Why R and Not Other Languages

Introduction

R is consistenly ranked as one of the best programming languages in the industry, an impressive feat given that it is a language created only for the purposes of Statistics by statisticians. In this post, I want to further explore the differences in R and other languages that are widely used for Data analysis such as Python. I personally have experience working with various sets of data with both languages, and want to try my best to highlight some of the major differences and advantages and disadvantages of both languages. It is my wish that this post will be helpful for my understanding of not just the two languages discussed, programming languages in general, so that the readers and I can go on to make smart choices in the industry when we are tasked with grueling datasets. It would be good to know that the language that we are spending a lot of our time on is a language that will be used for many more years in the future and developed further to make other derivative languages.

Background & Differences

Let’s talk about the background of each language a little bit, along with the differences in their usage in data analysis.

Python:

- Written with simple syntax and structure in mind, meaning that the learning curve for new users is very friendly.
- Used excessively in the production world, Python is very versatile and general-purpose.
- Python’s data analysis libraries are developing rapidly, with the already full-fledged libraries like NumPy and Pandas leading the way in Pythonic way of statistics.
- It is pretty accepted that in the world of machine learning, Python is ahead of other languages in terms of availability of machine learning libraries like Tensorflow, scikit-learn, and others. If the future of data science is actually in machine learning, learning Python could be a good investment.
- Graphing in Python is ok. Matplotlib is their major graphics library, and there are other libraries like Seaborn that are built on top of it.

R:

- Written by hard-core statisticians, so be warned if you are stuck learning it for the first time by yourself without the help of R gurus who have walked the paths of learning before.
- Meant to be used for data analysis on a single machine usually. Not really optimized for speed or efficiency, but supposed to be used with hundreds of libraries that are available through Cran.
- Speaking of libraries, statistically models are brought into existence with a line of code and some arguments. Statistical research made easy.
- What about machine learning in R? It’s supported with libraries like caret, but they are not at the level of Python’s libraries.
- Graphing in R is made pretty easy through ggplot2 and other libraries. Most people in the industry think ggplot is one of the most elegant ways of transforming data into visual elements with very easy-to-work-with syntax, and graphs that are just visually pleasing.

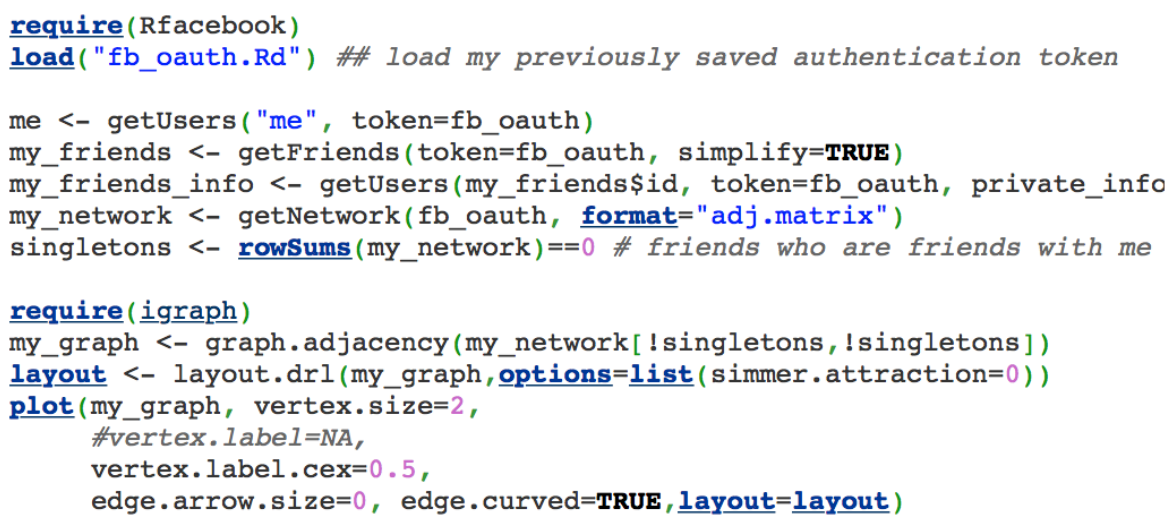
Examples

Let’s dive into some examples!

facebook
December 2010

Facebook Friends in R

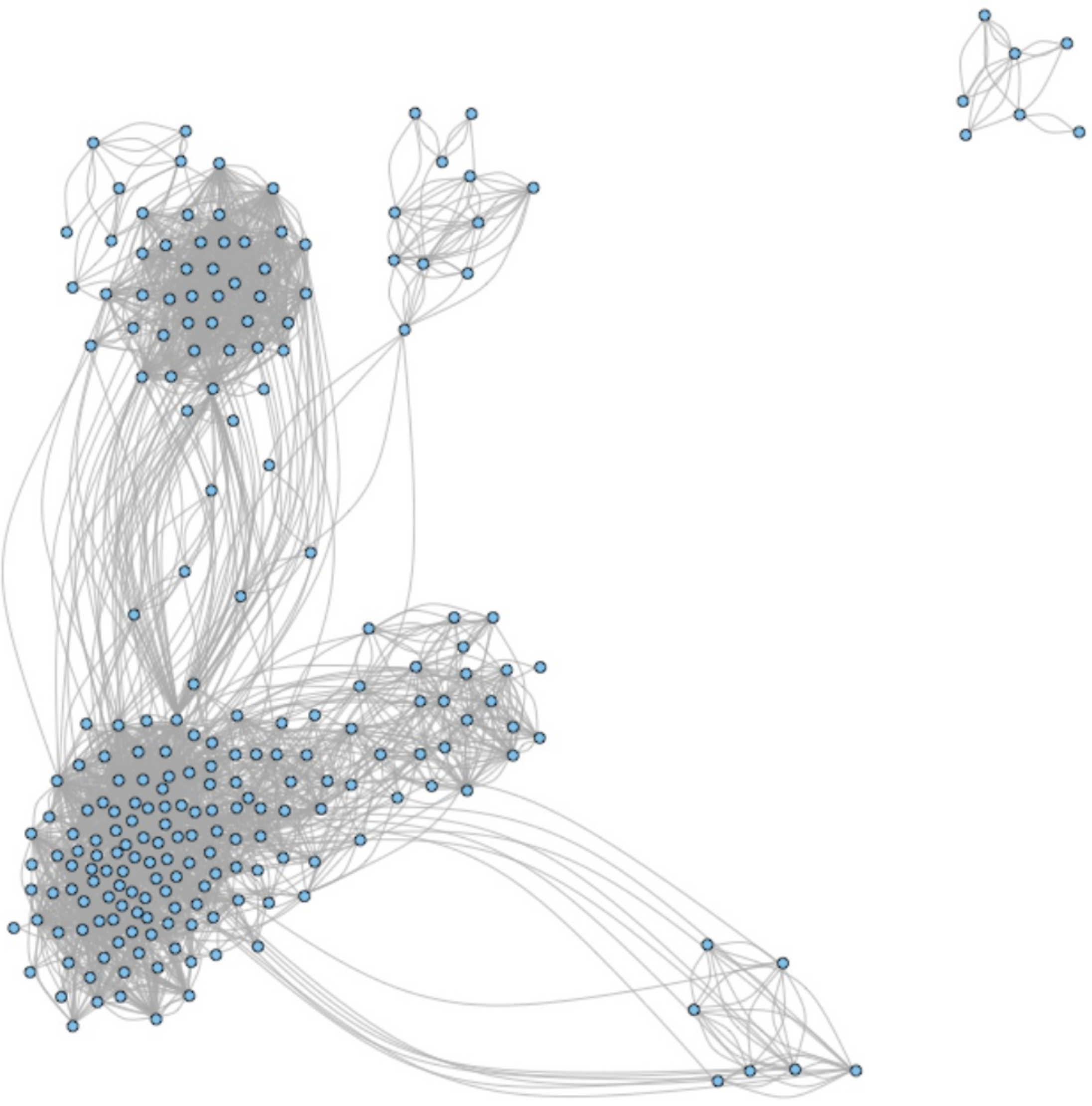
This picture became famous on the web for providing a cool way of visualizing the web of friendships on the world’s biggest social media website and it also attracted attention because it was done all in R!


```
require(Rfacebook)
load("fb_oauth.Rd") ## load my previously saved authentication token

me <- getUsers("me", token=fb_oauth)
my_friends <- getFriends(token=fb_oauth, simplify=TRUE)
my_friends_info <- getUsers(my_friends$id, token=fb_oauth, private_info
my_network <- getNetwork(fb_oauth, format="adj.matrix")
singletons <- rowSums(my_network)==0 # friends who are friends with me

require(igraph)
my_graph <- graph.adjacency(my_network[!singletons,!singletons])
layout <- layout.drl(my_graph,options=list(simmer.attraction=0))
plot(my_graph, vertex.size=2,
     #vertex.label=NA,
     vertex.label.cex=0.5,
     edge.arrow.size=0, edge.curved=TRUE,layout=layout)
```

This is the R code (Reference 7) to draw out your friend network and as you can see, it is pretty elegant. (I was not able to do one for myself cuz I have deactivated my Facebook!) You can see that the top half is actually grabbing the data necessary from Facebook's R library, and only the bottom half is creating the graph. A library called igraph is used, providing an elegant way of providing arguments to the graph, producing:



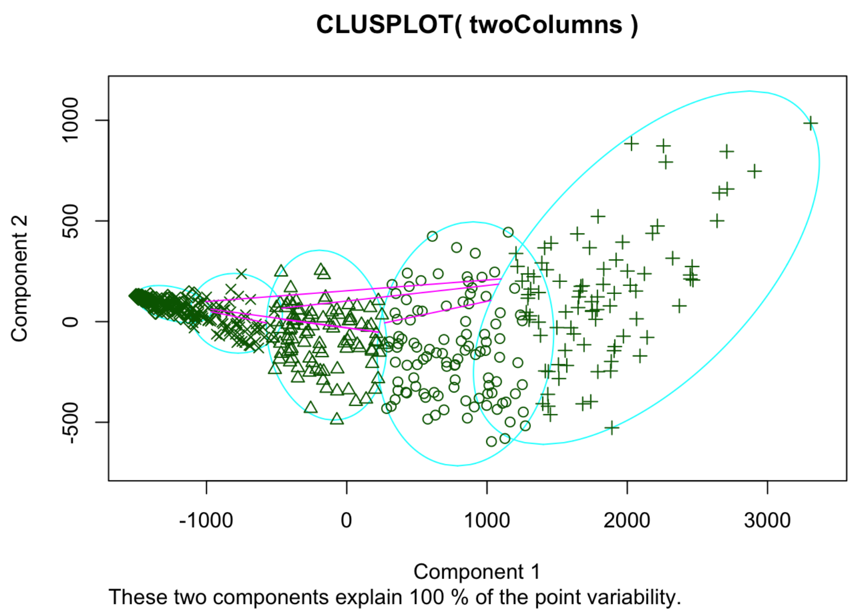
a picture like this.

Let's dive into some actual code chunks.(Reference 6)

```r
#install.package(GGally)
nba <- read.csv("/Users/Jin_Lee/google_drive/berkeley/stat133/stat133-hws-fall17/post01/nba_2013.csv")

library(cluster)
set.seed(1)
isGoodCol <- function(col){
   sum(is.na(col)) == 0 && is.numeric(col)
}
goodCols <- sapply(nba, isGoodCol)
clusters <- kmeans(nba[,goodCols], centers=5)
labels <- clusters$cluster

nba2d <- prcomp(nba[,goodCols], center=TRUE)
twoColumns <- nba2d$x[,1:2]
clusplot(twoColumns, labels)
```



As shown in this example, in the R programming environment, the user can just simply load up a library and start input-ing the right elements. In Python, the user has to rely on a part of the scikit-learn library that deals with clustering for this example; therefore, the function used is not as specialized as that of R.

Conclusions

Through this post, I was able to see that the R language definitely has its strengths in the statistics world because of the prevalence of just so many built-in and CRAN libraries out there. Python is lagging behind at least for now, so R definitely has its edge over Python in this aspec. Also, R has one of the leading ways to graph data through libraries like ggplot, and often times leads to a cleaner implementation of making graphs than Python, as illustrated by the Facebook example.

References

1. https://www.quora.com/Which-is-better-for-data-analysis-R-or-Python-Is-R-still-a-better-data-analysis-language-than-Python-Has-anyone-else-used-Python-with-Pandas-to-a-large-extent-in-data-analysis-projects
2. https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis
3. https://shiring.github.io/r_vs_python/2017/01/22/R_vs_Py_post
4. https://paulbutler.org/archives/visualizing-facebook-friends/
5. https://www.r-bloggers.com/visualize-your-facebook-friends-network-with-r/
6. https://www.dataquest.io/blog/python-vs-r/
7. http://blog.revolutionanalytics.com/2013/11/how-to-analyze-you-facebook-friends-network-with-r.html