HW04

Deblina Mukherjee

April 23, 2019

What a `for` loops look like conceptually

1. Pre-allocating space specifying data type of vector and it's length
2. Call for loop and specify element in list to iterate on
3. Specify function and output
4. Create output

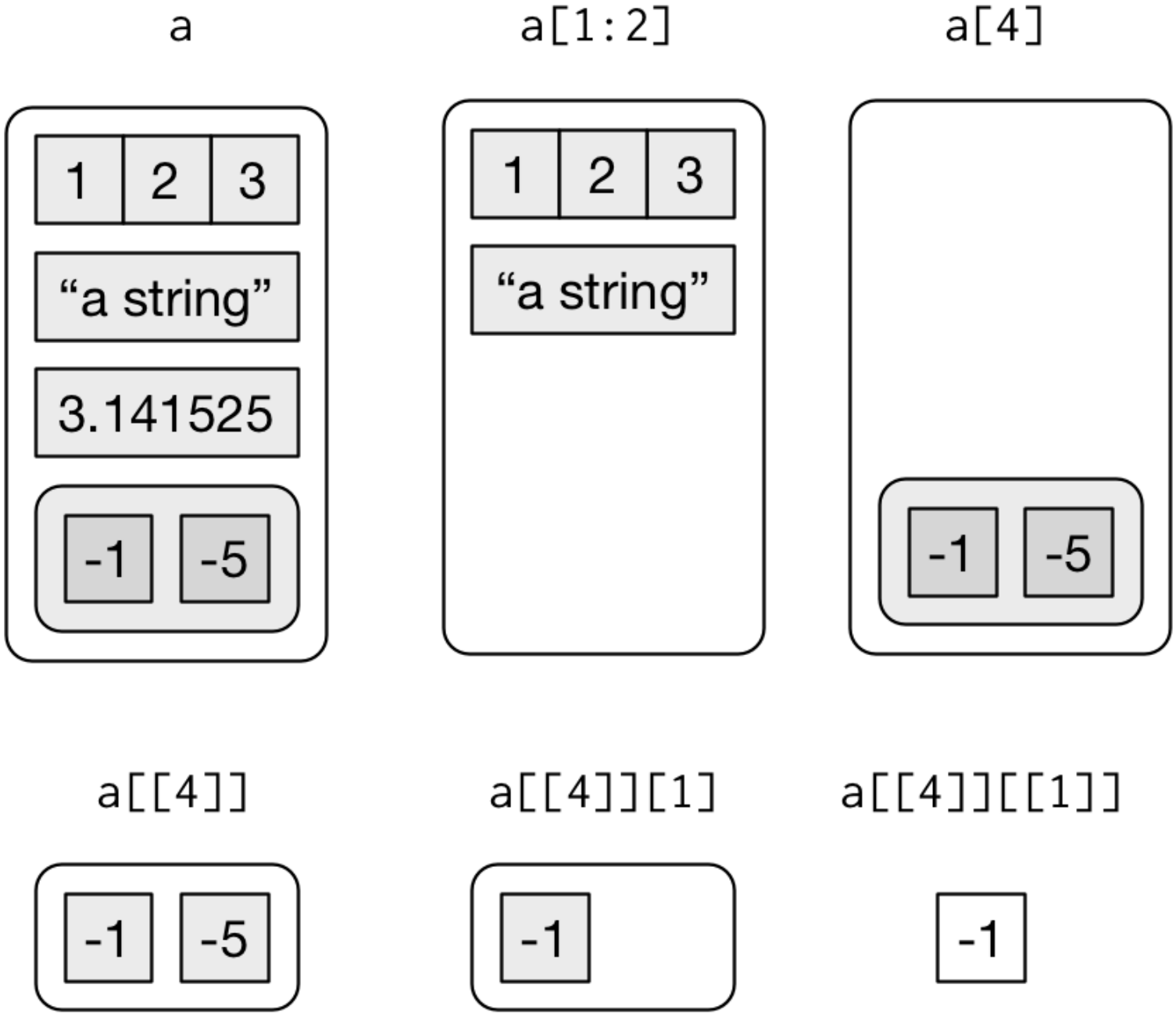
What a `for` loop looks like practically

```
output <- vector("numeric", ncol(mtcars))

for(i in seq_along(mtcars)){
  output[[i]] <- length(unique(mtcars[[i]]))
}

output
```

Subsetting Lists



Anonymous Functions

```
map_int(mtcars, function(x) length(unique(x)))
```

Pythagorean Problem

- Conceptual Roadmap:
  - Inputs: too many? too few? all numeric?
  - Sort the inputs: what sides of the triangle do we have?
  - Perform the operation (depending on what sides you have)

Pythagorean Problem (cont.)

- Functions you may find helpful:
  - `sort()`
  - `if()`
  - `length()`
  - `unique()`
  - `stop()`
  - `else if()`

An example of the `stop()` function

```
if(!is.numeric(x)){
    stop("At least one argument is not numeric.
         Only provide numbers.")
}
```