Writing a Grind model and simplifying that by using with()

Grind uses the R-package deSolve [1] for numerical integration, and that package requires a function returning the derivatives of the differential equations as a list. The form of that function is \( f(t, s, p) \), where \( t \) is time (for ODEs that depend on time), \( s \) is the state, and \( p \) are the parameters. For the Lotka-Volterra model, \( \frac{dR}{dt} = rR(1 - R/K) - aRN \), and \( \frac{dN}{dt} = caN - \delta N \), the state is a vector of two variables, \( R \) and \( N \), and the parameter setting is a vector of 5 values. In R these can be defined as named vectors by typing \( s \leftarrow c(R=1,N=0.01) \) and \( p \leftarrow c(r=1,K=1,a=1,c=1,\delta=0.5) \), respectively.

Having defined \( s \) and \( p \) one could write the model function, \( f(t, s, p) \), as follows:

```r
model <- function(t, s, p) {
  dR <- p["r"] * s["R"] * (1 - s["R"] / p["K"] ) - p["a"] * s["R"] * s["N"]
  dN <- p["c"] * p["a"] * s["R"] * s["N"] - p["\delta"] * s["N"]
  return(list(c(dR, dN)))
}
```

and call the model by \( \text{model}(0, s, p) \), which gives the derivatives for the initial condition \( t = 0 \), or solve the model numerically by calling the Grind function `run()`. Importantly, the derivatives \( dR \) and \( dN \) should be returned as a list, and they should be in the same order as the corresponding variables in the state (this is required by the deSolve package).

Although this works fine, it is somewhat cumbersome to index the state and parameter vectors by the named elements. Fortunately, R provides the `with()` function to evaluate expressions within a particular environment. For instance, typing `with(as.list(p),r+K)` returns the value 2 because \( p["r"] + p["K"] = 2 \). Note that `with()` needs the environment to be a list, i.e., `with(p,r+K)` gives an error message. If we now want to use the names of variables and parameters in the model, we just combine the two named vectors into one, e.g., `with(as.list(c(s,p)),r+R)`. Note that by doing so, there should be no overlap in the names of the variables and the parameters.

Now that we understand `with()` we can write the Lotka-Volterra model in the much more readable, and therefore less error prone, notation

```r
model <- function(t, state, parms) {
  with(as.list(c(state,parms)), {
    dR <- r * s["R"] * (1 - s["R"] / p["K"] ) - a * s["R"] * s["N"]
    dN <- c * a * s["R"] * s["N"] - \delta * s["N"]
    return(list(c(dR, dN)))
  })
}
```

Note that we are returning from within the call to `with` because \( dR \) and \( dN \) are only defined in the scope of that environment.

August 23, 2019, Rob J. de Boer

References