Assignment 8 - CT5102

*Programming with S3 Classes - Extending the **lm** class*

The aim of this assignment is to explore how S3 can be used to extend existing class functionality. The goal is to create a new class **rep_lm** which stores additonal information relating to a linear regression task. The **rep_lm** function takes a number of parameters:

- The name of the model
- The linear regression equation (in string form)
- The data frame containing the data (in string form)

Here is an example of how the function is called.

```
m <- rep_lm("My Model", "eruptions~waiting","faithful")
```

The function returns a **repl_lm** object that has the following structure (one new list element has been added):

```
str(m)
```

```
## List of 13
##  $ coefficients : Named num [1:2] -1.874 0.0756
##   ..- attr(*, "names")= chr [1:2] "(Intercept)" "waiting"
##  $ residuals    : Named num [1:272] -0.5006 -0.4099 -0.3895 -0.5319 -0.0214 ...
##   ..- attr(*, "names")= chr [1:272] "1" "2" "3" "4" ...
##  $ effects      : Named num [1:272] -57.522 16.9256 -0.365 -0.4885 -0.0143 ...
##   ..- attr(*, "names")= chr [1:272] "(Intercept)" "waiting" "" "" ...
##  $ rank         : int 2
##  $ fitted.values: Named num [1:272] 4.1 2.21 3.72 2.81 4.55 ...
##   ..- attr(*, "names")= chr [1:272] "1" "2" "3" "4" ...
##  $ assign       : int [1:2] 0 1
##  $ qr           :List of 5
##   ..$ qr   : num [1:272, 1:2] -16.4924 0.0606 0.0606 0.0606 0.0606 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:272] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:2] "(Intercept)" "waiting"
##   .. ..- attr(*, "assign")= int [1:2] 0 1
##   ..$ qraux: num [1:2] 1.06 1.08
##   ..$ pivot: int [1:2] 1 2
##   ..$ tol  : num 1e-07
##   ..$ rank : int 2
##   ..- attr(*, "class")= chr "qr"
##  $ df.residual  : int 270
##  $ xlevels      : Named list()
##  $ call         : language lm(formula = as.formula(model), data = get(data_f_name))
##  $ terms        :Classes 'terms', 'formula'  language eruptions ~ waiting
##   .. ..- attr(*, "variables")= language list(eruptions, waiting)
##   .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:2] "eruptions" "waiting"
##   .. .. .. ..$ : chr "waiting"
##   .. ..- attr(*, "term.labels")= chr "waiting"
##   .. ..- attr(*, "order")= int 1
##   .. ..- attr(*, "intercept")= int 1
```

```
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: 0x7ffe1ed7f528>
##   .. ..- attr(*, "predvars")= language list(eruptions, waiting)
##   .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. ..- attr(*, "names")= chr [1:2] "eruptions" "waiting"
##  $ model        :'data.frame':   272 obs. of  2 variables:
##   ..$ eruptions: num [1:272] 3.6 1.8 3.33 2.28 4.53 ...
##   ..$ waiting  : num [1:272] 79 54 74 62 85 55 88 85 51 85 ...
##   ..- attr(*, "terms")=Classes 'terms', 'formula'  language eruptions ~ waiting
##   .. .. ..- attr(*, "variables")= language list(eruptions, waiting)
##   .. .. ..- attr(*, "factors")= int [1:2, 1] 0 1
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr [1:2] "eruptions" "waiting"
##   .. .. .. .. ..$ : chr "waiting"
##   .. .. ..- attr(*, "term.labels")= chr "waiting"
##   .. .. ..- attr(*, "order")= int 1
##   .. .. ..- attr(*, "intercept")= int 1
##   .. .. ..- attr(*, "response")= int 1
##   .. .. ..- attr(*, ".Environment")=<environment: 0x7ffe1ed7f528>
##   .. .. ..- attr(*, "predvars")= language list(eruptions, waiting)
##   .. .. ..- attr(*, "dataClasses")= Named chr [1:2] "numeric" "numeric"
##   .. .. .. ..- attr(*, "names")= chr [1:2] "eruptions" "waiting"
##  $ Information  :List of 6
##   ..$ Name        : chr "My Model"
##   ..$ DateRun     : chr "Wed Oct 30 11:47:13 2019"
##   ..$ LinearModel : chr "eruptions~waiting"
##   ..$ DataSource  : chr "faithful"
##   ..$ Columns     : chr [1:2] "eruptions" "waiting"
##   ..$ Observations: int 272
##  - attr(*, "class")= chr [1:2] "rep_lm" "lm"
```

In addition to writing the constructor function, a new **summary** function should be written for the new class, which provides the following output:

```
summary(m)
```

```
## (1) rep_lm class summary

## Model Name: My Model     Date of Run: Wed Oct 30 11:47:13 2019

## Linear Model: eruptions~waiting  Data Source: faithful

## Columns: eruptions
## Columns: waiting

## Observations: 272

## (2) lm class summary

##
## Call:
## lm(formula = as.formula(model), data = get(data_f_name))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29917 -0.37689  0.03508  0.34909  1.19329
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.874016   0.160143  -11.70   <2e-16 ***
## waiting      0.075628   0.002219   34.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4965 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
```

In order to implement the function, the following additional functions are needed:

- **as.formula()**
- **get()**