Bite Size R

Factors

Keith Hurley

April 24, 2019

Factors

Factors are a special data type that helps us work with categorical data. We all know that computers are very inefficient when working with character data and very efficient when working with integer data. Factors are a data type that take categorical data and express the values as a number while maintaining textual labels for each category. This provides much better use of both computer memory and computational power when using R.

Let's take a look at an example. Let's create a vector of character strings that are obviously categorical data. Obviously, not only is it an efficient use of memory and storage, but there's a lot of opportunities for typo errors here.

```r
myBirds<-c("Snow Owl", "Barn Owl", "Redtail Hawk", "Golden Eagle",
           "Screech Owl", "Barn Owl", "Osprey", "Redtail Hawk",
           "Redtail Hawk", "Barn Owl")

myBirds
```

```
##  [1] "Snow Owl"     "Barn Owl"     "Redtail Hawk" "Golden Eagle"
##  [5] "Screech Owl"  "Barn Owl"     "Osprey"       "Redtail Hawk"
##  [9] "Redtail Hawk" "Barn Owl"
```

Let's take that variable and create a factor variable from it.

```r
myBirds_factor<-factor(myBirds)

myBirds_factor
```

```
##  [1] Snow Owl     Barn Owl     Redtail Hawk Golden Eagle Screech Owl
##  [6] Barn Owl     Osprey       Redtail Hawk Redtail Hawk Barn Owl
## 6 Levels: Barn Owl Golden Eagle Osprey Redtail Hawk ... Snow Owl
```

Notice how there are no longer quotes around each data item. That is because R is just displaying the label…not the actual value being stored. We can see this if we ask for the numeric value of the factor variable.

```r
as.numeric(myBirds_factor)
```

```
##  [1] 6 1 4 2 5 1 3 4 4 1
```

If we want to get the labels rather than the numeric values, we use the "as.character" function.

```r
as.character(myBirds_factor)
```

```
##  [1] "Snow Owl"     "Barn Owl"     "Redtail Hawk" "Golden Eagle"
##  [5] "Screech Owl"  "Barn Owl"     "Osprey"       "Redtail Hawk"
##  [9] "Redtail Hawk" "Barn Owl"
```

If we just want to know all possible levels available in the factor, we use the "levels" function.

```
levels(myBirds_factor)
```

```
## [1] "Barn Owl"     "Golden Eagle" "Osprey"       "Redtail Hawk"
## [5] "Screech Owl"  "Snow Owl"
```

While R will automatically create the factor levels, it's also possible to define the levels of a factor when you create. This is beneficial if you wish to sort the factor in orders other than alphabetical, which is the default by R (see above example). It's also helpful if there are levels you wish to include that don't currently exist in the data, or if you want to drop data that doesn't exist in the list of levels you provide. Let's create a similar factor that has: 1) a sort order for the levels that's not alphabetical, 2) a level with no matching data (Bald Eagle), and 3) data that does not have a matching level in the list we provide (Kestrel).

```
myBirdLevels<-c("Snow Owl", "Screech Owl", "Barn Owl", "Redtail Hawk",
                "Golden Eagle", "Bald Eagle", "Osprey")

myBirds<-c("Snow Owl", "Barn Owl", "Redtail Hawk", "Golden Eagle",
           "Screech Owl", "Barn Owl", "Osprey", "Redtail Hawk",
           "Redtail Hawk", "Barn Owl", "Kestrel")

myBirds_factor<-factor(myBirds, levels=myBirdLevels)

myBirds_factor
```

```
##  [1] Snow Owl     Barn Owl     Redtail Hawk Golden Eagle Screech Owl
##  [6] Barn Owl     Osprey       Redtail Hawk Redtail Hawk Barn Owl
## [11] <NA>
## 7 Levels: Snow Owl Screech Owl Barn Owl Redtail Hawk ... Osprey
```

Notice that Kestrel, which wasn't on the list, is converted to NA. If we look at the levels, you will see that they now sort according to the order we provided rather than in alphabetical order. There is also now a Bald Eagle level even though there is no data in that level.

```
levels(myBirds_factor)
```

```
## [1] "Snow Owl"     "Screech Owl"  "Barn Owl"     "Redtail Hawk"
## [5] "Golden Eagle" "Bald Eagle"   "Osprey"
```

```
table(myBirds_factor, useNA="always")
```

```
## myBirds_factor
##     Snow Owl  Screech Owl     Barn Owl Redtail Hawk Golden Eagle
##            1            1            3            3            1
##   Bald Eagle       Osprey         <NA>
##            0            1            1
```

By default, R likes to convert any character values into factors whenever it can. For example, when you import data from a csv file or spreadsheet, R will automatically make factors out of text data. In many cases, this is not ideal (like comment fields where every data item is different). You may also wish factors to be created using different levels or orders than R might intuit for you. For these reasons, the first line in all of my scripts turns OFF the automatic creation of factors for character data. I have found that I prefer to control this process myself and do not like R altering my data for me. While I am not alone in this opinion, many people feel the opposite as well. The choice is yours! To turn off the automatic creates on factors for character data, the code is:

```
options(stringsAsFactors = FALSE)
```

That is our whirlwind tour of factors in R. Happy snacking!