CWRU DSCI351-451: Dataframe 7 Ways and Violent Crime

Roger H. French, JiQi Liu

07 October, 2018

Contents

0.1.0.1 Examine a Data Frame in R with 7 Basic Functions

When one first started learning R,

- it seems way more complicated than what you were used to
- with looking at spreadsheets in Microsoft Excel.

When I started working with data frames in R,

- it didn't seem quite as easy to know what I was looking at.

I've since come to see the light.

- While there is a bit of a learning curve to get a handle on it,
- viewing data in R is infinitely more flexible than doing so in Excel.

In this discussion, I'll cover the most basic R functions

- for examining a data set and
- explain why they're important.

Understanding how to get a simple overview of the data set

- has become a huge time saver for me.
- If you aren't familiar with these functions, you need to be.
- If you're anything like me,
  - you'll use them first for every single data set you consider.

All of the functions I'm discussing here come in the base R Utils package,

- so there's no need to install any additional packages.

Here are the functions, with links to their documentation:

- dim(): shows the dimensions of the data frame by row and column
- str(): shows the structure of the data frame
- summary(): provides summary statistics on the columns of the data frame
- colnames(): shows the name of each column in the data frame
- head(): shows the first 6 rows of the data frame
- tail(): shows the last 6 rows of the data frame
- View(): shows a spreadsheet-like display of the entire data frame

0.1.0.1.1 Now, let's import a data set see how each of these functions works.

First, here's the code:

0.1.1 Import a data set on violent crime by state and assign it to the data frame "crime"

```
crime <- read.csv("http://vincentarelbundock.github.io/Rdatasets/csv/datasets/USArrests.csv", stringsAs
```

Now, let's take a look at the output,

- so we can see what happens when the code is run.

First, we'll look at the

- dim(),
- str(),
- summary(), and
- colnames() functions:

```
### Call the functions on crime to examine the data frame
dim(crime)
```

```
## [1] 50  5
```

- dim(): In the crime data set, we can see immediately that
  - there are only 50 rows and 5 columns.
  - This function is useful, because it tells us whether it would be okay to print the entire data frame to the console. With this data set, it's probably okay.
  - If, however, there were 5,000 rows and 50 columns, we'd definitely want to view the data frame in smaller chunks.

```
### Call the functions on crime to examine the data frame
str(crime)
```

```
## 'data.frame':    50 obs. of  5 variables:
##  $ X       : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

- str(): The structure of the crime data set also
  - tells us the number of rows (observations) and columns (variables),
  - but it provides even more information.
    - It tells us the column names,
    - the class of each column (what kind of data is stored in it),
    - and the first few observations of each variable.

```
### Call the functions on crime to examine the data frame
summary(crime)
```

```
##       X                  Murder          Assault         UrbanPop
##  Length:50          Min.   : 0.800   Min.   : 45.0   Min.   :32.00
##  Class :character   1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50
##  Mode  :character   Median : 7.250   Median :159.0   Median :66.00
##                     Mean   : 7.788   Mean   :170.8   Mean   :65.54
##                     3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75
##                     Max.   :17.400   Max.   :337.0   Max.   :91.00
##       Rape
```

```
##  Min.   : 7.30
##  1st Qu.:15.07
##  Median :20.10
##  Mean   :21.23
##  3rd Qu.:26.18
##  Max.   :46.00
```

- summary(): The summary provides descriptive statistics including
  - the min, max, mean, median, and quartiles of each column.
  - For example, we can see in the crime data set that
    - the average murder rate across all states is 7.8 for every 100k people.

```
### Call the functions on crime to examine the data frame
colnames(crime)
```

```
## [1] "X"        "Murder"   "Assault"  "UrbanPop" "Rape"
```

- colnames(): This function prints a vector of the column names,
  - which can be useful if you're trying to reference a particular column.
  - For the crime data set, we can see that the state column has no name.
  - Knowing this, we may want to assign it a name before going forward in our analysis.

Now, let's take a look at the head() and tail() functions:

```
### The head() and tail() functions default to 6 rows, but we can adjust the number of rows using the "n
head(crime, n = 10)
```

```
##              X Murder Assault UrbanPop Rape
## 1      Alabama   13.2     236       58 21.2
## 2       Alaska   10.0     263       48 44.5
## 3      Arizona    8.1     294       80 31.0
## 4     Arkansas    8.8     190       50 19.5
## 5   California    9.0     276       91 40.6
## 6     Colorado    7.9     204       78 38.7
## 7  Connecticut    3.3     110       77 11.1
## 8     Delaware    5.9     238       72 15.8
## 9      Florida   15.4     335       80 31.9
## 10     Georgia   17.4     211       60 25.8
```

```
tail(crime, n = 5)
```

```
##               X Murder Assault UrbanPop Rape
## 46     Virginia    8.5     156       63 20.7
## 47   Washington    4.0     145       73 26.2
## 48 West Virginia    5.7      81       39  9.3
## 49    Wisconsin    2.6      53       66 10.8
## 50      Wyoming    6.8     161       60 15.6
```

- head(): This function defaults to printing the first 6 rows, -but we've decided to call the first 10.
  - In the crime data set, this gives us the data on states Alabama through Georgia.
- tail(): The same as head(),
  - except this function prints the end of the data frame.
  - In this case, we've called the last 5 observations,
    - so we can see the data on Virginia through Wyoming.

Finally, let's take a look at the window that appears when we call the View() function:

```
### While the first 6 functions are printed to the console, the View() function opens a table in anothe
View(crime)
```

- View(): This window provides vertical and horizontal
  - (if enough columns to justify)
  - scroll bars for you to browse the entire data set.
  - It looks exactly like an Excel spreadsheet'
    - you just can't manipulate any of the data.
    - (Note: make sure you use a capital 'V' when calling this function; it's case sensitive).

That's it! Getting comfortable with these functions should make it easier for you to work with data frames in a more logical and efficient manner.

0.1.1.1 Cites

- Douglas E Rice https://www.r-bloggers.com/examine-a-data-frame-in-r-with-7-basic-functions/amp/