

12. Looking at Data

Solutions to Swirl's R Programming Exercises

07-08-2022

Acknowledgements: R Language Concepts and code questions (with minor modifications) are used here from the swirl package. <https://www.r-project.org/nosvn/pandoc/swirl.html>

Important note: We don't require to use `library(swirl)` and `swirl()` here because we are not going to run the R script in RStudio Console.

Whenever you're working with a new dataset, the first thing you should do is look at it! What is the format of the data? What are the dimensions? What are the variable names? How are the variables stored? Are there missing data? Are there any flaws in the data?

This lesson will teach you how to answer these questions and more using R's built-in functions. We'll be using a dataset constructed from the United States Department of Agriculture's PLANTS Database (http://plants.usda.gov/adv_search.html).

```
load("C:/r-basics/Data/plants.rdata")
```

Type `ls()` to list the variables in your workspace, among which should be plants.

```
ls()
```

```
## [1] "plants"
```

Let's begin by checking the class of the plants variable with `class(plants)`. This will give us a clue as to the overall structure of the data.

```
class(plants)
```

```
## [1] "data.frame"
```

It's very common for data to be stored in a data frame. It is the default class for data read into R using functions

Since the dataset is stored in a data frame, we know it is rectangular. In other words, it has two dimensions (rows and columns) and fits neatly into a table or spreadsheet. Use `dim(plants)` to see exactly how many rows and columns we're dealing with.

```
dim(plants)
```

```
## [1] 5166 10
```

The first number you see (5166) is the number of rows (observations) and the second number (10) is the number of columns (variables).

You can also use `nrow(plants)` to see only the number of rows. Try it out.

```
nrow(plants)
```

```
## [1] 5166
```

... And `ncol(plants)` to see only the number of columns.

```
ncol(plants)
```

```
## [1] 10
```

If you are curious as to how much space the dataset is occupying in memory, you can use `object.size(plants)`.

```
object.size(plants)
```

```
## 745944 bytes
```

Now that we have a sense of the shape and size of the dataset, let's get a feel for what's inside. `names(plants)` will return a character vector of column (i.e. variable) names. Give it a shot.

```
names(plants)
```

```
## [1] "Scientific_Name"      "Duration"             "Active_Growth_Period"
## [4] "Foliage_Color"       "pH_Min"              "pH_Max"
## [7] "Precip_Min"         "Precip_Max"          "Shade_Tolerance"
## [10] "Temp_Min_F"
```

We've applied fairly descriptive variable names to this dataset, but that won't always be the case. A logical next step is to peek at the actual data. However, our dataset contains over 5000 observations (rows), so it's impractical to view the whole thing all at once.

The `head()` function allows you to preview the top of the dataset. Give it a try with only one argument.

```
head(plants)
```

```
##           Scientific_Name      Duration Active_Growth_Period
## 1           Abelmoschus           <NA>           <NA>
## 2 Abelmoschus esculentus Annual, Perennial           <NA>
## 3                Abies           <NA>           <NA>
## 4      Abies balsamea      Perennial      Spring and Summer
## 5 Abies balsamea var. balsamea      Perennial           <NA>
## 6             Abutilon           <NA>           <NA>
##  Foliage_Color pH_Min pH_Max Precip_Min Precip_Max Shade_Tolerance Temp_Min_F
## 1           <NA>    NA    NA         NA         NA           <NA>         NA
## 2           <NA>    NA    NA         NA         NA           <NA>         NA
## 3           <NA>    NA    NA         NA         NA           <NA>         NA
## 4      Green      4      6         13         60      Tolerant       -43
## 5           <NA>    NA    NA         NA         NA           <NA>         NA
## 6           <NA>    NA    NA         NA         NA           <NA>         NA
```

Take a minute to look through and understand the output above. Each row is labeled with the observation number and each column with the variable name. Your screen is probably not wide enough to view all 10 columns side-by-side, in which case R displays as many columns as it can on each line before continuing on the next.

By default, `head()` shows you the first six rows of the data. You can alter this behavior by passing as a second argument the number of rows you'd like to view. Use `head()` to preview the first 10 rows of plants.

```
head(plants, 10)
```

```
##           Scientific_Name      Duration Active_Growth_Period
## 1           Abelmoschus          <NA>          <NA>
## 2       Abelmoschus esculentus Annual, Perennial          <NA>
## 3              Abies            <NA>          <NA>
## 4       Abies balsamea      Perennial    Spring and Summer
## 5   Abies balsamea var. balsamea      Perennial          <NA>
## 6              Abutilon          <NA>          <NA>
## 7       Abutilon theophrasti      Annual          <NA>
## 8              Acacia          <NA>          <NA>
## 9       Acacia constricta      Perennial    Spring and Summer
## 10  Acacia constricta var. constricta      Perennial          <NA>
##  Foliage_Color pH_Min pH_Max Precip_Min Precip_Max Shade_Tolerance Temp_Min_F
## 1           <NA>    NA    NA         NA         NA           <NA>         NA
## 2           <NA>    NA    NA         NA         NA           <NA>         NA
## 3           <NA>    NA    NA         NA         NA           <NA>         NA
## 4           Green     4    6.0        13        60       Tolerant       -43
## 5           <NA>    NA    NA         NA         NA           <NA>         NA
## 6           <NA>    NA    NA         NA         NA           <NA>         NA
## 7           <NA>    NA    NA         NA         NA           <NA>         NA
## 8           <NA>    NA    NA         NA         NA           <NA>         NA
## 9           Green     7    8.5         4        20      Intolerant       -13
## 10          <NA>    NA    NA         NA         NA           <NA>         NA
```

The same applies for using `tail()` to preview the end of the dataset. Use `tail()` to view the last 15 rows.

```
tail(plants, 15)
```

```
##           Scientific_Name      Duration Active_Growth_Period
## 5152           Zizania          <NA>          <NA>
## 5153       Zizania aquatica      Annual          Spring
## 5154   Zizania aquatica var. aquatica      Annual          <NA>
## 5155           Zizania palustris      Annual          <NA>
## 5156  Zizania palustris var. palustris      Annual          <NA>
## 5157           Zizaniopsis          <NA>          <NA>
## 5158       Zizaniopsis miliacea      Perennial    Spring and Summer
## 5159              Zizia          <NA>          <NA>
## 5160       Zizia aptera      Perennial          <NA>
## 5161       Zizia aurea      Perennial          <NA>
## 5162       Zizia trifoliata      Perennial          <NA>
## 5163              Zostera          <NA>          <NA>
## 5164       Zostera marina      Perennial          <NA>
## 5165              Zoysia          <NA>          <NA>
## 5166       Zoysia japonica      Perennial          <NA>
```

```
##      Foliage_Color pH_Min pH_Max Precip_Min Precip_Max Shade_Tolerance
## 5152      <NA>      NA      NA      NA      NA      <NA>
## 5153      Green    6.4    7.4     30     50      Intolerant
## 5154      <NA>      NA      NA      NA      NA      <NA>
## 5155      <NA>      NA      NA      NA      NA      <NA>
## 5156      <NA>      NA      NA      NA      NA      <NA>
## 5157      <NA>      NA      NA      NA      NA      <NA>
## 5158      Green    4.3    9.0     35     70      Intolerant
## 5159      <NA>      NA      NA      NA      NA      <NA>
## 5160      <NA>      NA      NA      NA      NA      <NA>
## 5161      <NA>      NA      NA      NA      NA      <NA>
## 5162      <NA>      NA      NA      NA      NA      <NA>
## 5163      <NA>      NA      NA      NA      NA      <NA>
## 5164      <NA>      NA      NA      NA      NA      <NA>
## 5165      <NA>      NA      NA      NA      NA      <NA>
## 5166      <NA>      NA      NA      NA      NA      <NA>
##      Temp_Min_F
## 5152      NA
## 5153      32
## 5154      NA
## 5155      NA
## 5156      NA
## 5157      NA
## 5158      12
## 5159      NA
## 5160      NA
## 5161      NA
## 5162      NA
## 5163      NA
## 5164      NA
## 5165      NA
## 5166      NA
```

After previewing the top and bottom of the data, you probably noticed lots of NAs, which are R's placeholders for missing values. Use `summary(plants)` to get a better feel for how each variable is distributed and how much of the dataset is missing.

```
summary(plants)
```

```
## Scientific_Name      Duration      Active_Growth_Period Foliage_Color
## Length:5166      Length:5166      Length:5166      Length:5166
## Class :character  Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character  Mode  :character
##
##
##
##      pH_Min      pH_Max      Precip_Min      Precip_Max
## Min.   :3.000   Min.   : 5.100   Min.   : 4.00   Min.   : 16.00
## 1st Qu.:4.500   1st Qu.: 7.000   1st Qu.:16.75   1st Qu.: 55.00
## Median :5.000   Median : 7.300   Median :28.00   Median : 60.00
## Mean   :4.997   Mean   : 7.344   Mean   :25.57   Mean   : 58.73
## 3rd Qu.:5.500   3rd Qu.: 7.800   3rd Qu.:32.00   3rd Qu.: 60.00
```

```
## Max.      :7.000   Max.      :10.000   Max.      :60.00   Max.      :200.00
## NA's      :4327   NA's      :4327   NA's      :4338   NA's      :4338
## Shade_Tolerance      Temp_Min_F
## Length:5166      Min.      :-79.00
## Class :character   1st Qu.: -38.00
## Mode  :character   Median   :-33.00
##                      Mean     :-22.53
##                      3rd Qu.: -18.00
##                      Max.      : 52.00
##                      NA's      :4328
```

summary() provides different output for each variable, depending on its class. For numeric data such as Precip_Min, summary() displays the minimum, 1st quartile, median, mean, 3rd quartile, and maximum. These values help us understand how the data are distributed. You can see that R truncated the summary for Active_Growth_Period by including a catch-all category called 'Other'. Since it is a categorical/factor variable, we can see how many times each value actually occurs in the data with table(plants\$Active_Growth_Period).

```
table(plants$Active_Growth_Period)
```

```
##
## Fall, Winter and Spring      Spring      Spring and Fall
##              15              144              10
##      Spring and Summer      Spring, Summer, Fall      Summer
##              447              95              92
##      Summer and Fall      Year Round
##              24              5
```

Each of the functions we've introduced so far has its place in helping you to better understand the structure of your data. However, we've left the best for last...

Perhaps the most useful and concise function for understanding the *structure* of your data is str(). Give it a try now.

```
str(plants)
```

```
## 'data.frame':   5166 obs. of  10 variables:
## $ Scientific_Name      : chr  "Abelmoschus" "Abelmoschus esculentus" "Abies" "Abies balsamea" ...
## $ Duration             : chr  NA "Annual, Perennial" NA "Perennial" ...
## $ Active_Growth_Period: chr  NA NA NA "Spring and Summer" ...
## $ Foliage_Color         : chr  NA NA NA "Green" ...
## $ pH_Min               : num  NA NA NA 4 NA NA NA NA 7 NA ...
## $ pH_Max               : num  NA NA NA 6 NA NA NA NA 8.5 NA ...
## $ Precip_Min           : int  NA NA NA 13 NA NA NA NA 4 NA ...
## $ Precip_Max           : int  NA NA NA 60 NA NA NA NA 20 NA ...
## $ Shade_Tolerance      : chr  NA NA NA "Tolerant" ...
## $ Temp_Min_F           : int  NA NA NA -43 NA NA NA NA -13 NA ...
```

The beauty of str() is that it combines many of the features of the other functions you've already seen, all in a concise and readable format. At the very top, it tells us that the class of plants is 'data.frame' and that it has 5166 observations and 10 variables. It then gives us the name and class of each variable, as well as a preview of its contents.

str() is actually a very general function that you can use on most objects in R. Any time you want to understand the

In this lesson, you learned how to get a feel for the structure and contents of a new dataset using a collection of simple and useful functions. Taking the time to do this upfront can save you time and frustration later on in your analysis.