

# Advanced R programming: solutions 2

Dr Colin Gillespie

## 1 S3 objects

1. Following the cohort example in the notes, suppose we want to create a mean method.

- List all S3 methods associated with the mean function.

```
methods("mean")

## [1] mean,ANY-method      mean,Cohort-method
## [3] mean.cohort           mean.Date
## [5] mean.default          mean.difftime
## [7] mean.POSIXct          mean.POSIXlt
## see '?methods' for accessing help and source code
```

- Examine the source code of mean.

```
body("mean")
```

- What are the arguments of mean?

```
args("mean")

## function (x, ...)
## NULL
```

- Create a function called mean.cohort that returns a vector containing the mean weight and mean height.<sup>1</sup>

```
mean.cohort = function(x, ...) {
  m1 = mean(x$details[,1], ...)
  m2 = mean(x$details[,2], ...)
  return(c(m1, m2))
}
```

<sup>1</sup> Ensure that you can pass in the standard mean arguments, i.e. na.rm.

2. Let's now make a similar function for the standard deviation

- Look at the arguments of the sd function.
- Create an function call sd.cohort that returns a vector containing the weight and height standard deviation.<sup>2</sup>
- Create a default sd function. Look at cor.default in the notes for a hint.

```
sd = function(x, ...) UseMethod("sd")
sd.default = function(x, ...) stats::sd(x, ...)
sd.cohort = function(x, ...) {
  s1 = sd(x$details[,1], ...)
  s2 = sd(x$details[,2], ...)
  return(c(s1, s2))
}
```

<sup>2</sup> Ensure that you can pass in the standard sd arguments, i.e. na.rm.

3. Create a summary method for the cohort class. When the summary function is called on a cohort object it should call the base summary on the details element.
  - Use the body function to check if the function is already a generic function.
  - Use the args function to determine the arguments.
  - Create a summary.cohort function

```
## summary is already a generic
body(summary)

## standardGeneric("summary")

## Match the args
args(summary)

## function (object, ...)
## NULL

## Function
summary.cohort = function(object, ...) summary(object$details, ...)
```

4. Create a hist method for the cohort class. When the hist function is called on a cohort object, it should produce a single plot showing two histograms - one for height and another for weight.

```
## hist is already a generic
body(hist)

## standardGeneric("hist")

## Match the args
args(hist)

## function (x, ...)
## NULL

## Function
hist.cohort = function(x, ...) {
  op = par(mfrow=c(1, 2))
  hist(x$details[,1], main="Weight")
  hist(x$details[,2], main="Height")
  par(op)
}
```

5. Create a [ method for the cohort class. This method should return a cohort object, but with the relevant rows sub setted. For example, if cc was a cohort object, then

```
cc[1:3,]
```

would return the first three rows of the data frame.

```
## Lots of methods available.
methods('[')

## [1] [,Cohort-method      [,nonStructure-method
## [3] [,.acf*                [,.AsIs
## [5] [,.bibentry*          [,.cohort
## [7] [,.data.frame         [,.Date
## [9] [,.difftime           [,.Dlist
## [11] [,.factor              [,.formula*
## [13] [,.fseq*               [,.getAnywhere*
## [15] [,.hexmode             [,.listof
## [17] [,.noquote             [,.numeric_version
## [19] [,.octmode             [,.pdf_doc*
## [21] [,.person*            [,.POSIXct
## [23] [,.POSIXlt            [,.raster*
## [25] [,.roman*             [,.simple.list
## [27] [,.table               [,.terms*
## [29] [,.ts*                [,.tskernel*
## [31] [,.warnings
## see '?methods' for accessing help and source code

## Examine [,.data.frame
args('[')

## function (x, i, j, drop = if (missing(i)) TRUE else length(cols) ==
##      1)
## NULL

'[,.cohort' = function(x, ...){
  x$details = x$details[...]
  x
}
```

6. Create a [`<-`] method for the cohort class. This method should allow us to replace values in the details data frame, i.e.

```
cc[1,1] = 10
```

```
## Lots of methods available.
methods('[')

## [1] [,<-,Cohort-method      [,<-,data.frame-method
## [3] [,<-.cohort             [,<-.data.frame
## [5] [,<-.Date               [,<-.factor
```

```
## [7] [<-.numeric_version    [<-.POSIXct
## [9] [<-.POSIXlt              [<-.raster*
## [11] [<-.ts*
## see '?methods' for accessing help and source code

## Examine [.data.frame
args(' [<-.data.frame')

## function (x, i, j, value)
## NULL

' [<-.cohort' = function(x, i, j, value){
  x$details[i, j] = value
  x
}
cc[1:3, ] = 55
```

## Solutions

Solutions are contained within the course package

```
library("nclRadvanced")
vignette("solutions2", package="nclRadvanced")
```