Random Shuffles

Here is a suprising claim. If you shuffle a standard deck of cards seven times, with almost total certainty you can claim that the exact ordering of cards has never been seen! Wow! Let's explore. We can ask this question formally as: What is the probability that in the n shuffles seen since the start of time, yours is unique?

Orderings of 52 Cards

Our adventure starts with a simple observation: there are **very** many ways to order 52 cards. But exactly how many unique orderings of a standard deck are there?

There are 52! ways to order a standard deck of cards. Since each card is unique (each card has a unique suit, value combination) then we can apply the rule for Permutations of Distinct Objects:

$$\text{Num. Unique Orderings} = 52!$$

That is a humongous number. 52! equals
80658175170943878571660636856403766975289505440883277824000000000000.
That is over $8 \cdot 10^{67}$. Recall it is estimated that there are around 10^{82} atoms in the observable universe

Number of Shuffles Ever Seen

Of course we don't know what the value of n is — nobody has been counting how many times humans have shuffled cards. We can come up with a reasonable overestimate. Assume k = 7 billion people have been shuffling cards once a second since cards were invented. Playing cards may have been invented as far back as the Tang dynasty in the 9th century. To the best of my knowledge the oldest set of 52 cards is the Topkapı deck of cards in Istanbul around the 15th century ad. That is about s = 16,472,828,422 seconds ago. As such our overestimate is $n = s \cdot k \approx 10^{20}$.

Next let's calculate the probability that none of those n historical shuffles matches your particular ordering of 52 cards. There are two valid approaches: using equally likely outcomes, and using independence.

Equally Likely Outcomes

One way to the probability that your ordering of 52 cards is unique in history is to use Equally Likely Outcomes. Consider the sample space of all the possible ordering of all the cards ever dealt. Each outcome in this set will have n card decks each with their own ordering. As such the size of the sample space is $|S| = (52!)^n$. Note that all outcomes in the sample space are equally likely — we can convince ourselves of this by symmetry — no ordering is more likely than any other. Out of that sample space we want to count the number of outcomes where none of the orderings matches yours. There are $52! - 1$ ways to order 52 cards that are not yours. We can construct the event space by steps: for each of the n shuffles in history select any one of those $52! - 1$ orderings. Thus $|E| = (52! - 1)^n$.

Let U be the event that your particular ordering of 52 cards is unique

$$\begin{aligned}
\mathrm{P}(U) &= \frac{|E|}{|S|} && \text{Equally Likely Outcomes} \\
&= \frac{(52! - 1)^n}{(52!)^n} \\
&= \frac{(52! - 1)^{10^{20}}}{(52!)^{10^{20}}} && n = 10^{20} \\
&= \left(\frac{52! - 1}{52!}\right)^{10^{20}}
\end{aligned}$$

In theory that is the correct answer, but those numbers are so big, its not clear how to evaluate it, even when using a computer. One good idea is to first compute the log probability:

$$\begin{aligned}
\log \mathrm{P}(U) &= \log\left[\left(\frac{52!-1}{52!}\right)^{10^{20}}\right] \\
&= 10^{20} \cdot \log\left(\frac{52!-1}{52!}\right) \\
&= 10^{20} \cdot \Big[\log(52!-1) - \log(52!)\Big] \\
&= 10^{20} \cdot (-1.24 \times 10^{-68}) \\
&= -1.24 \times 10^{-48}
\end{aligned}$$

Now if we undo the log (and use the fact that e^{-x} is very close to $1-x$ for small values of x):

$$\begin{aligned}
\mathrm{P}(U) &= e^{-1.24 \times 10^{-48}} \\
&\approx 1 - 1.24 \times 10^{-48}
\end{aligned}$$

So the probability that your particular ordering is unique is very close to 1, and the probability that someone else got the same ordering, $1 - P(U)$, is a number with 47 zeros after the decimal point. It is safe to say your ordering is unique.

In python, you can use a special library called decimal to compute very small probabilities. Here is an example of how to compute $\log \frac{52!-1}{52!}$:

```
from decimal import *
import math

n = math.pow(10, 20)
card_perms = math.factorial(52)
denominator = card_perms
numerator = card_perms - 1

# decimal library because these are tiny numbers
getcontext().prec = 100 # increase precision
log_numer = Decimal(numerator).ln()
log_denom = Decimal(denominator).ln()
log_pr = log_numer - log_denom

# approximately -1.24E-68
print(log_pr)
```

We can also check our result using the binomial approximation.

For small values of x, the value $(1-x)^n$ is very close to $1 - nx$, and this gives us another way to compute $P(U)$:

$$\begin{aligned}
\mathrm{P}(U) &= \frac{(52!-1)^n}{(52!)^n} \\
&= \left(1 - \frac{1}{52!}\right)^{10^{20}} \qquad\qquad n = 10^{20} \\
&\approx 1 - \frac{10^{20}}{52!} \\
&\approx 1 - 1.24 \times 10^{-48}
\end{aligned}$$

This agrees with the result we got using python's decimal library.

Independence

Another approach is to define events D_i that the ith card shuffle is different than yours. Because we assume each shuffle is independent, then $P(U) = \prod_i P(D_i)$. What is the probability of (D_i)? If you think of the sample space of D_i, it is 52! ways of ordering a deck cards. The event space is the 52! - 1 outcomes which are not your ordering.

$$
\begin{aligned}
\mathrm{P}(U) &= \prod_{i=1}^{n} \mathrm{P}(D_i) \\
\log \mathrm{P}(U) &= \sum_{i=1}^{n} \log \mathrm{P}(D_i) \\
&= n \cdot \log \mathrm{P}(D_i) \\
&= 10^{20} \cdot \log \frac{52! - 1}{52!} \\
&\approx 10^{20} \cdot -1.24 \times 10^{-68}
\end{aligned}
$$

Which is the same answer we got with the other approach for $\log \mathrm{P}(U)$

How Random is your Shuffle?

A final question we can look into. How do you get a truly random ordering of cards? Dave Bayer and Persi Diaconis in 1992 worked through this problem and published their results in the article Trailing the Dovetail Shuffle to its Lair. They showed that if you shuffle a deck of cards seven times using a riffle shuffle also known as the dovetail shuffle, you are almost garunteed a random ordering of cards. The methodology used paved the way for studying psuedo random numbers produced by computers.