Reading course homework

Goal

Take the exercises for STAT 244 and turn them into computational problems, to get a better sense of what's going on.

How to do the homework

For each exercise, produce an `Rmd` file documenting the code and the mathematics. Run simulations etc. to solve the exercise. An example is provided below.

Example: HW1.1

Three cards are drawn randomly from a standard 52 card deck.

First, we need to create a deck of cards, along with their suit and value. I chose to encode them into a `data.frame`:

```
deck <- data.frame(id = 1:52,
                   suit = c(rep("H", 13), rep("D", 13),
                            rep("C", 13), rep("S", 13)),
                   value = rep(1:13, 4))
```

The structure looks like:

```
head(deck)
```

```
##   id suit value
## 1  1    H     1
## 2  2    H     2
## 3  3    H     3
## 4  4    H     4
## 5  5    H     5
## 6  6    H     6
```

Next, we need to write a function that draws cards from the deck without replacement:

```
draw_k_cards <- function(deck, k){
  # draw cards without repetition
  drawn <- sample(deck$id, k)
  return(deck[deck$id %in% drawn,])
}
```

For example:

```
draw_k_cards(deck, 3)
```

```
##    id suit value
## 7   7    H     7
## 20 20    D     7
## 48 48    S     9
```

Now we can attack the first question:

(a) What is the probability that all three cards are from the same suit (i.e. all hearts, or all diamonds, or all clubs, or all spades)?

We write a function that estimates this probability by repeatedly drawing k cards, and checking whether they're all of the same suit:

```
prob_all_same_suit <- function(deck, k, nreps){
  tot_all_same <- 0
  for (i in 1:nreps){
    drawn <- draw_k_cards(deck, k)
    if (length(unique(drawn$suit)) == 1) {
      tot_all_same <- tot_all_same + 1
    }
  }
  return(tot_all_same / nreps)
}
```

If we run this for a sufficiently large number of times, it should converge to the actual probability (note: this might take a few seconds to run):

```
estimate_p_all_same_suit <- prob_all_same_suit(deck, 3, 50000)
estimate_p_all_same_suit
```

```
## [1] 0.05084
```

Suppose that we want to compute the probability of drawing 3 hearts. Then we have probability 13/52 of drawing a hearth as a first card; probability 12/51 of drawing a hearth as a second card (provided that we've drawn a hearth first); and finally we have probability 11/50 of drawing the third hearth. Multiplying we have $1716/132600 \approx 0.129$. Because we only care if we've got cards of the same suit, we can multiply this probability by four, obtaining 0.05176.

The same calculation can be done in a slightly more elegant manner. Suppose all the cards are now balls, of two colors: the hearths are black balls, and the other cards are white balls. Then, the probability of sampling without replacement k black balls is given by the Hypergeometric distribution.

```
n <- 13 # number of black balls
m <- 52 - n # number of white balls
k <- 3 # how many balls we draw
x <- 3 # how many are black (hearths)
p_all_hearths <- dhyper(x, n, m, k)
p_all_hearths
```

```
## [1] 0.01294118
```

And therefore the answer is:

```
p_all_same_suit <- 4 * dhyper(x, n, m, k)
p_all_same_suit
```

```
## [1] 0.05176471
```

We can do something very similar for the second point:

(b) What is the probability that the three cards come from three different suits?

```
prob_all_different_suit <- function(deck, k, nreps){
  tot_all_different <- 0
  for (i in 1:nreps){
    drawn <- draw_k_cards(deck, k)
    if (length(unique(drawn$suit)) == 3) {
      tot_all_different <- tot_all_different + 1
    }
  }
```

```r
  return(tot_all_different / nreps)
}
# The solution should be close to
prob_all_different_suit(deck, 3, 50000)
```

```
## [1] 0.39854
```

We draw a first card; the probability that the second card is not of the suit of the first is 39/51 (we can choose any of the 39 cards out of 51 belonging to the other three suits); the probability that the third card will not be of one of the suits we've seen before is 26/50 (i.e., we can choose from the remaining two suits). As such the answer is

```r
(39 / 51) * (26 / 50)
```

```
## [1] 0.3976471
```