PROBABILITY PUZZLES IN R

Texas Hold'em

Peter Chi
Assistant Professor of Statistics
Villanova University

Community Cards in Texas Hold'em

Two cards left to come

Outs: Cards that improve hand from losing to winning

Complement Rule: $P(A) = 1 - P(A^c)$

```
p_lose <- choose(10-3,2) / choose(10,2)
p_win <- 1 - p_lose
p_win
 [1] 0.5333333
```

Calculating for different outs simultaneously

Using `choose` on a vector:

```
outs <- c(0,1,2,3)

p_lose <- choose(10-outs,2) / choose(10,2)
p_win <- 1 - p_lose
p_win
 [1] 0.0000000 0.2000000 0.3777778 0.5333333
```

Expected values

$$E(X) = \sum_{\text{all values}} x \cdot P(X = x)$$

Coin flip wager:

```
probs <- c(0.5, 0.5)
values <- c(-2, 3)
probs * values
 [1] -1  1.5
```

```
sum(probs * values)
 [1] 0.5
```

PROBABILITY PUZZLES IN R

Let's do this!

PROBABILITY PUZZLES IN R

Consecutive Cashes in the World Series of Poker

Peter Chi
Assistant Professor of Statistics
Villanova University

World Series of Poker

- ~ 6000 entrants in recent years
- Prize money awarded to top 10% - cashing

Poker Professional Ronnie Bardah:

- Cashed in 2010, 2011, 2012, 2013, 2014

Simplifying assumptions

- 6000 players every year
- Same players each year
- All player have identical ability

The intersection function

```
players <- c(1:20)
```

```
cash_year1 <- sample(players, 4)
cash_year1
 [1] 20  4 11 14
```

```
cash_year2 <- sample(players, 4)
cash_year2
 [1] 18  7 19  4
```

```
intersect(cash_year1, cash_year2)
 [1] 4
```

Storing cashes as a matrix

```
players <- c(1:20)
```

```
cashes <- replicate(3, sample(players, 4))
cashes
      [,1] [,2] [,3]
 [1,]    4    5    1
 [2,]   11   19    6
 [3,]   18   16    9
 [4,]   12   14   17
```

The Reduce function

```
cashes <- replicate(3, sample(players, 4))
cashes
      [,1] [,2] [,3]
 [1,]    4    5    1
 [2,]   11   19    6
 [3,]   18   16    9
 [4,]   12   14   17
```

```
in_all_three <- Reduce(intersect, list(cashes[, 1], cashes[, 2], cashes[, 3]))
```

```
in_all_three
integer(0)
```

```
length(in_all_three)
 [1] 0
```

PROBABILITY PUZZLES IN R

Let's simulate the World Series of Poker!

PROBABILITY PUZZLES IN R

The von Neumann Model

Peter Chi
Assistant Professor of Statistics
Villanova University

Uniform Random Variables

Each "hand" drawn from a Uniform(0,1) distribution.

```
runif(n, min = 0, max = 1)
```

```
runif(n = 1)
 [1] 0.5888486
```

```
playerA <- runif(n = 1)
playerB <- runif(n = 1)
playerA > playerB
 [1] TRUE

playerA
 [1] 0.6575921
playerB
 [1] 0.3587836
```

Betting under the von Neumann model

Player B observes value, decides whether to wager $1 or not

- If wagered, players compare values. Higher value wins $1 from other player
- If not wagered: no money won or lost by either player

The ifelse function

Condition is true:

```
x <- 4
result <- ifelse(x > 0, sqrt(x), sqrt(-x))
result
 [1] 2
```

Condition is false:

```
x <- (-4)
result <- ifelse(x > 0, sqrt(x), sqrt(-x))
result
 [1] 2
```

The mean function, revisited

```
values <- replicate(10, roll_dice(3))
values
 [1]  5 10 14  5 12 12  4 15  7  9
```

```
mean(values)
 [1] 9.3
```

PROBABILITY PUZZLES IN R

Your turn!

PROBABILITY PUZZLES IN R

Congratulations!

Peter Chi
Assistant Professor of Statistics
Villanova University

Tools that you have learned

- Combinatorics
  - `choose(n,k)`
  - `factorial(n)`
- Simulation
  - `sample()`
  - `replicate()`
  - `runif()`
- Other functions
  - `else`, `ifelse`
  - `identical()`
  - `Reduce()`
- And more...

Where to go next

- More complex combinatorics questions
- Monte Carlo techniques
    - Markov chain Monte Carlo (MCMC)

Other DataCamp courses:

- Bayesian Modeling with RJAGS
- Statistical Simulation in Python

PROBABILITY PUZZLES IN R

Thank you!