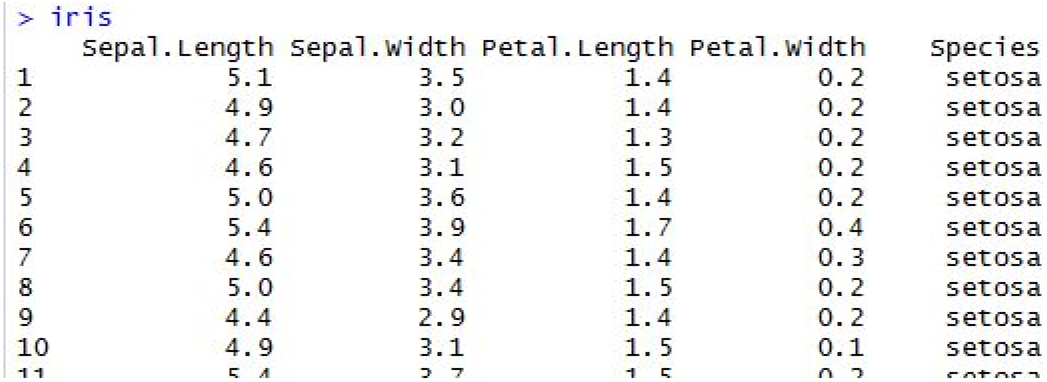
Problem 1:

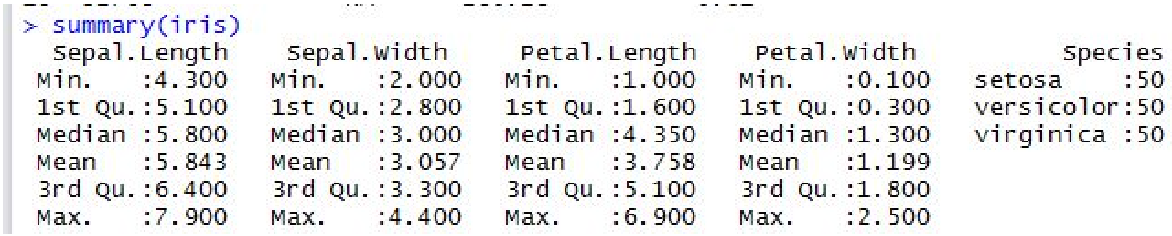
Insights:

There is some pre-loaded data in the R environment, which can be printed. For example, iris:



```
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
11           5.4         3.7          1.5         0.2     setosa
```

Usually, when the data is too big, we can use summary() to see the statistical metrics of the dataset.

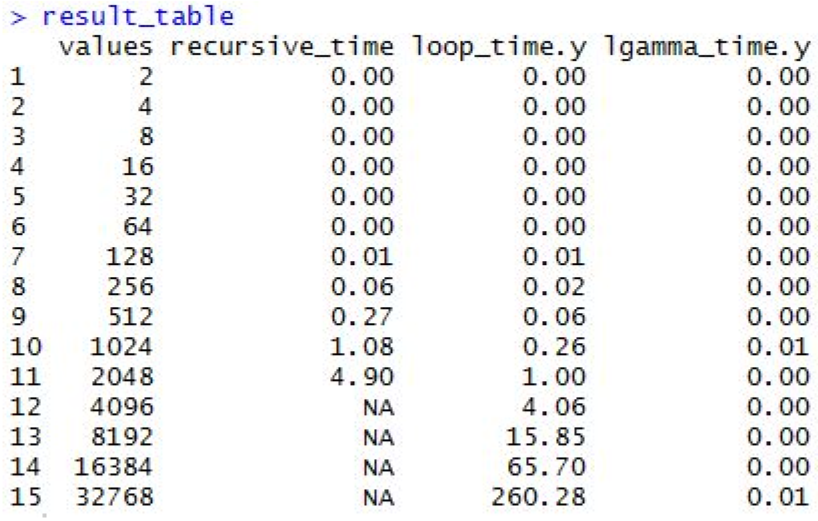


```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

Problem 5:

I use the values of exponential results of 2 to compare the running time. When the value reached to 4096, it indicates “stack overflow”.

Obviously the recursive function is the slowest, the loop function is the second fastest and the built-in lgamma function is the most efficient one.



```
> result_table
   values recursive_time loop_time.y lgamma_time.y
1       2           0.00        0.00          0.00
2       4           0.00        0.00          0.00
3       8           0.00        0.00          0.00
4      16           0.00        0.00          0.00
5      32           0.00        0.00          0.00
6      64           0.00        0.00          0.00
7     128           0.01        0.01          0.00
8     256           0.06        0.02          0.00
9     512           0.27        0.06          0.00
10   1024           1.08        0.26          0.01
11   2048           4.90        1.00          0.00
12   4096             NA        4.06          0.00
13   8192             NA       15.85          0.00
14  16384             NA       65.70          0.00
15  32768             NA      260.28          0.01
```