Python/R for Data Science

Lecture notes for 2022 Fall at Arkansas Tech University

Xinli Xiao

2022-08-01T16:04:09-05:00

Table of contents

Preface

This is the lecture notes for STAT 2304 Programming languages for Data Science 2022 Fall at ATU. If you have any comments/suggetions/concers about the notes please contact me at my email xxiao@atu.edu.

Part I

Part I: Python

1 Python Fundamentals

1.1 Why Python?

1.1.1 Python is easy to use

Programmers familiar with traditional languages will find it easy to learn Python. All of the familiar constructs—loops, conditional statements, arrays, and so forth—are included, but many are easier to use in Python. Here are a few of the reasons why:

- Types are associated with objects, not variables. A variable can be assigned a value of any type, and a list can contain objects of many types. This also means that type casting usually isn't necessary and that your code isn't locked into the straitjacket of predeclared types.
- Python typically operates at a much higher level of abstraction. This is partly the result of the way the language is built and partly the result of an extensive standard code library that comes with the Python distribution. A program to download a web page can be written in two or three lines!
- Syntax rules are very simple. Although becoming an expert Pythonista takes time and effort, even beginners can absorb enough Python syntax to write useful code quickly.

Python is well suited for rapid application development. It isn't unusual for coding an application in Python to take one-fifth the time it would in C or Java and to take as little as one-fifth the number of lines of the equivalent C program. This depends on the particular application, of course; for a numerical algorithm performing mostly integer arithmetic in for loops, there would be much less of a productivity gain. For the average application, the productivity gain can be significant.

1.1.2 Python is expressive

1.1.3 Python is readable

[1] [2] [3] [4] [5] [6]

1.2 Exercises

Exercise 1.1 (Indentation)**.** Please tell the differences between the following codes. If you don't understand `for` don't worry about it. Just focus on the indentation and try to understand how the codes work.

```
for i in range(5):
    print('Hello world!')
print('Hello world!')
```

```
for i in range(5):
    print('Hello world!')
    print('Hello world!')
```

```
for i in range(5):
print('Hello world!')
print('Hello world!')
```

```
for i in range(5):
    pass
print('Hello world!')
print('Hello world!')
```

Exercise 1.2 (Play with built-in data types)**.**

```
print(True and True)
print(True or True)
print(False and True)
print((1+1>2) or (1-1<1))
```

```
True
True
False
True
```

Exercise 1.3 (Play with strings)**.**

1.3 Projects

jupyter notebook

1.3.1 Gnomic data

lists of data

References

[1] Klosterman, S. (2021). *Data science projects with python: A case study approach to gaining valuable insights from real data with machine learning*. Packt Publishing, Limited.

[2] Ramalho, L. (2015). *Fluent python : Clear, concise, and effective programming: Clear, concise, and effective programming*. O'Reilly Media.

[3] Shaw, Z. A. (2017). *Learn python 3 the hard way*. Addison Wesley.

[4] Shaw, Z. A. (2017). *Learn more python 3 the hard way: The next step for new python programmers*. Addison Wesley.

[5] Ceder, N. R. (2018). *The quick python book*. MANNING PUBN.

[6] Youens-Clark, K. (2020). *Tiny python projects*. Manning Publications.