Using Objects in R

CMSC 702
Zia Khan
Department of Computer Science
Center for Bioinformatics and Computational Biology
OOP in e.g. C++, Python, Java, Ocaml, etc.

- An easy to understand “message passing” style interface.
- Can look up class declaration and/or documentation to figure out how to use the class.

```python
class MyClass:
    def __init__(self):
        self.variableA = 1
        self.variableB = 2

    def hello(self):
        return "Hello"

    def goodbye(self):
        return "Goodbye"

    def ohDear(self):
        return "Have no clue"
```
OOP in R

- Based on generic functions.
- Confusing.
- Many different OOP systems:
  - S3
  - S4
  - Reference classes
  - proto package
  - R6 classes
- I know. This sucks.
- Will try to give you basics for using S3 and S4 classes today since they’re commonly used in Bioconductor.
- Once you understand how to use them, maybe you’ll write some yourself one day.
How do I know if R object is an S3 object?

• Use the pryr package
  – `typeof(myobject)`

• Use R functions.
  – `is.object(myobject) == TRUE & isS4(myobject) == FALSE`
Generic Functions for S3 objects

• To get the class of an S3 object
  – `cname = attr(myobject, "class")`

• To get methods that can be applied to a class with a given name.
  – `methods(class=came)`

• To get S3 classes a method can be applied to e.g
  – `methods("summary")`

• .class portion of function name is used to match class with function
Internal Structure of an S3 object

• Just use str()
  – str(myS3object)
• Usually will be a list type with output like:

  > str(myS3object)
  List of 2
  $ blah : chr "A"
  $ fudge: num [1:3] 2 3 4

• Can be used to read and update members.
• Sometimes structure is complex. Lists of lists.
S4 classes in R

• Add validation mechanisms. Must be created using a constructor function.
• Add inheritance.
• Multiple class types can be matched in generic functions.
• Special @ operator for extracting and updating fields (or slots) from objects.
How do I know if R object is an S4 object?

• Use the pryr package
  – otype(myobject)

• Use R functions.
  – is.object(myobject) == TRUE & isS4(myobject) == TRUE
What’s in this S4 class?

• Get first level of slots.
  – `str(myobject, max.level=2)`

• Get all levels of slots.
  – `str(myobject)`

• To update a slot value.
  – `myobject@a.slot.i.found = “new value”`

• To get class name and names of parent classes.
  – `is(myobject)`
S4 Class Generic Methods

• List methods relevant to a class.
  – `showMethods(class="MyClassName")`

• Call `help(MethodName)` to learn more about generic method/function.
A Few Final Comments

- Reference and R6 classes are closer to message passing classes and easier to use, but upset functional programmers.
- S3 and S4 are most common.
- Look at what they are `is()` and what they store `str()` combined with `methods()` or `showMethods()` and `help()` to figure out how to use them.