Assessment 05 - Random Variables and Sampling Models

Gabriele Mineo - Harvard Data Science Professional

American Roulette probabilities

An American roulette wheel has 18 red, 18 black, and 2 green pockets. Each red and black pocket is associated with a number from 1 to 36. The two remaining green slots feature "0" and "00". Players place bets on which pocket they think a ball will land in after the wheel is spun. Players can bet on a specific number (0, 00, 1-36) or color (red, black, or green).

What are the chances that the ball lands in a green pocket?

Instructions

- Define a variable `p_green` as the probability of the ball landing in a green pocket.
- Print the value of `p_green`.

```
# The variables `green`, `black`, and `red` contain the number of pockets for each color
green <- 2
black <- 18
red <- 18

# Assign a variable `p_green` as the probability of the ball landing in a green pocket
p_green <- green/ (green+black+red)

# Print the variable `p_green` to the console
print(p_green)
```

```
## [1] 0.05263158
```

American Roulette payout

In American roulette, the payout for winning on green is \$17. This means that if you bet \$1 and it lands on green, you get \$17 as a prize.

Create a model to predict your winnings from betting on green.

Instructions

- Use the `sample` function return a random value from a specified range of values.
- Use the `prob =` argument in the `sample` function to specify a vector of probabilities for returning each of the values contained in the vector of values being sampled.

```
# Use the `set.seed` function to make sure your answer matches the expected result after random sampling
set.seed(1)

# The variables 'green', 'black', and 'red' contain the number of pockets for each color
green <- 2
black <- 18
red <- 18

# Assign a variable `p_green` as the probability of the ball landing in a green pocket
p_green <- green / (green+black+red)
```

```r
# Assign a variable `p_not_green` as the probability of the ball not landing in a green pocket
p_not_green <- 1-p_green

#Create a model to predict the random variable `X`, your winnings from betting on green.
X <- sample(c(17,-1),1,prob=c(p_green, p_not_green))

# Print the value of `X` to the console
print(X)
```

```
## [1] -1
```

American Roulette expected value

In American roulette, the payout for winning on green is $17. This means that if you bet $1 and it lands on green, you get $17 as a prize.In the previous exercise, you created a model to predict your winnings from betting on green.

Now, compute the expected value of X, the random variable you generated previously.

Instructions

- Using the chances of winning $17 (`p_green`) and the chances of losing $1 (`p_not_green`), calculate the expected outcome of a bet that the ball will land in a green pocket.

```r
# The variables 'green', 'black', and 'red' contain the number of pockets for each color
green <- 2
black <- 18
red <- 18

# Assign a variable `p_green` as the probability of the ball landing in a green pocket
p_green <- green / (green+black+red)

# Assign a variable `p_not_green` as the probability of the ball not landing in a green pocket
p_not_green <- 1-p_green

# Calculate the expected outcome if you win $17 if the ball lands on green and you lose $1 if the ball
17*p_green + -1*p_not_green
```

```
## [1] -0.05263158
```

American Roulette standard error

The standard error of a random variable X tells us the difference between a random variable and its expected value. You calculated a random variable X in exercise 2 and the expected value of that random variable in exercise 3.

Now, compute the standard error of that random variable, which represents a single outcome after one spin of the roulette wheel.

Instructions

- Compute the standard error of the random variable you generated in exercise 2, or the outcome of any one spin of the roulette wheel.

```r
# The variables 'green', 'black', and 'red' contain the number of pockets for each color
green <- 2
black <- 18
red <- 18

# Assign a variable `p_green` as the probability of the ball landing in a green pocket
p_green <- green / (green+black+red)

# Assign a variable `p_not_green` as the probability of the ball not landing in a green pocket
p_not_green <- 1-p_green

# Compute the standard error of the random variable
abs((17 - -1))*sqrt(p_green*p_not_green)
```

```
## [1] 4.019344
```

American Roulette sum of winnings

You modeled the outcome of a single spin of the roulette wheel, X, in exercise 2.

Now create a random variable S that sums your winnings after betting on green 1,000 times.

Instructions

- Use `set.seed` to make sure the result of your random operation matches the expected answer for this problem.
- Specify the number of times you want to sample from the possible outcomes.
- Use the `sample` function to return a random value from a vector of possible values.
- Be sure to assign a probability to each outcome and to indicate that you are sampling with replacement.

```r
# The variables 'green', 'black', and 'red' contain the number of pockets for each color
green <- 2
black <- 18
red <- 18

# Assign a variable `p_green` as the probability of the ball landing in a green pocket
p_green <- green / (green+black+red)

# Assign a variable `p_not_green` as the probability of the ball not landing in a green pocket
p_not_green <- 1-p_green

# Use the `set.seed` function to make sure your answer matches the expected result after random samplin
set.seed(1)

# Define the number of bets using the variable 'n'
n <- 1000

# Create a vector called 'X' that contains the outcomes of 1000 samples
X <- sample(c(17, -1),n, prob=c(p_green, p_not_green),replace=TRUE)

# Assign the sum of all 1000 outcomes to the variable 'S'
S <- sum(X)

# Print the value of 'S' to the console
print(S)
```

```
## [1] -10
```

American Roulette winnings expected value

In the previous exercise, you generated a vector of random outcomes, S, after betting on green 1,000 times.

What is the expected value of S?

Instructions

- Using the chances of winning $17 (`p_green`) and the chances of losing $1 (`p_not_green`), calculate the expected outcome of a bet that the ball will land in a green pocket over 1,000 bets.

```r
# The variables 'green', 'black', and 'red' contain the number of pockets for each color
green <- 2
black <- 18
red <- 18

# Assign a variable `p_green` as the probability of the ball landing in a green pocket
p_green <- green / (green+black+red)

# Assign a variable `p_not_green` as the probability of the ball not landing in a green pocket
p_not_green <- 1-p_green

# Define the number of bets using the variable 'n'
n <- 1000

# Calculate the expected outcome of 1,000 spins if you win $17 when the ball lands on green and you los
n * (17*p_green + -1*p_not_green)
```

```
## [1] -52.63158
```

American Roulette winnings expected value

You generated the expected value of S, the outcomes of 1,000 bets that the ball lands in the green pocket, in the previous exercise.

What is the standard error of S?

Instructions

- Compute the standard error of the random variable you generated in exercise 5, or the outcomes of 1,000 spins of the roulette wheel.

```r
# The variables 'green', 'black', and 'red' contain the number of pockets for each color
green <- 2
black <- 18
red <- 18

# Assign a variable `p_green` as the probability of the ball landing in a green pocket
p_green <- green / (green+black+red)

# Assign a variable `p_not_green` as the probability of the ball not landing in a green pocket
p_not_green <- 1-p_green

# Define the number of bets using the variable 'n'
```

```r
n <- 1000

# Compute the standard error of the sum of 1,000 outcomes
sqrt(n) * abs((17 - -1))*sqrt(p_green*p_not_green)
```

```
## [1] 127.1028
```