Week 4 - Looking at Data

Sally Longmore

30/01/2020

Whenever you are working with a new dataset, the first thing you should do is look at it and ask yourself some questions. - 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 will be using a dataset constructed from the United States Department of Agriculture's PLANTS datbase.

(http://plants.usda.gov/adv_search.html)

The data is stored in a variable called plants.

```
plants <- readRDS("plants.rds")
```

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 like read.csv() and read.table(), which you will learn about in another lesson.

Since the dataset is stored in a data frame, we know it is retangular. 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 are 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.

```
nrow(plants)
```

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

And you can run ncol(plants) to see the number of columns.

```
ncol(plants)
```

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

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

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

```
## 686080 bytes
```

Now that we have a sense of the shape and size of the dataset, let's get a feel for what is inside. we can use names(plants) to return a character vector of column (i.e. variable) names.

```
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 have applied fairly descriptive variable names to this dataset, but that won't always be the case. A logical next step is to peak at the actual data. However, our dataset contains over 5,000 observations (rows), so it is impractical to view the whole dataset at once.

The head() function allows you to preview the top of the dataset.

```
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 labelled 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 to the next.

By default, head() shows you the first six rows of data. You can alter this behaviour by passing as a second argument the number of rows you would 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.

```r
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   
##  Abelmoschus                 :   1   Perennial        :3031  
##  Abelmoschus esculentus      :   1   Annual           : 682  
##  Abies                       :   1   Annual, Perennial: 179  
##  Abies balsamea              :   1   Annual, Biennial :  95  
##  Abies balsamea var. balsamea:   1   Biennial         :  57  
##  Abutilon                    :   1   (Other)          :  92  
##  (Other)                     :5160   NA's             :1030  
##            Active_Growth_Period      Foliage_Color      pH_Min     
##  Spring and Summer   : 447      Dark Green  :  82   Min.   :3.000  
##  Spring              : 144      Gray-Green  :  25   1st Qu.:4.500  
##  Spring, Summer, Fall:  95      Green       : 692   Median :5.000  
##  Summer              :  92      Red         :   4   Mean   :4.997  
##  Summer and Fall     :  24      White-Gray  :   9   3rd Qu.:5.500  
##  (Other)             :  30      Yellow-Green:  20   Max.   :7.000  
##  NA's                :4334      NA's        :4334   NA's   :4327   
##      pH_Max         Precip_Min      Precip_Max         Shade_Tolerance
##  Min.   : 5.100   Min.   : 4.00   Min.   : 16.00   Intermediate: 242  
##  1st Qu.: 7.000   1st Qu.:16.75   1st Qu.: 55.00   Intolerant  : 349  
##  Median : 7.300   Median :28.00   Median : 60.00   Tolerant    : 246  
##  Mean   : 7.344   Mean   :25.57   Mean   : 58.73   NA's        :4329  
##  3rd Qu.: 7.800   3rd Qu.:32.00   3rd Qu.: 60.00                      
##  Max.   :10.000   Max.   :60.00   Max.   :200.00                      
##  NA's   :4327     NA's   :4338    NA's   :4338                        
##    Temp_Min_F    
##  Min.   :-79.00  
##  1st Qu.:-38.00  
##  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 distrubted.

For categorical variables (called 'factor' variables in R), summary() displays the number of times each value (or 'level') occurs in the data. For exmaple, each value of **Scientific_Name** only appears once, since it is unique to specific plants. In contrast, the summary for Duration (also a factor variable) tells us that our dataset contains 3031 Perennial plants, 682 Annual plants etc.

You can see that R truncated the summary for **Active_Growth_Period** by including a catch-all category called **Other**. Since it is a vategorical/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 have introduced so far has its place in helping you to better understand the structure of your data. However, we have left the best for last.

Perhaps the most useful and concise function for understanding the **str**ucture of your data is the str() function.

```
str(plants)
```

```
## 'data.frame':    5166 obs. of  10 variables:
##  $ Scientific_Name     : Factor w/ 5166 levels "Abelmoschus",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Duration            : Factor w/ 8 levels "Annual","Annual, Biennial",..: NA 4 NA 7 7 NA 1 NA 7 7
##  $ Active_Growth_Period: Factor w/ 8 levels "Fall, Winter and Spring",..: NA NA NA 4 NA NA NA NA 4 NA
##  $ Foliage_Color       : Factor w/ 6 levels "Dark Green","Gray-Green",..: NA NA NA 3 NA NA NA NA 3 NA
##  $ 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     : Factor w/ 3 levels "Intermediate",..: NA NA NA 3 NA NA NA NA 2 NA ...
##  $ 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 have 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 it's contents.

str() is actually a very general function that you can use on most objects in R. Any time you want to understand the structure of something (a dataset, function, etc.), str() is a good place to start.