Programming for Artificial Intelligence (Python)
Homework 2
Due: March 22 before class

1 Question 1

In class, we learned that the estimates of the parameters of the model $y = f(x; a, b) = a + bx$ include

$$\hat{b} = \frac{s_{xy}}{s_{xx}},$$

where $s_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$, n is the number of cases in the data, $\bar{x} = 1/n \sum_{i=1}^{n} x_i$, and $\bar{y} = 1/n \sum_{i=1}^{n} y_i$.

What is the estimate of a?

2 Question 2

Choose one to answer from 2.1 and 2.2. You can also answer both if you want.

2.1

In Wednesday's class, we learned to create a *set* using the `set` function.

However, we also see some "weird" behavior of the `set` function and the `{}` operator:

```
set(1,2,3)      # raises an error
set((1,2,3))    # creates a set
```

```
set(1)          # raises an error
set((1))        # raises an error
set(1,)         # raises an error
set((1,))       # creates a set
{1,2,3}         # creates a set of three elements
{(1,2,3)}       # creates a set of one element
```

Can you please do some experiments and summarise how to correctly use the `set()` function and the `{}` operator? How do you create the set `{(1,2,3)}` with the `set()` function?

Hint: You can find some clues from

`https://docs.python.org/3/tutorial/datastructures.html#sets`

2.2

Given a dict `dict1`, we can take out an element using `dict1["key1"]` or `dict1.get("key1")`. When the *key* of the element is not present. `dict1["key1"]` raises a "`KeyError`" but `dict1.get("key1")` will not. Try to write a function `myget(d:dict, key:str)→str` so that you can take the value corresponding to the key if the key is in the dictionary but will not raise an error if the key is not in the dictionary. You should not use the `dict1.get` already defined in Python.

For example,:

```
dict1 = {"key1": "value1",
         "key2": "value2"}
def myget(d:dict, key:str)→str:
    pass
myget(dict1, "key1") # should return "value1"
myget(dict1, "key3") # should not raise an error
```

Hint: Use the `try-except` structure.