Object Oriented Programming in R

The topics that are covered in this is:

- Object Oriented Programming: Class and Objects in R

1. Object Oriented Programming

The basic idea of Object Oriented Programming (OOP) is simple.

The first fundamental notion is **object**, which can be anything. For instance, each person is an object. Each object has a number of **fields**, which store information about the object. For instance, the fields of a person include the name, the age, the gender and the height of the person.

The second fundamental notion is **class**, which can be thought as a group of objects which share common fields. For instance, a class can be a group of people.

While you may not have realized, in R everything is an object which belongs to some class.

To retrieve the class of an object, use the class function.

```
a <- 2021
b <- FALSE
c <- "Data Analysis"
class(a)
```

```
## [1] "numeric"
```

```
class(b)
```

```
## [1] "logical"
```

```
class(c)
```

```
## [1] "character"
```

We can declare our own class using **setClass** function. You need to specify two arguments:

- the **name** of the class, and
- **slots**, which is a list of fields.

For each field, you need to specify the class it belongs to. Then you can create an object under the class using **the new function**.

```
setClass( "person", slots=list( name="character", age="numeric", gender="character", height="numeric"))
```

```
person1 <- new("person", name="Sheldon",age=28,gender="M",height=188)
```

To retrieve a field of an object, use `@`.

```
person1@age
```

```
## [1] 28
```

We can write functions that take objects under our own class as arguments.

```
is.adult <- function(x){
  if (x@age >= 18){
    print(paste(x@name,"is an adult."))
    } else {
      print(paste(x@name,"is not an adult."))
    }
}
is.adult(person1)
```

```
## [1] "Sheldon is an adult."
```

```
compare.age <- function(x1,x2){
  if (x1@age == x2@age){
    print(paste(x1@name,"and",x2@name,"are of the same age."))
  } else if (x1@age > x2@age){
    print(paste(x1@name," is older than ",x2@name,".", sep=" "))
  } else {
    print(paste(x2@name," is older than",x1@name,".", sep=" "))
  }
}

person2 <- new("person",name="Amy",age=30,gender="F",height=173)

compare.age(person1,person2)
```

```
## [1] "Amy  is older than Sheldon ."
```

```
compare.age(person2,person2)
```

```
## [1] "Amy and Amy are of the same age."
```