Project 2: The Number of Different Faces Seen in \(n\) Rolls of a Die

*Six rolls of a fair die.* A fair die is rolled \(n = 6\) times. What is the probability all six faces are seen? What is the distribution of the number \(X\) of faces seen?

The probability that all six faces are seen is \(P\{X = 6\} = \frac{6!}{6^6}\). The R code \(\text{prod}(\{6:1\}/6)\) returns 0.0154321. Also, \(P\{X = 1\} = \frac{6}{6^6} = \frac{1}{6^5} = 0.0001286\).

The entire distribution of \(X\) is not so easily computed by combinatorial methods, but it is easily simulated. The R code `sample(1:6, 6, repl=T)` simulates six rolls of a fair die. Here we use the parameter `repl=T` to indicate that, in repeated "drawing" from the population of elements \{1, 2, 3, 4, 5, 6\}, an element may be drawn more than once. As in the birthday problem in the main paper, the `unique` function can then be used to count the number of distinct results, and the length of the resulting vector is the number \(X\) of faces seen. Four simulations of the 6-roll experiment gave the results shown below.

```r
rolls <- sample(1:6, 6, repl=T); rolls
faces <- unique(rolls); faces
length(faces)
```

```r
[1] 3 6 6 1 3 6
> faces <- unique(rolls); faces
[1] 3 6 1
> length(faces)
[1] 3
```

```r
rolls <- sample(1:6, 6, repl=T); rolls
faces <- unique(rolls); faces
length(faces)
```

```r
[1] 2 4 5 5 4 1
> faces <- unique(rolls); faces
[1] 2 4 5 1
> length(faces)
[1] 4
```

```r
rolls <- sample(1:6, 6, repl=T); rolls
faces <- unique(rolls); faces
length(faces)
```

```r
[1] 6 4 3 5 2 3
> faces <- unique(rolls); faces
[1] 6 4 3 5 2
> length(faces)
[1] 5
```

```r
rolls <- sample(1:6, 6, repl=T); rolls
faces <- unique(rolls); faces
length(faces)
```

```r
[1] 1 2 1 6 2 3
> faces <- unique(rolls); faces
[1] 1 2 6 3
> length(faces)
[1] 4
```
If we find the realizations of $X$ in $m = 40000$ repetitions of the 6-roll experiment, then we can use the sampling distribution as a reasonable approximation of the theoretical distribution of $X$.

```r
m <- 40000
n <- 6
x <- numeric(m)
for (i in 1:m)
{
  x[i] <- length(unique(sample(1:6, n, repl=T)))
}

hist(x, breaks=.5:6.5, prob=T)
mean(x==6)  # Simulates P{X=6}
mean(x)     # Simulates E(X)
summary(as.factor(x))
summary(as.factor(x))/m

> mean(x==6)  # Simulates P{X=6}
[1] 0.0148
> mean(x)     # Simulates E(X)
[1] 3.9866
> summary(as.factor(x))
          1   2   3   4   5   6
   6 754 9408 20026 9214 592
> summary(as.factor(x))/m
       1   2   3   4   5   6
0.00015 0.01885 0.23520 0.50065 0.23035 0.01480
```

Notice that $P\{X = 6\}$ is approximated as 0.01480, which is reasonably near the exact answer 0.0154321 obtained by our earlier combinatorial computation. Two or three place accuracy can be expected.
Ten rolls of a fair die. For ten rolls of a fair die the combinatorial approach becomes messy, but simulation is no more difficult than for six rolls. Changing \( n \) to 10 in the code above we obtained the following results for one run.

\[
> \text{mean}(x==6) \quad \# \text{Simulates } P\{X=6\}
\]

\[ \begin{array}{c}
1 \quad 0.2726 \\
\end{array} \]

\[
> \text{mean}(x) \quad \# \text{Simulates } E(X)
\]

\[ \begin{array}{c}
1 \quad 5.03025 \\
\end{array} \]

\[
> \text{summary(as.factor(x))}
\]

\[ \begin{array}{cccccc}
2 & 3 & 4 & 5 & 6 \\
7 & 754 & 8165 & 20170 & 10904 \\
\end{array} \]

\[
> \text{summary(as.factor(x))/m}
\]

\[ \begin{array}{cccccc}
2 & 3 & 4 & 5 & 6 \\
0.000175 & 0.018850 & 0.204125 & 0.504250 & 0.272600 \\
\end{array} \]

It is not surprising that with more rolls we are more likely to see all six faces of the die.

Six rolls of an unfair die. Now suppose the die is biased against 1s in favor of 6s, specifically let \( P(1) = 1/12, \ P(2) = P(3) = P(4) = P(5) = 1/6 \) and \( P(6) = 3/12 \). Here it is not difficult to see that \( P\{X = 6\} = 6!(1/12)(1/6)^4(3/12) = 0.01157407 \). Notice that it is less likely to see all six faces in six rolls of this die than for the fair die (recall the result 0.0154321 from above for a fair die). This is because it is harder to get 1s to complete the "set" of six faces. The probability of seeing only one face is \( P\{X = 1\} = (1/12)^6 + 4(1/6)^6 + (3/12)^6 = 0.0003302 \), somewhat larger than for a fair die because of the increased probability of getting all 6s.
As for the birthday problem in the main paper, in order to simulate this situation, we introduce a weighting vector \( w <- c(1, 2, 2, 2, 2, 3)/12 \) to control the probabilities of choosing the various faces on each roll of the die. (Strictly speaking, the division by 12 is unnecessary because R automatically adjusts a weight vector so that its components sum to unity.) Thus the R code to find the distribution of \( X \) is changed slightly as shown below.

\[
\begin{align*}
m &<- 40000 \\
n &<- 6 \\
w &<- c(1, 2, 2, 2, 2, 3)/12 \\
x &<- \text{numeric}(m) \\
\text{for} \ (i \ \text{in} \ 1:m) \\
\quad \{ \\
\quad \quad x[i] \leftarrow \text{length(unique(sample(1:6, n, prob=w, repl=T)))} \\
\quad \}
\end{align*}
\]

\[
\text{hist(x, breaks=.5:6.5, prob=T)} \\
\text{mean(x==6)} \quad \# \text{Simulates } P\{X=6\} \\
\text{mean(x)} \quad \# \text{Simulates } E(X) \\
\text{summary(as.factor(x))} \\
\text{summary(as.factor(x))/m}
\]

Results from one run were as follows.

\[
\begin{align*}
> \text{mean(x==6)} \quad \# \text{Simulates } P\{X=6\} \\
[1] \ 0.0111 \\
> \text{mean(x)} \quad \# \text{Simulates } E(X) \\
[1] \ 3.8869 \\
> \text{summary(as.factor(x))} \\
\quad \begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
9 & 1219 & 10835 & 19605 & 7888 & 444
\end{array} \\
> \text{summary(as.factor(x))/m} \\
\quad \begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
0.000225 & 0.030475 & 0.270875 & 0.490125 & 0.197200 & 0.011100
\end{array}
\end{align*}
\]

Notice that \( P\{X = 6\} = 0.01157407 \) is reasonably well simulated as 0.011100. The 2-to-3-place accuracy of simulations yielded by \( m = 40000 \) iterations is adequate to demonstrate the difference in the probability of seeing all six faces for this unfair die as compared with a fair die. With slightly modified code, we ran simulations for fair and unfair dice again and plotted the resulting histograms side by side (see the top of the next page).

**Exercises:**
(a) The R code for ten rolls of this unfair die was given in the main paper, and we leave it to you to do a run and to compare the results with those for ten rolls of a *fair* die.

(b) Explore six and ten rolls of a die severely biased so that \( P(1) = 1/2 \) and the other faces are equally likely.
Varying the number of rolls of a fair die. As the number of rolls of a fair die increases, we expect that \( P\{X = 6\} \) and \( E(X) \) will both increase. The idea is similar to Figure 1 of the main paper, but here each point is to be found by simulation. The implementation, in terms of simulation, is somewhat in the spirit of Figure 4 and Display 5 of the main paper.

\[
m \gets 10000\\
x \gets \text{numeric}(m)\\
mn \gets \text{numeric}(30) \quad \# \text{ vector of } E(X)\\
pr \gets \text{numeric}(30) \quad \# \text{ vector of } P(X=6)\\
\text{for (n in 1:30) } \quad \# \text{ loop for number or rolls}\\
\text{\{ for (i in 1:m) } \quad \# \text{ inner loop simulating realizations of X}\\
\text{\{ x[i] <- length(unique(sample(1:6, n, repl=T))) \}}\\
mn[n] <- \text{mean}(x)\\
pr[n] <- \text{mean}(x==6)\\
\text{\}}
\]

\text{par(mfrow=c(1,2)) } \quad \# \text{ Puts two plots side by side}\\
\text{plot(1:30, mn, xlab="Rolls of Die", ylab="E(X)", ylim=c(1,6),}\\
\text{\quad main = "E(Number of Faces Seen)"})\\
\text{plot(1:30, pr, xlab="Rolls of Die", ylab="P(X=6)", ylim=c(0,1),}\\
\text{\quad main = "P(Seeing All 6 Faces)"})}
\text{par(mfrow=c(1,1)) } \quad \# \text{ Returns to default plotting format (important!)}
A larger value of $m$ would have given a smoother plot at the upper right of the graph for $P\{X = 6\}$. Of course, $P\{X = 6\} = 0$ for fewer than $n = 6$ rolls.

**Advanced Exercise:** For six rolls of a biased die plot $E(X)$ and $P(X = 6)$ against the "degree of bias." One easily programmed model for degree of bias is that $P(1)$ takes values $0, 1/24, 2/24, 3/24, \ldots, 1$ — with the other five faces equally likely. (When $P(1) = 4/24 = 1/6$, the die is fair.)

As your instructor requests, you might do the same exploration for ten rolls of a biased die and/or explore alternate models for the degree of bias. (One additional model might be to let $P(1) = 1/6, 1/8, 1/10, 1/12, \ldots, 1/36$, $P(2) = P(3) = P(4) = P(5) = 1/6$, and let $P(6)$ take the value required for unit total.) We suggest you use $m = 500$ in debugging your code; then switch to a suitably large value for the serious runs.