JPL-Caltech Virtual Summer School

Big Data Analytics

September 2 – 12, 2014

Ashish Mahabal
California Institute of Technology

R - V

Jet Propulsion Laboratory

Keck INSTITUTE FOR SPACE STUDIES

Sponsored by the Jet Propulsion Laboratory and by the Keck Institute for Space Studies, California Institute of Technology

Caltech

Classes in R

- Google's R Style Guide
  - http://google-styleguide.googlecode.com/svn-history/r106/trunk/google-r-style.html
  - S3 methods are more interactive and flexible
  - S4 methods are more formal and rigorous
  - Avoid mixing S3 and S4:
    - S4 methods ignore S3 inheritance and vice-versa.
- Also base type (using C), and Reference classes (in-place modifications)
  - http://adv-r.had.co.nz/OO-essentials.html

S3 classes

- essentially a list with OO functionality (methods, inheritance)
- generic and casual – not like C++
- No error checking
  - assigning to variables not in class possible (say, by mistake)

Galaxy class

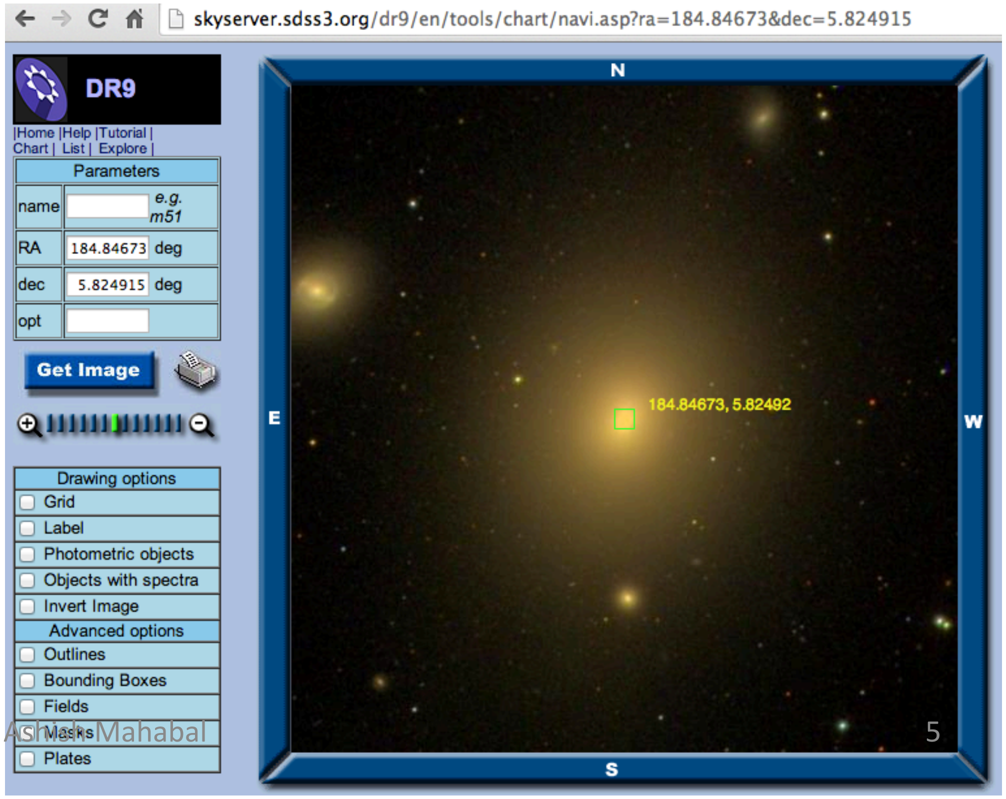
```
R> g <- list(Name = “NGC 4261”, RA = 184.84673,
   Declination = 5.824915, mag = 10.4)
R> class(g) <- “Galaxy”

# Let us check the attributes
R> attributes(g)
$names
[1] "Name"        "RA"          "Declination" "mag"

$class
[1] "Galaxy”
```

Printing info

```
R> g
$Name
[1] "NGC 4261"

$RA
[1] 184.8467

$Declination
[1] 5.824915

$mag
[1] 10.4

attr(,"class")
[1] "Galaxy"
```

Modifying default print

```
#   browseURL(url, browser = getOption("browser"))
# Define a new print for our class “Galaxy” to display an image
R> print.Galaxy <- function(g) {
+     baseurl <- "http://skyserver.sdss3.org/dr9/en/tools/chart/navi.asp?"
+     cat(g$Name,”\n”)
+     g_url <- paste(baseurl,"ra=",g$RA,"&dec=",g$Declination,sep="")
+     browseURL(g_url)
+     }
R> methods(,"Galaxy")
[1] print.Galaxy
R> g         # or print(g)
NGC 4261
# and opens the image in a browser
```



skyserver.sdss3.org/dr9/en/tools/chart/navi.asp?ra=184.84673&dec=5.824915

Parameters		
name		*e.g. m51*
RA	184.84673	deg
dec	5.824915	deg
opt		

If we had misspelt any of the attribute names, R would not have complained.

S4 objects are more rigorous

```
R> setClass("GalaxyS4",
+   representation(
+     Name="character",
+     RA="numeric",
+     Declination="numeric",
+     mag="numeric")
+ )
R> gS4 <- new("GalaxyS4",Name="NGC
   4261",RA=184.84673,Declination=5.824915,mag
   =10.4)
```

```
> gS4
An object of class "GalaxyS4"
Slot "Name":
[1] "NGC 4261"

Slot "RA":
[1] 184.8467

Slot "Declination":
[1] 5.824915

Slot "mag":
[1] 10.4
```

“slots” in S4 are accessible using @ instead of S3’s $

```
> g$Name
[1] "NGC 4261"
> gS4@Name
[1] "NGC 4261"
```

What is ‘print’? It depends on the context

```
# We were able to redefine print
R> print
function (x, ...)
UseMethod("print")
<bytecode: 0x10881b550>
<environment: namespace:base>
R> print.Galaxy
function(g) {
baseurl <- "http://skyserver.sdss3.org/dr9/en/tools/chart/navi.asp?"
cat(g$Name,"\n")
g_url <- paste(baseurl,"ra=",g$RA,"&dec=",g$Declination,sep="")
browseURL(g_url)
}
```

‘Different’ prints (* are in other namespaces)

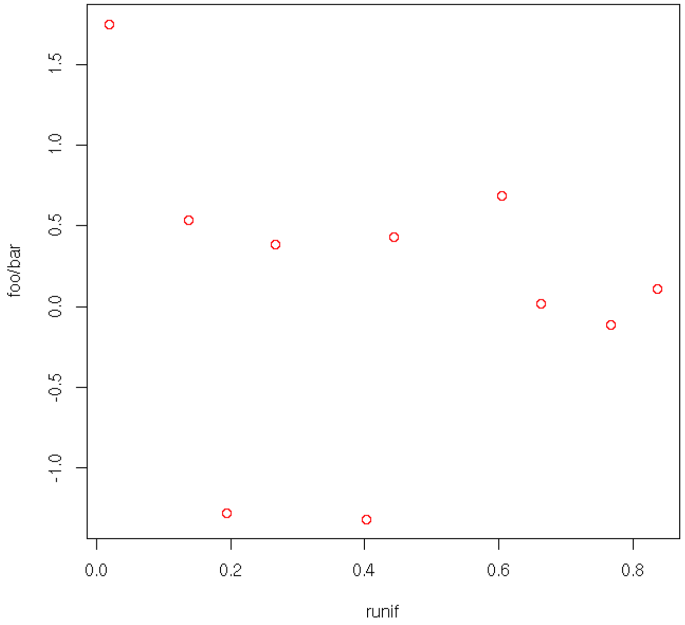
```
> methods(print)
  [1] print.abbrev*
  [2] print.acf*
  [3] print.AES*
  [4] print.anova
  [5] print.Anova*
  [6] print.anova.loglm*
  [7] print.aov*
  [8] print.aovlist*
  [9] print.ar*
 [10] print.Arima*
 [11] print.arima0*
 [12] print.AsIs
 [13] print.aspell*
 [14] print.aspell_inspect_context*
 [15] print.bibentry*
 [16] print.Bibtex*
 [17] print.browseVignettes*
 [18] print.by
```

```
R> print.by          # yes
R> print.aspell      # no
R> aspell            # yes
```

Using R through python

- import **rpy2**.robjects as robjects
- r = robjects.r

- x = robjects.IntVector(range(10))
- y = r.rnorm(10)
- r.X11()



- r.layout(r.matrix(robjects.IntVector([1,2,3,2]), nrow=2, ncol=2))
- r.plot(r.runif(10), y, xlab="runif", ylab="foo/bar", col="red")

Imprecise?

```
R> gS4 <- new("GalaxyS4",Name="NGC 4261",RA=184.84673,Declination=5.824915,mag=10.4)
```

```
R> gS4
```

- Slot "RA":
- [1] 184.8467

```
# Where did the last digit vanish?
```

About precision in R

(from http://stackoverflow.com/questions/2287616/controlling-digits-in-r)

```
R> options(digits=10)          # a suggestion (default 7; 1-22)
R> sprintf("%.100f",pi)        # inaccurate after about 15 places
[1] "3.141592653589793115997963468544185161590576171875000000…"
R> .Machine$double.xmin   # small numbers are not an issue really
[1] 2.225073859e-308
# Can install gmp, rcdd
```

Very small numbers versus significant digits:

- x1 <- rnorm(50, 1, 1e-15)
- y1 <- rnorm(50, 1 + 1e-15, 1e-15)
- t.test(x1, y1) #Should throw an error as numbers almost constant

- x2 <- rnorm(50, 0, 1e-15)
- y2 <- rnorm(50, 1e-15, 1e-15)
- t.test(x2, y2) #ok

Next time ..

- ggplot …