

Basics

1. Describe the difference between `t.test()` and `t.data.frame()`. When is each function called?
2. Make a list of commonly used base R functions that contain `.` in their name but are not S3 methods.
3. What does the `as.data.frame.data.frame()` method do? Why is it confusing? How could you avoid this confusion in your own code?
4. Describe the difference in behaviour in these two calls.

```
set.seed(1014)
some_days <- as.Date("2017-01-31") + sample(10, 5)

mean(some_days)
#> [1] "2017-02-06"
mean(unclass(some_days))
#> [1] 17203
```

5. What class of object does the following code return? What base type is it built on? What attributes does it use?

```
x <- ecdf(rpois(100, 10))
x
#> Empirical CDF
#> Call: ecdf(rpois(100, 10))
#> x[1:18] = 2, 3, 4, ..., 2e+01, 2e+01
```

6. What class of object does the following code return? What base type is it built on? What attributes does it use?

```
x <- table(rpois(100, 5))
x
#>
#> 1 2 3 4 5 6 7 8 9 10
#> 7 5 18 14 15 15 14 4 5 3
```

Classes

1. Write a constructor for `data.frame` objects. What base type is a data frame built on? What attributes does it use? What are the restrictions placed on the individual elements? What about the names?
2. Enhance my `factor()` helper to have better behaviour when one or more values is not found in levels. What does `base::factor()` do in this situation?
3. Carefully read the source code of `factor()`. What does it do that my constructor does not?
4. Factors have an optional “contrasts” attribute. Read the help for `C()`, and briefly describe the purpose of the attribute. What type should it have? Rewrite the `new_factor()` constructor to include this attribute.
5. Read the documentation for `utils::as.roman()`. How would you write a constructor for this class? Does it need a validator? What might a helper do?

Generics and methods

1. Read the source code for `t()` and `t.test()` and confirm that `t.test()` is an S3 generic and not an S3 method. What happens if you create an object with class `test` and call `t()` with it? Why?

```
x <- structure(1:10, class = "test")
t(x)
```

2. What generics does the table class have methods for?
3. What generics does the ecdf class have methods for?
4. Which base generic has the greatest number of defined methods?
5. Carefully read the documentation for UseMethod() and explain why the following code returns the results that it does. What two usual rules of function evaluation does UseMethod() violate?

```
g <- function(x) {
  x <- 10
  y <- 10
  UseMethod("g")
}
g.default <- function(x) c(x = x, y = y)

x <- 1
y <- 1
g(x)
#> x y
#> 1 10
```

6. What are the arguments to [? Why is this a hard question to answer?

Object styles

1. Categorise the objects returned by lm(), factor(), table(), as.Date(), as.POSIXct(), ecdf(), ordered(), I() into the styles described above.
2. What would a constructor function for lm objects, new_lm(), look like? Use ?lm and experimentation to figure out the required fields and their types.

Inheritance

1. How does [.Date support subclasses? How does it fail to support subclasses?
2. R has two classes for representing date time data, POSIXct and POSIXlt, which both inherit from POSIXt. Which generics have different behaviours for the two classes? Which generics share the same behaviour?
3. What do you expect this code to return? What does it actually return? Why?

```
generic2 <- function(x) UseMethod("generic2")
generic2.a1 <- function(x) "a1"
generic2.a2 <- function(x) "a2"
generic2.b <- function(x) {
  class(x) <- "a1"
  NextMethod()
}

generic2(structure(list(), class = c("b", "a2")))
```

Dispatch details

1. Explain the differences in dispatch below:

```
length.integer <- function(x) 10

x1 <- 1:5
class(x1)
#> [1] "integer"
```

```
s3_dispatch(length(x1))
#> * length.integer
#> length.numeric
#> length.default
#> => length (internal)

x2 <- structure(x1, class = "integer")
class(x2)
#> [1] "integer"
s3_dispatch(length(x2))
#> => length.integer
#> length.default
#> * length (internal)
```

2. What classes have a method for the Math group generic in base R? Read the source code. How do the methods work?
3. Math.difftime() is more complicated than I described. Why?