Marina E. Johnson - mjohnson393

HW 1: R Programming

Problem 1:

- Observation 1: R omits white spaces. Unlike compiled languages such as Java or C++ or an interpreted language such as Python, R omits white spaces.
    - **Code**

```
a = 10; b = 5; c = 1
if (a < b) {
  d=1
} else if (a == b) {
  d=2
} else {
  d=3
}
d
```

    - **Output**

```
3
```

- The code above has indentation and will spit out 3 as its output. The code below without indentation will run in R as well. It will give the same exact output. This will not be possible in many other languages
    - **Code**

```
a = 10; b = 5; c = 1
if (a < b) {
d=1
} else if (a == b) {
d=2
} else {
d=3
}
D
```

    - **Output**

```
3
```

- I also observe that we can even do one liners with loops as seen below.
    - **Code**

```
a = 10; b = 5; c = 1
if (a < b) {d=1} else if (a == b) {d=2} else {d=3}
d
```

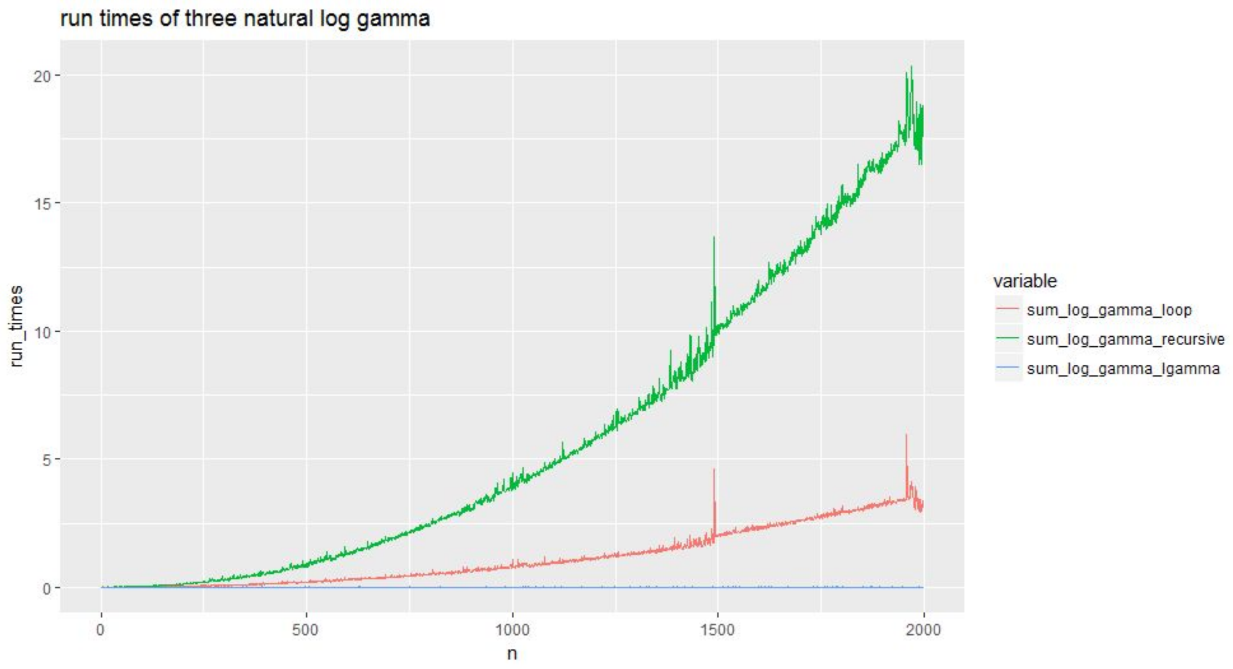
    - **Output**

```
3
```

- We do not have to define variable types like we do in Java and C++. R knows if a variable is integer or string, etc. From this perspective, it is similar to python.
  - **Code**
    ```
    a=3.2 ; b="b is a string"
    print(a) ; print(b)
    ```
  - **Output**
    ```
    [1] 3.2
    [1] "b is a string"
    ```

- If you want to select a variable in a “data frame”, you need to use “$” sign. This is little it complicated because in other languages you normally use “.” operation.
  - **Code**
    ```
    data("iris")
    iris$Sepal.Length[1:4]
    ```
  - **Output**
    ```
    5.1 4.9 4.7 4.6
    ```

Problem 5:



As seen in the graph above, lgamma function performs significantly better than the other two functions that I coded. Regardless of the value n, lgamma can run very fast. (This is maybe due to the fact that it might be written in C). The recursive function is the slowest one. It seems that the performances of the three algorithms do not differ until n reaches 200 or so. We can say that, when n is very small, it does not matter which function we use because their speed is almost the same. However, if n is large, we have to use sum_log_gamma_lgamma because it is much faster.