PROBABILITY PUZZLES IN R

Introduction to the Course

Peter Chi
Assistant Professor of Statistics
Villanova University

Course overview

- Chapter 1: The Classics
  - The Birthday Problem
  - Monty Hall
- Chapter 2: Games with Dice
  - Yahtzee
  - Settlers of Catan
  - Craps
- Chapter 3: Inspired by the Web
  - iPhone Passcode Combinations
  - Sign Error Cancellation
  - Factoring a Quadratic
- Chapter 4: Poker Games
  - Texas Hold'em Hole Cards
  - Consecutive Cashes in the WSOP
  - von Neumann Model of Poker

Combinatorics

- `factorial(n)`
  - `factorial(3)` $= 3! = 3 \times 2 \times 1$

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

- `choose(n,k)`
  - `choose(5,3)` $= \binom{5}{3} = \frac{5!}{3! \times (5-3)!}$

```
choose(5,3)
 [1] 10
```

Simulations

- Select an object at random - `sample()`
- Simulate a coinflip - `rbinom()`
- Repeat a process
  - `replicate()`
  - Loops: `for`, `while`
- Set a seed - `set.seed()`

More details on for loops

```
for(i in 1:10){
  sum(sample(x = c(1,2,3,4,5,6), size = 2, replace = TRUE))
}
```

Storing the results:

```
rolls <- rep(NA, 10)
for(i in 1:10){
  rolls[i] <- sum(sample(x = c(1,2,3,4,5,6), size = 2, replace = TRUE))
}
```

```
rolls
 [1]  3  6  3  9  9  3  6 11  9 10
```

Functions

```
my_function <- function(n){
  answer <- n^3
  return(answer)
}
```

```
my_function(10)
[1] 1000
```

PROBABILITY PUZZLES IN R

Let's practice!

PROBABILITY PUZZLES IN R

The Birthday Problem

Peter Chi
Assistant Professor of Statistics
Villanova University

Problem overview

Setup

- Room with n people in it
- What is the probability that anyone shares the same birthday?

Assumptions

- Ignore February 29th
- Birthdays are uniformly distributed across the remaining 365 days
- Each individual in the room is independent

Defining a counter

Simulating the probability of rolling a 12

```
counter <- 0
```

```
roll <- roll_dice(2)
roll
 [1] 12
```

```
if(roll == 12){
  counter <- counter + 1
}
```

```
counter
 [1] 1
```

```
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)
}
```

Incrementing a counter in a loop

Simulating the probability of rolling a 12

```
counter <- 0

for(i in 1:10000){
  roll <- roll_dice(2)
  if(roll == 12){
    counter <- counter + 1
  }
}

p_twelve <- counter / 10000
```

```
print(p_twelve)
 [1] 0.0278
```

```
1/36
 [1] 0.02777778
```

The pbirthday function

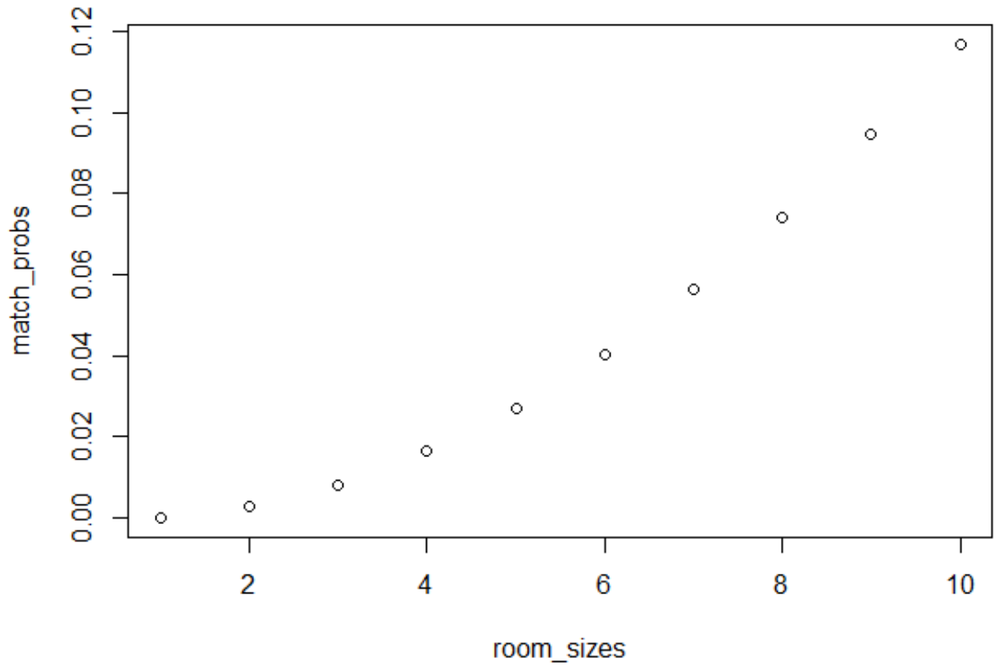
```
pbirthday(10)
  [1] 0.1169482
```

```
room_sizes <- c(1:10)
match_probs <- sapply(room_sizes, pbirthday)

print(match_probs)
 [1] 0.000000000 0.002739726 0.008204166 0.016355912 0.027135574
 [6] 0.040462484 0.056235703 0.074335292 0.094623834 0.116948178
```

Plotting

```
plot(match_probs ~ room_size)
```



PROBABILITY PUZZLES IN R

Let's solve it!

PROBABILITY PUZZLES IN R

Monty Hall

Peter Chi
Assistant Professor of Statistics
Villanova University

Choose one of the doors

One door is revealed

Revealing a door with reverse indexing

The `doors` object:

```
doors
[1] 1 2 3
```

Suppose that Door #1 is chosen, and Door #2 contains the prize...

Revealing the remaining door:

```
reveal <- doors[-c(1,2)]
reveal
 [1] 3
```

Revealing a door at random

The `doors` object:

```
doors
[1] 1 2 3
```

Suppose that Door #1 is chosen, and Door #1 in fact contains the prize...

Revealing the remaining door:

```
reveal <- sample(x = doors[-1], size = 1)
```

PROBABILITY PUZZLES IN R

Let's try it!