Stats 506 PS5

Alyssa Yang

Problem 1: OOP Programming

1a

```
# Constructor
rational <- setClass("rational",
              slots = c(a = "numeric",
                        b = "numeric"))
```

```
# Validator
setValidity("rational", function(object) {
  if (object@b == 0) {
    stop("Denominator cannot be 0")
  }
  return(TRUE)
})
```

```
Class "rational" [in ".GlobalEnv"]

Slots:

Name:        a       b
Class: numeric numeric
```

```
# Show method
setMethod("show", "rational",
  function(object) {
    cat(paste0(object@a, "/", object@b, "\n"))
    return(invisible(object))
```

```
  }
)
```

```
# GCD and LCM in RCpp
library(Rcpp)

cppFunction("
  #include <numeric>`
  int C_gcd(int a, int b) {
    return std::gcd(a, b);
  }")

cppFunction("
  #include <numeric>
  int C_lcm(int a, int b) {
    return std::lcm(a, b);
  }")
```

```
# Simplify method
setGeneric("simplify",
  function(object) {
    standardGeneric("simplify")
  })
```

```
[1] "simplify"
```

```
setMethod("simplify", "rational",
  function(object) {
    gcd <- C_gcd(object@a, object@b)
    object@a <- object@a / gcd
    object@b <- object@b / gcd
    show(object)
    return(invisible(object))
  })
```

```
# Quotient method
setGeneric("quotient",
  function(object, digits = 4) {
    standardGeneric("quotient")
  })
```

```
[1] "quotient"
```

```
setMethod("quotient", "rational",
  function(object, digits = 4) {
    digits <- tryCatch(
      { as.integer(digits) },
      error = function(e) {
        stop("Digits must be an integer.")
      })

    result <- object@a / object@b
    print(format(result, digits = digits))
    return(invisible(result))
  })
```

```
# +, -, *, /
setMethod("+", signature(e1 = "rational",
                         e2 = "rational"),
          function(e1, e2) {
            lcm <- C_lcm(e1@b, e2@b)
            num1 <- (lcm / e1@b) * e1@a
            num2 <- (lcm / e2@b) * e2@a
            return(rational(a = num1 + num2, b = lcm))
          })

setMethod("-", signature(e1 = "rational",
                         e2 = "rational"),
          function(e1, e2) {
            lcm <- C_lcm(e1@b, e2@b)
            num1 <- (lcm / e1@b) * e1@a
            num2 <- (lcm / e2@b) * e2@a
            return(rational(a = num1 - num2, b = lcm))
          })

setMethod("*", signature(e1 = "rational",
                         e2 = "rational"),
          function(e1, e2) {
            return(rational(a = e1@a * e2@a, b = e1@b * e2@b))
          })

setMethod("/", signature(e1 = "rational",
                         e2 = "rational"),
```

```
          function(e1, e2) {
            return(rational(a = e1@a * e2@b, b = e1@b * e2@a))
          })
```

1b

```
r1 <- rational(a = 24, b = 6)
r2 <- rational(a = 7, b = 230)
r3 <- rational(a = 0, b = 4)
```

```
r1
```

```
24/6
```

```
r3
```

```
0/4
```

```
r1 + r2
```

```
2781/690
```

```
r1 - r2
```

```
2739/690
```

```
r1 * r2
```

```
168/1380
```

```
r1 / r2
```

```
5520/42
```

```
r1 + r3
```

```
48/12
```

```
r1 * r3
```

```
0/24
```

```
r2 / r3
```

```
Error in validityMethod(object): Denominator cannot be 0
```

```
quotient(r1)
```

```
[1] "4"
```

```
quotient(r2)
```

```
[1] "0.03043"
```

```
quotient(r2, digits = 3)
```

```
[1] "0.0304"
```

```
quotient(r2, digits = 3.14)
```

```
[1] "0.0304"
```

```
quotient(r2, digits = "avocado")
```

```
Warning in doTryCatch(return(expr), name, parentenv, handler): NAs introduced
by coercion
```

```
Error in prettyNum(.Internal(format(x, trim, digits, nsmall, width, 3L, : invalid value -214
```

```r
q2 <- quotient(r2, digits = 3)
```

```
[1] "0.0304"
```

```r
q2
```

```
[1] 0.03043478
```

```r
quotient(r3)
```

```
[1] "0"
```

```r
simplify(r1)
```

```
4/1
```

```r
simplify(r2)
```

```
7/230
```

```r
simplify(r3)
```

```
0/1
```