Bite Size R

Forcats Package

Keith Hurley

April 24, 2019

Remember, by default R likes to convert any character values into factors whenever it can. To turn off the automatic creates on factors for character data, the code is:

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

Before we get started, let's create the data we'll need in this snack.

```
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
```

Forcats Package

Obviously, there can be a whole bunch of manipulations and operations that we may need to do to work with factors. Good thing there's a package for that! Forcats is a package that's loaded as part of the tidyverse. We'll cover the tidyverse in other snacks but let's take a look at some of the functions in the Forcats package. Make sure that you have installed either the Tidyverse package or the Forcats package on your computer.

```
#load the forcats package
library(forcats)
```

First, let's drop an levels that don't have any data in them (Bald Eagle).

```
myBirds_factor<-fct_drop(myBirds_factor)

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

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

We can also reorder the levels. . . lets put the eagle first.

```
myBirds_factor<-fct_relevel(myBirds_factor, "Golden Eagle")

levels(myBirds_factor)
```

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

How about reversing the order of the factors?

```
myBirds_factor<-fct_rev(myBirds_factor)

levels(myBirds_factor)
```

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

We can also order the factors by their frequency in the data.

```
myBirds_factor<-fct_infreq(myBirds_factor)

levels(myBirds_factor)
```

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

There is also functions that let us reorder factors based on other variables in our data. All this factor reordering may not seem useful at the moment, but it will really come into play as we start graphing and visualizing our data; for example, ordering bar plots in descending order.

There's also a function to collapse factor levels. Let's combine all the owls into a level.

```
myBirds_factor<-fct_collapse(myBirds_factor, owls=c("Barn Owl",
                                                    "Screech Owl",
                                                    "Snow Owl"))

table(myBirds_factor)
```

```
## myBirds_factor
## Redtail Hawk         owls       Osprey Golden Eagle
##            3            5            1            1
```

Using factors appropriately can be a real asset when working with data in R. The Forcats package has a whole bunch of other functions that can be useful and I encourage you to check them out. Happy snacking!