Online CDS 101 *Introduction to Computational and Data Sciences*

Data transformation

Boolean logic (or, how to make a rule)

Building rules

Building rules

- The `filter` function takes logical rules as input

Building rules

- The `filter` function takes logical rules as input
- Logical rules must evaluate to one of two outputs, either `TRUE` or `FALSE`

Building rules

- The `filter` function takes logical rules as input
- Logical rules must evaluate to one of two outputs, either `TRUE` or `FALSE`
- Chaining rules together is allowed, so long as the final result is simply `TRUE` or `FALSE`

Building rules

- The `filter` function takes logical rules as input
- Logical rules must evaluate to one of two outputs, either `TRUE` or `FALSE`
- Chaining rules together is allowed, so long as the final result is simply `TRUE` or `FALSE`
- Working with `TRUE` and `FALSE` values in this way is a form of **Boolean logic**

Comparisons

Comparisons

Simple comparisons can be made using the following symbols

Comparisons

Simple comparisons can be made using the following symbols

- `>`: greater than

Comparisons

Simple comparisons can be made using the following symbols

- `>`: greater than
- `>=`: greater than or equal to

Comparisons

Simple comparisons can be made using the following symbols

- `>`: greater than
- `>=`: greater than or equal to
- `<`: less than

Comparisons

Simple comparisons can be made using the following symbols

- `>`: greater than
- `>=`: greater than or equal to
- `<`: less than
- `<=`: less than or equal to

Comparisons

Simple comparisons can be made using the following symbols

- `>`: greater than
- `>=`: greater than or equal to
- `<`: less than
- `<=`: less than or equal to
- `!=`: not equal

Comparisons

Simple comparisons can be made using the following symbols

- `>`: greater than
- `>=`: greater than or equal to
- `<`: less than
- `<=`: less than or equal to
- `!=`: not equal
- `==`: equal

Logical operators

Logical operators

- `&`: *and* Boolean operator

Logical operators

- `&`: *and* Boolean operator
- `|`: *or* Boolean operator

Logical operators

- `&`: *and* Boolean operator
- `|`: *or* Boolean operator
- `!`: *not* Boolean operator

Logical operators

- `&`: *and* Boolean operator
- `|`: *or* Boolean operator
- `!`: *not* Boolean operator
- `xor`: *exclusive or* Boolean operator

Using the *and* operator

```
TRUE & TRUE
```

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

```
TRUE & FALSE
```

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

```
FALSE & TRUE
```

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

```
FALSE & FALSE
```

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

Using the *or* operator

```
TRUE | TRUE
```

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

```
TRUE | FALSE
```

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

```
FALSE | TRUE
```

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

```
FALSE | FALSE
```

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

Using the *exclusive or* operator

```
xor(TRUE, TRUE)
```

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

```
xor(TRUE, FALSE)
```

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

```
xor(FALSE, TRUE)
```

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

```
xor(FALSE, FALSE)
```

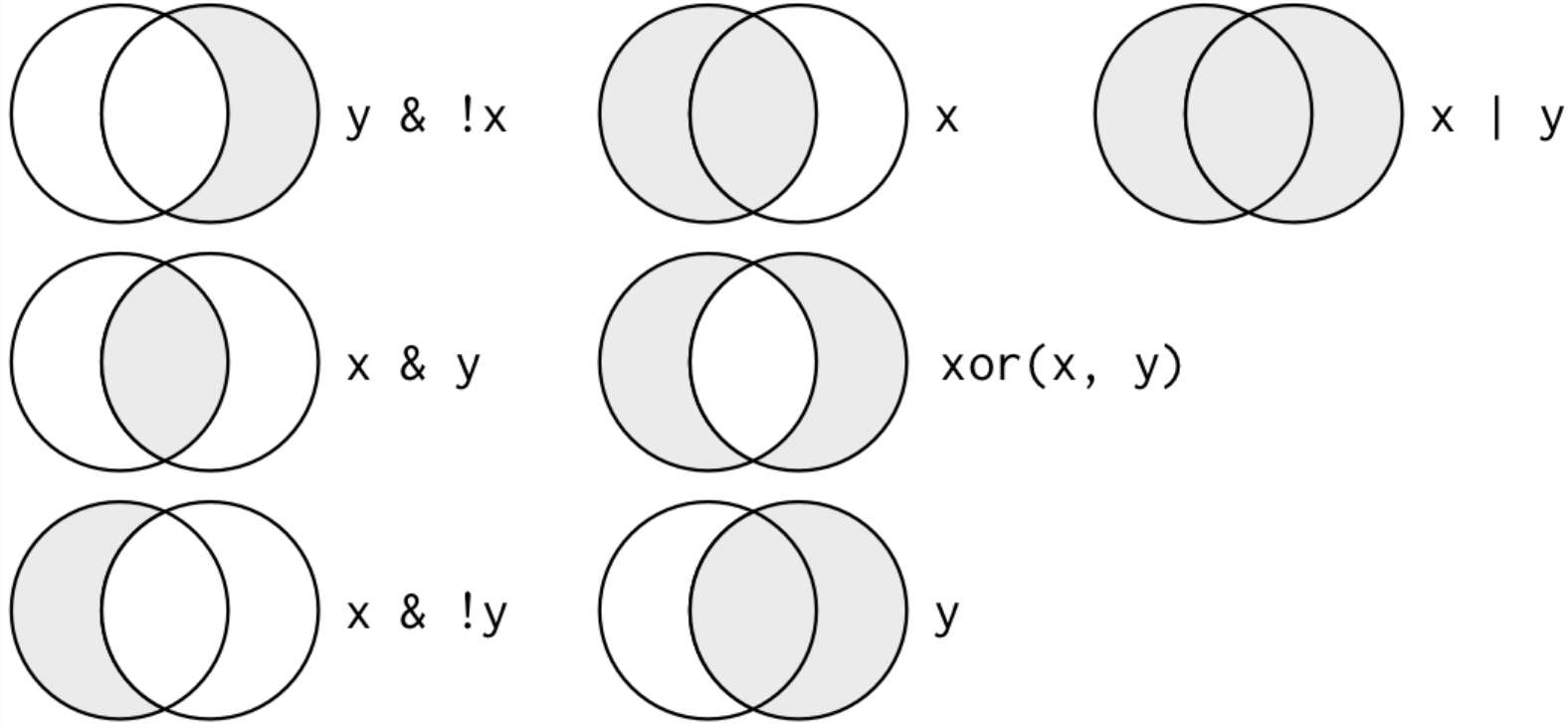
```
## [1] FALSE
```

Using the *not* operator

```
TRUE & !FALSE
```

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

Logical operators schematic



Source: Digital image of logical operations, Digital image on *r4ds.had.co.nz*, accessed September 20, 2017, r4ds.had.co.nz/transform.html#logical-operators

Credits

License