Scripting for Data Science in Python and R

SMU Interdisciplinary Master's Degree in Data Science

Unit 6 - I. an introduction to the week

Eric C. Larson, Lyle School of Engineering,
Computer Science and Engineering, Southern Methodist University

Scripting for Data Science in Python and R

SMU Interdisciplinary Master's Degree in Data Science

Unit 6 - II. data.frames and S4 classes in R

Eric C. Larson, Lyle School of Engineering,
Computer Science and Engineering, Southern Methodist University

lists and data frames in R

- similar to pandas series and data frame


```
> v = list(bob=c(2, 3, 5),
           john=c("aa", "bb"))
> v
$bob
[1] 2 3 5

$john
[1] "aa" "bb"

> v$bob
[1] 2 3 5

> v$what
NULL
```


```
> n = c(2, 3, 5)
> s = c("aa", "bb", "cc")
> b = c(TRUE, FALSE, TRUE)
> df = data.frame(n, s, b)
> df
  n  s     b
1 2 aa  TRUE
2 3 bb FALSE
3 5 cc  TRUE

> mtcars[1, 2]
[1] 6
> mtcars["Mazda RX4", "cyl"]
[1] 6
```

S4 classes in R

- S4 classes are the preferred method of class creation in R
- Caveat: object oriented in R is **awkward**
  - **not a strong suit** of the language
  - actually… it was an **afterthought** made to fit current design
  - its incredibly **convoluted**, and you should **not be happy** with the overall implementation
  - as a programmer it makes me want to 😥

S4 classes in R: initialization

- step one: define class name and properties

```
# now let's create an S4 object of the article
Article <- setClass("Article", slots=list(data='list',keys='character'))
```

- step two: create initializer, if necessary

```
#override the default initializer
setMethod(f="initialize",
          signature="Article",
          definition=function(.Object,data){
              .Object@data <- data # keep the data
              .Object@keys <- names(data)
              return(.Object)
          })
```

- class “Article” is now ready to use if you want:

```
a <- Article(some_article)
```

S4 classes in R: class methods

- step three: define class method as a **global function** 😒

```
#create a new function and define it
setGeneric('getParam',
           def=function(object,
                        param='type_of_material'){standardGeneric('getParam')})
```

- step four: define function

```
setMethod(f='getParam',signature='Article',
          definition=function(object,param){
              if(param %in% object@keys){
                  return(object@data[[param]])
              }
              else{
                  return(c(''))
              }
          })
```

- calling class methods, you need to provide the class instance 😒

```
a <- Article(some_article)
getParam(a,'lead_paragraph')
```

S4 classes in R: inheritance

- step five: define define a subclass of Article

```
ArticleWithMedia <- setClass("ArticleWithMedia",
                             slots=list(data='list',keys='character',media='list'),
                             contains="Article")
```

- step six: initializing inherited class

```
setMethod(f="initialize",
          signature="ArticleWithMedia",
          definition=function(.Object,data){
              .Object@media <- vector("list", length(data$multimedia))
              for(i in 1:length(data$multimedia)){
                  .Object@media[i] <- NYMedia(data$multimedia[[i]])
              }

              # now call the inherited initializer for the class
              callNextMethod(.Object,data)
          })
```

- you have **no control** over initialization hierarchy 😒

Scripting for Data Science in Python and R

SMU Interdisciplinary Master's Degree in Data Science

Unit 6 - III. putting it all together in R

Eric C. Larson, Lyle School of Engineering,
Computer Science and Engineering, Southern Methodist University

extended example

analyzing the New York Times in **R**