Simulating Conditional Probabilities

Recall, one interpretation of *probability* is long-term relative frequency. Suppose you repeat a random experiment a large number of time, say N, and observe n number of event A The relative frequency

$$\frac{n}{N}$$

is approximately the probability $P(A)$.

Example: Rolling an 8-Sided Fair Die Twice

$$A = \text{sum of the throws is divisible by 4}$$

$$B = \text{the two throws are the same}$$

Recall we can generate N simulations of a single roll with:

```
roll <- sample(x = 1:8, size = N, replace = TRUE)
```

We can use 2 separate `sample` calls to mimic 2 rolls and then combine them using `cbind`.

Code | Start Over | Run Code

```
N <- 10 # number of simulations
roll1 <- sample(1:8, N, replace = TRUE)
roll2 <- sample(1:8, N, replace = TRUE)
# use `cbind` to combine them into a matrix
tworolls <- cbind(roll1, roll2)
# print(tworolls)
print(roll1)
```

```
[1] 2 5 8 1 5 8 2 8 1 7
```


```
[1] 2 5 8 1 5 8 2 8 1 7
```


Using the `N=1,000,000`, compute $P(A)$, $P(B)$, $P(A \cap B)$, $P(A|B)$, and $P(B|A)$.

$$P(A)$$

Code | Start Over | Run Code

```
N <- 10^6 # number of simulations
roll1 <- sample(1:8, N, replace = TRUE)
roll2 <- sample(1:8, N, replace = TRUE)
# use `cbind` to combine them into a matrix
# tworolls <- cbind(roll1, roll2)
# A: sum is divisible by 4
# check whether the modulus is 0
divisible4 <- ((roll1 + roll2) %% 4) == 0
cat("sim:", sum(divisible4)/N) # relative frequency (by simulation)
## check by counting
```

```
# table of outcomes
m <- matrix(rep(1:8, each = 8), nrow = 8) + (1:8)
cat("theory:", sum(m %% 4 == 0)/64) # theoretical probability by counting

```

```
sim: 0.249599
```

```
theory: 0.25
```

$$P(B)$$

Code Start Over Run Code

```
N <- 10^6 # number of simulations
roll1 <- sample(1:8, N, replace = TRUE)
roll2 <- sample(1:8, N, replace = TRUE)
# use `cbind` to combine them into a matrix
tworolls <- cbind(roll1, roll2)
```

$$P(A \cap B)$$

Code Start Over Run Code

```
N <- 10^6 # number of simulations
roll1 <- sample(1:8, N, replace = TRUE)
roll2 <- sample(1:8, N, replace = TRUE)
# use `cbind` to combine them into a matrix
tworolls <- cbind(roll1, roll2)
```

$$P(A|B)$$

Code Start Over Run Code

```
N <- 10^6 # number of simulations
roll1 <- sample(1:8, N, replace = TRUE)
```

```
roll2 <- sample(1:8, N, replace = TRUE)
# use `cbind` to combine them into a matrix
tworolls <- cbind(roll1, roll2)
# P(A and B)/P(B)
# select those that satisfy B
tworolls_givenb <- tworolls[roll1 == roll2,]
n_aandb_givenb <- sum(
  (tworolls_givenb[,1] + tworolls_givenb[,2]) %% 4 == 0)
print(n_aandb_givenb / sum(roll1 == roll2))
print(n_aandb_givenb / nrow(tworolls_givenb))

```

```
[1] 0.499752
```

```
[1] 0.499752
```

$$P(B|A)$$

Code Start Over Run Code

```
N <- 10^6 # number of simulations
roll1 <- sample(1:8, N, replace = TRUE)
roll2 <- sample(1:8, N, replace = TRUE)
# use `cbind` to combine them into a matrix
tworolls <- cbind(roll1, roll2)
```

Next Topic