T)      # n random birthdays in ith room
  x[i] = n - length(unique(b))      # no. of matches in ith room
}
mean(x == 0); mean(x)               # approximates P{X=0}; E(X)
cutp = (0:(max(x)+1)) - .5          # break points for histogram
hist(x, breaks=cutp, prob=T)        # relative freq. histogram
```

The parameter **prob=T** puts a density scale on the vertical axis. Thus we can see that the bar of the histogram at $x = 0$ is approximately 0.43 units high. It also seems reasonable that the balance point of the histogram is consistent with $E(X) = 0.81$. (We used the parameter **breaks=cutp** to specify breakpoints for the histogram.

Example 4 *Estimating the Probability that a Die Shows a Six.*

Suppose that n binomial trials with $\pi = P(\text{Success})$ result in X Successes. As an elementary illustration of confidence intervals made with formula $p \pm 1.96\sqrt{\frac{p(1-p)}{n}}$, where the point estimate $p = X/n$ of the parameter π.

Suppose 20 students in a class were each asked to roll a die 30 times. We note the number X of 6's observed and find the corresponding confidence interval.

Now we determine which of the 31 confidence intervals cover the value $\pi = 0.8$. The coverage probability is also computed. It is the sum of the probabilities corresponding to values of x that yield intervals covering π.

```
n = 30                                      # number of trials
x = 0:n; sp = x/n                           # n+1 possible outcomes
m.err = 1.96*sqrt(sp*(1-sp)/n)              # n+1 Margins of error
lcl = sp - m.err                            # n+1 Lower conf. limits
ucl = sp + m.err                            # n+1 Upper conf. limits
pp = .80                                    # pp = P(Success)
prob = dbinom(x, n, pp)                     # distribution vector
cover = (pp >= lcl) & (pp <= ucl)           # vector of 0s and 1s

> round(cbind(x, sp, lcl, ucl, prob, cover), 4)   # 4-place printout

x sp lcl ucl prob cover
...
[18,] 17 0.5667 0.3893 0.7440 0.0022 0
[19,] 18 0.6000 0.4247 0.7753 0.0064 0
[20,] 19 0.6333 0.4609 0.8058 0.0161 1
[21,] 20 0.6667 0.4980 0.8354 0.0355 1
[22,] 21 0.7000 0.5360 0.8640 0.0676 1
[23,] 22 0.7333 0.5751 0.8916 0.1106 1
[24,] 23 0.7667 0.6153 0.9180 0.1538 1
[25,] 24 0.8000 0.6569 0.9431 0.1795 1
[26,] 25 0.8333 0.7000 0.9667 0.1723 1
[27,] 26 0.8667 0.7450 0.9883 0.1325 1
[28,] 27 0.9000 0.7926 1.0074 0.0785 1
[29,] 28 0.9333 0.8441 1.0226 0.0337 0
[30,] 29 0.9667 0.9024 1.0309 0.0093 0
[31,] 30 1.0000 1.0000 1.0000 0.0012 0

> sum(dbinom(x[cover], n, pp))                  # total cov. prob. at pp
[1] 0.9463279
```

Thus the total coverage probability for $\pi = 0.30$ is

$$\begin{aligned} P(\text{Cover}) &= P(X = 19) + P(X = 20) + \cdots + P(X = 27) \\ &= 0.0161 + 0.0355 + 0.0676 + \cdots + 0.0785 = 0.9463. \end{aligned}$$

Example 5 *Two Thousand Coverage Probabilities.*

To get a more comprehensive view of the performance of confidence intervals, we step through two thousand values of π from near 0 to near 1. For each value of π, we go through a procedure like that shown in Example above. Finally, we plot the coverage probabilities against π.

```
n = 30                                      # number of trials
alpha = .05; k = qnorm(1-alpha/2)           # conf level = 1-alpha
adj = 0                                     # (2 for Agresti-Coull)
x = 0:n; sp = (x + adj)/(n + 2*adj)         # vectors of
m.err = k*sqrt(sp*(1 - sp)/(n + 2*adj))     # length
lcl = sp - m.err                            # n + 1
ucl = sp + m.err                            #
m = 2000                                    # no. of values of pp
pp = seq(1/n, 1 - 1/n, length=m)            # vectors
p.cov = numeric(m)                          # of length m
for (i in 1:m)                              # loop (values of pp)
{                                           # for each pp:
 cover = (pp[i] >= lcl) & (pp[i] <= ucl) # 1 if cover, else 0
 p.rel = dbinom(x[cover], n, pp[i])      # relevant probs.
 p.cov[i] = sum(p.rel)                   # total coverage prob.
}
plot(pp, p.cov, type="l", ylim=c(1-4*alpha,1))
lines(c(.01,.99), c(1-alpha,1-alpha))
```

It is clear from the resulting plot that it is not unusual for the coverage probabilities of intervals to vary rapidly as π varies in $(0, 1)$. More regrettably, the true coverage probabilities are often much lower than the claimed 95%. Furthermore, this tendency for coverage probabilities to be too low persists even for moderately large n

These plots show that, for values of π and n frequently encountered in practice, the traditional confidence intervals cannot be relied upon to provide the promised level of confidence unless π is close to 1/2.

Note that if we substitute **adj = 2**, the program shows coverage probabilities for 95% Agresti-Coull confidence intervals when $n = 30$.

Problems

1. The random variable X of Example 1 has a hypergeometric distribution. In R, the hypergeometric probabilities $P(X = x)$ can be computed using the function **dhyper**. Its parameters are, in order, the number x of good items seen in the sample, the number of good items in the population, the number of bad items in the population, and the number of items selected without replacement. Thus each of the statements **dhyper(5, 90, 10, 5)** and **dhyper(0, 10, 90, 5)** returns 0.5837524.

   (a) What is the relationship between **sample(1:100, 5)** and **choose(100, 5)**?

   (b) Compute $P(X = 2)$ using **dhyper** and then again using **choose**.

   (c) Run the program of Example 1 followed by the statements **mean(good)** and **var(good)**. What are the numerical results of these two statements? In terms of the random variable X, what do they approximate and how good is the agreement?

   (d) Execute **sum((0:5)*dhyper(0:5, 90, 10, 5))**? How many terms are being summed? What numerical result is returned? What is its connection with part (c)?

   Notes: If n items are drawn at random without replacement from a box with b Bad items, g Good items, and $T = g + b$, then $E(X) = ng/T$, $V(X) = n(\frac{g}{T})(1 - \frac{g}{T})(\frac{T-n}{T-1})$.

2. **Item matching**. There are ten letters and ten envelopes, a proper one for each letter. A very tired administrative assistant puts the letters into the envelopes at random. We seek the probability that no letter is put into its proper envelope and the expected number of letters put into their proper envelopes. Explain, statement by statement, how the program below approximates these quantities by simulation. Run the program with $n = 10$, then with $n = 5$, and again with $n = 20$. Report and compare the results.

```
m = 100000; n = 10; x = numeric(m)
for (i in 1:m) {perm = sample(1:n, n); x[i] = sum(1:n==perm)}
cutp = (-1:n) + .5; hist(x, breaks=cutp, prob=T)
mean(x == 0); mean(x); sd(x)
```

Notes: Let X be the number correct. For n envelopes, a combinatorial argument gives $P(X = 0) = 1/2! - 1/3! + \cdots + (-1)^n/n!$. In R,

```
i = 0:10;
sum((-1)^i/factorial(i)).
```

For any $n > 1$, $P(X = n - 1) = 0$, $P(X > n) = 0$, $E(X) = 1$, and $V(X) = 1$. For large n, X is approximately $POIS(1)$. Even for n as small as 10, this approximation is good to two places; to verify this, run the program above, followed by **points(0:10, dpois(0:10, 1))**.

3. A poker hand consists of five cards selected at random from a deck of 52 cards. (There are four Aces in the deck.)

   (a) Use combinatorial methods to express the probability that a poker hand has no Aces. Use R to find the numerical answer correct to five places.

   (b) Modify the program of Example 1 to approximate the probability in part (a) by simulation.

4. If X is BINOM$(n; \pi)$, then one can show that its mode is $[(n+1)\pi]$; that is, the greatest integer in $(n+1)\pi$. Except that if $(n+1)\pi$ is an integer, then there is a "double mode": values $k = (n + 1)\pi$ and $(n + 1)\pi + 1$ have the same probability. Run the following program for $n = 6$ and $\pi = 1/5$ (as shown); for $n = 7$ and $\pi = 1/2$; and for $n = 18$ and $\pi = 1/3$. Explain the code and interpret the answers in terms of the facts stated above about binomial random variables. (If necessary, use **?dbinom** to get explanations of **dbinom, pbinom**, and **rbinom**.)

```
n = 6; pp = 1/5; k = 0:n
pdf = dbinom(k, n, pp); sum(k*pdf)
cdf = pbinom(k, n, pp); sum(1 - cdf)
```

```
mean(rbinom(100000, n, pp))
n*pp; round(cbind(k, pdf, cumsum(pdf), cdf), 4)
k[pdf==max(pdf)]; floor((n+1)*pp)
```

5. In the R program of Example 5, set **adj = 2** and leave $n = 30$. This adjustment implements the Agresti-Coull type of 95% confidence interval. The formula is similar to traditional one, except that one begins by "adding two successes and two failures" to the data. [Example: If we see 20 Successes in 30 trials, the 95% Agresti-Coull interval is centered at $22/34 = 0.6471$ with margin of error $1.96\sqrt{(22)(12)/34^3} = 0.1606$, and the interval is $(0.4864, 0.8077)$.]

   Run the modified program, and compare your plot with Figures 1.6 (p12) and 1.8 (p21) (Suess). For what values of π are such intervals too "conservative"—too long and with coverage probabilities far above 95%? Also make plots for 90% and 99% and comment. (See Problem 1.17 (Suess) for more on this type of interval.)

6. **Average lengths of confidence intervals.** For given n and π the length of a confidence interval is a random variable because the margin of error depends on the number of Successes observed. The program below illustrates the computation and finds the expected length.

```
n = 30; pp = .2                                  # binomial parameters
alpha = .05; kappa = qnorm(1-alpha/2)          # level is 1 - alpha
adj = 0                     # 0 for traditional; 2 for Agresti-Coull
x = 0:n; sp = (x + adj)/(n + 2*adj)
CI.len = 2*kappa*sqrt(sp*(1 - sp)/(n + 2*adj))
Prob = dbinom(x, n, pp); Prod = CI.len*Prob
round(cbind(x, CI.len, Prob, Prod), 4)        # displays computation
sum(Prod)                                          # expected length
```

   (a) Explain each statement in this program, and state the length of each named vector. (Consider a constant as a vector of length 1.)

   (b) Run the program as it is to find the average length of intervals based on traditional one when $\pi = 0.1, 0.2$, and 0.5. Then use **adj = 2** to do the same for Agresti-Coull intervals.

(c) Figure 1.9 was made by looping through about 200 values of π? Use it to verify your answers in part (b). Compare the lengths of the two kinds of confidence intervals and explain.

(d) Write a program to make a plot similar to Figure 1.9. Use the program of Example 4 as a rough guide to the structure. You can use **plot** for the first curve and **lines** to overlay the second curve.

Note: This program includes the entire length of any CI extending outside $(0, 1)$.