In-class Assignment 2

Andrew Shao

2024-09-19

2.4.4 Q1

mix_1 will be "integer" because it is the sum of two "integer"s (int_1 and int_2 have "L" at the end) so using the coercion rule the storage type will be of the most complex type which is "integer".
mix_2 will be "double" because it is the sum of an "integer" and a "double" (num_1 doesn't have the "L"), so the most complex type is "double" and we invoke the coercion rule like before.
mix_3 will be "double" because it is the quotient of two "integer"s that isn't a whole number so it must be stored as a "double".
mix_4 will be "character" because it contains a "double" and a "character", and "character" is the more complex type and we can invoke the coercion rule.
mix_5 will be "double" because it contains a "double" and a "logical", and "double" is the more complex and we invoke the coercion rule.
mix_6 will be "character" because it contains a "double", "character", and "logical", and "character" is the most complex of the three and we invoke the coercion rule.

```r
int_1 <- 5L
int_2 <- 6L
num_1 <- 2
char_1 <- "pig"
logi_1 <- TRUE
mix_1 <- int_1 + int_2
mix_2 <- int_1 + num_1
mix_3 <- int_1/int_2
mix_4 <- c(num_1, char_1)
mix_5 <- c(num_1, logi_1)
mix_6 <- c(num_1, char_1, logi_1)
```

```r
typeof(mix_1)
```

```
## [1] "integer"
```

```r
typeof(mix_2)
```

```
## [1] "double"
```

```r
typeof(mix_3)
```

```
## [1] "double"
```

```r
typeof(mix_4)
```

```
## [1] "character"
```

```r
typeof(mix_5)
```

```
## [1] "double"
```

```r
typeof(mix_6)
```

```
## [1] "character"
```

2.4.4 Q2

`3 * logi_2 + logi_3` will result in a vector of storage type `"double"` with values 4, 1, and 4.
`logi_2 - logi_3` will result in a vector of storage type `"double"` with values 0, -1, and 0.
This is because the logical vectors will be treated as vectors of 0's and 1's in math operations.

2.5.3 Q1

```r
v1 <- c(7, 2, 4, 9, 7)
v2 <- c(6, 2, 8, 7, 9)
v3 <- 1:50
c(1, 2, 3, 4, 5)[v1 < v2]
```

```
## [1] 3 5
```

2.5.3 Q2

```r
v2[v1 > 5]
```

```
## [1] 6 7 9
```

2.5.3 Q3

```r
v3[v3 %% 7 == 0]
```

```
## [1]  7 14 21 28 35 42 49
```

2.5.3 Q4

```r
v3[v3 %% 8 == 0] <- 100
v3
```

```
##  [1]   1   2   3   4   5   6   7 100   9  10  11  12  13  14  15 100  17  18  19
## [20]  20  21  22  23 100  25  26  27  28  29  30  31 100  33  34  35  36  37  38
## [39]  39 100  41  42  43  44  45  46  47 100  49  50
```

2.7.4 Q1

```r
exe <- c(2, 0, -3, 0, 5, 6)
sort(exe, decreasing = T)
```

```
## [1]  6  5  2  0  0 -3
```

2.7.4 Q2

```r
rank(exe, ties.method = 'min')[exe %in% c(2, 0)]
```

```
## [1] 4 2 2
```

2.7.4 Q3

```r
order(exe)
```

```
## [1] 3 2 4 1 5 6
```

2.13.6 Q1

```r
v1 <- seq(from = 1, to = 100, by = 3)
v2 <- sqrt(v1)
v1s <- v1[v1 > 30 & v1 < 60]
v1s
```

```
##  [1] 31 34 37 40 43 46 49 52 55 58
```

2.13.6 Q2

```r
v2[v1 < 20 | v1 > 50]
```

```
##  [1]  1.000000  2.000000  2.645751  3.162278  3.605551  4.000000  4.358899
##  [8]  7.211103  7.416198  7.615773  7.810250  8.000000  8.185353  8.366600
## [15]  8.544004  8.717798  8.888194  9.055385  9.219544  9.380832  9.539392
## [22]  9.695360  9.848858 10.000000
```

2.13.6 Q3

```r
a <- c(T, F, T, F)
b <- c(T, T, F, F)
xor(a, b) == (!a & b) | (a & !b)
```

```
## [1] TRUE TRUE TRUE TRUE
```

2.15.3 Q1

```r
x <- rep(c(1, 2, NA), 3:5)
```

a.

```r
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   1.000   1.000   2.000   1.571   2.000   2.000       5
```

```r
min(x, na.rm = T)
```

```
## [1] 1
```

```r
quantile(x, 0.25, na.rm = T)
```

```
## 25% 
##   1
```

```r
median(x, na.rm = T)
```

```
## [1] 2
```

```r
mean(x, na.rm = T)
```

```
## [1] 1.571429
```

```r
quantile(x, 0.75, na.rm = T)
```

```
## 75% 
##   2
```

```r
max(x, na.rm = T)
```

```
## [1] 2
```

```r
sum(is.na(x))
```

```
## [1] 5
```

b.

```r
which(is.na(x))
```

```
## [1]  8  9 10 11 12
```

c.

```r
x_no_na <- na.omit(x)
x_no_na
```

```
## [1] 1 1 1 2 2 2 2
## attr(,"na.action")
## [1]  8  9 10 11 12
## attr(,"class")
## [1] "omit"
```

d.

```r
x[is.na(x)] <- median(x, na.rm = T)
x
```

```
##  [1] 1 1 1 2 2 2 2 2 2 2 2 2
```

2.15.3 Q2

```r
y <- rep(c("N", 2, "A"), 5:3)
y[y %in% c('N', 'A')] <- NA
y
```

```
##  [1] NA  NA  NA  NA  NA  "2" "2" "2" "2" NA  NA  NA
```

```r
which(is.na(y))
```

```
## [1]  1  2  3  4  5 10 11 12
```