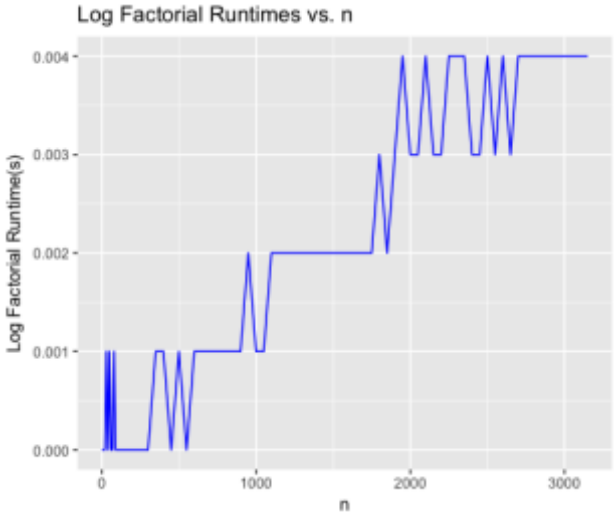


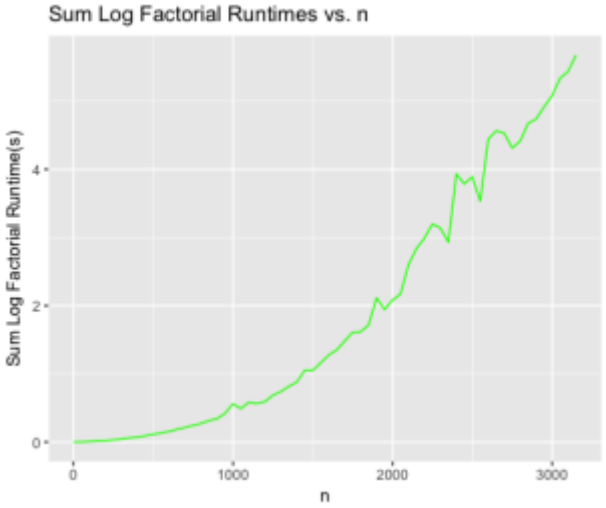


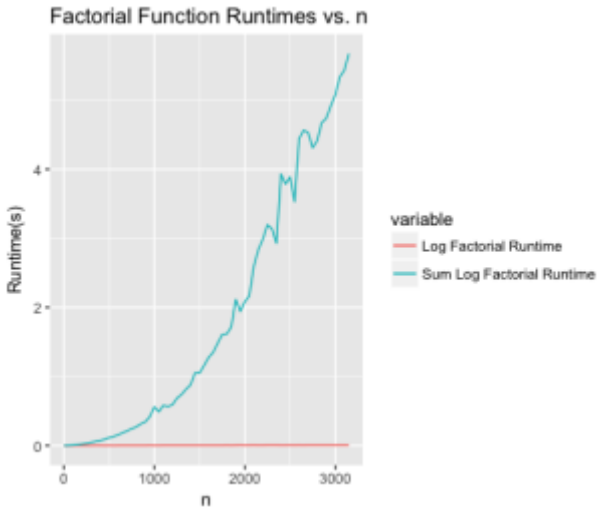


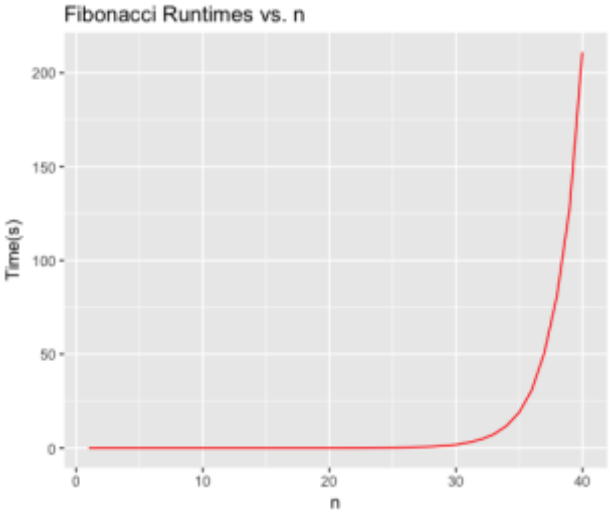


log_factorial → O(n)
sum_log_factorial → $O(n^2)$
Fibonacci → $O(2^n)$

```r
make_log_fact_df <- function(){
  N_logfact = c(1,seq(10,100, by = 10),
        seq(100,3150,by=50))

  time_logfact = array(0, dim = c(length(N_logfact)))
  time_sumlogfact = array(0, dim = c(length(N_logfact)))
  options(expressions=500000)
  for(n in seq(1,length(N_logfact))){
    time_logfact[n] = system.time(log_factorial(N_logfact[n]))
    time_sumlogfact[n] = system.time(sum_log_factorial(N_logfact[n]))
    print(paste('n =',N_logfact[n],' ---',round(N_logfact[n]/3150*100,1),'%'))
  }
  df_logfact = data.frame(n = N_logfact, time_logfact = time_logfact, time_sumlogfact = time_sumlogfact)
  return(df_logfact)
}

make_fibonacci_df <- function(){
  N_fib = seq(1,40)
  time_fib = array(0, dim = c(length(N_fib)))

  for(n in seq(1,length(N_fib))){
    time_fib[n] = system.time(fibonacci(N_fib[n]))
    print(paste('n =',N_fib[n],' ---',round(N_fib[n]/40*100,1),'%'))
  }
  df_fib = data.frame(n = N_fib, time_fib = time_fib)
  return(df_fib)
}
```