

Advanced R programming: solutions 3

Dr Colin Gillespie

S4 objects

1. Following the Cohort example in the notes, suppose we want to make a generic for the mean function.

- Using the `isGeneric` function, determine if the mean function is an S4 generic. If not, use `setGeneric` to create an S4 generic.

```
isGeneric("mean")  
  
## [1] TRUE  
  
setGeneric("mean")  
  
## [1] "mean"
```

I've intentionally mirrored the functions from previous practical to highlight the differences.

- Using `setMethod`, create a mean method for the Cohort class.¹

```
setMethod("mean", signature=c("Cohort"),  
          definition=function(x, ...) {  
            m1 = mean(x@details[,1], ...)  
            m2 = mean(x@details[,2], ...)  
            return(c(m1, m2))  
          })  
  
## [1] "mean"
```

¹ Be careful to match the arguments.

2. Repeat the above steps for the sd function.

```
isGeneric("sd")  
  
## [1] TRUE  
  
setGeneric("sd")  
  
## [1] "sd"  
  
setMethod("sd", signature=c("Cohort"),  
          definition=function(x, na.rm=FALSE) {  
            m1 = sd(x@details[,1], na.rm=na.rm)  
            m2 = sd(x@details[,2], na.rm=na.rm)  
            return(c(m1, m2))  
          })  
  
## [1] "sd"
```

3. Create a summary method for the cohort class

- Use `isGeneric` to determine if an S4 generic exists.
- Use `setGeneric` to set the generic method (if necessary).
- Create an S4 summary method.

```
isGeneric("summary")

## [1] TRUE

setGeneric("summary")

## [1] "summary"

setMethod("summary", signature=c("Cohort"),
          definition=function(object, ...) {
            summary(object@details)
          })

## [1] "summary"
```

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

```
isGeneric("hist")

## [1] TRUE

setGeneric("hist")

## [1] "hist"

setMethod("hist", signature=c("Cohort"),
          definition=function(x, ...) {
            op = par(mfrow=c(1, 2))
            hist(x@details[,1], main="Weight", ...)
            hist(x@details[,2], main="Height", ...)
            par(op)
          })

## [1] "hist"
```

5. Create a `[` method for the cohort class. This method should return a cohort object, but with the relevant rows sub setted.

```

isGeneric("[")

## [1] TRUE

getGeneric('[')

## standardGeneric for "[" defined from package "base"
##
## function (x, i, j, ..., drop = TRUE)
## standardGeneric("[" , .Primitive("["))
## <bytecode: 0x338edb8>
## <environment: 0x3397c98>
## Methods may be defined for arguments: x, i, j, drop
## Use  showMethods("[")  for currently available ones.

## Can you determine what drop does?
setMethod("[" , signature=c("Cohort"),
          definition=function(x, i, j, ..., drop = TRUE) {
            x@details = x@details[i, j, ..., drop=drop]
            x
          }
)

## [1] "["

```

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

```

isGeneric("[<-")

## [1] TRUE

setGeneric('[<-')

## [1] "[<-"

setMethod("[<-", signature=c("Cohort"),
          definition=function(x, i, j, value) {
            x@details[i, j] = value
            x
          }
)

## [1] "[<-"

coh_s4[1,]= 5

```

Solutions

Solutions are contained within the course package

```
library("nclRadvanced")  
vignette("solutions3", package="nclRadvanced")
```