R Objects, Workflow, and Functions

Lanette Tyler

Vectors

Create a Vector

```
set.seed(42)
my_runif <- runif(30)
is.vector(my_runif)
```

```
[1] TRUE
```

Subset object my_unif

```
my_runif[1:10]
```

```
 [1] 0.9148060 0.9370754 0.2861395 0.8304476 0.6417455 0.5190959 0.7365883
 [8] 0.1346666 0.6569923 0.7050648
```

```
my_runif[c(1:3,15:17)]
```

```
[1] 0.9148060 0.9370754 0.2861395 0.4622928 0.9400145 0.9782264
```

Sort the vector (my_unif)

```
sort(my_runif)
```

```
 [1] 0.08243756 0.11748736 0.13466660 0.13871017 0.25542882 0.28613953
 [7] 0.39020347 0.44696963 0.45774178 0.46229282 0.47499708 0.51421178
[13] 0.51909595 0.56033275 0.64174552 0.65699229 0.70506478 0.71911225
[19] 0.73658831 0.83044763 0.83600426 0.90403139 0.90573813 0.91480604
[25] 0.93467225 0.93707541 0.94001452 0.94666823 0.97822643 0.98889173
```

Create a vector with character strings in it and explore sorting conventions

```
char_vector <- c("hix","abd","z4","xyz","42t","Xyz")
sort(char_vector)
```

```
[1] "42t" "abd" "hix" "xyz" "Xyz" "z4"
```

Data Frames

List data sets currently loaded into environment:

```
data()
```

List of data sets in environment opens in a file (R data sets)

Load trees data set into environment:

```
data(trees)
```

Loads into environment as "promise"

Output data set:

```
trees
```

```
   Girth Height Volume
1    8.3     70   10.3
2    8.6     65   10.3
3    8.8     63   10.2
4   10.5     72   16.4
5   10.7     81   18.8
6   10.8     83   19.7
7   11.0     66   15.6
8   11.0     75   18.2
9   11.1     80   22.6
10  11.2     75   19.9
11  11.3     79   24.2
12  11.4     76   21.0
13  11.4     76   21.4
14  11.7     69   21.3
15  12.0     75   19.1
```

```
16  12.9     74   22.2
17  12.9     85   33.8
18  13.3     86   27.4
19  13.7     71   25.7
20  13.8     64   24.9
21  14.0     78   34.5
22  14.2     80   31.7
23  14.5     74   36.3
24  16.0     72   38.3
25  16.3     77   42.6
26  17.3     81   55.4
27  17.5     82   55.7
28  17.9     80   58.3
29  18.0     80   51.5
30  18.0     80   51.0
31  20.6     87   77.0
```

Outputs to console or qmd file, depending on settings; changes from "promise" to listed in env

Figure out the type of R object:

```
str(trees)
```

```
'data.frame':   31 obs. of  3 variables:
 $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
 $ Height: num  70 65 63 72 81 83 66 75 80 75 ...
 $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
```

It's a data frame!

Subset a column (two ways):

```
trees$Height
```

```
 [1] 70 65 63 72 81 83 66 75 80 75 79 76 76 69 75 74 85 86 71 64 78 80 74 72 77
[26] 81 82 80 80 80 87
```

```
trees[,2]
```

```
 [1] 70 65 63 72 81 83 66 75 80 75 79 76 76 69 75 74 85 86 71 64 78 80 74 72 77
[26] 81 82 80 80 80 87
```

ouput columns (both are the same)

Get attributes from the data frame, especially names, names lists are output as vectors, so we can subset those vectors:

```
attributes(trees)
```

```
$names
[1] "Girth"  "Height" "Volume"

$class
[1] "data.frame"

$row.names
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30 31
```

```
names(trees)
```

```
[1] "Girth"  "Height" "Volume"
```

```
colnames(trees)
```

```
[1] "Girth"  "Height" "Volume"
```

```
names(trees)[2:3]
```

```
[1] "Height" "Volume"
```

Lists

Investigate date frame trees:

```
is.list(trees)
```

```
[1] TRUE
```

```r
is.data.frame(trees)
```

```
[1] TRUE
```

Because trees is a list as well as a data frame, we can subset like a list:

```r
trees[1] #output is a data frame since single brackets preserve
```

```
   Girth
1    8.3
2    8.6
3    8.8
4   10.5
5   10.7
6   10.8
7   11.0
8   11.0
9   11.1
10  11.2
11  11.3
12  11.4
13  11.4
14  11.7
15  12.0
16  12.9
17  12.9
18  13.3
19  13.7
20  13.8
21  14.0
22  14.2
23  14.5
24  16.0
25  16.3
26  17.3
27  17.5
28  17.9
29  18.0
30  18.0
31  20.6
```

```r
trees[[1]] #output is a vector, since double brackets simplify
```

```
 [1]  8.3  8.6  8.8 10.5 10.7 10.8 11.0 11.0 11.1 11.2 11.3 11.4 11.4 11.7 12.0
[16] 12.9 12.9 13.3 13.7 13.8 14.0 14.2 14.5 16.0 16.3 17.3 17.5 17.9 18.0 18.0
[31] 20.6
```

```r
#double brackets are for sinlge elememt/one column
trees[1:2] #first two columns
```

```
   Girth Height
1    8.3     70
2    8.6     65
3    8.8     63
4   10.5     72
5   10.7     81
6   10.8     83
7   11.0     66
8   11.0     75
9   11.1     80
10  11.2     75
11  11.3     79
12  11.4     76
13  11.4     76
14  11.7     69
15  12.0     75
16  12.9     74
17  12.9     85
18  13.3     86
19  13.7     71
20  13.8     64
21  14.0     78
22  14.2     80
23  14.5     74
24  16.0     72
25  16.3     77
26  17.3     81
27  17.5     82
28  17.9     80
29  18.0     80
30  18.0     80
31  20.6     87
```

```r
trees[1,3] #single element
```

```
[1] 10.3
```

Lists are very convenient for storage. Moedling function outputs are often store as lists. Let's look at a linear model's output:

```r
fit <- lm(Volume~Girth+Height,data =trees)
str(fit) #overwhelming amount of info
```

```
List of 12
 $ coefficients : Named num [1:3] -57.988 4.708 0.339
  ..- attr(*, "names")= chr [1:3] "(Intercept)" "Girth" "Height"
 $ residuals    : Named num [1:31] 5.462 5.746 5.383 0.526 -1.069 ...
  ..- attr(*, "names")= chr [1:31] "1" "2" "3" "4" ...
 $ effects      : Named num [1:31] -167.985 87.073 10.118 -0.812 -1.489 ...
  ..- attr(*, "names")= chr [1:31] "(Intercept)" "Girth" "Height" "" ...
 $ rank         : int 3
 $ fitted.values: Named num [1:31] 4.84 4.55 4.82 15.87 19.87 ...
  ..- attr(*, "names")= chr [1:31] "1" "2" "3" "4" ...
 $ assign       : int [1:3] 0 1 2
 $ qr           :List of 5
  ..$ qr   : num [1:31, 1:3] -5.57 0.18 0.18 0.18 0.18 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:31] "1" "2" "3" "4" ...
  .. .. ..$ : chr [1:3] "(Intercept)" "Girth" "Height"
  .. ..- attr(*, "assign")= int [1:3] 0 1 2
  ..$ qraux: num [1:3] 1.18 1.23 1.24
  ..$ pivot: int [1:3] 1 2 3
  ..$ tol  : num 1e-07
  ..$ rank : int 3
  ..- attr(*, "class")= chr "qr"
 $ df.residual  : int 28
 $ xlevels      : Named list()
 $ call         : language lm(formula = Volume ~ Girth + Height, data = trees)
 $ terms        :Classes 'terms', 'formula'  language Volume ~ Girth + Height
  .. ..- attr(*, "variables")= language list(Volume, Girth, Height)
  .. ..- attr(*, "factors")= int [1:3, 1:2] 0 1 0 0 0 1
  .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. ..$ : chr [1:3] "Volume" "Girth" "Height"
  .. .. .. ..$ : chr [1:2] "Girth" "Height"
```

```
  .. ..- attr(*, "term.labels")= chr [1:2] "Girth" "Height"
  .. ..- attr(*, "order")= int [1:2] 1 1
  .. ..- attr(*, "intercept")= int 1
  .. ..- attr(*, "response")= int 1
  .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
  .. ..- attr(*, "predvars")= language list(Volume, Girth, Height)
  .. ..- attr(*, "dataClasses")= Named chr [1:3] "numeric" "numeric" "numeric"
  .. .. ..- attr(*, "names")= chr [1:3] "Volume" "Girth" "Height"
 $ model        :'data.frame':  31 obs. of  3 variables:
  ..$ Volume: num [1:31] 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
  ..$ Girth : num [1:31] 8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
  ..$ Height: num [1:31] 70 65 63 72 81 83 66 75 80 75 ...
  ..- attr(*, "terms")=Classes 'terms', 'formula'  language Volume ~ Girth + Height
  .. .. ..- attr(*, "variables")= language list(Volume, Girth, Height)
  .. .. ..- attr(*, "factors")= int [1:3, 1:2] 0 1 0 0 0 1
  .. .. .. ..- attr(*, "dimnames")=List of 2
  .. .. .. .. ..$ : chr [1:3] "Volume" "Girth" "Height"
  .. .. .. .. ..$ : chr [1:2] "Girth" "Height"
  .. .. ..- attr(*, "term.labels")= chr [1:2] "Girth" "Height"
  .. .. ..- attr(*, "order")= int [1:2] 1 1
  .. .. ..- attr(*, "intercept")= int 1
  .. .. ..- attr(*, "response")= int 1
  .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
  .. .. ..- attr(*, "predvars")= language list(Volume, Girth, Height)
  .. .. ..- attr(*, "dataClasses")= Named chr [1:3] "numeric" "numeric" "numeric"
  .. .. .. ..- attr(*, "names")= chr [1:3] "Volume" "Girth" "Height"
 - attr(*, "class")= chr "lm"
```

```
str(fit$coefficients) #one way to limit info, one element at a time
```

```
 Named num [1:3] -57.988 4.708 0.339
 - attr(*, "names")= chr [1:3] "(Intercept)" "Girth" "Height"
```

```
str(fit,max.level=1)
```

```
List of 12
 $ coefficients : Named num [1:3] -57.988 4.708 0.339
  ..- attr(*, "names")= chr [1:3] "(Intercept)" "Girth" "Height"
 $ residuals    : Named num [1:31] 5.462 5.746 5.383 0.526 -1.069 ...
  ..- attr(*, "names")= chr [1:31] "1" "2" "3" "4" ...
 $ effects      : Named num [1:31] -167.985 87.073 10.118 -0.812 -1.489 ...
```

```
 ..- attr(*, "names")= chr [1:31] "(Intercept)" "Girth" "Height" "" ...
$ rank         : int 3
$ fitted.values: Named num [1:31] 4.84 4.55 4.82 15.87 19.87 ...
 ..- attr(*, "names")= chr [1:31] "1" "2" "3" "4" ...
$ assign       : int [1:3] 0 1 2
$ qr           :List of 5
 ..- attr(*, "class")= chr "qr"
$ df.residual  : int 28
$ xlevels      : Named list()
$ call         : language lm(formula = Volume ~ Girth + Height, data = trees)
$ terms        :Classes 'terms', 'formula'  language Volume ~ Girth + Height
 .. ..- attr(*, "variables")= language list(Volume, Girth, Height)
 .. ..- attr(*, "factors")= int [1:3, 1:2] 0 1 0 0 0 1
 .. .. ..- attr(*, "dimnames")=List of 2
 .. ..- attr(*, "term.labels")= chr [1:2] "Girth" "Height"
 .. ..- attr(*, "order")= int [1:2] 1 1
 .. ..- attr(*, "intercept")= int 1
 .. ..- attr(*, "response")= int 1
 .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
 .. ..- attr(*, "predvars")= language list(Volume, Girth, Height)
 .. ..- attr(*, "dataClasses")= Named chr [1:3] "numeric" "numeric" "numeric"
 .. .. ..- attr(*, "names")= chr [1:3] "Volume" "Girth" "Height"
$ model        :'data.frame':  31 obs. of  3 variables:
 ..- attr(*, "terms")=Classes 'terms', 'formula'  language Volume ~ Girth + Height
 .. .. ..- attr(*, "variables")= language list(Volume, Girth, Height)
 .. .. ..- attr(*, "factors")= int [1:3, 1:2] 0 1 0 0 0 1
 .. .. .. ..- attr(*, "dimnames")=List of 2
 .. .. ..- attr(*, "term.labels")= chr [1:2] "Girth" "Height"
 .. .. ..- attr(*, "order")= int [1:2] 1 1
 .. .. ..- attr(*, "intercept")= int 1
 .. .. ..- attr(*, "response")= int 1
 .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
 .. .. ..- attr(*, "predvars")= language list(Volume, Girth, Height)
 .. .. ..- attr(*, "dataClasses")= Named chr [1:3] "numeric" "numeric" "numeric"
 .. .. .. ..- attr(*, "names")= chr [1:3] "Volume" "Girth" "Height"
- attr(*, "class")= chr "lm"
```

Some of these fit objects have helper functions:

```
coef(fit)
```

```
(Intercept)       Girth      Height
```

```
-57.9876589    4.7081605    0.3392512
```

```
residuals(fit)
```

```
          1           2           3           4           5           6
 5.46234035  5.74614837  5.38301873  0.52588477 -1.06900844 -1.31832696
          7           8           9          10          11          12
-0.59268807 -1.04594918  1.18697860 -0.28758128  2.18459773 -0.46846462
         13          14          15          16          17          18
-0.06846462  0.79384587 -4.85410969 -5.65220290  2.21603352 -6.40648192
         19          20          21          22          23          24
-4.90097760 -3.79703501  0.11181561 -4.30831896  0.91474029 -3.46899800
         25          26          27          28          29          30
-2.27770232  4.45713224  3.47624891  4.87148717 -2.39932888 -2.89932888
         31
 8.48469518
```

```
effects(fit)
```

```
 (Intercept)        Girth       Height
-167.9848390   87.0734249   10.1183585   -0.8124686   -1.4891024   -1.5288330

  -2.4801701   -2.0350805    0.7068854   -1.2568038    1.6245965   -1.3179617

  -0.9179617   -0.7245053   -5.7436973   -6.5520180    2.4142026   -6.0686786

  -6.0206079   -5.6054282   -0.2792344   -4.4798268    0.1741951   -4.2598611

  -2.5396186    4.6940266    3.8328687    5.0682910   -2.1925707   -2.6925707

   9.6489841
attr(,"assign")
[1] 0 1 2
attr(,"class")
[1] "coef"
```