Practical 2

Jumping Rivers

S3 objects

1. Following the cohort example in the notes, suppose we want to create a `mean` method.
2. List all S3 methods associated with the `mean` function.

```
methods("mean")
```

```
##  [1] mean,ANY-method
##  [2] mean,Matrix-method
##  [3] mean,sparseMatrix-method
##  [4] mean,sparseVector-method
##  [5] mean.Date
##  [6] mean.POSIXct
##  [7] mean.POSIXlt
##  [8] mean.default
##  [9] mean.difftime
## [10] mean.quosure*
## see '?methods' for accessing help and source code
```

3. Examine the source code of `mean`.

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

4. What are the arguments of `mean`?

```
args("mean")
```

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

5. Create a function called `mean.cohort` that returns a vector containing the mean weight and mean height. Ensure that you can pass in the standard `mean` arguments, i.e.`na.rm`.‘

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

6. Let’s now make a similar function for the standard deviation
7. Look at the arguments of the `sd` function.
8. Create an function call `sd.cohort` that returns a vector containing the weight and height standard deviation. Ensure that you can pass in the standard `sd` arguments, i.e.`na.rm`.‘

9. 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))
}
```

10. 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.
11. Use the `body` function to check if the function is already a generic function.
12. Use the `args` function to determine the arguments.
13. 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,
    ...)
```

14. 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. You can either use base or **ggplot2**, again we'll be using **ggplot2**. To get both plots onto one plotting window, try using the `grid.arrange()` function from **gridExtra** package.

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

## UseMethod("hist")

## Match the args
args(hist)
```

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

```
## Function
hist.cohort = function(x, ...) {
    dd = x$details
    Weight = ggplot(dd, aes(x = weight)) + geom_histogram() +
        labs(title = "Weight")
    Height = ggplot(dd, aes(x = height)) + geom_histogram() +
        labs(title = "Height")
    gridExtra::grid.arrange(Weight, Height)
}
```

15. 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] [,CsparseMatrix,index,index,logical-method
##  [2] [,CsparseMatrix,index,missing,logical-method
##  [3] [,CsparseMatrix,missing,index,logical-method
##  [4] [,Matrix,ANY,ANY,ANY-method
##  [5] [,Matrix,index,index,missing-method
##  [6] [,Matrix,index,missing,missing-method
##  [7] [,Matrix,lMatrix,missing,missing-method
##  [8] [,Matrix,logical,missing,missing-method
##  [9] [,Matrix,matrix,missing,ANY-method
## [10] [,Matrix,matrix,missing,missing-method
## [11] [,Matrix,missing,index,missing-method
## [12] [,Matrix,missing,missing,ANY-method
## [13] [,TsparseMatrix,index,index,logical-method
## [14] [,TsparseMatrix,index,missing,logical-method
## [15] [,TsparseMatrix,missing,index,logical-method
## [16] [,abIndex,index,ANY,ANY-method
## [17] [,color,ANY,ANY,ANY-method
## [18] [,denseMatrix,index,index,logical-method
## [19] [,denseMatrix,index,missing,logical-method
## [20] [,denseMatrix,matrix,missing,ANY-method
## [21] [,denseMatrix,matrix,missing,missing-method
## [22] [,denseMatrix,missing,index,logical-method
## [23] [,diagonalMatrix,index,index,logical-method
## [24] [,diagonalMatrix,index,missing,logical-method
## [25] [,diagonalMatrix,missing,index,logical-method
## [26] [,indMatrix,index,missing,logical-method
## [27] [,nonStructure,ANY,ANY,ANY-method
## [28] [,sparseMatrix,index,index,logical-method
## [29] [,sparseMatrix,index,missing,logical-method
## [30] [,sparseMatrix,missing,index,logical-method
## [31] [,sparseVector,index,ANY,ANY-method
## [32] [,sparseVector,lsparseVector,ANY,ANY-method
## [33] [,sparseVector,nsparseVector,ANY,ANY-method
## [34] [.AsIs
## [35] [.DLLInfoList
## [36] [.Date
## [37] [.Dlist
## [38] [.POSIXct
## [39] [.POSIXlt
## [40] [.acf*
## [41] [.arrow*
## [42] [.bibentry*
## [43] [.check_details_changes*
## [44] [.data.frame
## [45] [.difftime
## [46] [.envlist*
## [47] [.factor
## [48] [.formula*
## [49] [.fs_bytes*
## [50] [.fs_path*
## [51] [.fs_perms*
## [52] [.fseq*
## [53] [.fun_list*
## [54] [.gList*
## [55] [.getAnywhere*
## [56] [.glue*
## [57] [.gpar*
## [58] [.grouped_df*
## [59] [.gtable*
## [60] [.hexmode
## [61] [.lazy_dots*
## [62] [.listof
## [63] [.matrix.coo*
## [64] [.matrix.csr*
## [65] [.matrix.diag.csr*
## [66] [.news_db*
## [67] [.noquote
## [68] [.numeric_version
## [69] [.octmode
## [70] [.person*
## [71] [.quosure*
## [72] [.quosures*
## [73] [.raster*
## [74] [.rlang_data_pronoun*
## [75] [.rlang_envs*
## [76] [.roman*
## [77] [.shingle*
## [78] [.simple.list
## [79] [.stack*
## [80] [.table
## [81] [.tbl_df*
## [82] [.terms*
## [83] [.trellis*
## [84] [.ts*
## [85] [.tskernel*
## [86] [.uneval*
## [87] [.unit*
## [88] [.unit.arithmetic*
## [89] [.unit.list*
## [90] [.vpPath*
## [91] [.warnings
## [92] [.xml_missing*
## [93] [.xml_nodeset*
## see '?methods' for accessing help and source code
```

```
## Examine [.data.frame
args("[.data.frame")
```

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

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

16. 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("[<-")

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

"[<-.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("jrOOP")
vignette("solutions2", package = "jrOOP")
```