# Introduction to Social Data Analytics
# Week 6: Class 12

Arushi Kaushik

arkaushi@ucsd.edu

# Today: logic and subsetting in R

By the end of today's lecture, you should be able to:

- Write logic statements in R and identify the relevant Boolean operators
- Generate subsets of data using logic operators and $

Open class12.R if you haven't already.

# Some Basic logical statements in R

```
"Triton" != "triton"
```

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

```
5<3 | FALSE
```

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

```
vec1 <- c(TRUE, TRUE, FALSE, TRUE); sum(vec1) > 3
```

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

# Creating lists and sequences

- Suppose you want a list of numbers increasing by 1

```
list1 <- 1:5
list1
```

```
## [1] 1 2 3 4 5
```

```
list1 > 3
```

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

```
(list1 > 1) & (list1 < 5)
```

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

# Now suppose you want a list of numbers increasing by 0.5

- Use the command `seq(start, end, increment)`

```
sequence1 <- seq(1,5,0.5)
sequence1
```

```
## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

```
(sequence1 > 1) & (sequence1 < 5)
```

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

- Now generate a sequence with six elements starting at 10 and ending at 20.

# Some "class" examples

```
x = 5
class(x)
```

```
## [1] "numeric"
```

```
y = "My name is King Triton."
class(y)
```

```
## [1] "character"
```

```
z = TRUE
class(z)
```

```
## [1] "logical"
```

# The *class* function outputs the class of the object you insert

- Separate classes cannot in general be combined unless you convert them first
- We can coerce objects to be recognized as different classes

# Objects: coercion with "as"

```
z.str <- c("1", "2", "602")
z.num <- as.numeric(z.str)
z.str
```

```
## [1] "1"   "2"   "602"
```

```
z.num
```

```
## [1]   1   2 602
```

```
# z.str + c(1,1,1) will give an error
z.num + c(1,1,1)
```

```
## [1]   2   3 603
```

# More "as" functions

```
a_num <- c(1,2,3, 3498)
a_str <- as.character(a_num)
a_num
```

```
## [1]    1    2    3 3498
```

```
a_str
```

```
## [1] "1"    "2"    "3"    "3498"
```

```
a_num + c(1,1,1,1)
```

```
## [1]    2    3    4 3499
```

```
# a_str + c(1,1,1,1) will give an error
```

# But, "is" it what you think?

```
is.character(z.num)
```

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

```
is.character(z.str)
```

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

```
is.numeric(a_num)
```

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

```
is.numeric(a_str)
```

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

# How can we tell if something `isTRUE`?

```
isTRUE(TRUE)
```

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

```
isTRUE(FALSE)
```

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

```
isTRUE(T)
```

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

```
isTRUE(1)
```

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

# But be careful...

```
as.integer(TRUE)
```

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

```
T <- 1
isTRUE(T)
```

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

- T is a predefined "global" that is equivalent to TRUE, but can be changed.

# Also...

```
logic <- c(TRUE, TRUE, FALSE, TRUE)
mean(logic)
```

```
## [1] 0.75
```

```
sum(logic)
```

```
## [1] 3
```

# Coercion

- Coercion is when you call a variable of the wrong type and R tries to fix it for you.

```
vec2 <- c(1,2,3)
vec2
```

```
## [1] 1 2 3
```

```
typeof(vec2)
```

```
## [1] "double"
```

```
class(vec2)
```

```
## [1] "numeric"
```

# But ...

```
vec2[1] = "text"
class(vec2)
```

```
## [1] "character"
```

```
vec2
```

```
## [1] "text" "2"    "3"
```

```
# or
TRUE == "TRUE"
```

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

- Both are examples of coercion.

# Subsetting

- `[]` is for subsets
- `$` is for extracting by name (so what you want must be named to be used...)
- What does `resume$race[1:5]` give us?
- What does `resume[,"race"]` give us?
- What does `mean(resume$call[resume$race == "black"])` give us?

# Using subset()

- R has a function that can help generate subsets of dataframes

```
black <- subset(resume, select = c("firstname", "sex",
        "call"), subset = (race == "black"))
```

# Here are the commands/operators we covered today:

- `&`, `|`, `!`, `==`
- `:`, `seq()`
- `class()`, `typeof()`
- `as.numeric()`, `as.character()`
- `is.numeric()`, `is.character()`, `isTRUE()`
- `subset()`