**Module # 7 assignment**

Download any type of data (from the web or use datasets package) or create your own set.

Then, on the second step, determine if generic function as discussed in this module can be assigned to your data set, and if not, why? (Example, here is list of data set in R)  
data("mtcars")  
head (mtcars, 6)  
list(mtcars, 6)

In third and last step, explore if S3 and S4 can be assigned to your data set.  
   
In your blog, discuss the following questions:

* 1. How do you tell what OO system (S3 vs. S4) an object is associated with?
  2. How do you determine the base type (like integer or list) of an object?
  3. What is a generic function?
  4. What are the main differences between S3 and S4?
  5. In your GitHub, create two examples of S3 and S4.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

1. **Download any type of data (from the web or use datasets package) or create your own set.**

# Provide a data set

data**(**"USArrests"**)**

myData **<-** USArrests

I used one of R’s default data packages called USArrests which contains statistics about violent crime rates by us states. I set it to myData.

1. **Then, on the second step, determine if generic function as discussed in this module can be assigned to your data set, and if not, why? (Example, here is list of data set in R)  
   data("mtcars")  
   head (mtcars, 6)  
   list(mtcars, 6)**

# Test for generic functions

head**(**myData**)**

str**(**myData**)**

class**(**myData**)**

# Both use the print() function and are generic

USArrests

print**(**USArrests**)**

To test generic functions on the data set I used head(), str(), and class(). Each function is a generic functions as they all complete a simple function and returns what the UseMethod() declared within the function.

1. **In third and last step, explore if S3 and S4 can be assigned to your data set.**

# Assign S3 to data with in data set

mys3 **<-** list**(**State **=** rownames**(**myData**[**1,**])**, Murder **=** myData**[**1,1**]**, Assualt **=** myData**[**1,2**]**, UrbanPop **=** myData**[**1,3**]**, Rape **=** myData**[**1,4**])**

mys3

**>** mys3

**$**State

**[**1**]** "Alabama"

**$**Murder

**[**1**]** 13.2

**$**Assualt

**[**1**]** 236

**$**UrbanPop

**[**1**]** 58

**$**Rape

**[**1**]** 21.2

#Assign s4 to your data set

setClass**(**"Crime", representation**(**State **=** "character", Murder **=** "numeric", Assualt **=** "numeric", UrbanPop **=** "numeric", Rape **=** "numeric"**))**

mys4 **<-** new**(**"Crime", State **=** rownames**(**myData**[**1,**])**, Murder **=** myData**[**1,1**]**, Assualt **=** myData**[**1,2**]**, UrbanPop **=** myData**[**1,3**]**, Rape **=** myData**[**1,4**])**

mys4

**>** mys4

An object of class "Crime"

Slot "State"**:**

**[**1**]** "Alabama"

Slot "Murder"**:**

**[**1**]** 13.2

Slot "Assualt"**:**

**[**1**]** 236

Slot "UrbanPop"**:**

**[**1**]** 58

Slot "Rape"**:**

**[**1**]** 21.2

As you can tell from my results you are able to assign data from the data frame to S3 and S4 OO. For s3 I used a simple list() as per the example you gave and I had the values correspond to the first entry in the data frame. I used the otype() function to see what data type it’s associated with, however the data type returned is ‘base’ which is technically not s3, but you used list() as an example of a s3. Base is considered a building block for OO.

For s4 I used the new() function you gave to make one. I first however, needed to make a new class. I used the setClass() function to just that with it being called Crime I set the data type of each of the variables associated with the data frame. Afterwards I was able to construct it and got ‘S4’ as a data type returned.

**4.1) How do you tell what OO system (S3 vs. S4) an object is associated with?**

# Install needed packages

install.packages**(**"pryr"**)**

library**(**pryr**)**

# Use otype() to check what data type object is associated with.

otype**(**myData**)**

**[**1**]** "S3"

**>** otype**(**mys3**)**

**[**1**]** "base"

**>** otype**(**mys4**)**

**[**1**]** "S4"

As mentioned before I used the otype() function from the pryr package to do so.

**4.2) How do you determine the base type (like integer or list) of an object?**

# Use class() to determine what data type an object is.

**>** examp **<-** myData**[**2,2**]**

**>** examp

**[**1**]** 263

**>** class**(**examp**)**

**[**1**]** "integer"

Similar to using otype() we just need to use the class() function. I took a random data entry from the data frame to show as an example.

**4.3) What is a generic function?**

A Generic function is an s3 related object that is a function that preforms a simple task by using a dispatcher such as UseMethod() to take an input and run it through a predefined method.

**4.4) What are the main differences between S3 and S4?**

The main differences between the S3 and S4 comes down to two different aspects. S4 contains everything that S3 contains but an S4 OO is defined by a class. Which help it with inheritance with the ability to use special functions for defining other methods. The other is that S4 can use multiple dispatchers. Meaning depending on the inputted data type it can correctly use the right method needed, not limited to only one type.

**4.5) In your GitHub, create two examples of S3 and S4.**

# Create an S3 object

**>** s3 **<-** list**(**Country **=** "USA", State **=** "Florida", City **=** "Tampa"**)**

**>** s3

**$**Country

**[**1**]** "USA"

**$**State

**[**1**]** "Florida"

**$**City

**[**1**]** "Tampa"

**>** otype**(**s3**)**

**[**1**]** "base"

# Or

**>** s3\_2.0 **<-** matrix**(**c**(**1, 2, 3, 4**)**, nrow **=** 2**)**

**>** s3\_2.0

**[**,1**]** **[**,2**]**

**[**1,**]** 1 3

**[**2,**]** 2 4

**>** otype**(**s3\_2.0**)**

**[**1**]** "base"

# S4 object

**>** setClass**(**"location", representation**(**Country **=** "character", State **=** "character", City **=** "character"**))**

**>** s4 **<-** new**(**"location", Country **=** "USA", State **=** "Florida", City **=** "Tampa"**)**

**>** s4

An object of class "location"

Slot "Country"**:**

**[**1**]** "USA"

Slot "State"**:**

**[**1**]** "Florida"

Slot "City"**:**

**[**1**]** "Tampa"

**>** otype**(**s4**)**

**[**1**]** "S4"

For the last question I did pretty much the same formula that I did earlier but with my own data I made up. I also gave a matrix as another example of a s3 type OO, although it was returned as a ‘base’ type object from the otype() function.h