Data Structures in R: Arrays and Factors

Stat 133 with Gaston Sanchez
Creative Commons Attribution Share-Alike 4.0 International CC BY-SA
Data Types & Vectors recap
Data Types (primitives)

1L    # integer
2.5   # double (real)
TRUE  # logical
"hello" # character
1 + 3i # complex
Fundamental concepts

Atomic objects

Coercion

Subsetting or Bracket Notation

Vectorization

Recycling
Atomic objects

**Atomic**: all elements must be of the same data type.

In other words: you cannot have an atomic object containing elements of different types.
Matrices and Arrays
single data type

Vector

1D

Matrix

2D

Array

nD

Atomic objects
Arrays and matrices

You can transform a vector in an n-dimensional array by giving it a `dimensions` attribute.

```r
x <- 1:8
dim(x) <- c(2, 4)
```
Arrays and matrices

The **dimensions** attribute is a numeric vector with as many elements as desired dimensions

```r
x <- 1:8
dim(x) <- c(2, 2, 2)
```

*In practice, we don’t really create matrices and arrays via `dim()`*
Arrays and matrices

To have more control about how a matrix is filled, we use the function `matrix()`

```r
a <- 1:8
A <- matrix(a, nrow = 2, ncol = 4)
```
About R matrices

R stores matrices as vectors.
Which means that matrices are also atomic.
Matrices in R are stored column-major (i.e. by columns).
This is like Fortran, Matlab, and Julia, but not like C or Python (e.g. numpy).
Arrays and matrices

If you want to fill a matrix by rows use

\texttt{byrow = TRUE}

\begin{verbatim}
b <- 1:8
B <- matrix(a, nrow = 2, ncol = 4, byrow = TRUE)
\end{verbatim}
Vector

1D

Matrix

2D

Array

nD

Object

\[ \text{object}[i] \]

\[ \text{object}[i,j] \]

\[ \text{object}[i,j,k] \]

\[ \text{object}[i,j,k,l] \]
So far

Vectors, matrices, and arrays are atomic objects (they can only store one type of data)

Many operations in R need atomic structures to make sure all values are of the same mode

In real life, however, many datasets contain multiple types of information

R provides other data structures for this purpose
Factors
R factors

Another data structure in R are factors

A factor is designed to handle categorical data

The name “factor” comes from “Analysis of Variance” (ANOVA) terminology
R factors

To create a factor, typically you pass a vector to the function `factor()

```r
size <- c("sm", "md", "lg", "md")
size <- factor(size)
```
About R factors

Factors are excellent for working with categorical data, especially data with an “ordinal” scale.

Factors are internally stored as vectors of integers.

Factors behave a lot like vectors.

But factors have their own special properties.
Codification issues
Codification

It is very common that we (humans) codify information (e.g. data, variables) in many interesting ways.

It can make completely sense to us.

But not necessarily to the computer.
**Binary scale variable**

<table>
<thead>
<tr>
<th>Example</th>
<th>R mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE, FALSE</td>
<td>logical</td>
</tr>
<tr>
<td>0, 1</td>
<td>numeric</td>
</tr>
<tr>
<td>&quot;yes&quot;, &quot;no&quot;</td>
<td>character</td>
</tr>
<tr>
<td>yes, no</td>
<td>factor</td>
</tr>
</tbody>
</table>
# Nominal scale variable

<table>
<thead>
<tr>
<th>Example</th>
<th>R mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3</td>
<td>numeric</td>
</tr>
<tr>
<td>&quot;blue&quot;, &quot;white&quot;, &quot;red&quot;</td>
<td>character</td>
</tr>
<tr>
<td>blue, white, red</td>
<td>factor</td>
</tr>
</tbody>
</table>
Odinal scale variable

<table>
<thead>
<tr>
<th>Example</th>
<th>R mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3</td>
<td>numeric</td>
</tr>
<tr>
<td>&quot;small&quot;, &quot;medium&quot;, &quot;large&quot;</td>
<td>character</td>
</tr>
<tr>
<td>small, medium, large</td>
<td>factor</td>
</tr>
</tbody>
</table>
## Interval / Ratio scale variables in R

<table>
<thead>
<tr>
<th>Example</th>
<th>R mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1, -2.5, 100</td>
<td>numeric</td>
</tr>
<tr>
<td>1/4, pi, exp(1)</td>
<td>numeric</td>
</tr>
</tbody>
</table>
## Missing Values

<table>
<thead>
<tr>
<th>Example</th>
<th>R mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>logical</td>
</tr>
<tr>
<td>-999</td>
<td>numeric</td>
</tr>
<tr>
<td>-99999</td>
<td>numeric</td>
</tr>
<tr>
<td>&quot;?&quot;</td>
<td>character</td>
</tr>
<tr>
<td>&quot; &quot;</td>
<td>character</td>
</tr>
<tr>
<td>&quot;na&quot;</td>
<td>character</td>
</tr>
</tbody>
</table>
single data type

Vector

1D

Matrix

2D

Array

nD

multiple data types

List

Data Frame

non-atomic structures