Working with Logicals

R

Logical operators

Logical operators are used to phrase yes/no questions in terms of R objects and create logical objects.

- `!` : the "not" operator
- `&` : the "and" operator
- `|` : the "or" operator (Shift+backslash)
- `==` : the "equals" operator
- `!=` : the "not equal" operator
- `>` : the "greater than" operator
- `>=` : the "greater than or equal to" operator
- `<` : the "less than" operator
- `<=` : the "less than or equal to" operator
- `%in%` : the "contained in" operator

Example setup

Student ages:

```
ages <- c(12, 17, 16, 13, 14)
```

Age cutoff for activity 1 (common to all students):

```
common_cutoff <- 13
```

Age cutoffs for activity 2 (individualized):

```
indiv_cutoffs <- c(12, 12, 14, 14, 14)
```

Example: ==

Do the students' ages equal the common cutoff?

```
ages <- c(12, 17, 16, 13, 14)
common_cutoff <- 13
indiv_cutoffs <- c(12, 12, 14, 14, 14)
```

```
> ages==common_cutoff
[1] FALSE FALSE FALSE  TRUE FALSE
```

Do the students' ages equal the individualized cutoffs?

```
> ages == indiv_cutoffs
[1]  TRUE FALSE FALSE FALSE  TRUE
```

Example: > and >=

Do the students' ages exceed the common cutoff? The individualized cutoffs?

```
ages <- c(12, 17, 16, 13, 14)
common_cutoff <- 13
indiv_cutoffs <- c(12, 12, 14, 14, 14)
```

```
> ages > common_cutoff
[1] FALSE  TRUE  TRUE FALSE  TRUE
> ages > indiv_cutoffs
[1] FALSE  TRUE  TRUE FALSE FALSE
```

Are the students' at least as old as the common cutoff? The individualized cutoffs?

```
> ages >= common_cutoff
[1] FALSE  TRUE  TRUE  TRUE  TRUE
> ages >= indiv_cutoffs
[1]  TRUE  TRUE  TRUE FALSE  TRUE
```

Example: < and <=

Are the students' ages less than the common cutoff? The individualized cutoffs?

```
ages <- c(12, 17, 16, 13, 14)
common_cutoff <- 13
indiv_cutoffs <- c(12, 12, 14, 14, 14)
```

```
> ages < common_cutoff
[1]  TRUE FALSE FALSE FALSE FALSE
> ages < indiv_cutoffs
[1] FALSE FALSE FALSE  TRUE FALSE
```

Are the students' no older than the common cutoff? The individualized cutoffs?

```
> ages <= common_cutoff
[1]  TRUE FALSE FALSE  TRUE FALSE
> ages <= indiv_cutoffs
[1]  TRUE FALSE FALSE  TRUE  TRUE
```

Example: `&` and `|`

Are the students older than the common cutoff **and** the individualized cutoffs?

```
ages <- c(12, 17, 16, 13, 14)
common_cutoff <- 13
indiv_cutoffs <- c(12, 12, 14, 14, 14)
```

```
> ages > common_cutoff & ages > indiv_cutoffs
[1] FALSE  TRUE  TRUE FALSE FALSE
```

Are the students older than the common cutoff **or** the individualized cutoffs?

```
> ages > common_cutoff | ages > indiv_cutoffs
[1] FALSE  TRUE  TRUE FALSE  TRUE
```

Example: !

Are the students older than the common cutoff **but not** the individualized cutoffs?

```
ages <- c(12, 17, 16, 13, 14)
common_cutoff <- 13
indiv_cutoffs <- c(12, 12, 14, 14, 14)
```

```
> ages > common_cutoff & !(ages > indiv_cutoffs)
[1] FALSE FALSE FALSE FALSE  TRUE
```

Parentheses recommended for clarity and correctness

Another way to get the same result:

```
> ages > common_cutoff & ages <= indiv_cutoffs
[1] FALSE FALSE FALSE FALSE  TRUE
```

Example: storing intermediate objects

Are the students older than the common cutoff **but not** the individualized cutoffs?

```
ages <- c(12, 17, 16, 13, 14)
common_cutoff <- 13
indiv_cutoffs <- c(12, 12, 14, 14, 14)
```

```
> meets_common_cut <- ages > common_cutoff
> not_meets_indiv_cut <- !(ages > indiv_cutoffs)
> meets_common_cut
[1] FALSE  TRUE  TRUE FALSE  TRUE
> not_meets_indiv_cut
[1]  TRUE FALSE FALSE  TRUE  TRUE
> meets_common_cut & not_meets_indiv_cut
[1] FALSE FALSE FALSE FALSE  TRUE
```

Intermediate steps

More easily readable expression

Example: character objects

Are the colors “red”?

```
> colors <- c("red", "red", "green", "orange", "blue")
> colors == "red"
[1]  TRUE  TRUE FALSE FALSE FALSE
```

Are the colors not “blue”?

```
> colors != "blue"
[1]  TRUE  TRUE  TRUE  TRUE FALSE
```

Example: `%in%`

Are “red” and “purple” contained in this set of colors?

```
> c("red", "purple") %in% colors
[1]  TRUE FALSE
```

length(left hand side) = length(output) = 2

Are the colors in the `colors` object contained in the red and purple set?

```
> colors %in% c("red", "purple")
[1]  TRUE  TRUE FALSE FALSE FALSE
```

length(left hand side) = length(output) = 5

Example: factor objects

```
> height_factor <- factor(c(2,1,2,3,1), levels = 1:3,
labels = c("short", "average", "tall"))
> height_factor
[1] average short   average tall    short
Levels: short average tall
```

Are the heights equal to the shortest level (short)?

```
> height_factor == 1
[1] FALSE FALSE FALSE FALSE FALSE
> as.integer(height_factor)
[1] 2 1 2 3 1
> as.integer(height_factor) == 1
[1] FALSE  TRUE FALSE FALSE  TRUE
> height_factor == "short"
[1] FALSE  TRUE FALSE FALSE  TRUE
```

Doesn't work as desired

Equivalent

Logical functions

Functions in R facilitate answering natural questions concerning logicals:

- Do all units meet the condition?
- Do any units meet the condition?
- Who meets the condition?
- How many units meet the condition?

`all` function

Are all student ages equal to the individual cutoffs?

```
> all(ages == indiv_cutoffs)
[1] FALSE
```

Are all ages greater than or equal to zero?

```
> all(ages >= 0)
[1] TRUE
```

`any` function

Are any of the student ages equal to the individual cutoffs?

```
> any(ages == common_cutoff)
[1] TRUE
```

Are any ages greater than 100?

```
> any(ages > 100)
[1] FALSE
```

`which` function

Provides the indices for TRUE values.

Which colors are contained within the red and purple set?

```
> colors <- c("red", "red", "green", "orange", "blue")
> colors %in% c("red", "purple")
[1]  TRUE  TRUE FALSE FALSE FALSE
> which(colors %in% c("red", "purple"))
[1] 1 2
```

Functions for answering “How many?”

The `sum` function can count the number of TRUEs.

The `mean` function can compute the fraction of TRUEs.

```
> meets_indiv_cut <- ages >= indiv_cutoffs
> meets_indiv_cut
[1]  TRUE  TRUE  TRUE FALSE  TRUE
> sum(meets_indiv_cut)
[1] 4
> mean(meets_indiv_cut)
[1] 0.8
```