The Versatility of Data Frames

Saalar Aghili
10/29/2017

Introduction

Data frames are one of the most common forms of data sets for a lot of programming softwares, including RStudio. Most of our class' work with data frames came in the form of manipulating data frames using certain packages dedicated to data frame manipulation. This part of the class seems to be most applicable to my professional career path into economics. Economics research uses data drames frequently and manipulates data based on the models being used. I want to explore data frames further to see what they offer specifically in the realm of economics work with data.

Basics

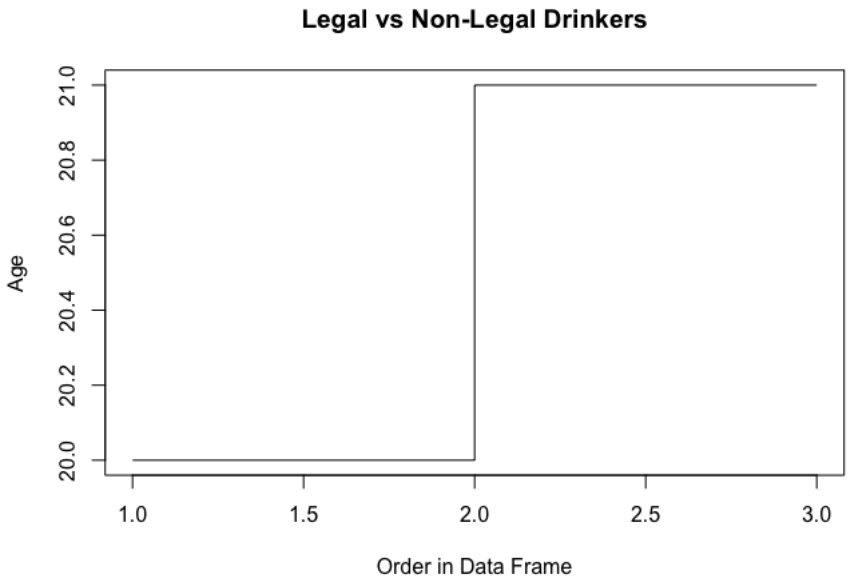
Data frames can facilitate a data set by being able to name each column. One of the beauties of data frames are that they are **non-atomic**, meaning that they can include columns of different data types:

```
table1 <- data.frame(name = c("Saalar", "Turner", "Alexei"), age = c(20, 21, 21), legal_drinker = c(FALSE,
table1
```

```
##     name age legal_drinker
## 1 Saalar  20         FALSE
## 2 Turner  21          TRUE
## 3 Alexei  21          TRUE
```

In the data frame above, I have made a table with named columns to show if the listed names can drink legally in the US. Notice that each column is a different data type. names are characters, ages are double, and legal_drinker logical. One of the biggest tips with data frames is that R will automatically turn the class of one of your columns into a factor One of the things that you can do with data frames is extract columns to be visualized into a graph. Using the table above:

```
plot(table1$age, main = "Legal vs Non-Legal Drinkers", xlab = "Order in Data Frame", ylab = "Age", type = '
```



Here I extracted the age column from the data frame to make a simple line plot visualizing the data frame. It shows that up until person 2, the people are not of legal drinking age. Specifically for columns, data frames can use "$" to retrieve the name of the column associated with the part of the data frame specified in the command. As you can see, I used $ to extract the respective column, age, that I was looking for. Some of the most fundamental tools in R for manipulation is bracket notation. This can help you extract any part of the data frame that is desired.

```
table1[1, 2]
```

```
## [1] 20
```

You can extract parts of your data frame using bracket notation with respect to **rows** first, then **columns**.

Importing Data Frames

Data Frames also exist among other softwares, like excel. Importing data frames from other softwares can be difficult. In the case of excel files, they are enriched. One way to go about properly importing data from an excel file is by having the right package to do so. One of the packages that can be used to import a ".xls" file, or excel file, is with package "gdata" or "XLConnect"."gdata" will provide the function **read.xls** while "XLConnect" uses the **loadWorkbook**. Other files that are imported into R can be Minitab, SPSS, or CSV files. We have used **read.csv** before in class for comma-separated variables. Some of the packages and functions associated with data frames include:

- dplyr
- reshape2
- foreign
- readr
- googlesheets

Timeseries as a package

One of the packages not used in the class that has a lot applicability to my personal and professional interests has to do with timeseries objects. The packages most commonly used for timeseries objects in a data frame are "ts" (Timeseries) and "xts" (extensible Time Series). This package allows for dataframes to be converted into a timeseries object in order to easily manipulate data to show growth rates and returns over time. One of the abilities associated with timeseries objects is the use of **date as a class**.
Like data frames, timeseries objects offer the ability to manipulate the timeseries objects to your liking. Like data frames, timeseries objects can add/delete columns (like how we used mutate in dplyr), add rows, and subset is unique about timeseries is the use of date as a class. Effectively, the frequencies of each data type designated as date can be changed based on weeks, months, years, days, etc. In terms of economics, this is applicable to working with data in the long run (which can cover decades in which years are more approprate), or in the short run (which can cover days or months). Being able to change the frequency of dates is important to analyze the data once you have it all organized. The function used to properly convert dates is the function **as.Date** Overall, R serves as a platform for students of many academic backgrounds to capitalize on the data that they have. Data frames are a strong component of working with data in R and the amount of data frames we encounter, even beyond R, prove how versatile they are. From excel, to google sheets, to different packages within R, data frames can take many shapes and be used for different functions. I found an interest mostly in timeseries because of the applicability it has to growth rates.

Works Cited

- Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.
- Mukherjee, Pinaki M.,R For Economics and Finance: Data Manipulation. rstudio-pubs-static.s3.amazonaws.com/40873_5fbe3860854a47c38a58aabd01f9cf9d.html.
- Machlis, Sharon. “Great R Packages for Data Import, Wrangling and Visualization.” Computerworld, Computerworld, 5 Oct. 2017, www.computerworld.com/article/2921176/business-intelligence/great-r-packages-for-data-import-wrangling-visualization.html.
- robk@statmethods.net, Robert Kabacoff -. “Time Series and Forecasting.” Quick-R: Time Series, www.statmethods.net/advstats/timeseries.html.
- “Data Import.” Data Import | R Tutorial, www.r-tutor.com/r-introduction/data-frame/data-import.
- R: DataFrame Objects, web.mit.edu/~r/current/arch/i386_linux26/lib/R/library/S4Vectors/html/DataFrame-class.html.
- “Data Frame Column Slice.” Data Frame Column Slice | R Tutorial, www.r-tutor.com/r-introduction/data-frame/data-frame-column-slice.