PROBABILITY PUZZLES IN R

Yahtzee

Peter Chi
Assistant Professor of Statistics
Villanova University

Yahtzee scoring

Multiplication Rule

- k independent processes
- n_i possibilities for each

Total number of possibilities:

$$n_1 \times n_2 \times \ldots \times n_k$$

Example. Roll three dice. Total number of configurations:

$$6 \times 6 \times 6 = 6^3$$

```
6^3
 [1] 216
```

Permutations

- k objects
- n total possibilities
- Each possibility used once at most

Total number of configurations:

$$n \times (n-1) \times ... \times (n-k+1) = \frac{n!}{(n-k)!}$$

Example. Number of ways for three dice to land as {2,3,4}:

$$3 \times 2 \times 1 = \frac{3!}{(3-3)!} = 3!$$

```
factorial(3)
 [1] 6
```

Addition Rule

Given disjoint events A and B::

$$P(A \cup B) = P(A) + P(B)$$

Example 1. Probability of rolling {2,3,4} or {3,4,5} with three dice

```
factorial(3)/6^3 + factorial(3)/6^3
 [1] 0.05555556
```

Example 2. Probability of rolling three dice landing on the same denomination

```
1/6^3 + 1/6^3 + 1/6^3 + 1/6^3 + 1/6^3 + 1/6^3
 [1] 0.02777778
```

Combinations

- n total objects
- Choose k of them; order does not matter

Total number of ways:

$$\binom{n}{k} = \frac{n!}{k! \times (n-k)!}$$

Example. Number of ways to choose 2 dice out of 3:

$$\binom{3}{2} = \frac{3!}{2! \times (3-2)!} = 3$$

```
choose(3,2)
 [1] 3
```

Combining rules

Example. Roll 10 dice

Number of ways to roll 5 of one denomination and 5 of another:

```
n_denom <- factorial(6) / factorial(4)
```

```
n_groupings <- choose(10,5) * choose(5,5)
```

```
n_total <- n_denom * n_groupings
n_total
  [1] 7560
```

PROBABILITY PUZZLES IN R

Let's calculate it!

PROBABILITY PUZZLES IN R

Settlers of Catan

Peter Chi
Assistant Professor of Statistics
Villanova University

Introduction to the game

Simulating dice rolls

```
roll_dice <- function(k){
  all_rolls <- sample(c(1,2,3,4,5,6),
                      k,
                      replace = TRUE)
  final_answer <- sum(all_rolls)
  return(final_answer)
}
```

```
roll_dice(2)
 [1] 7
```

```
replicate(10, roll_dice(2))
  [1]  8 10 10  2 11  5  4  6 11  7
```

The table function

```
rolls <- replicate(10, roll_dice(2))
rolls
  [1]  8 10 10  2 11  5  4  6 11  7
```

```
table(rolls)
rolls
 2  4  5  6  7  8 10 11
 1  1  1  1  1  1  2  2
```

Counting the number of occurrences

```
rolls <- replicate(100, roll_dice(1))
sum(rolls == 3)
  [1] 22
```

```
if(sum(rolls == 3) > 17){
  print("The value of 3 was rolled more than 17 times")
}

  [1] "The value of 3 was rolled more than 17 times"
```

```
if(sum(rolls == 3) > 17 | sum(rolls == 4) > 17){
  print("The value of 3 or 4 was rolled more than 17 times")
}

  [1] "The value of 3 or 4 was rolled more than 17 times"
```

PROBABILITY PUZZLES IN R

Let's try it!

PROBABILITY PUZZLES IN R

Craps

Peter Chi
Assistant Professor of Statistics
Villanova University

Introduction to craps

- Pass line bet - wager made at beginning of a round of play
- Shooter rolls first

The first roll:

- 7 or 11: win bet
- 2, 3, 12: lose bet
- Any other value establishes point

When a point is established

- Shooter's first roll: 5 = point
- Shooter continues to roll
- If 5 rolled before 7 - pass line bet won
- If 7 rolled before 5 - pass line bet lost
- Shooter continues to roll until 5 or 7

While loop

```
roll <- 0

while(roll != 6){
  roll <- roll_dice(1)
  print(roll)
}
 [1] 5
 [1] 2
 [1] 5
 [1] 6
```

Compound condition in a while loop

```
roll <- 0

while( (roll != 6) & (roll != 5) ){
  roll <- roll_dice(1)
  print(roll)
}
 [1] 2
 [1] 4
 [1] 5
```

The %in% operator

```
roll <- roll_dice(1)
if(roll %in% c(2,4,6) ){
  print("The roll is even")
}
 [1] "The roll is even"

roll
 [1] 2
```

PROBABILITY PUZZLES IN R

Let's play some Craps!