CEFET/RJ

Classes S3

Eduardo Ogasawara
eduardo.ogasawara@cefet-rj.br
https://eic.cefet-rj.br/~eogasawara

Introduction to S3 class

- S3 class is the most popular way of build classes in R
- Most of the classes that come predefined in R are of this type
  - It is simple and easy to build
- A class is a list that is marked as a class

```
# creates an object of class "polygon"
obj <- list(n = n)
# class can be set using class() or attr() function
attr(obj, "class") <- "polygon"
```

"Constructor"

- It is a function that returns a created object of the name of the class

```
polygon <- function(n) {
  if(n <= 0)  stop("number of vertices should be greater than zero")
  obj <- list(n = n)
  # class can be set using class() or attr() function
  attr(obj, "class") <- "polygon"
  return(obj)
}
```

"Inheritance"

- Create a constructor that sets the name of the class and append it upper level hierarchy

```
rectangle <- function(w, h) {
  obj <- polygon(4)
  obj$w <- w
  obj$h <- h
  class(obj) <- append("rectangle", class(obj))
  return(obj)
}
```

Implementing an “interface”

- Adding the support for a previous published interface for a particular class

```
print.polygon <- function(obj) {
  cat(obj$n, "\n")
}

print.rectangle <- function(obj) {
  cat(obj$w, ",", obj$h, "\n")
}
```

Implementing an "interface"

- Adding the support for a previous published interface for a particular class

```
print.polygon <- function(obj) {
  cat(obj$n, "\n")
}

print.rectangle <- function(obj) {
  cat(obj$w, ",", obj$h, "\n")
}
```

Creating an "interface"

- Defines an interface
- Implement a general implementation
- "Override" the implementation in a specific class

```r
area <- function(obj) {
  UseMethod("area")
}

area.default <- function(obj) {
  return(0)
}

area.rectangle <- function(obj) {
  return(obj$w * obj$h)
}
```

Creating an "interface"

- Defines an interface
- Implement a general implementation
- "Override" the implementation in a specific class

```
area <- function(obj) {
  UseMethod("area")
}

area.default <- function(obj) {
  return(0)
}

area.rectangle <- function(obj) {
  return(obj$w * obj$h)
}
```

Finding published interfaces

- Before creating an interface, it is a good practice to check first if it is not already defined

```
methods(class="default")
```

Exploring the classes

```
a <- 3
p <- polygon(5)
r <- rectangle(3, 10)

print(a)
print(p)
print(r)

print(area(a))
print(area(p))
print(area(r))
```

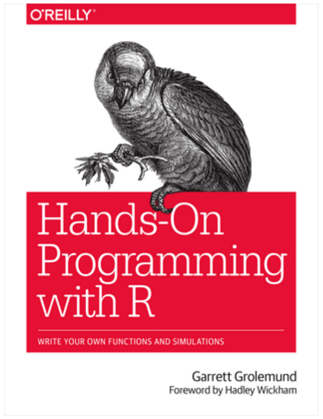
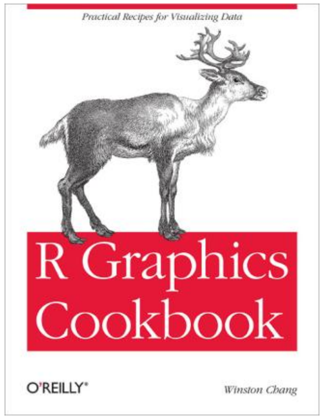
```
[1] 3
5
3 , 10
[1] 0
[1] 0
[1] 30
```

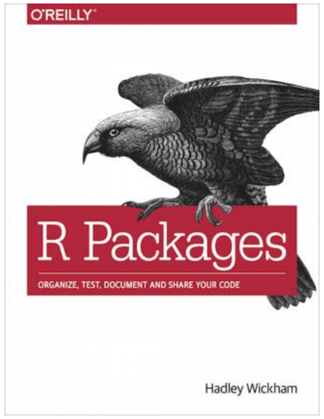
Practicing

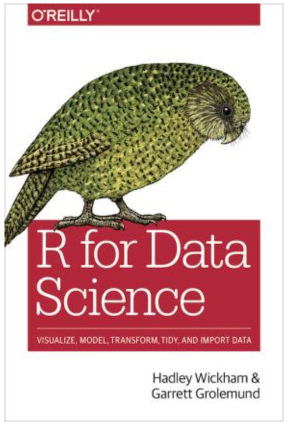
- Take some time to practice the example
  - https://github.com/eogasawara/R/tree/main/14-S3-Classes
- Exercise
  - Implement the classes square and hexagon compute their areas

Referências

Material: https://eic.cefet-rj.br/~eogasawara/tutorial-r


O'REILLY
Hands-On Programming with R
WRITE YOUR OWN FUNCTIONS AND SIMULATIONS
Garrett Grolemund
Foreword by Hadley Wickham






Hands-on Programming with R: https://rstudio-education.github.io/hopr/index.html
R Graphics Cookbook: https://r-graphics.org
R Packages: https://r-pkgs.org/index.html
R for Data Science: https://r4ds.had.co.nz

https://rstudio-education.github.io/hopr/basics.html

[1] G. Grolemund, 2014, Hands-On Programming with R: Write Your Own Functions and Simulations. O'Reilly Media, Inc.