



15-110 PRINCIPLES OF COMPUTING – F18

LAB 1: PYTHON BASICS

TEACHER:
GIANNI A. DI CARO

Python summary so far

- Scalar type literal objects:
 - **int: Integer relative numbers (\mathbb{Z})**, 1,2,-3
 - **float: Real numbers (\mathbb{R})**, 1.3, 2.0,-3.1
 - **complex: Complex numbers (\mathbb{C})**, 1+2j,1-3j
 - **bool: Boolean (logical) values**, True, False
 - **None: Type with a single value**
- Variables:
 - `x = 2, y = 3.5, okay = False,`
`car_color="red", a = 3//2, a = b + 2,`
`c = a + b, x = y = 2`
- Functions:
 - `def name_of_function(parameters):`
 custom instructions ...
 `return` values
 - `print(): print(2.5), print(x),`
`print(x, y, 3.5), print(x+1, type(x))`
- Non-Scalar type literal objects:
 - **str: String of characters (non-numeric text)**, "Hi", "abc", "Hello!", 'z', 'abc', '_wow_', "I'm Joe", 'Say "hello!" to her'
- Operators (numeric types) objects:
 - **Sum:** `i+j`
 - **Difference:** `i-j`
 - **Product:** `i*j`
 - **Integer division:** `i//j`, $i \div j = j \cdot n + r$
 - **Division:** `i/j` (float)
 - **Modulus:** `i%j`, $i \div j = j \cdot n + r$
 - **Power raising:** `i**j`
- Conversions (implicit and explicit) :
 - `x = 2, y = 3.5 → x = y`
`x = "hello" → x = 2.0, x = 1 + 2/3`
 - `x = int(y), z = float(x)`

Reserved language keywords

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Style writing programs

- **Readability** and **clarity** of a program are essential, for you and for whoever will read / use your programs
- Moreover, **you'll be graded also based on readability and clarity of your homework!**
 - Even if you will pass all tests (i.e., your program is correct and does the expected job), you might lose points if the code is not *properly* written
- *First* rules of thumb of writing clear and readable programs:
 - ✓ Give brief but self-explanatory names to variables and functions
 - E.g., a variable representing interest rate, shouldn't be called, a or b, but rather `interest_rate`
 - A variable representing a color, should be called `color`, not something (unclear) else
 - A function returning a DNA sequence, should be defined as `dna_sequence()`
 - ✓ Use the form with lower letters and underscores to define variables' and functions' names (upper case letters will be used later on for defining classes, for instance, or global variables)
 - Use `car_velocity` instead of `CarVelocity` or `carVelocity`, or other possible forms
 - ✓ Use spaces in statements: `x = 2 + y` is more readable than `x=2+y`, or, `print(x, y, z)` is better than `print(x,y,y)` ...
 - ✓ Use comments using # and ''' to document your code if necessary (we'll see this next time)

A practical problem



- **3 regions**, $R_i, i = 1,2,3$ to where make an assessment of water quality
- **5 water samples** per region, $w_{ij}, i = 1,2,3; j = 1,2,3,4,5$
- Each water sample is a **real number** between 0 and 100, $w_{ij} \in [0,100]$ expressing the presence of a specific pollutant

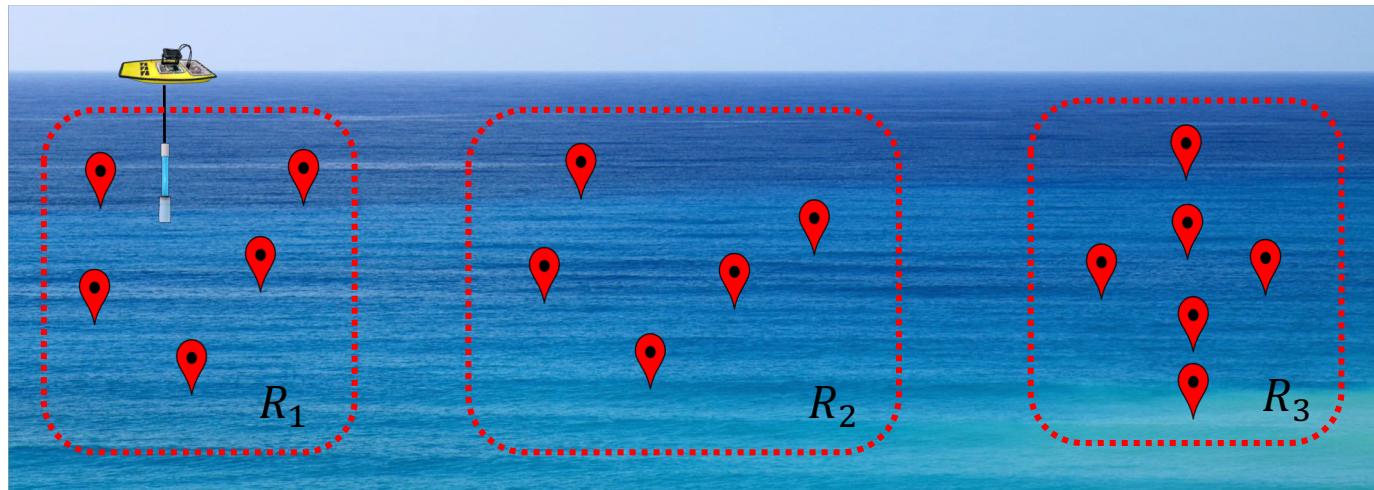
1. For each region $R_i, i = 1,2,3$ I want to compute an estimate of the average presence of a pollutant, $p \rightarrow$ **Arithmetic average!**
2. For the whole zone including the three regions $R_i, i = 1,2,3$ I want to compute an estimate of the average presence of a pollutant, p , averaged over the 3 regions
3. The sample averages for each region I want to see it a **real number**, while the one for the entire zone under monitoring I want to see it as an **integer** since I do not expect decimals playing an important role, but I want to compute also the standard deviation of estimate

$$p_i = \frac{\sum_{j=1}^5 w_{ij}}{5}$$

$$\bar{p} = \frac{\sum_{i=1}^3 p_i}{3}$$

$$\sigma_p = \sqrt{\frac{\sum_{i=1}^3 (p_i - \bar{p})^2}{3}}$$

A practical problem: Data



1.5	12.5	8.9
10.6	11.6	7.6
20.0	21.0	11.0
25.6	15.6	19.6
33.1	30.1	20.1

A practical problem: Code for questions 1 and 2

The screenshot shows the Spyder Python IDE interface. The code editor on the left contains a file named 'temp.py' with the following content:

```
# -*- coding: utf-8 -*-
sum_region_1 = 1.5 + 10.6 + 20.0 + 25.6 + 33.1
sum_region_2 = 12.5 + 11.6 + 21.0 + 15.6 + 30.1
sum_region_3 = 8.5 + 7.6 + 11.0 + 19.6 + 20.1
#print(region_1, region_2, region_3)
avg_region_1 = sum_region_1 / 5
print(avg_region_1)
avg_region_2 = sum_region_2 / 5
print(avg_region_2)
avg_region_3 = sum_region_3 / 5
print(avg_region_3)
avg_all_regions = int((avg_region_1 + avg_region_2 + avg_region_3) / 3)
```

The Variable explorer on the right shows the following variables:

Name	Type	Size	Value
a	NoneType	1	NoneType object of builtins module
avg_all_regions	int	1	16
avg_region_1	float	1	18.160000000000004
avg_region_2	float	1	18.160000000000004
avg_region_3	float	1	13.360000000000003
region_1	float	1	90.80000000000001
region_2	float	1	90.80000000000001
region_3	float	1	66.80000000000001
sum_region_1	float	1	90.80000000000001
sum_region_2	float	1	90.80000000000001

The IPython console at the bottom shows the following output:

```
In [24]:  
In [24]: runfile('/Users/giannidicaro/.spyder-py3/temp.py', wdir='/Users/giannidicaro/.spyder-py3')  
18.160000000000004  
18.160000000000004  
13.360000000000003  
In [25]:
```