HW 3

Due: Tuesday February 13, 2pm

Instructions

- Produce your assignment as a RMarkdown document rendered to pdf (knit to pdf).
- Also submit your Rmd file (it will not be graded but we want it for reference purposes).
- Show all the code (use `echo=TRUE` as option in R chunks) as well as the results.
- 100 total points.
- See Syllabus for HW policies.

1 [20]

Using paste (or paste0), create the following Latex formula: c_{1}X_{1}^0 + c_{2}X_{2}^1 + c_{3}X_{3}^2 + c_{4}X_{4}^3 + c_{5}X_{5}^4 + c_{6}X_{6}^5 + c_{7}X_{7}^6 + c_{8}X_{8}^7 + c_{9}X_{9}^8 + c_{10}X_{10}^9

which displayed looks like:

$$c_1X_1^0 + c_2X_2^1 + c_3X_3^2 + c_4X_4^3 + c_5X_5^4 + c_6X_6^5 + c_7X_7^6 + c_8X_8^7 + c_9X_9^8 + c_{10}X_{10}^9$$

- Note: only one `paste` or `paste0` can be used. Your solution should contain vectors. No loops allowed.

2 [20]

- In a tank containing 10 fishes, there are three yellow and seven black fishes. We select three fishes at random. What is the probability that exactly one yellow fish gets selected? Using classical approach, the exact solution is:

$$\frac{\binom{3}{1}\binom{7}{2}}{\binom{10}{3}} = \frac{21}{40} = 0.525$$

Solve the problem using three (numerical) techniques:

- 1. using a for loop with an if() clause and accumulator (Hint: The strategy is to sample 3 fishes without replacement from `sample_space`,

```r
sample_space <- c(rep("Y",3),rep("B",7))
n <- 1000000
```

and count how many fishes were yellow. If only one is yellow, then this is a successful case. In this case, add 1 to the accumulator. Repeat the procedure n times. The frequentist result will be accumulator/n.

- 2. using a for loop with an ifelse() function and accumulator

- 3. Vectorial approach using sapply (no loops, no if() or ifelse(), no accumulator).

- Note: in all cases time how long it takes to perform the calculations. You can use the following technique:

```r
ptm <- proc.time()
## Your code
proc.time() - ptm
```

- Comment on the timing of the different techniques.

3 [20]

Using only vectorial approach, determine:

- 1. What is the probability that at most one yellow fish gets selected?
- 2. What is the probability that at least one yellow fish gets selected?

4 [10]

Modify `open.account()` in `06-functions-closures.R` so it is able to:

1. Keep track of the name of the account's holder.
2. print the balance the first time it is called.
3. accept withdrawals.
4. perform a transfer between two accounts.

- It should be invoked as below (producing the corresponding results):

```r
lily <- open.account(200, "Lily")
```

```
## Balance (Lily):200. You made 1 deposits and 0 withdrawals.
```

```r
ross <- open.account(100, "Ross")
```

```
## Balance (Ross):100. You made 1 deposits and 0 withdrawals.
```

```r
lily$transfer(70,ross)
```

```
## 70 withdrawn. Balance (Lily):130. You made 1 deposits and 1 withdrawals.
## 70 deposited. Balance (Ross):170. You made 2 deposits and 0 withdrawals.
```

```r
ross$withdraw(50)
```

```
## 50 withdrawn. Balance (Ross):120. You made 2 deposits and 1 withdrawals.
```

```r
lily$deposit(30)
```

```
## 30 deposited. Balance (Lily):160. You made 2 deposits and 1 withdrawals.
```

```r
ross$transfer(100,lily)
```

```
## 100 withdrawn. Balance (Ross):20. You made 2 deposits and 2 withdrawals.
## 100 deposited. Balance (Lily):260. You made 3 deposits and 1 withdrawals.
```

```r
lily$balance()
```

```
## Balance (Lily):260. You made 3 deposits and 1 withdrawals.
```

```r
ross$balance()
```

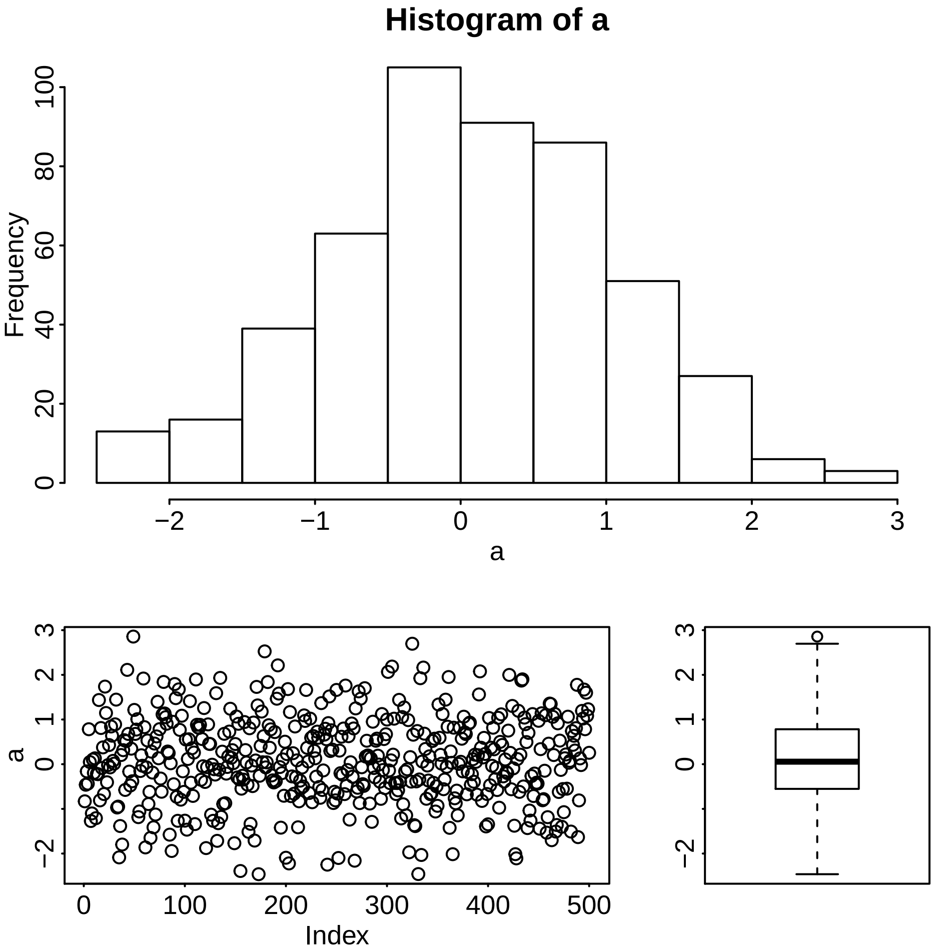
```
## Balance (Ross):20. You made 2 deposits and 2 withdrawals.
```

5 [20]

- In class we have done the following:

```r
par(mar = c(2, 2, 2, 1),
    mgp = c(1.1, 0.15, 0),
    tck = -0.01)
a <- rnorm(500)
```

```r
layout(mat = matrix(c(2, 2,
                      1, 3), 2, byrow = TRUE),
       widths = c(2, 1),
       heights = c(3, 2))
plot(a)
hist(a)
boxplot(a)
```

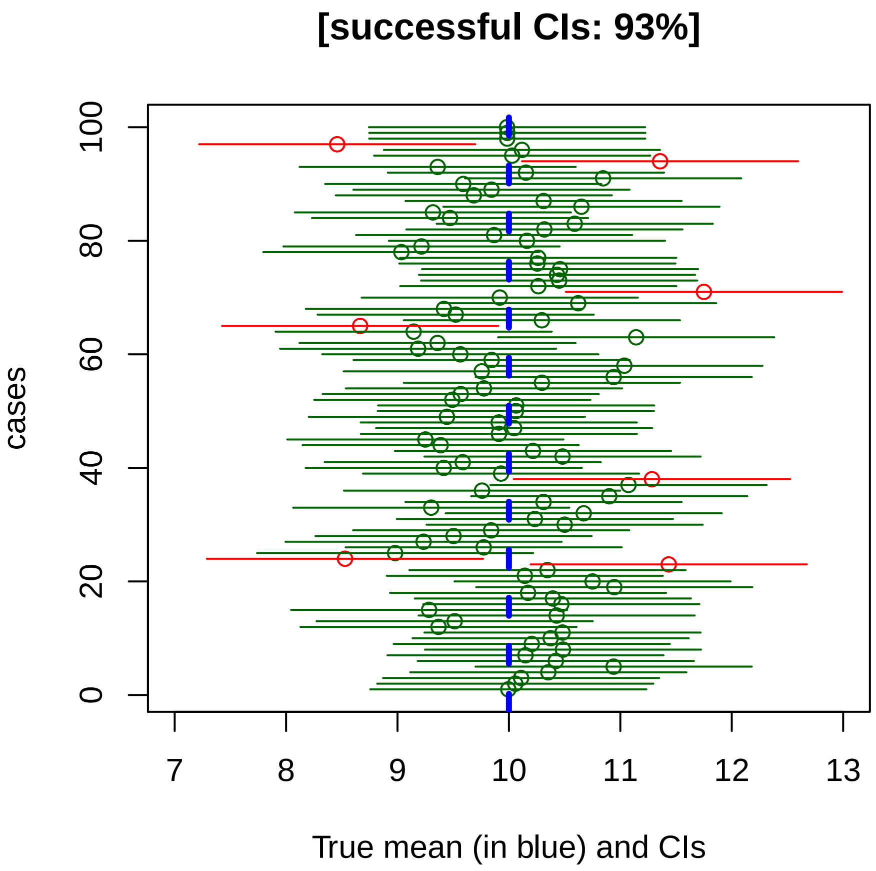


- Achieve the same result without using widths nor heights (only mat) (Note: there is a slight difference on the results of both approaches. So they are almost but not exactly the same).

6. Plot from scratch [10]

- Create a plot that will illustrate how confidence intervals need to be interpreted (from a frequentist approach). For that end, draw 100 samples of size 10 each from a normal distribution with population mean = 10 and population variance = 4. Then plot each 95% confidence interval for the mean. When constructing the intervals, you can assume the population variance is known (of course you cannot assume that the population mean is known). For plotting you can only use `plot`, `abline`, and `segments`. Your plot should look similar to the example below where in green are the cases that captured the true mean while in red the ones that failed. Each case has the interval and the sample mean (the dot in the middle of the interval). Your code must be in a function to be called as shown below:

```
CI_sigma_known(n = 10,               ## sample size
               conf.coeff = .95,     ## confidence level
               real.mu = 10,         ## population mean
               real.var = 4,         ## population variance
               B = 100               ## number of intervals
               )
```


[successful CIs: 93%]
cases
True mean (in blue) and CIs