Please make sure that you have

- Installed Anaconda for Python 3.7 on your laptop (refer to the announcement on ICON)
- Downloaded all class materials from the Module 1 on ICON

MSCI:6040
Data Programming in Python

Introduction

Kang-Pyo Lee

Outline

- Introduction to the Instructor
- Introduction to the Course
- Python and Jupyter Notebook
- Getting Familiar with Jupyter Notebook
- Project Announcement

Instructor

Name: Kang-Pyo Lee

Motto: "Learn from data!"

Education: Seoul National University, Ph.D. in Computer Science

Previous Work: Data Scientist at Samsung Big Data Center

Current Work: Lecturer at Business Analytics, Tippie College of Business

Data Scientist at Informatics Initiative and ITS Research Services

Adjunct Assistant Professor at Biostatistics, College of Public Health

Research Interests: data science, social media analytics, text analytics, machine learning

Courses and Workshops

Credit courses

- Data Programming in Python (Business Analytics)
- Big Data Analysis with Python (Biostatics)
- Text Analytics (Business Analytics)

Training workshops

- Introduction to Python Data Analytics (Sep 2019)
- Machine Learning with Python (Nov 2019)
- Web Scraping with Python (Oct 2019)
- Social Media Analytics with Python (~~Jun 2019~~)

Goal & Scope of This Course

This course aims to introduce the principles and practices of data programming, or more specifically, handling, cleaning, processing, and visualizing data, using the Python programming language

Goal & Scope of This Course

The main topics include:

- Introduction to Python and Jupyter Notebook/Hub
- Python basics
- Data manipulation and analysis
- Files and external data sources
- Text processing
- Data visualization
- A glimpse of machine learning and deep learning

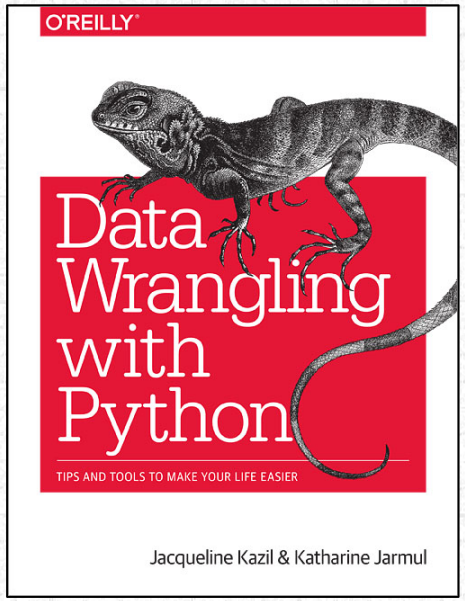
Course Calendar

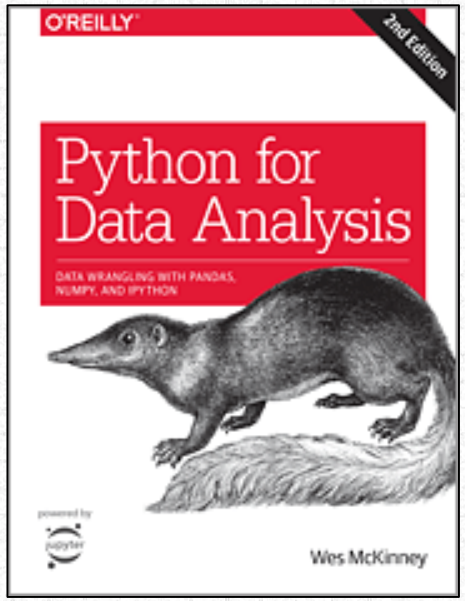
Week	Topic	Due
1 (Aug 26)	Introduction to Python and Jupyter Notebook Group Project Announcement	
2 (Sep 2)	No class (Labor Day)	
3 (Sep 9)	Python Basics Part 1: Data Types, Built-in Functions, and Operators	
4 (Sep 16)	Python Basics Part 2: Flow Control, Functions, Modules and Packages, and Exceptions	HW 1
5 (Sep 23)	Handling Numbers with NumPy Introduction to JupyterHub and UI Interactive Data Analytics Service (IDAS)	HW 2
6 (Sep 30)	Test 1	HW 3 (Sep 29)
7 (Oct 7)	Data Manipulation and Analysis with Pandas	Project Proposal
8 (Oct 14)	Files and External Data Sources Text Processing with NLTK and TextBlob	HW 4
9 (Oct 21)	Midterm Mingle Week Data Visualization with Matplotlib and Ipywidgets	HW 5
10 (Oct 28)	A Glimpse of Advanced Data Analytics: Machine Learning with Scikit-Learn and Deep Learning with TensorFlow	HW 6
11 (Nov 4)	Test 2	HW 7 (Nov 3)
12 (Nov 11)	Group Project Presentations	Project (Nov 10)

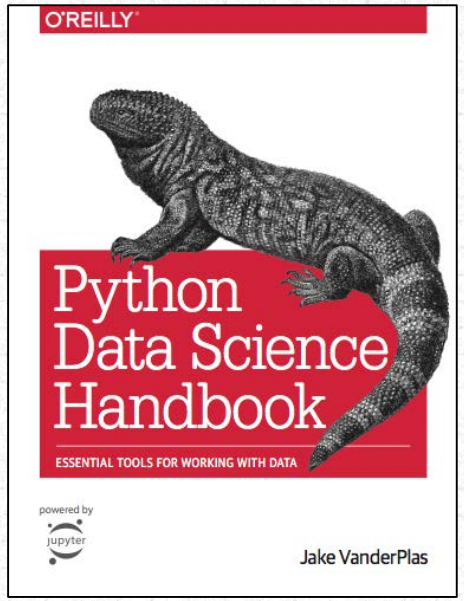
Textbook/Materials

No required textbooks

A few references:







Course Activities

8 formal and active-learning lectures

Individual in-class exercise

7 individual homeworks

2 individual tests

1 group project

Coursework

30% for 7 homeworks

(each week)

50% for 2 tests

(two in-class exams, equally weighted)

20% group project

Final Letter Grades

A: ≈ 50% of students

B: ≈ 50% of students

C, D, F: as needed

The A and B ranges will be equally divided into +/- designations

Late Assignments

- **All assignments are expected on time**
- **You may turn in an assignment late, but you will receive a 20% deduction for each day that it is late, including the first/same day**

Make-Up Exams

- **All students are expected to take tests during the scheduled testing period**

- **Refer to the syllabus**
  - **In the event that you must miss a test**
  - **If you have specific accommodations that have been approved by the university**

Media/System Requirements

- **Please check the ICON course website frequently for announcements, assignments, etc.**

- **You should have access to a laptop computer that you can bring to each class**

Weather Policy

- If bad weather occurs on a class day, please watch the ICON course website and your university email for updated information
- Unless the bad weather occurs suddenly, a decision whether or not to cancel class will be made by about 3:00 PM on that day
- In the event that class is cancelled due to weather, we will use the Zoom system during the regularly scheduled class time
- You can also play back recordings of sessions that you miss

Class Size

47 enrolled students

with 1 instructor and no TAs

Class Rules and Expectations

- **No attendance policy**
- **You may miss regular sessions, but must discuss with me in advance when you'll need to miss any of the tests or group presentation**
- **On homeworks, feel free to discuss, but do not share code**

Office Hours

- **As this is an off-campus course, office hours will be held before or after class**

- **I will also be available via e-mail at kangpyo-lee@uiowa.edu or using Zoom**

Course Syllabus

Refer to the full text of the course syllabus on the ICON course website

Why Data Analytics Tool?

Have your favorite data analytics tool that you feel comfortable using

That will make a difference!

Why Data Analytics Tool?

vs.

Data Analytics Tool

Data Scientist

Python as a Programming Language

python™ is a general-purpose high-level programming language

Python as a Programming Language

Python is a general-purpose high-level programming language

Can be used to build just about anything:

Web development
Data analysis and artificial intelligence
Networking
Scientific computing
Building productivity tools, games, and desktop apps

Python as a Programming Language

Python is a general-purpose high-level programming language

Written in a form that is close to our human language, enabling programmers to just focus on the problem being solved

```
a = "I'm learning Python data analytics."
a.replace("Python", "R")
```

Python is a general-purpose high-level programming language

Advantages:

Easier to modify as it uses English-like statements
Easier/faster to write code as it uses English-like statements
Easier to debug during development due to English-like statements
Portable code – not designed to run on just one type of machine

The nature of Python makes it a perfect-fit for data analytics

Easy to understand and learn
Readable code
Flexible (→ slow)
Easy integration with other apps
Open access to an extensive set of libraries
Active community & ecosystem

Python as a Data Analytics Tool

Python has chosen productivity sacrificing performance

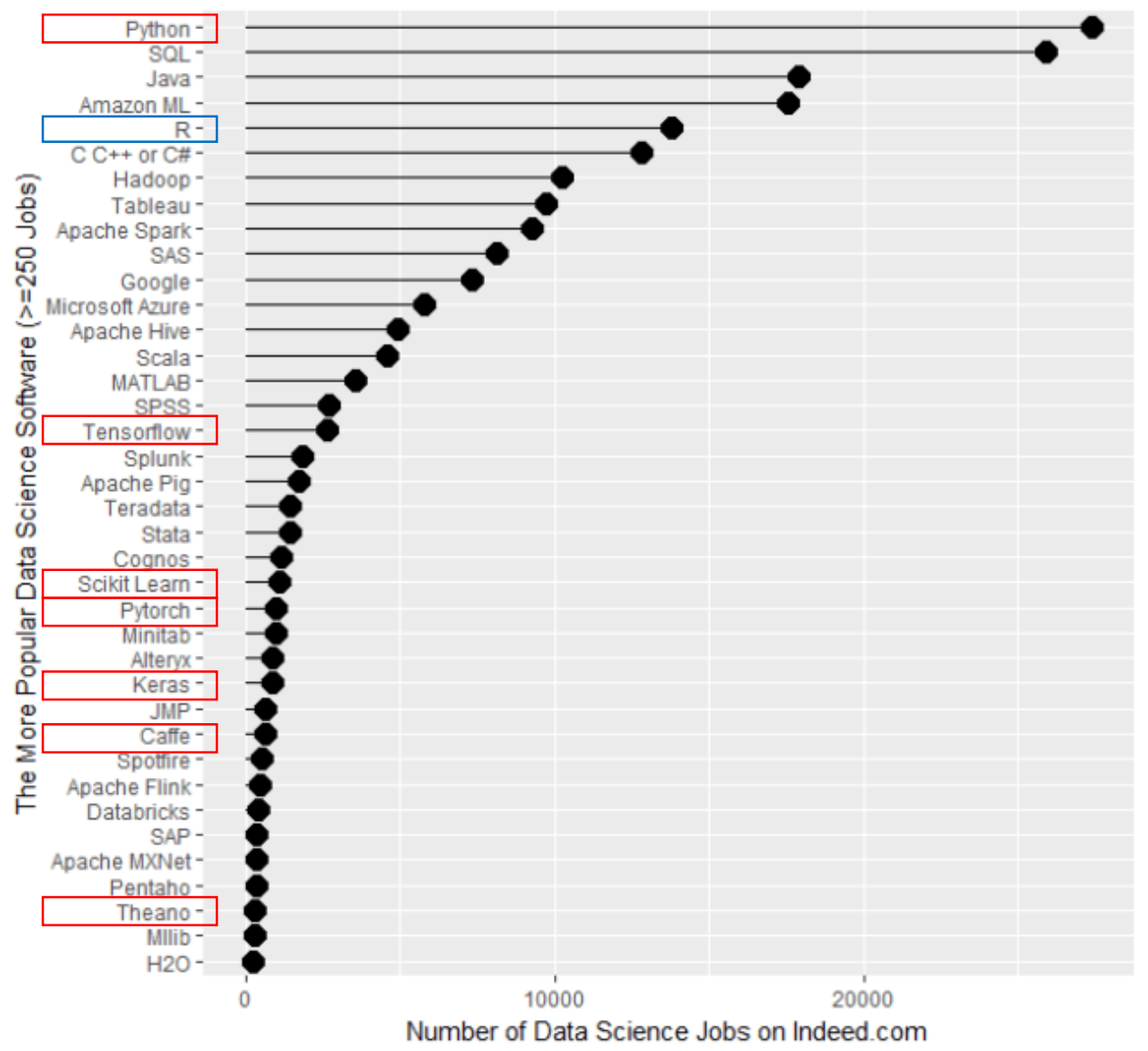
Yes, Python is Slow, and I Don't Care

Python vs. R

python™ **vs.** R

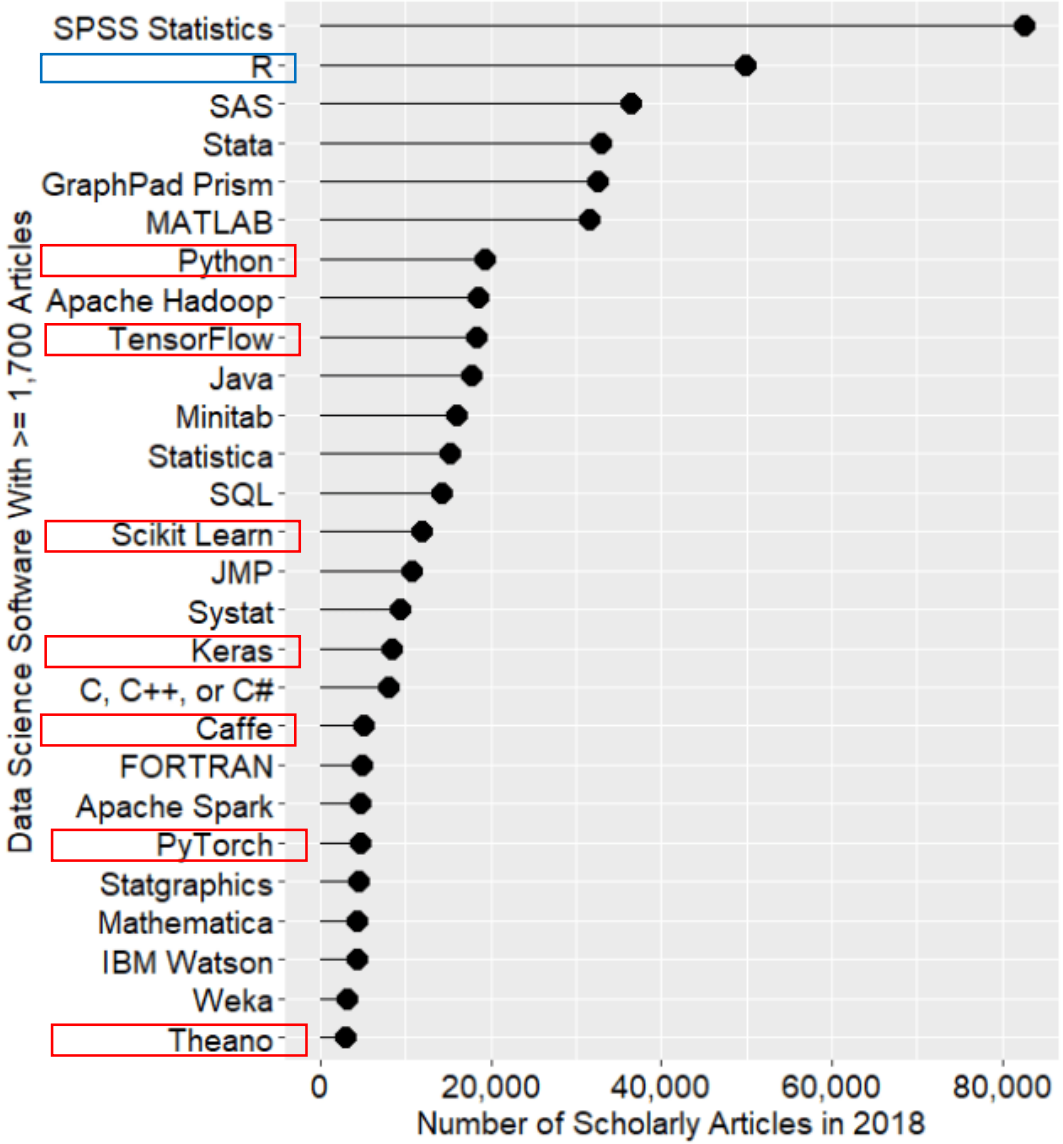
Python vs. R: Why Python and R?

The Popularity of Data Science Software by Robert A. Muenchen



Python vs. R: Why Python and R?

The Popularity of Data Science Software by Robert A. Muenchen



Python vs. R: High-Level Description

Python and R are both <u>open-source</u>, <u>high-level</u> programming languages, <u>actively supported</u> by many developers and users

Python vs. R: Origins and History

	Python	R
Release Year	1991	1995
Creator(s)	Guido Van Rossum (programmer)	Ross Ihaka and Robert Gentleman (statisticians)
Area of Origin	Computer Science	Statistics
Origin of Name	From the "Monty Python's Flying Circus" comedy series	An implementation of the S programming language
Language Type	General-purpose programming language	Special-purpose programming language
Purpose	Productivity and code readability	Better, user-friendly data analysis, statistics and graphics
Target Users	Programmers and developers	Statisticians and scholars
Governing Body	Python Software Foundation (PSF)	R Foundation
Code Repository	PyPI (Python Package Index)	CRAN (Comprehensive R Archive Network)
Current Version	3.7.4 / 2.7.16	3.6.1

Python vs. R: Libraries

Python and R each provide richer libraries in their more specialized areas

E.g.,
Scikit-learn and TensorFlow for Python
Tidyverse and data.table for R

Python vs. R: Libraries

	Python	R
Numerical/scientific computing	numpy, scipy	matrix, optimx
Data manipulation	pandas	dplyr, data.table
Machine learning	mlpy, scikit-learn	e1071, rpart, nnet
Deep learning	keras, tensorflow, theano	keras, kerasR, tensorflow
Text processing	nltk, gensim	tm, tidytext
Statistical analysis	statsmodels	car, zoo
Network analysis	networkx	igraph
Visualization	bokeh, matplotlib, plotly, seaborn	ggplot2, plotly, ggVis, htmlwidgets, shiny
Web scraping	beautifulsoup, scrapy, selenium	rvest, Rselenium, xml2

Python vs. R: Performance

Python and R, as high-level, dynamically-typed languages, are known to be generally <u>slower</u> than other lower-level, statically-typed languages

Both focus on <u>productivity</u>

Python vs. R: Popularity

- Among general-purpose programming languages, Python is significantly more popular than R
  - The RedMonk Programming Language Rankings (June 2019)
  - TIOBE Index (July 2019)
- About 55% of data scientists use both Python and R
  - O'Reilly Data Science Survey (2016)
- In the past, R has enjoyed more success in analytics, but Python usage has recently eclipsed R
  - KDNuggets: Top Software for Analytics, Data Science, Machine Learning (2018)

Python vs. R: Strengths of Python and R

python™ **vs.** R

- Can easily be integrated with other applications and systems
- Useful throughout entire data-analysis process
- Well suited for advanced engineering & computing, such as big-data analytics, AI, and GPU's

- Specialized for statistical analysis and data handling, visualization, and reporting
- Provides a rich set of libraries

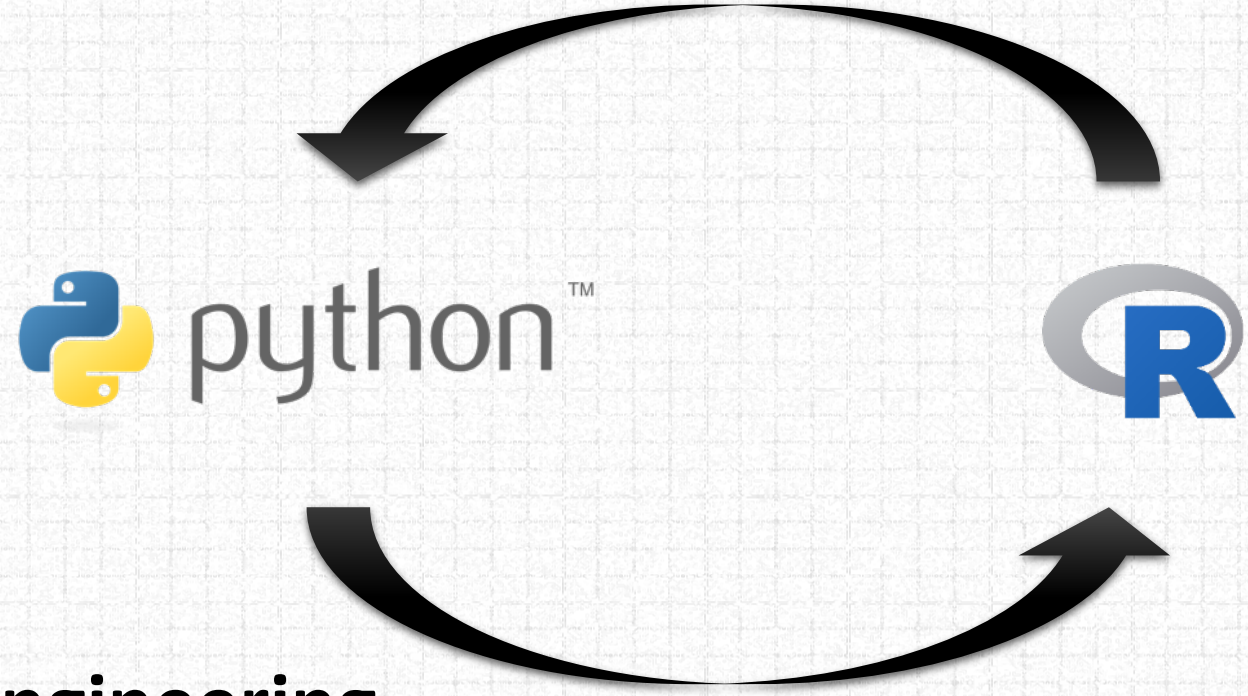
Python vs. R: Weaknesses of Python and R

python™ **vs.** R

- **Slower release of cutting-edge libraries for statistical analysis and visualization**

- **Less use in data science, especially in advanced engineering & computing areas**
- **Limited integration and communication with other applications and systems**

Python vs. R: Feedback Loop



Statistical analysis, visualization & reporting

Advanced engineering & computing

You might choose Python if you…

- Prefer a more traditional programming language and setting
- Focus on machine learning and AI in your analysis
- Want to integrate with other tools (e.g., GPU's, IoT, etc.)
- Need to deploy your analysis at scale
- Want to avoid "base R vs. tidyverse." After all, you just finished transitioning to Python 3

python™

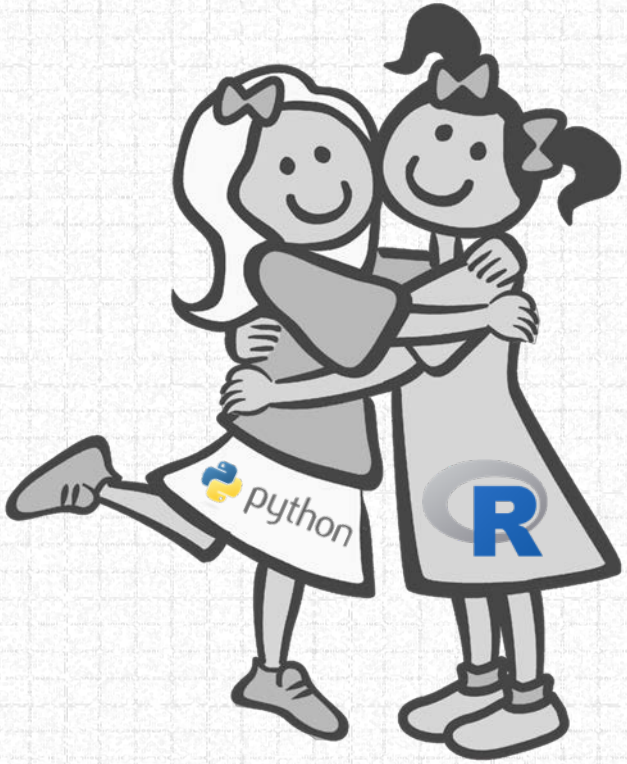
You might choose R if you…

- Prefer an environment crafted specifically for data analysis
- Rely on statistics more than machine learning
- Conduct analysis that is mostly isolated from other systems
- Have connections with academia
- Like RStudio Desktop/Server, tidyverse, and the philosophy of RStudio

Python vs. R: Conclusion

Be ready and willing to use both!

Use what fit your needs!



Comparison with Other Data Science Software

	Proprietary	Open-Source
Traditional	SPSS, sas, STATA, MATLAB, Minitab	python, jupyter, WEKA The University of Waikato, R, RStudio
Latest	rapidminer, tableau, Power BI	Open for Innovation KNIME, orange, H2O.ai

Python Script

A Python script is a text file that contains the statements comprising a Python program

A first way to write and run a Python script

1. Install Python on your computer
2. Write a Python script using any text editor
3. Save the script as a file with the file extension .py
4. Open a command line tool and move to the directory of the script file
5. Type the following and press enter:

```
python filename.py
```

Writing a Python Script

get_evens_odds.py

```
import numpy as np

random_integers = np.random.randint(0, 100, 10)

evens, odds = [], []
for integer in random_integers:
    if integer % 2 == 0:
        evens.append(integer)
    else:
        odds.append(integer)

print(evens, odds)
```

Running a Python Script

Mac

```
kangplee — -bash — 109×11
Last login: Fri May 10 13:53:16 on ttys000
(base) wireless-nat-inside:~ kangplee$ python get_evens_odds.py
[52, 62, 36, 64] [63, 19, 39, 49, 41, 15]
(base) wireless-nat-inside:~ kangplee$
```

Windows

```
Select Anaconda Powershell Prompt
(base) PS>python get_evens_odds.py
[54, 44, 38, 16, 50] [23, 15, 81, 31, 37]
(base) PS>
```

Using Python Shell

Mac

```
kangplee — python — 111×11
Last login: Fri May 10 13:51:15 on ttys000
(base) wireless-nat-inside:~ kangplee$ python
Python 3.7.3 (default, Mar 27 2019, 16:54:48)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Windows

```
Anaconda Powershell Prompt
(base) PS>python
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

iPython & Jupyter Notebook

iPython is a Python command shell for interactive computing

Jupyter Notebook (formerly iPython Notebook) is a web-based interactive data analysis environment that supports iPython

Why Jupyter Notebook?

Interactive

Easy to share

Jupyter Notebook

get_evens_odds.ipynb

Get Random Even/Odd Numbers

- Written by Kang Lee
- Last updated on May 13, 2019

Import modules

In [1]:
```
import numpy as np     # random number generation
```

Generate random integers

In [2]:
```
random_integers = np.random.randint(0, 100, 10)     # generate 10 random integers between 0 and 99
random_integers
```

Out[2]:
```
array([86, 13, 80, 62, 98,  4,  2, 82, 55,  9])
```

Distinguish between even and odd numbers

In [3]:
```
evens, odds = [], []
```

In [4]:
```
for integer in random_integers:
    if integer % 2 == 0:          # if the integer is an even number,
        evens.append(integer)     # add it to the evens
    else:                         # if the integer is an odd number
        odds.append(integer)      # add it to the odds
```

Print the two lists of even and odd numbers

In [5]:
```
print(evens, odds)
```

```
[86, 80, 62, 98, 4, 2, 82] [13, 55, 9]
```

Jupyter Notebook Examples

Delirium Prediction Using EEG Data

Twitter Analysis on the Game of Thrones

Python Data Analytics Libraries/Packages

Useful to know:

- Each library has its own purpose and usage
- In both Python and R, a library takes the form of a package
- Python and R each have their own library repository: PyPI and CRAN
- A library is typically developed, maintained, and upgraded by a team/organization of developers (versioning and dependencies are important!)
- Installing a package is a one-time process. You just load it after installation

Python Data Analytics Libraries/Packages

Python Package

Input → Black Box → **Output**

Parameters

Something you expect from the library

You do not have to implement each component yourself!
All you have to care about is to find a right package and use it in a right way

PyPI (Python Package Index): a repository of software for the Python programming language

Python Data Analytics Libraries/Packages

Reasons you should use commonly-used Python packages rather than writing the code yourself

Convenient to use

Often well-tested

Possibly faster than your code

Popular Python Data Analytics Libraries/Packages

Package	Usage
numpy, scipy	Numerical & scientific computing
pandas	Data manipulation & aggregation
mlpy, scikit-learn	Machine learning
keras, tensorflow, theano	Deep learning
statsmodels	Statistical analysis
nltk, gensim, textblob	Text processing
networkx	Network analysis
bokeh, matplotlib, plotly, seaborn	Visualization
beautifulsoup, scrapy, selenium	Web scraping

Installing Python Packages

On the command line, type and run:

```
pip install [options] PACKAGE_NAME
```

Data Analytics Settings for This Course

Component	Name
Python version	Python 3 (vs. Python 2)
Data analytics environment	Jupyter Notebook (vs. Wing IDE, PyCharm, PyDev, Spyder)
Data analytics software toolkit	Anaconda (vs. Enthought Canopy)
Data analytics libraries	numpy & pandas for data analysis nltk & textblob for text processing matplotlib & ipywidgets for visualization sklearn & tensorflow for machine learning

Getting Familiar with Jupyter Notebook

Run Jupyter Notebook
Handle a notebook
Use R on Jupyter Notebook

https://docs.google.com/document/d/1fxcVd01uKmSkihT-W5UUJGxHcPU8FmYvsHekxn_MCec/edit?usp=sharing

Useful Resources for Learning Jupyter Notebook

Jupyter Notebook for Beginners: A Tutorial
https://towardsdatascience.com/jupyter-notebook-for-beginners-a-tutorial-f55b57c23ada

Advanced Jupyter Notebooks: A Tutorial
https://towardsdatascience.com/advanced-jupyter-notebooks-a-tutorial-3569d8153057

Jupyter Notebook for Beginners: A Tutorial
https://www.dataquest.io/blog/jupyter-notebook-tutorial/

28 Jupyter Notebook Tips, Tricks, and Shortcuts
https://www.dataquest.io/blog/jupyter-notebook-tips-tricks-shortcuts/