Lab 7: Python vs. R

Zoe Vernon, Andrew Vaughn, James Duncan

2022-10-14

Table of contents

PDF

Comparing the Behavior of Python and R

This week, we will be exploring the behavior of Python for several common actions that we have previously covered in R. The final product of today's work will be a PDF which is uploaded as a group submission to Gradescope.

Instructions

Please carefully read the full instructions before starting the assignment, as you will need to decide with your group how to organize and combine your efforts.

Your group is expected to work on this for ~**60 minutes**

It's OK if you don't get through all of the questions in one hour (i.e., you can just submit what you have at the end of section), but it should be clear that you put some thought and effort into the questions your group worked on.

> Tip
>
> Do your best to document your efforts and insights as you go.

1. We will separate into groups of (ideally) 2 or 3. I'll try to make sure that each group has at least one person with some R experience.

2. Groups should try to answer every question in the Main Questions section. Here are some **options** for working through these questions:
    - Option A – **Many minds, one task**: The group works together, discussing and answering each question sequentially.
    - Option B – **Divide, consult, conquer**: Individual group members work on different questions, consulting with each other as they go or as needed.
3. At the end of section (say, the last ~10-15 minutes), combine the solutions into **one PDF**. Each group member will then **individually submit** copies of this PDF on Gradescope.
    - Make sure to include the name of each group member at the top of the PDF.
    - How you create the PDF is up to you but some **options** are:
        a. (recommended) Using **Google Docs**.
            - *Separate documents, then combine*: Each group member uses a separate Google Doc, then one person combines those documents into a final shared document.
            - *One document*: Everyone adds to a single document at the same time.
            - You can take screenshots of code / outputs that you run in Jupyter or RStudio (alternatively, the IPython or R consoles), or just copy paste code and outputs directly to the document.
            - When your final shared document is ready, each group member will export it to PDF and make individual submission to Gradescope.
        b. Create an **R Markdown** file with the code and render to PDF. This may be more of a headache to combine if each person is working separately.

No shows

If you miss lab this week, you can form a group of 2-3 students and submit this assignment on Gradescope by Monday October 17th at 10pm.

Questions

Main Questions

> **Tip**
>
> Ideally, you will make it through all of these, although if you run out of time that is okay.

1. Do R functions behave like pass-by-value or pass-by-reference? In other words, if you pass in an object and modify it, does that affect the value of the object in the environment from which the function was called? Check this for a scalar, a list, and an R vector.
2. Can R lists and vectors be modified in place, without copying the object?

    For this the function `.Internal(inspect)` will be helpful. Here's an example for a list.

```
#| eval: false
x <- list(7, c('abc', 'def'), rnorm(5))
.Internal(inspect(x))
```

```
@5652776540f8 19 VECSXP g0c3 [REF(1)] (len=3, tl=0)
@5652776b9740 14 REALSXP g0c1 [REF(3)] (len=1, tl=0) 7
@56527580b0d8 16 STRSXP g0c2 [REF(1)] (len=2, tl=0)
@5652776b97b0 09 CHARSXP g0c1 [REF(4),gp=0x60] [ASCII] [cached] "abc"
@565275b60168 09 CHARSXP g0c1 [MARK,REF(14),gp=0x61] [ASCII] [cached] "def"
@565275a97478 14 REALSXP g0c4 [REF(1)] (len=5, tl=0) -0.38248,-0.100364,-0.485605,1.15111,-0.111647

#`5652776540f8` is the address of the overall list.
#`5652776b9740` of the 1-element vector containing '7'.
#`56527580b0d8` is the address of the character vector.
#`565275a97478` is the address of the vector of random numbers.
```

3. Does R behave similarly to Python in terms of storing strings, as seen in PS4?
4. If you make a copy of an R vector does it use the same memory as the original vector and does changing an element of the original vector affect the copy of the vector?
5. How does variable scoping work in R - does it use lexical scoping and look for variables in the environment where a function was defined?
6. Can you create a closure with embedded data, like we did in Python?

Additional Questions

> **Tip**
>
> Work on these if you finish quickly/are curious

1. Consider the relative efficiency of `for` loops versus vectorized calculations vs. `apply` for numeric vectors in R and see how it compares to the equivalent operations in python.
2. Can you determine if the speed of looking up values in a named vector varies with the size of the dictionary (this will indicate if something like hashing is going on or if the lookup has to scan through all the elements).