

An Introduction to Object Oriented Programming with MATLAB

Emil Nejstgaard

University of Copenhagen, Department of Economics

December 10th 2012

Object Oriented Programming (OOP)

Wikipedia's definition of object oriented programming:

"Object-oriented programming (OOP) is a programming paradigm that represents concepts as "objects" that have data fields and associated procedures known as methods."

An example of an object: A *hand*.

- ▶ We have two instances of that object, left and right
- ▶ The hand has a number of fields defining it: palm, fingers, left hand, right hand etc.
- ▶ It furthermore has methods working on the fields, e.g. fold and stretch fingers:



- ▶ The brain (another object) interacts with the hand through the interface that is the nerves in the arms and shoulders

OOP Compared to classical programming

Procedural Programming

- ▶ Data (structures) and code (functions) are kept separate, data is often treated as global variables.
- ▶ Functions manipulate the input data and create new data in a chronological fashion.

Object Oriented Programming

- ▶ Data (properties) and code (methods) are collected in objects.
- ▶ Objects are (preferably) small independent entities performing a series of specific tasks. Small machines.
- ▶ Data manipulations and construction of new objects and data is not necessarily chronological.

OOP Central Concepts and Definitions

An *object*: A thing that performs a set of related activities.

A *class*:

- ▶ A class is a representation of a given object.
- ▶ It is the blueprint for an object describing all the properties and methods of an object.
- ▶ To create a concrete object, one makes an instance of a class.

Objects communicate with each other through messages using methods called *getters* and *setters*

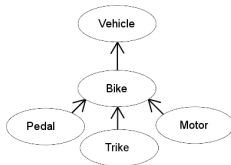
If objects are constructed properly, they can communicate with each other without having further knowledge specific object.

Creating well-functioning classes is an art and is the core of OOP.

OOP Central Concepts and Definitions

The so-called “gods” of OOP:

- ▶ *Inheritance*: Constructing subclasses from super-classes:



- ▶ *Encapsulation*: An object contains all the properties and methods it needs and exposes only the methods used to communicate with other objects.
- ▶ *Abstraction*: Create code that emphasizes on the abstract concept of an object, making it easier to write general code.
- ▶ *Polymorphism*: Allows for method overloading. (similar to generic functions)

Is OOP needed in mathematical programming?

MAYBE, MAYBE NOT. Everything you can do with OOP can be done by classical means! OOP offers a different design paradigm.

My own experience:

- ▶ Advantages

- ▶ Code for tasks that are often repeated (e.g. bootstrap and simulation experiments) can be reused easily
- ▶ Helps to think abstractly about the programming problem at hand
- ▶ Inheritance properties makes it easy to develop code for variations of a model

- ▶ Disadvantages

- ▶ Statistical models as classes tend to become large and clumsy to manage (- i.e. my capabilities of abstract thought are insufficient)
- ▶ We often need to calculate specific things in specific models and code re-useability is less of a concern. - We are not building complex software here!

References

- ▶ Object oriented programming with MATLAB:
 - ▶ [MATLAB classes](#)
- ▶ An introduction to object oriented programming:
 - ▶ www.codeproject.com

MATLAB example

I create an abstract model super-class and two subclasses, one for each of the following models:

$$x_t = \rho_1 x_{t-1} + \varepsilon_t \quad \text{AR}(1)$$

$$x_t = \varepsilon_t + \alpha \varepsilon_{t-1}, \quad \text{MA}(1)$$

Each class contain getters and setters to communicate with other classes as well as estimation and generation routines. (estimation and generation could be their own classes)

Furthermore, I create a simulation class that can use a model class object to simulate parameter distributions.

Note: This is a ridiculous exercise, since MATLAB already contains a well defined class to estimate ARIMA models. You could wrap that class into an object that fits the structure of your program.

See [MATLAB arima class](#)