Assignment 1.2P

2022 Programming in Psychological Science

Contents

Assignment description

Work through the following 10 Python problems. This assignment is worth 2 points (.2 points per problem). You should also complete Assignment 1.1 for 8 points. The total of Assignments 1.1 (R) and your choice of either 1.2P (this Python assignment) or 1.2R (R challenge) are worth 10 points and 15% of your final course grade. You should try to complete both 1.1 and a 1.2 assignment for full points. You must work individually, but feel free to ask questions in class and on Slack!

You must submit the problems as a Python script (.py file). All answers must have comments and be labeled with the problem number. Some answers will only be comments without associated Python code. IPython terminal-specific commands should be reported as a comment. For questions about output, report your output in a comment under the relevant line of code. Note that all comments have # in front of them so that your Python interpreter does not run this line of code.

```
# Q1.2P.0
# This is the answer the the example problem Q1.2P.0 in a comment
this_variable = 1
this_variable
# Out[3]: 1
```

Problems

Q1.2P.1

```
my_numpy_array = np.array([5, 6, 1])
```

This python code produces an error that the name 'np' is not defined. What line of code did I need to import numpy before this line or at the beginning of my Python script?

Q1.2P.2

Assume Numpy is loaded before all following questions.

```
another_array = np.zeros((2, 4, 6))
```

Return the first element of the first dimension, the 4th element of the second dimension, and the last element of the 3rd dimension using the Python shortcut for a last element.

Q1.2P.3

```
another_array = np.zeros((2, 4, 6))
new_array = another_array
new_array[1, 2, 2] = 1
print(another_array[1, 2, 2])
```

What went wrong in this code? How is this different from R? How can we make a true new copy of the numpy array?

Q1.2P.4

Note that the iPython terminal is a specific **interactive** terminal program that will save Python variables in a workspace like R. Other Windows, Mac, and Linux terminals can run Python scripts but are not as interactive.

Try copying some code and then pasting it into a iPython terminal using the magic function `%paste`. Does the `%paste` function work in a Python script? Why or why not?

Q1.2P.5

What are the magic functions in the iPython terminal to change the working directory, print the current working directory, list the contents of the working directory, and to list current workspace variables?

Q1.2P.6

What is the command that can be run in a terminal or iPython terminal to install the package `pip-install-test`?

Q1.2P.7

```
sample_scores = np.array([1, 6, 7, 8, 9, np.nan])
print(np.mean(sample_scores))
```

This returns `nan`, why? Can you compute the mean with the missing value removed? Check `?np.mean`

Q.1.2P.8

Make a four dimensional numpy array of 2 by 2 by 2 by 2 with as elements the squares of 1 to 16 (e.g., 1^2, 2^2, 3^2, 4^2 etc).

Q.1.2P.9

Refer to problem Q.1.1.32 in Assignment 1.1. Import the InsectSprays .csv file you created into Python as a Python `dictionary`. Note that one method is to use `pandas.read_csv` with one additional step. How can you return the variables or `keys` of this dictionary in Python?

Q.1.2P.10

We used this data to simulate the speed of R and Python on various operations.

```
np.random.seed(1234)  # Set the random seed
speed_sec = np.zeros(10)
sim_speed = np.random.uniform(size=5)  # Speed simulation in seconds
speed_sec[::2] = sim_speed * 10
speed_sec[1::2] = sim_speed
```

Make the following `pandas.DataFrame` without typing out the first two variables. You can use the variable `speed_sec` for the last variable. Note that you can first make a Python `dictionary` before converting to a `pandas.DataFrame`.

```
##   language code_type  speed_sec
## 0        R  forloop1   1.915195
## 1   Python  forloop1   0.191519
## 2        R  forloop2   6.221088
## 3   Python  forloop2   0.622109
## 4        R  forloop3   4.377277
## 5   Python  forloop3   0.437728
## 6        R  forloop4   7.853586
## 7   Python  forloop4   0.785359
## 8        R  forloop5   7.799758
## 9   Python  forloop5   0.779976
```