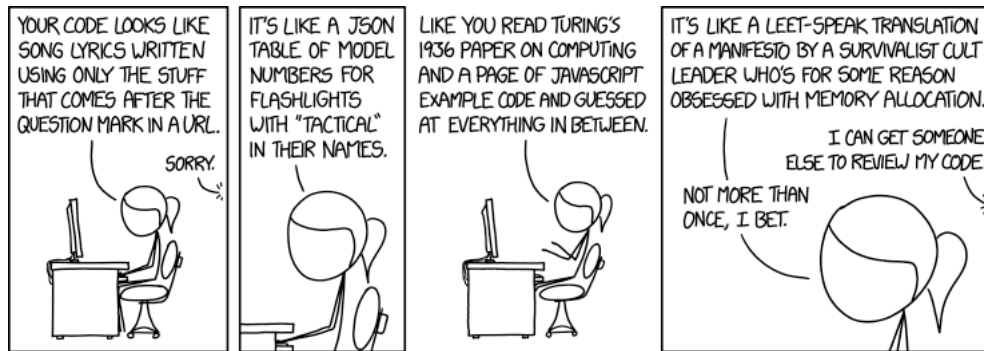


This Class Should be Taught in Python

Idan Hovav

10/31/2017



The average response to any program written in R.

Flaws And Problems in the R Programming Language, and why Python is Better

The R programming language has gaping issues that are elegantly solved in the Python programming language. If the goal of STAT 133 is to focus on Concepts of Computing with Data, as its title purports, R takes away much more often than it aides.

For the believers, mathematical at heart, I offer a one-line proof of my bodacious claim:

```
what.is.a.better.programming.language('r', 'python')
```

```
## [1] "python"
```

For those skeptics, who have either not had the privilege of getting lost in the manual pages of one of the “_apply” functions, let’s dive into why R is not the business, and python deserves your attention. For our first point:

Is R Easy To Learn? A Syntactic Analysis

```
is.r.easy.to.learn("vote on your phones!!!!!!one!")
```

```
## [1] FALSE
```

The R programming language’s learning curve is infamous in the data science world. One especially enlightened scientist, Tal Yarkoni, offers:

```
"R gives you a certain amount of rope to work with. If you don't know what you're doing at all, you will most likely end up accidentally hanging yourself[.]"
```

```
## [1] "R gives you a certain amount of rope to work with. If you don't know what you're doing at all, you will most likely end up accidentally hanging yourself[.]"
```

Let’s dive into why he would dare say such a thing. Good programming languages offer the programmer the ability to be clear, effective, and concise in their programs. They offer beautiful things such as uniformity. Let’s give an example of a programming language that does this well. Oh I don’t know, let’s look at python perhaps.

Indexing in Python

When indexing in python, there is one (1) indexing syntax. In python, regardless of if you’re using a dictionary or a list, you use brackets to retrieve elements of the data structure. The same is the case for an arr- oh wait, an array is the same thing as a list in Python. Hm, maybe that could be our next point. Backtracking to our current criticism, let’s show some examples of Python indexing:

Indexing with a python list:

```
pythonIsTheBest = [0,2,4,6,8]
print(pythonIsTheBest[3])
```

```
## 6
```

Indexing with a python dictionary:

```
pythonRulesRDrools = {"yes": "python rocks", "#1": ['p','y','t','h','o','n'], 0: "r"}
print(pythonRulesRDrools[0])
print(pythonRulesRDrools["yes"])
print(pythonRulesRDrools["#1"][0])
```

```
## r
## python rocks
## p
```

Notice in the above example that dictionaries can play the same role as a data frame in R.

Indexing in R

Now, let's consider how we can index in R. I am a lazy programmer, as I was taught by my gurus, and will use the built in data frame "mtcars":

```
structure(mtcars)
```

```
##           mpg  cyl  disp  hp drat   wt  qsec vs  am gear carb
## Mazda RX4      21.0    6  160.0 110 3.90 2.620 16.46 0  1    4    4
## Mazda RX4 Wag  21.0    6  160.0 110 3.90 2.875 17.02 0  1    4    4
## Datsun 710     22.8    4  108.0  93 3.85 2.320 18.61 1  1    4    1
## Hornet 4 Drive  21.4    6  258.0 110 3.08 3.215 19.44 1  0    3    1
## Hornet Sportabout 18.7    8  360.0 175 3.15 3.440 17.02 0  0    3    2
## Valiant        18.1    6  225.0 105 2.76 3.460 20.22 1  0    3    1
## Duster 360     14.3    8  360.0 245 3.21 3.570 15.84 0  0    3    4
## Merc 240D      24.4    4  146.7  62 3.69 3.190 20.00 1  0    4    2
## Merc 230       22.8    4  140.8  95 3.92 3.150 22.90 1  0    4    2
## Merc 280       19.2    6  167.6 123 3.92 3.440 18.30 1  0    4    4
## Merc 280C      17.8    6  167.6 123 3.92 3.440 18.90 1  0    4    4
## Merc 450SE     16.4    8  275.8 180 3.07 4.070 17.40 0  0    3    3
## Merc 450SL     17.3    8  275.8 180 3.07 3.730 17.60 0  0    3    3
## Merc 450SLC    15.2    8  275.8 180 3.07 3.780 18.00 0  0    3    3
## Cadillac Fleetwood 10.4    8  472.0 205 2.93 5.250 17.98 0  0    3    4
## Lincoln Continental 10.4    8  460.0 215 3.00 5.424 17.82 0  0    3    4
## Chrysler Imperial 14.7    8  440.0 230 3.23 5.345 17.42 0  0    3    4
## Fiat 128       32.4    4   78.7  66 4.08 2.200 19.47 1  1    4    1
## Honda Civic    30.4    4   75.7  52 4.93 1.615 18.52 1  1    4    2
## Toyota Corolla 33.9    4   71.1  65 4.22 1.835 19.90 1  1    4    1
## Toyota Corona  21.5    4  120.1  97 3.70 2.465 20.01 1  0    3    1
## Dodge Challenger 15.5    8  318.0 150 2.76 3.520 16.87 0  0    3    2
## AMC Javelin    15.2    8  304.0 150 3.15 3.435 17.30 0  0    3    2
## Camaro Z28     13.3    8  350.0 245 3.73 3.840 15.41 0  0    3    4
## Pontiac Firebird 19.2    8  400.0 175 3.08 3.845 17.05 0  0    3    2
## Fiat X1-9      27.3    4   79.0  66 4.08 1.935 18.90 1  1    4    1
## Porsche 914-2  26.0    4  120.3  91 4.43 2.140 16.70 0  1    5    2
## Lotus Europa   30.4    4   95.1 113 3.77 1.513 16.90 1  1    5    2
## Ford Pantera L  15.8    8  351.0 264 4.22 3.170 14.50 0  1    5    4
## Ferrari Dino   19.7    6  145.0 175 3.62 2.770 15.50 0  1    5    6
## Maserati Bora  15.0    8  301.0 335 3.54 3.570 14.60 0  1    5    8
## Volvo 142E     21.4    4  121.0 109 4.11 2.780 18.60 1  1    4    2
```

Let's see how many ways we can get to the miles per gallon, or mpg column of this data frame:

```
mtcars[,1]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4
## [29] 15.8 19.7 15.0 21.4
```

```
mtcars[,c('mpg')]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4
## [29] 15.8 19.7 15.0 21.4
```

```
mtcars$mpg
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4
## [29] 15.8 19.7 15.0 21.4
```

```
mtcars[['mpg']]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4
## [29] 15.8 19.7 15.0 21.4
```

```
mtcars[, (names(mtcars) %in% c('mpg'))]
```

```
## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4
## [29] 15.8 19.7 15.0 21.4
```

```
subset(mtcars, select=c('mpg'))
```

```
##                mpg
## Mazda RX4      21.0
## Mazda RX4 Wag  21.0
## Datsun 710     22.8
## Hornet 4 Drive 21.4
## Hornet Sportabout 18.7
## Valiant        18.1
## Duster 360     14.3
## Merc 240D      24.4
## Merc 230       22.8
## Merc 280       19.2
## Merc 280C      17.8
## Merc 450SE     16.4
## Merc 450SL     17.3
## Merc 450SLC    15.2
## Cadillac Fleetwood 10.4
## Lincoln Continental 10.4
## Chrysler Imperial 14.7
## Fiat 128       32.4
## Honda Civic    30.4
## Toyota Corolla 33.9
## Toyota Corona  21.5
## Dodge Challenger 15.5
## AMC Javelin    15.2
## Camaro Z28     13.3
## Pontiac Firebird 19.2
## Fiat X1-9      27.3
## Porsche 914-2  26.0
## Lotus Europa   30.4
## Ford Pantera L 15.8
## Ferrari Dino   19.7
## Maserati Bora   15.0
## Volvo 142E     21.4
```

```
select(mtcars, c('mpg'))
```

```
##                mpg
## Mazda RX4      21.0
## Mazda RX4 Wag  21.0
## Datsun 710     22.8
## Hornet 4 Drive 21.4
## Hornet Sportabout 18.7
## Valiant        18.1
## Duster 360     14.3
## Merc 240D      24.4
## Merc 230       22.8
## Merc 280       19.2
## Merc 280C      17.8
## Merc 450SE     16.4
## Merc 450SL     17.3
## Merc 450SLC    15.2
## Cadillac Fleetwood 10.4
## Lincoln Continental 10.4
## Chrysler Imperial 14.7
## Fiat 128       32.4
## Honda Civic    30.4
## Toyota Corolla 33.9
## Toyota Corona  21.5
## Dodge Challenger 15.5
## AMC Javelin    15.2
## Camaro Z28     13.3
## Pontiac Firebird 19.2
## Fiat X1-9      27.3
## Porsche 914-2  26.0
## Lotus Europa   30.4
## Ford Pantera L 15.8
## Ferrari Dino   19.7
## Maserati Bora   15.0
## Volvo 142E     21.4
```

```
select(mtcars, mpg)
```

```
##          mpg
## Mazda RX4          21.0
## Mazda RX4 Wag      21.0
## Datsun 710          22.8
## Hornet 4 Drive      21.4
## Hornet Sportabout   18.7
## Valiant             18.1
## Duster 360          14.3
## Merc 240D           24.4
## Merc 230            22.8
## Merc 280            19.2
## Merc 280C           17.8
## Merc 450SE          16.4
## Merc 450SL          17.3
## Merc 450SLC         15.2
## Cadillac Fleetwood  10.4
## Lincoln Continental 10.4
## Chrysler Imperial   14.7
## Fiat 128             32.4
## Honda Civic          30.4
## Toyota Corolla       33.9
## Toyota Corona        21.5
## Dodge Challenger     15.5
## AMC Javelin          15.2
## Camaro Z28           13.3
## Pontiac Firebird     19.2
## Fiat X1-9            27.3
## Porsche 914-2        26.0
## Lotus Europa         30.4
## Ford Pantera L       15.8
## Ferrari Dino         19.7
## Maserati Bora        15.0
## Volvo 142E           21.4
```

```
mtcars %>% select(mpg)
```

```
##          mpg
## Mazda RX4          21.0
## Mazda RX4 Wag      21.0
## Datsun 710          22.8
## Hornet 4 Drive      21.4
## Hornet Sportabout   18.7
## Valiant             18.1
## Duster 360          14.3
## Merc 240D           24.4
## Merc 230            22.8
## Merc 280            19.2
## Merc 280C           17.8
## Merc 450SE          16.4
## Merc 450SL          17.3
## Merc 450SLC         15.2
## Cadillac Fleetwood  10.4
## Lincoln Continental 10.4
## Chrysler Imperial   14.7
## Fiat 128             32.4
## Honda Civic          30.4
## Toyota Corolla       33.9
## Toyota Corona        21.5
## Dodge Challenger     15.5
## AMC Javelin          15.2
## Camaro Z28           13.3
## Pontiac Firebird     19.2
## Fiat X1-9            27.3
## Porsche 914-2        26.0
## Lotus Europa         30.4
## Ford Pantera L       15.8
## Ferrari Dino         19.7
## Maserati Bora        15.0
## Volvo 142E           21.4
```

With R, the possibilities are endless! With R, your program can be Unique™, it doesn't have to look like anybody else's program. With R, you can make a repeated operation look different every single time with ease! There's no better way to confuse novice programmers than by adding a million different options for every basic operation. Or, if you want to make your program even more difficult to read for anyone who doesn't live in your head, R will work for you too!

In seriousness, indexing is one of, if not the most basic data analysis and manipulation tool. By including superfluous indexing options, R announces two things to beginners from the get-go. One, there will be multiple ways to go about doing every single operation in this language. And two, uniformity will not be a priority in this programming language.

These are two very worrying and dejecting messages to receive when starting in on a new programming language. And R does not fail to dissappoint.

Assignments in R

Another prime example of overcomplication is with assignments:

```
this.is.valid <- TRUE
this.also.works <<- TRUE
so.does.this = TRUE
TRUE -> how.about.this
TRUE ->> even.this.is.allowed
```

Wow. The last time I saw such a bloated number of operators for one simple task was at the L&S Office of Undergraduate Advising.



Too many cooks in the kitchen ruin data analysis.

We have five (5) separate operators for the other most simple of simplistic operations in any programming language. The first operator evaluates the right side and binds it to the left. Great. We should be done here, but insteaadddddd

The second operator is called a scoping assignment. It evaluates the right hand side by looking in parent environments for relevant bindings.

The third operator evaluates the right hand side and binds it to the left hand side. The use of this operator is frowned upon outside of function calls. Why you ask? Great Question! Moving on,

The fourth operator evaluates the left hand side and binds it to the right hand side. Really? The fifth operator is called a scopin- wait, haven't I already wri- oh. Why is this necessary????

But wait, you might say, what if there's no better way?? What if we need all five (5)??

Well then, I might say, what if there was another programming language that did this better? With just ONE (1) operator?

Assignments in Python

Let's take a look at python's assignment operator(s):

```
muchBetter = True
theWayThingsShouldBe = True
teachPythonInstead = True
soUniform = True
```

```
try:
    True = noneOfThisSuperfluity
except NameError:
    print("this causes a NameError.")
```

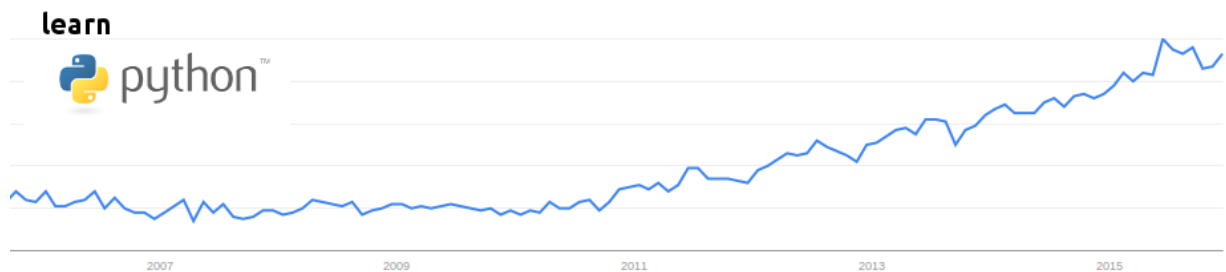
```
## this causes a NameError.
```

As you get deeper into the language, the possible combinations of the different operators quickly causes divergence among all scripts written in R, creating another obstacle to programmers who want to learn the language: reading other people's programs.

An already daunting tasks, diving into another programmer's mind is one of the most often complained about tasks in the industry, however it is imperative in learning any language quickly. By building up this syntactic obstacle to clarity, R successfully impedes beginners' attempts to read scripts written by others.

Conclusion of R's Syntax

In conclusion, R's syntax and plethora of options for uber-simple operations complicates more than it helps. As shown throughout this section, Python excels everywhere that R fails in this regard. Maybe that's why Python is gaining popularity so frequently:



Number of "learn python" search engine queries. There's a new queen in town.

A final point to consider in this argument is the fact that we are more than halfway through this course and a majority of our time has been spent learning R syntax. I propose that we would spend much less time if the language this class was taught in was python.

Also just a random thing that bugs me is allowing periods in variable names. Oh wait did I say random? I meant to say a transition into my next topic.

Is it Worth it To Learn R?

```
is.it.worth.it.to.learn.r
```

```
## [1] "tbh prob not fam"
```

Learning the R programming language is not conducive to learning other programming languages and only leads to a narrow area of application in industry.

Mr. Roger Peng, an R enthusiast and 18-year-long advocate admits that R is an "old technology in the way it was originally designed." Maybe it's outdated design explains its distance from almost any other programming language in common use today. For example, let's reopen the periods complaint. While every other programming language, from Python to Java to C++ to Javascript, uses periods to represent the accessing of a member of an object, r decides to just use it for variable names. Any student who learns r as a first programming language will get hit with confusion if they ever venture into a new language. And believe you me, there's no need to add even MORE ways to name variables. There's enough holy wars going on already:



Two of the countless variable naming conventions in actual programming languages.

Alienating decisions are made throughout the language. Let's highlight a few other key ones.

A Language Without A Paradigm



A pictorial representation of R.

R seems to be very indecisive about what programming paradigm it lives in. It's got a foot in functional programming and another foot in imperative programming. This is a weakness and not a strength for two reasons. Firstly it makes learning and working in the language difficult. Secondly, once a programmer has learned how to program in R, the skills they learned are not transferable to working in other languages. While other languages make a very clear stance in what a paradigm they work in, R fails to cover multiple paradigms for no obvious benefit.

A Language With Underdeveloped Types

The data types in R are aggressively limited, up to the point where it doesn't feel like a programming language at all. Other languages, such as Python, offer a much better variety of data types, and also offer more helpful errors when types are misused. On the other hand, R tends to fail silently or contort and bend the data to fit the situation. These approaches are unhelpful at best and debilitating at worst.

A Language Unfit for the World Wide Web

© Randy Glasbergen
www.glasbergen.com



Another issue with R is its lack of security. This is beyond the scope

of this course, but I bring it up to point out that because of its security weakness, R is rarely run on the open web. Compare this with Python, which has a serious level of security and is used all over the internet. For students who want to be employable in companies that work with the internet, the choice is a no-brainer.

Conclusion

Every computing class must choose a language to be taught in. However, the great classes realize that the language they choose is simply a medium towards an end goal: learning the syntax of a language is merely a means to an end. That end being lofty ideas such as artificial intelligence or efficient data storage. In this course, our lofty end goal is concepts in computing with data. We aim to analyze data, manipulate data, and learn to program. However, this course has been held back by the incessant demands of the R programming language. We spend too much time learning syntax when we should be learning overarching concepts.

During this post, I asked two questions: Is it easy to learn R, and Is it worth it to learn R. Those of you who have followed me here to the bottom of my post can surely agree that the answer to both questions is a resounding "NO". If this course were taught in Python, I believe that the two new questions I'd ask, Is it easy to learn Python, and Is it worth it to learn Python, the answers would be a resounding "YES". That would mean a more valuable course, and a better investment.

Thank you for your time, Idan Hovav

Sources: <https://stat.ethz.ch/R-manual/R-devel/library/base/html/assignOps.html>

<https://www.infoworld.com/article/2940864/application-development/r-programming-language-statistical-data-analysis.html>

<https://www.quora.com/Whats-wrong-with-R-statistical-scripting-language-Can-these-problems-be-fixed-How>

<https://www.talyarkoni.org/blog/2012/06/08/r-the-master-troll-of-statistical-languages/comment-page-1/>

<https://readwrite.com/2013/11/25/python-displacing-r-as-the-programming-language-for-data-science/>

<https://www.r-bloggers.com/good-advice-for-security-with-r/>

<http://r4stats.com/2012/06/13/why-r-is-hard-to-learn/>