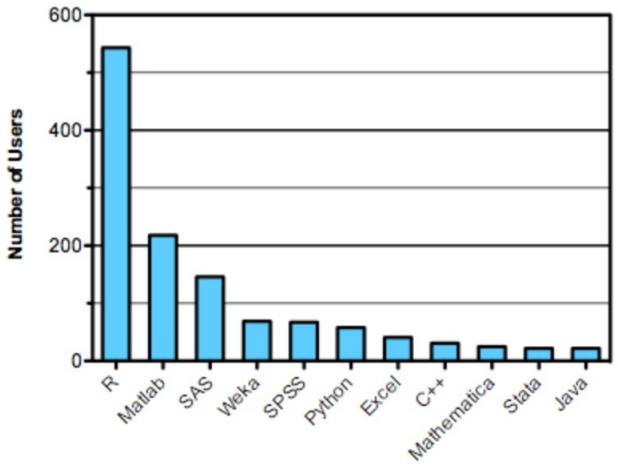
Nonparametric
Python and R programming languages Speed Differences
by Anthony Doan

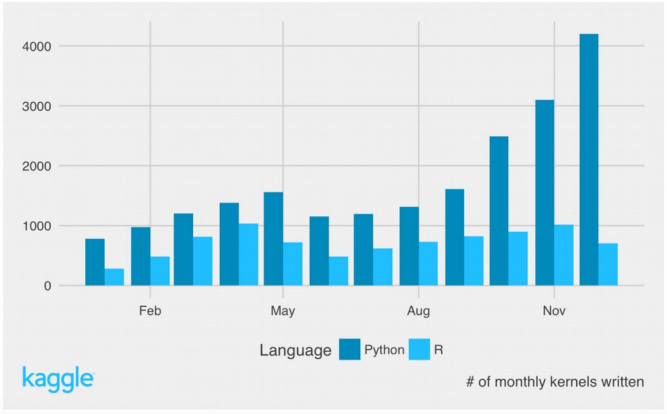
Professor - Dr. Olga Korosteleva
Spring 2017 - May 2, 2017
Class Project Writeup

The project is aim to show that there is a statistical significant difference in speed between R and Python when running classic computational solutions/algorithms.

R and Python are chosen because they are popular programming languages in the data science field.

R was the most used language in 2011 for Kaggle Competitions. In 2016, there is a huge shift toward Python, Python over takes R by a large margin toward the end of 2016.





The source for both Kaggle's graph:

1. http://blog.kaggle.com/2011/11/27/kagglers-favorite-tools/
2. http://blog.kaggle.com/2017/01/05/your-year-on-kaggle-most-memorable-community-stats-from-2016/

	Fortran gcc 5.1.1	Julia 0.4.0	Python 3.4.3	R 3.2.2	Matlab R2015b	Octave 4.0.0	Mathematica 10.2.0	JavaScript V8 3.28.71.19	Go go1.5	LuaJIT gsl-shell 2.3.1	Java 1.8.0_45
fib	0.70	2.11	77.76	533.52	26.89	9324.35	118.53	3.36	1.86	1.71	1.21
parse_int	5.05	1.45	17.02	45.73	802.52	9581.44	15.02	6.06	1.20	5.77	3.35
quicksort	1.31	1.15	32.89	264.54	4.92	1866.01	43.23	2.70	1.29	2.03	2.60
mandel	0.81	0.79	15.32	53.16	7.58	451.81	5.13	0.66	1.11	0.67	1.35
pi_sum	1.00	1.00	21.99	9.56	1.00	299.31	1.69	1.01	1.00	1.00	1.00
rand_mat_stat	1.45	1.66	17.93	14.56	14.52	30.93	5.95	2.30	2.96	3.27	3.92
rand_mat_mul	3.48	1.02	1.14	1.57	1.12	1.12	1.30	15.07	1.42	1.16	2.36

Figure: benchmark times relative to C (smaller is better, C performance = 1.0).

The data set was taken from Julia's website (www.julialang.org).

The code for these program benchmarks are on the websites.

1. For Python: https://github.com/JuliaLang/julia/blob/master/test/perf/micro/perf.py
2. For R: https://github.com/JuliaLang/julia/blob/master/test/perf/micro/perf.R
3. For Julia: https://github.com/JuliaLang/julia/blob/master/test/perf/micro/perf.jl

Friendman's Rank Test

Friedman's Rank Test was used to test for location parameters for Julia, Python and R dependent samples.

Null hypothesis for this test is that the parameters for Julia, Python, and Python are the same. The alternative hypothesis will be at least one of the parameter is not equal.

$$\mathcal{H}_{null} : \Theta_{Julia} = \Theta_{Python} = \Theta_{R}$$
$$\mathcal{H}_{alternative} : \text{at least one not equal.}$$

Friedman's Rank Test code for SAS & R:

1. https://github.com/mythicalprogrammer/nonparametric_datascience_language_speed_compare/blob/master/Friedman_rank_test.sas
2. https://github.com/mythicalprogrammer/nonparametric_datascience_language_speed_compare/blob/master/Friedman_rank_test.R

Summary Statistics for language by rank
Controlling for progproblem

Cochran-Mantel-Haenszel Statistics (Based on Table Scores)				
Statistic	Alternative Hypothesis	DF	Value	Prob
1	Nonzero Correlation	1	10.2857	0.0013
2	Row Mean Scores Differ	2	11.1429	0.0038
3	General Association	4	16.5714	0.0023

Total Sample Size = 21

From the result, looking at “Row Mean Scores Differ”, we can reject the null hypothesis and accept the alternative base on the data.

Wilcoxon's Signed-Rank Test

The main point of this project was to see if R and Python speed are statistically different from each other, that Python is faster than R. So Wilcoxon's Signed-Rank Test will be used to test if R is slower than Python.

$$\mathcal{H}_{\text{null}} : \Theta_R = \Theta_{Python}$$

$$\mathcal{H}_{\text{alternative}} : \Theta_R > \Theta_{Python}$$

Wilcoxon's Signed-Rank Test code for SAS & R:

1. https://github.com/mythicalprogrammer/nonparametric_datascience_language_speed_compare/blob/master/Wilcoxon_signed_rank.sas
2. https://github.com/mythicalprogrammer/nonparametric_datascience_language_speed_compare/blob/master/Wilcoxon_signed_rank_test.R

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	1.58608	Pr > |t|	0.1638
Sign	M	1.5	Pr >= |M|	0.4531
Signed Rank	S	9	Pr >= |S|	0.1563

The test show that we failed to reject the null hypothesis. The p-value is 0.1563 divided by 2, since it is one sided, we can see that it is above the 0.05 mark. If the test was conducted for 90% then there is a statistical difference. But the conclusion will be that there is no statistical speed difference between Python and R. We need more data.

Bootstrap

We can use Bootstrap to see if the average speed differences between Python and R.

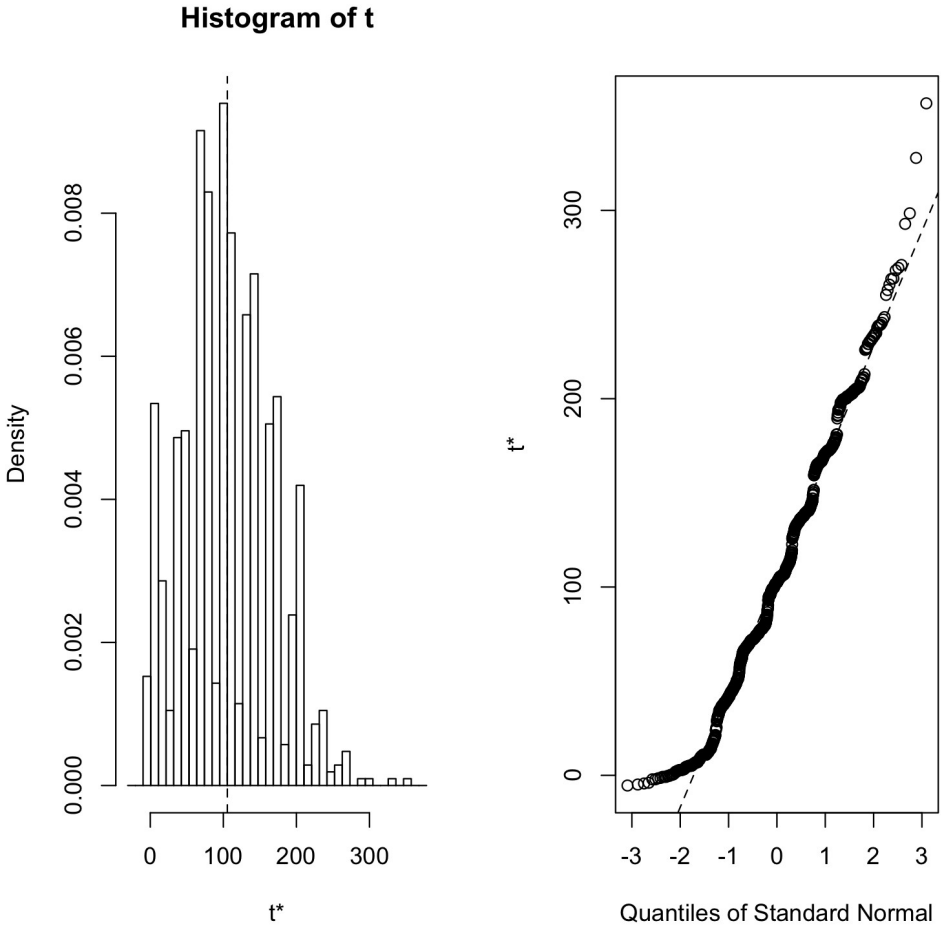
Bootstrap code for SAS & R:

1. https://github.com/mythicalprogrammer/nonparametric_datascience_language_speed_compare/blob/master/Efron_bootstrap.sas
2. https://github.com/mythicalprogrammer/nonparametric_datascience_language_speed_compare/blob/master/Efron_bootstrap.R

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = diff, statistic = samplemean, R = 1000)


Bootstrap Statistics :
    original   bias    std. error
t1* 105.5129 5.252626    61.52373
> plot(results)
> boot.ci(results, type="bca")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca")

Intervals :
Level       BCa
95%   ( 15.0, 269.7 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

The average difference between Python and R program speed is 105.5129. The confident interval for this average is between 15 to 269.7 with standard error of 61.52373.

A graph of bootstrap



Conclusion

We need more data for Wilcoxon Signed-Rank Test otherwise there is no difference in speed. It is interesting to see that the lack of data shows that is shortcoming the lack of power for our test especially regarding Wilcoxon Signed-Rank Test. The reason why Friedman's Rank Test worked was because Julia is such an outrageously fast language compare to R and Python that it can detect it. The difference is massive.

Unfortunately currently at the time of writing this paper, Julia's ecosystem is small and have not hit version 1.0 yet so the API is subjected to change. I cannot recommend Julia over the more established programming languages within the data science field.