Exploring integer partitioning applied to constraints on biodiversity

A. J. Rominger

February 13 2017

Background

The total number of species that can possibly exist is bounded above by the total number of individuals. If there are n individuals there can be at most $s = n$ species. However, the value of s is likely to be much smaller than n. In fact, from a purely combinatorial perspective there are $p_k(n)$ ways of allocating n individuals across $s = k$ species for abitraty $k \leq n$. This is the realm of integer partitioning. From a statistical mechanics perspective, the most likely number of species for a given n is the number that maximizes the microstates (i.e. the value of k that maximizes $p_k(n)$). Because our world is finite, we might expect some reasonable variation about this maximum entropy k. The probabilities for each k_i are equal to $p_{k_i}(n)/p(n)$, where $p(n) = \sum_k p_k(n)$ is the total number of partitions of n across all k.

Set-up

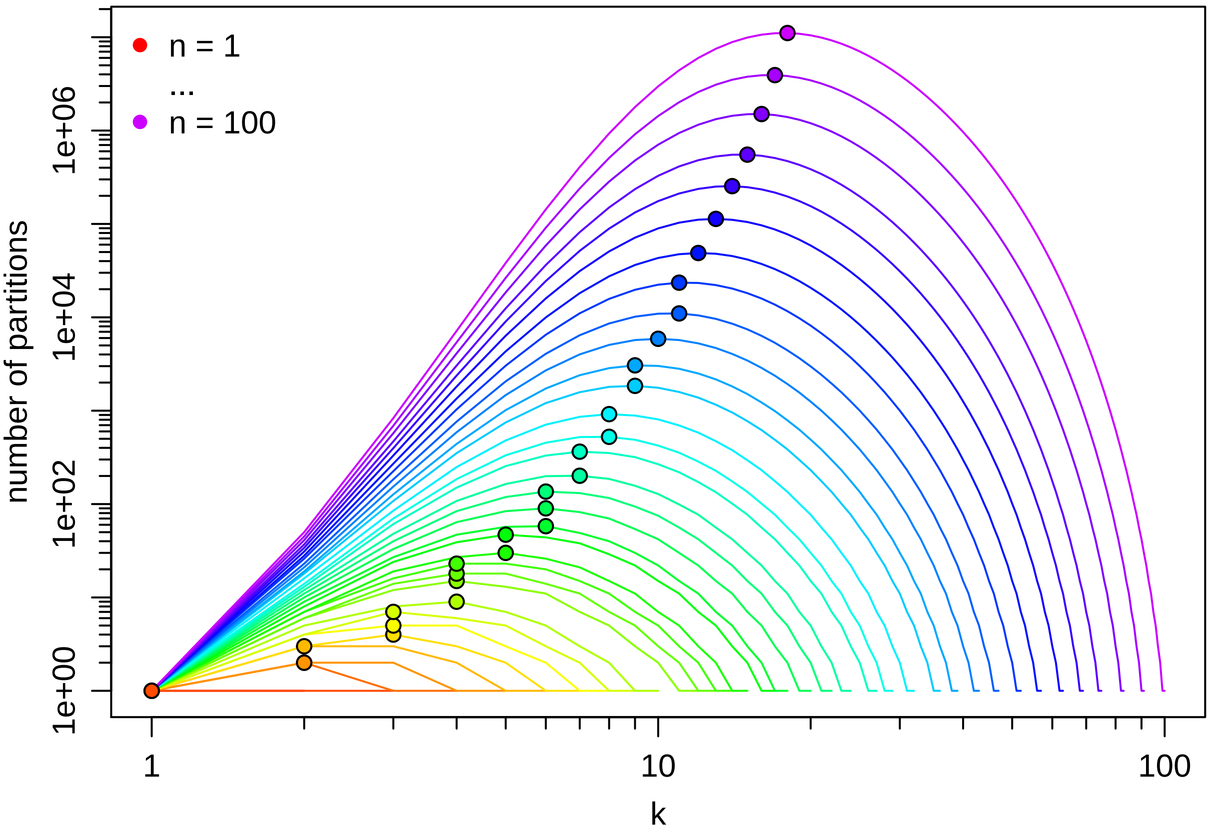
```r
library(partitions)
library(socorro)
```

Initial exploration

Explore how the number of ways to partition an integer n into k parts changes across values of n and k:

```r
## max of n
N <- 100

## loop over n values, for each evaluate k in 1:n
n <- unique(round(10^seq(0, log(N, 10), length = 50)))
npart <- sapply(n, function(ni) {
    out <- sapply(1:ni, function(k) R(k, ni))
    out <- c(out, rep(NA, N - length(out)))
    return(out)
})

## plotting
par(mar = c(3.5, 3.5, 0.5, 0.5), mgp = c(2, 0.75, 0))
matplot(npart, type = 'l', log = 'xy', axes = FALSE, frame.plot = TRUE,
        col = rainbow(length(n), end = 0.8), lty = 1,
        xlab = 'k', ylab = 'number of partitions')
legend('topleft',
       legend = c('n = 1', '...', 'n = 100'), pch = c(16, NA, 16),
       col = rainbow(length(n), end = 0.8)[c(1, 1, length(n))],
       bty = 'n')
logAxis(1)
logAxis(2)
```

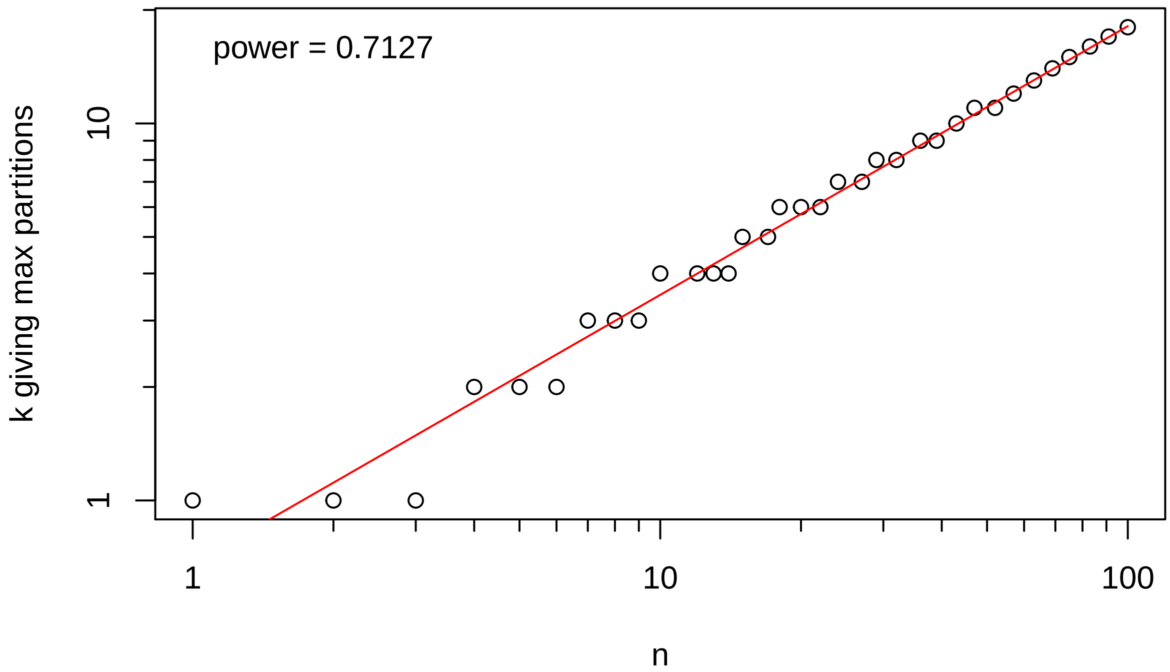
```r
## add the values of k that maximize number of parts
kmax <- apply(npart, 2, which.max)

points(kmax, npart[cbind(kmax, 1:length(kmax))], pch = 21,
       bg = rainbow(length(n), end = 0.8))
```



Note the color gradient starts at $n = 1$ in red, goes through the rainbow colors, and ends at $n = 100$ in purple. Thus each curve represents the number of partitions for a given n across the full range of k values $k \in \{1, \ldots, n\}$.

The filled dots show the maximum number of partitions achieved for each n. We could also look at this in terms of what k value maximizes the number of partitions for each n. Doing so we find a power law:

```r
plot(n, kmax, log = 'xy', axes = FALSE, frame.plot = TRUE,
     xlab = 'n', ylab = 'k giving max partitions')
logAxis(1)
logAxis(2)
mod <- lm(log(kmax) ~ log(n))
curve(exp(mod$coeff[1])*x^mod$coeff[2], add = TRUE, col = 'red')
legend('topleft', legend = paste('power =', round(mod$coeff[2], 4)), bty = 'n')
```


power = 0.7127
k giving max partitions
10
1
1
10
100
n