1. Create a univariate object and try out some of the available methods.

2. Methods to calculate the range and median were planned but not implemented.
    a. Implement one or both of these methods and check they work correctly.
    b. Update the print_summary method so that the median and range are printed when this method is called.

3. Magic methods / dunder (**d**ouble **under**score) methods. See Special method names.
    a. Try using the print() function on an instance of the Univariate class. The output isn't particularly useful.
    b. Have a quick look at the __str__ method in the documentation. Implement your own __str__ method for the Univariate class. You may do this however you please, but one option is to ask Python to print the 'data' attribute associated with an instance of Univariate.
    c. Run print() again on an instance of the Univariate class. Is the output more useful?
    d. What is the difference between the __str__ and __repr__ dunder methods?

4. More magic: addition.
    a. Create two numpy arrays (say, x and y) and then add them (x+y).
    b. Create two Univariate objects and add them using the '+' symbol. What is returned?
    c. Let's imagine we want addition to represent the concatenation of the underlying datasets (this is contrived at best, but I couldn't think of a better example in this context). For example, if u and v represent the data {1,2,3} and {4,5}, then u+v should represent the data {1,2,3,4,5}. Implement an __add__ method for the Univariate class.
    d. The binary operator '+' behaves differently depending on the input objects. Which OOP concept does this relate to? (Hint: check Adam's MF learning slides)

5. Bonus: Break the Univariate class
    a. Identify problems, edge cases, poor design choices, etc.
    b. Are there practices or Python features/libraries which can help fix any of the above?

6. Bonus: Can you think of a class or concept that we could build to help learn good programming practices? For example, we might create a SimSurvey class which reads a sampling frame provided by the user, simulates survey sampling, and calculating estimates under different sample designs.

7. Bonus: Tinker with conda and see what you can make it do.