object m61r

pv71u98h1

April 13, 2021

1 Introduction

Object m61r is an object that enables all the function present in m61r, and in addition, allows a sort of pipe.
The purpose of this package is informative.

2 Example 1: pipeline with 1 step cache

```
> co2 <- m61r(CO2)
> co2$filter(~Plant %in% c("Qn1","Qc3"))
> co2$mutate(z1=~uptake/conc,y=~conc/100)
> co2$group_by(~c(Type,Treatment))
> co2$summarise(foo=~mean(z1),bar=~sd(y))
> co2 # print results
```

```
    Type  Treatment       foo      bar
1 Quebec nonchilled 0.1079993 3.177263
2 Quebec    chilled 0.1009081 3.177263
```

```
> head(co2) # back to normal
```

```
  Plant   Type  Treatment conc uptake
1   Qn1 Quebec nonchilled   95   16.0
2   Qn1 Quebec nonchilled  175   30.4
3   Qn1 Quebec nonchilled  250   34.8
4   Qn1 Quebec nonchilled  350   37.2
5   Qn1 Quebec nonchilled  500   35.3
6   Qn1 Quebec nonchilled  675   39.2
```

```
>
>
```

3 Example 2: get only a data.frame as result

```
> co2 <- m61r(CO2)
> co2$filter(~Plant %in% c("Qn1","Qc3"))
> co2$transmutate(z1=~uptake/conc,y=~conc/100)
> tmp <- co2[] # get only the data.frame and not the whole m61r object
> head(tmp)
```

```
          z1    y
1 0.16842105 0.95
2 0.17371429 1.75
3 0.13920000 2.50
4 0.10628571 3.50
5 0.07060000 5.00
6 0.05807407 6.75
```

```
> class(tmp)
```

```
[1] "data.frame"
```

```
>
```

4 Example 3: manipulation of a m61r object

```
> co2 <- m61r(CO2)
> head(co2)
```

```
  Plant   Type  Treatment conc uptake
1   Qn1 Quebec nonchilled   95   16.0
2   Qn1 Quebec nonchilled  175   30.4
3   Qn1 Quebec nonchilled  250   34.8
4   Qn1 Quebec nonchilled  350   37.2
5   Qn1 Quebec nonchilled  500   35.3
6   Qn1 Quebec nonchilled  675   39.2
```

```
> names(co2)
```

```
[1] "Plant"     "Type"      "Treatment" "conc"      "uptake"
```

```
> dim(co2)
```

```
[1] 84  5
```

```
> co2[1,]
```

```
  Plant   Type  Treatment conc uptake
1   Qn1 Quebec nonchilled   95     16
```

```
> head(co2[,2:3])
```

```
    Type  Treatment
1 Quebec nonchilled
2 Quebec nonchilled
3 Quebec nonchilled
4 Quebec nonchilled
5 Quebec nonchilled
6 Quebec nonchilled
```

```
> co2[1:10,1:3]
```

```
   Plant   Type  Treatment
1     Qn1 Quebec nonchilled
2     Qn1 Quebec nonchilled
3     Qn1 Quebec nonchilled
4     Qn1 Quebec nonchilled
5     Qn1 Quebec nonchilled
6     Qn1 Quebec nonchilled
7     Qn1 Quebec nonchilled
8     Qn2 Quebec nonchilled
9     Qn2 Quebec nonchilled
10    Qn2 Quebec nonchilled
```

```
> co2[1,"Plant"]
```

```
  Plant
1   Qn1
```

```
> str(co2)
```

```
Classes 'm61r', 'environment' <environment: 0x42f8150>
```

```
> co2[1,"conc"] <- 100
> co2[1,] # w/temporary change
```

```
  Plant   Type  Treatment conc uptake
1   Qn1 Quebec nonchilled  100     16
```

```
> co2[1,] # back to normal
```

```
  Plant   Type  Treatment conc uptake
1   Qn1 Quebec nonchilled   95     16
```

```
> # WARNING: Keep the brackets to manipulate the intern data.frame
> co2[] <- co2[-1,]
> co2[1:3,] # temporary result

  Plant   Type  Treatment conc uptake
1   Qn1 Quebec nonchilled  175   30.4
2   Qn1 Quebec nonchilled  250   34.8
3   Qn1 Quebec nonchilled  350   37.2

> co2[1:3,] # back to normal

  Plant   Type  Treatment conc uptake
1   Qn1 Quebec nonchilled   95   16.0
2   Qn1 Quebec nonchilled  175   30.4
3   Qn1 Quebec nonchilled  250   34.8

> # ... OR you will destroy co2, and only keep the data.frame
> # co2 <- co2[-1,]
> # class(co2) # data.frame
>
> # cloning
> foo <- co2 # This will only create
>            # a second variable that points
>            # on the same object (i.e not cloning)
> str(co2)

Classes 'm61r', 'environment' <environment: 0x42f8150>

> str(foo)

Classes 'm61r', 'environment' <environment: 0x42f8150>

> # Instead, cloning into a new environment
> foo <- co2$clone()
> str(co2)

Classes 'm61r', 'environment' <environment: 0x42f8150>

> str(foo)

Classes 'm61r', 'environment' <environment: 0x3cb95a8>

>
```