Using data in R

Alexandre Courtiol

Leibniz Institute of Zoo and Wildlife Research

June 2018

Table of contents

1. Vector
    - Factors
2. Data frame
3. List

Table of contents

1. Vector
   - Factors
2. Data frame
3. List

Vector

A vector is a sequence of data elements of the same basic type

- Vectors allow the organisation of entities (e.g. numbers, characters. . .) along one dimension which can be indexed

```
height.girls <- c(178, 175, 159, 164, 183, 192)
height.boys <- c(181, 189, 174, 177)
```

```
height.girls[2]

## [1] 175

height.boys[3]

## [1] 174
```

Vector

- They can be combined:

```
(height <- c(height.boys, height.girls))

##  [1] 181 189 174 177 178 175 159 164 183 192
```

Vector continued

- They can be indexed logically (i.e. indexed by anything leading to a vector of booleans):

```
(height > 168)

##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE
## [10]  TRUE
```

```
height[height > 168]

## [1] 181 189 174 177 178 175 183 192

height[!(height == min(height))]

## [1] 181 189 174 177 178 175 164 183 192

height[height != min(height)]

## [1] 181 189 174 177 178 175 164 183 192
```

Factors

- They work with other things than numbers:

```r
sex <- c("girl","girl","girl","girl","girl", "girl",
"boy","boy","boy","boy")
sex <- factor(sex)
sex

##  [1] girl girl girl girl girl girl boy  boy  boy  boy
## Levels: boy girl
```

```r
# Or
sex <- factor(c(rep("girl", times = 6),
              rep("boy", times = 4)))
# Or

sex <- factor(c(rep("girl", times = length(height.girls)),
              rep("boy", times = length(height.boys))))
```

Changing the order of levels of a factor

You have:

```
my_factor1
```

```
## [1] A A B B C
## Levels: A B C
```

You want:

```
my_factor2
```

```
## [1] A A B B C
## Levels: C B A
```

Changing the order of levels of a factor

You have:

```
my_factor1
```

```
## [1] A A B B C
## Levels: A B C
```

You want:

```
my_factor2
```

```
## [1] A A B B C
## Levels: C B A
```

You do:

```
## Using base:
my_factor2 <- factor(my_factor1, levels(my_factor1)[c(3, 2, 1)])
my_factor2
```

```
## [1] A A B B C
## Levels: C B A
```

Changing the order of levels of a factor

You have:

```
my_factor1

## [1] A A B B C
## Levels: A B C
```

You want:

```
my_factor2

## [1] A A B B C
## Levels: C B A
```

You do:

```
## Using base:
my_factor2 <- factor(my_factor1, levels(my_factor1)[c(3, 2, 1)])
my_factor2

## [1] A A B B C
## Levels: C B A
```

Note: the order of levels influences the output of linear models and plotting functions (e.g. order in the legend of a ggplot) . . .

Changing the levels of a factor

You have:

```
my_factor1

## [1] A A B B C
## Levels: A B C
```

You want:

```
my_factor2

## [1] A A A A D
## Levels: A D
```

You do:

```
## Using base:
levels(my_factor1)

## [1] "A" "B" "C"

my_factor2 <- my_factor1
levels(my_factor2) <- c("A", "A", "D") ## in same order!
my_factor2

## [1] A A A A D
## Levels: A D
```

```
## Using dplyr:
my_factor2 <- recode(my_factor1, A = "A", B = "A", C = "D")
my_factor2

## [1] A A A A D
## Levels: A D
```

Data frame

List