

Getting started with R and Python for data analysis

- Tutors and organizers (1st Meeting) :
 - Felix Salim (Yamada Laboratory-Titech)
 - Mia Fitria Utami (Yamada Laboratory-Titech)
 - Pande Putu Erawijantari (Yamada Laboratory-Titech)



Pre-meeting requirements:

1. Install R and R studio (version + link):

<http://swcarpentry.github.io/r-novice-inflammation/setup.html>

2. Install python (suggestions: anaconda for convenience; miniconda if you have disk space problem):

<http://swcarpentry.github.io/python-novice-inflammation/setup/>

3. Download trial dataset: <http://swcarpentry.github.io/r-novice-inflammation/data/r-novice-inflammation-data.zip>

4. Your own problem set and/or tools/packages of interest :

Optional but very helpful to decide the topic for next meeting

Rundown (flexible)

- Introduction (15 min)
- Q and A (10 min)
- Break (5 min)
- Hands on (45 min): troubleshoot installations; load data; help manual search
- free discussion (15 min)

Who need to learn data analysis?

- Are you dealing with bunch of numbers from experiment(s)?
- Do you want to know more about "world"?
- Are you curious about some trivial things?

If one of the answer is YES! then you need to do data analysis

Gold rules:

NEVER EVER TRUST YOUR TOOLS (OR DATA). So who to trust?



14% respondents have learned R



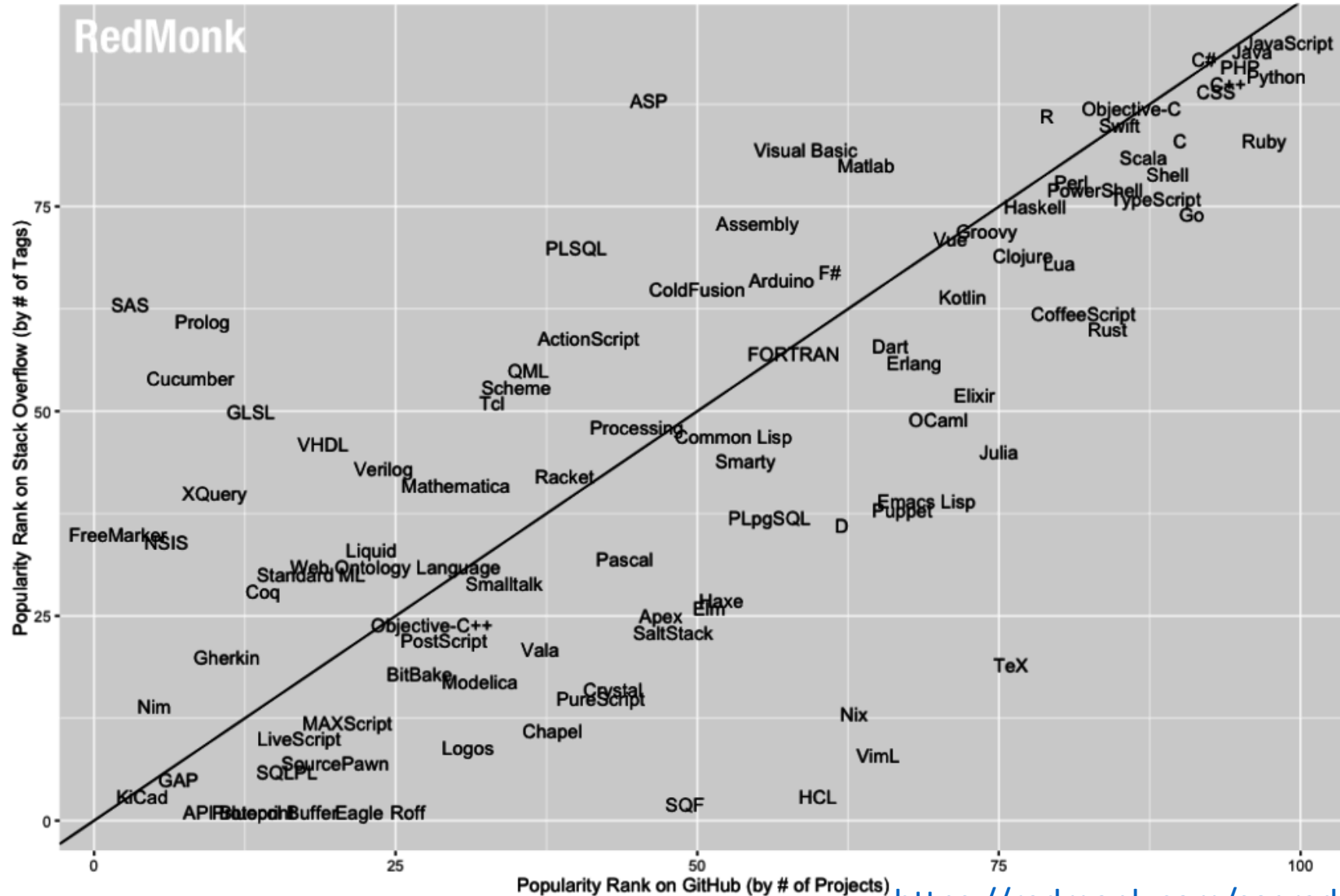
50% respondents have learned python

Why choose both?

It's FREE and POWERFUL !!!

Programing language popularity

RedMonk Q318 Programming Language Rankings





Why R?

- Free and open source
- High level language with widespread usage
- Programming language for statistical analysis and data visualization (can do other things, but not intended for them, consider Python instead)
- Loads of packages for many applications (tidyverse, tidymodels, shiny, etc.)
- Easy to do reproducible research and results sharing (Rmarkdown, Shiny, R packages or R projects)

Popular usages

- Statistical analysis: R is mainly used by statisticians and it has lots of support to do statistical analysis, such as statistical distribution, data modelling and data wrangling
- Data visualization: Publication ready plots can be produced straight from R
- Research sharing: R allows us to share data and script between collaborators easily by organizing it as a package or Shiny web application

R is used mostly in academic or research, especially those that relies on statistics (i.e. biology, healthcare, social studies, etc.)

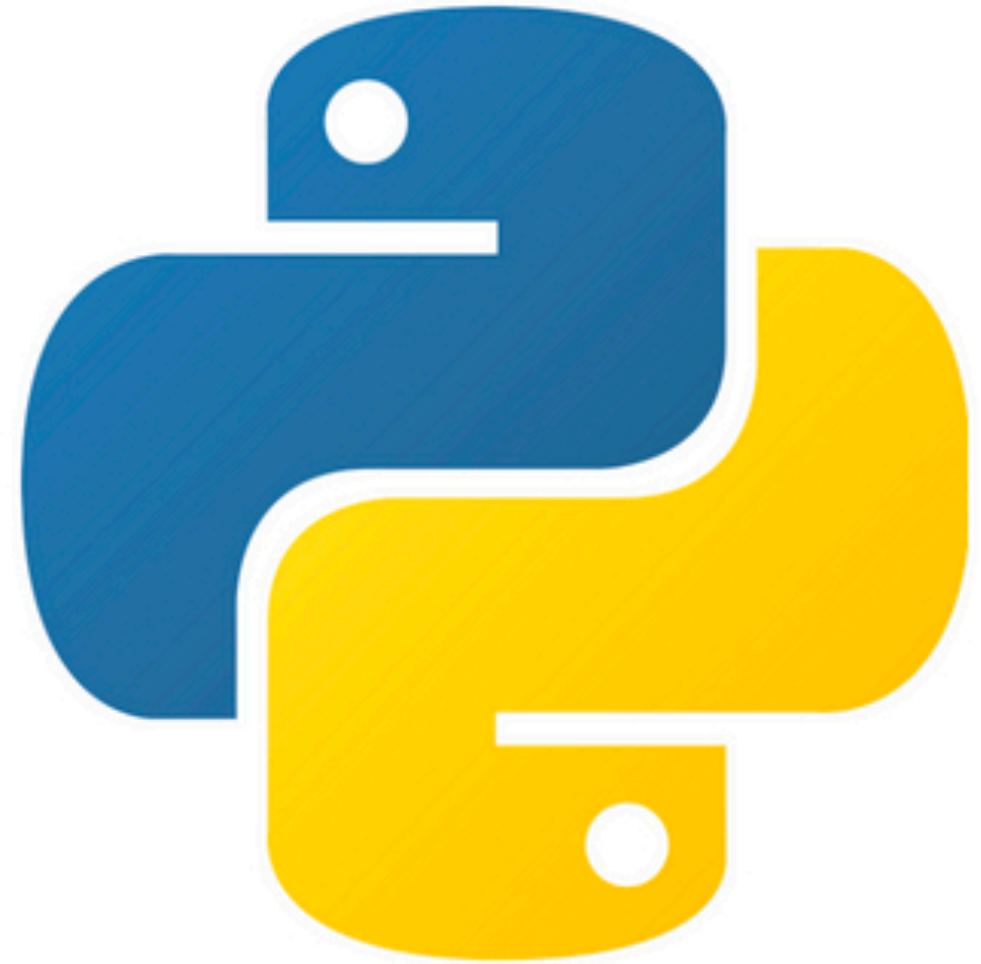
Intro to R studio



- + Friendlier UI compared to basic R
- + Lots of convenient functionality (file explorer, terminal, environment and plot viewer, etc.)
- + Accessible from web browser (Rstudio server)
- + Easy interactive analysis (write and execute)
- Harder to share compared to Jupyter notebooks
- May induce spaghetti code writing
- Multithreaded code may not run properly

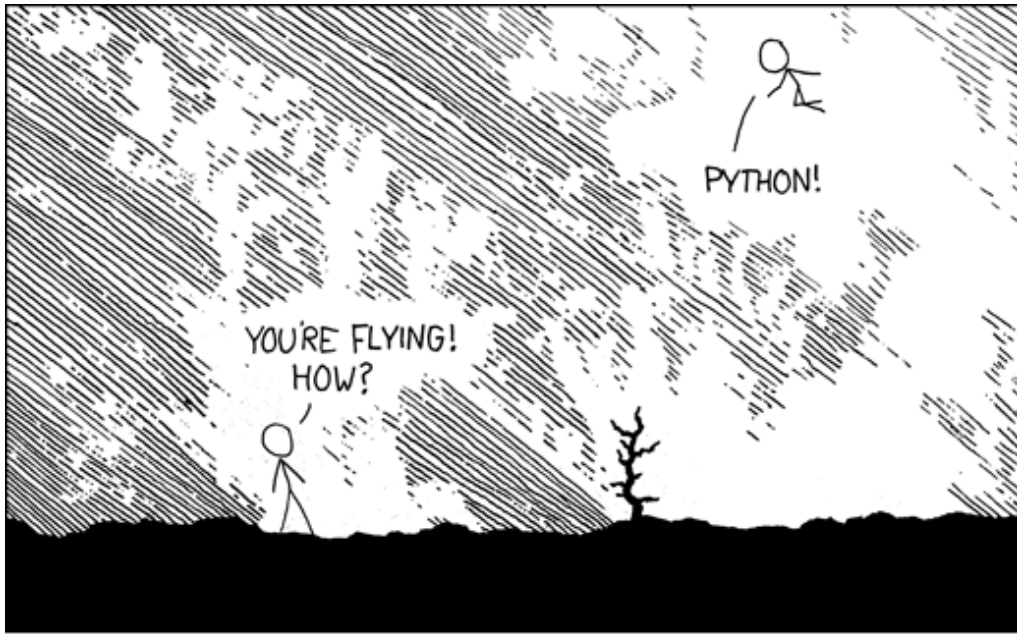
Favorite packages (Felix-Picks)





My Favorite free source (CC-BY license for uncommercial usage):
http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf

Why Python?

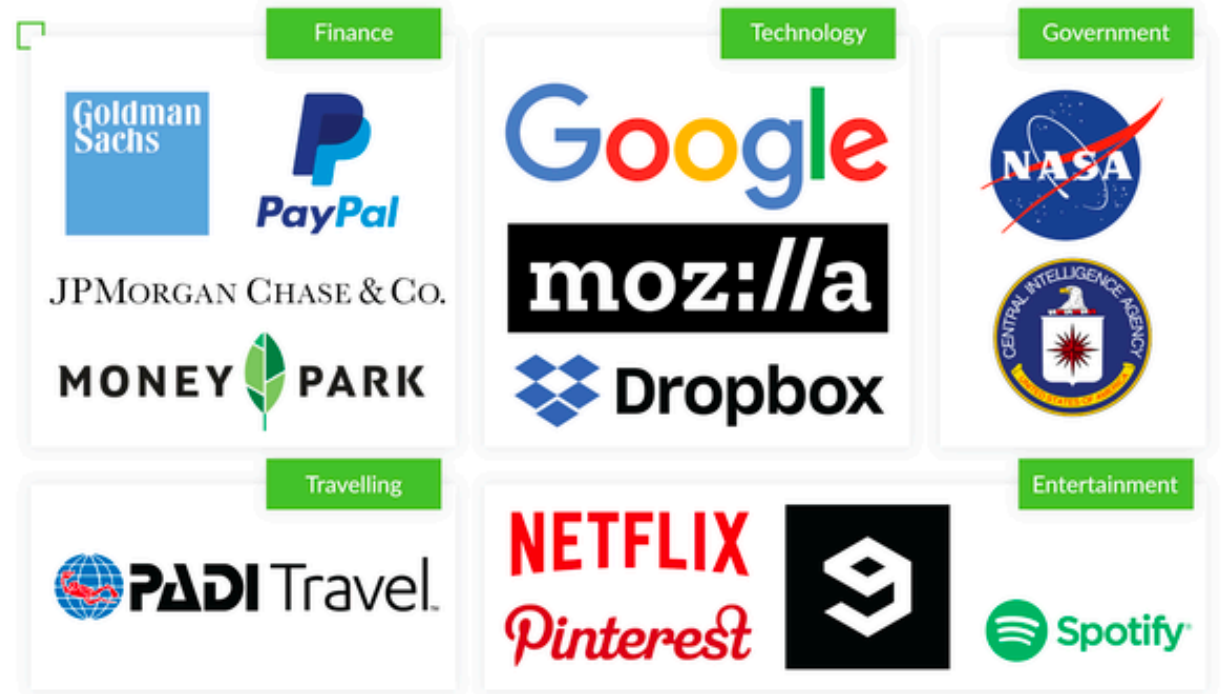


- Free and open source
- Python is a high-level language intended to be relatively straightforward for humans to read and write and for computers to read and process
- General-purpose programming language
- Extensive public libraries (numpy, pandas, scipy, scikit-learn, etc.)
- Python emphasizes productivity and readability

Popular usages

- **Scripting:** expressive and less bulky
- **Application Backends:** Django, Flask, and other server-side web frameworks
- **Scientific Computing:** SciPy/NumPy, Matplotlib, and Pandas
- **Desktop Applications**
- **Mobile Applications**

<https://www.quora.com/What-is-Python-primarily-used-for>



Intro to Jupyter (formerly, IPython) Notebook



The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Few examples of public Jupyter Notebook:

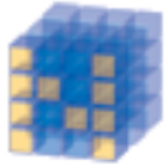
- [LIGO Gravitational Wave Data](#)
- [Satellite Imagery Analysis](#)
- [12 Steps to Navier-Stokes](#)
- [Computer Vision](#)
- [Machine Learning](#)

To run the notebook,
run the following command at the Terminal (Mac/Linux)
or Command Prompt (Windows):

```
jupyter notebook
```

- + Easy data explorations
- + Speak my language
- + Reproducibility
- + Easy online sharing
- Not as convenience as Rstudio

Personal favorite packages



NumPy

Base N-dimensional array package



SciPy library

Fundamental library for scientific computing

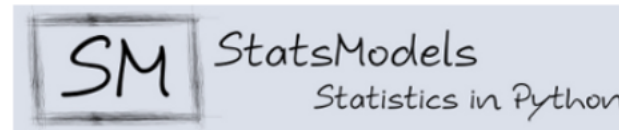
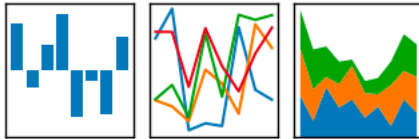


Matplotlib

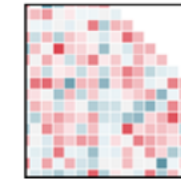
Comprehensive 2D Plotting

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



StatsModels
Statistics in Python



Seaborn





R or Python?

Opinion that I agree on the Internet (Cr: @vsbuffalo)

“Python's main advantage and disadvantage: it's a programming language written by computer scientists.

R's main advantage and disadvantage: it's a programming language written by statisticians.

Python's language features make it pretty great for a lot of things, and numpy is sort of an engineering marvel (I see why astronomy folks are so crazy for it). R has billion packages, tidyverse, bioc, and it's own nice language features (formulas!). In total:

- I ❤️ R
- I ❤️ Python”

So choose wisely:)

Practical use in research

[illegible]

e.g. ~5GB/sample; >100samples

Python + bash + HPC + tons of tools + stack overflow

	Control 1	Control 2	Case 3	Case 4
Feature 1	0.05127856	0.579605386	0.360309588	0.08239625	0.360309588
Feature 2	0.065766244	0.579605386	0.354275032	0.102283303	0.354275032
Feature 3	0.065766244	0.579605386	0.08239625	0.579605386	0.08239625
.....	0.102283303	0.579605386	0.130373517	0.664904935	0.130373517
.....				

Python + R + stack overflow

Visualizations

Let's have fun!!

‘Common error is simple, simple error is common’

Want some more?

- Self-paced get to know basic (vocabulary, simple technique)
 - codecademy
 - R: <https://www.codecademy.com/learn/learn-r>
 - Python: <https://www.codecademy.com/learn/learn-python-3>
- Self learning sources:
 - - Free online courses:
 - udemy
 - datacamp
 - coursera

Want some more? (2)

- Learning repositories:
 - The carpentries (our main source for meeting): <https://carpentries.org/> (data, software, library carpentry)
 - Kamis data (Indonesia): <https://github.com/indo-r/kamisdata>
 - Data is beautiful: <https://www.reddit.com/r/dataisbeautiful/>
 - twitter (cool UNIX/Linux command line tricks): @climagic
 - +one liner specific for bioinfo: <https://github.com/crazyhottommy/bioinformatics-one-liners/blob/master/README.md>
 - and many more....
- Challenge:
 - Kaggle: <https://www.kaggle.com>

Vote for next:

- Data wrangling (Text editing R and/or python) --> DNA Seq data will be given as default
- What statistic tools to choose in R and/or python --> multivariate tables with at least two group
- Data visualizations R vs python --> gapminder data?
-
-
- Any idea?