Stat 243: Lab, Monday Oct-17

OOP in R: Rolling a Die

The purpose of this lab is to practice Object-Oriented Programming (OOP) using either S3 or S4 classes in R.

The goal is to program two classes of objects: a regular `"die"` with six sides, and an object `"roll"` (i.e. the rolls of a `"die"` multiples times).

- Use an `.R` script file to write code for the class objects and methods
- If you are already familiar with S3 classes, try using S4 classes
- If you are new to both S3 and S4 classes, choose one of them

Object `"die"`

The object `"die"` should have two attributes:

- `sides`: vector with numbers 1, 2, 3, 4, 5, 6.
- `prob`: vector of probabilities for each side

For the `"die"` object write:

- a constructor function `die()` that creates a *fair* die by default
  - make sure that the argument `prob` has correct probability values
- a `"print"` method that displays the sides and the associated probabilities

You should be able to use `die()` like this:

```
# create a fair die by default
fair_die <- die()

# create a loaded die
loaded_die <- die(prob = c(0.075, 0.1, 0.125, 0.15, 0.20, 0.35))
```

Object `"roll"`

The object `"roll"` should have these attributes:

- `rolls`: vector with outputs of the rolls
- `sides`: vector with numbers 1, 2, 3, 4, 5, 6.
- `prob`: vector of probabilities for each side

For the `"roll"` object write:

- a constructor function that *rolls* a `"die"` a given number of `times`
  - give `times` a default value of 1
- a `print()` method that displays the sides and the associated probabilities
- a `summary()` method that displays the frequencies (i.e. counts) and the relative frequencies (i.e. percentage) of each side of the rolled `die`
- a `plot()` method using a barchart of frequencies (count of `1`'s, `2`'s, `3`'s, `4`'s, `5`'s, and `6`'s)
- an addition `"+"` method to add more rolls

```
# roll fair die
fair500 <- roll(fair_die, times = 500)

# summary method
summary(fair500)

# plot method
plot(fair, 500)

# adding 100 rolls
fair600 <- fair500 + 100
```

De Mere's problem

Use your objects `"die"` and `"roll()"` to simulate a series of 1000 games for the famous *Chevalier De Mere's* dice problems:

- One gambling problem that launched modern probability theory (by Dan Ma).

The first problem involves computing the probability of getting at least one "6" in four rolls of a die. This probability can be computed analytically as:

$$1 - (5/6)^4$$

The other problem involves computing the probability of of getting at least two "6" in 24 rolls of a pair of dice. This probability can be computed analytically as:

$$1 - (35/36)^{24}$$

The goal is to use your `roll()` function to simulate a series of 1000 games for both of De Mere's problems:

- one series of 1000 games should involve rolling a die four times, and then count the number of games in which there is at least one 6.
- the other series of 1000 games should involve rolling a pair of dice 24 times, and then count the number of games in which there is at least one double 6.