Representing data in R

Jeffrey Leek, Assistant Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

Important data types in R

Classes

- Character, Numeric, Integer, Logical

Objects

- Vectors, Matrices, Data frames, Lists, Factors, Missing values

---

Operations

- Subsetting, Logical subsetting

For more information:

- Data Types

Character

```
firstName = "jeff"
class(firstName)
```

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

```
firstName
```

```
## [1] "jeff"
```

Numeric

```
heightCM = 188.2
class(heightCM)
```

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

```
heightCM
```

```
## [1] 188.2
```

Integer

```
numberSons = 1L
class(numberSons)
```

```
## [1] "integer"
```

```
numberSons
```

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

Logical

```
teachingCoursera = TRUE
class(teachingCoursera)
```

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

```
teachingCoursera
```

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

Vectors

A set of values with the same class

```
heights = c(188.2, 181.3, 193.4)
heights
```

```
## [1] 188.2 181.3 193.4
```

```
firstNames = c("jeff", "roger", "andrew", "brian")
firstNames
```

```
## [1] "jeff"   "roger"  "andrew" "brian"
```

Lists

A vector of values of possibly different classes

```
vector1 = c(188.2, 181.3, 193.4)
vector2 = c("jeff", "roger", "andrew", "brian")
myList = list(heights = vector1, firstNames = vector2)
myList
```

```
## $heights
## [1] 188.2 181.3 193.4
##
## $firstNames
## [1] "jeff"   "roger"  "andrew" "brian"
```

Matrices

Vectors with multiple dimensions

```
myMatrix = matrix(c(1, 2, 3, 4), byrow = T, nrow = 2)
myMatrix
```

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

Data frames

Multiple vectors of possibly different classes, of the same length

```
vector1 = c(188.2, 181.3, 193.4)
vector2 = c("jeff", "roger", "andrew", "brian")
myDataFrame = data.frame(heights = vector1, firstNames = vector2)
```

```
## Error: arguments imply differing number of rows: 3, 4
```

```
myDataFrame
```

```
## Error: object 'myDataFrame' not found
```

Data frames

```
vector1 = c(188.2, 181.3, 193.4, 192.3)
vector2 = c("jeff", "roger", "andrew", "brian")
myDataFrame = data.frame(heights = vector1, firstNames = vector2)
myDataFrame
```

```
##   heights firstNames
## 1   188.2       jeff
## 2   181.3      roger
## 3   193.4     andrew
## 4   192.3      brian
```

Factors

Qualitative variables that can be included in models

```
smoker = c("yes", "no", "yes", "yes")
smokerFactor = as.factor(smoker)
smokerFactor
```

```
## [1] yes no  yes yes
## Levels: no yes
```

Missing values

In R they are usually coded NA

```
vector1 = c(188.2, 181.3, 193.4, NA)
vector1
```

```
## [1] 188.2 181.3 193.4    NA
```

```
is.na(vector1)
```

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

Subsetting

```
vector1 = c(188.2, 181.3, 193.4, 192.3)
vector2 = c("jeff", "roger", "andrew", "brian")
myDataFrame = data.frame(heights = vector1, firstNames = vector2)

vector1[1]
```

```
## [1] 188.2
```

```
vector1[c(1, 2, 4)]
```

```
## [1] 188.2 181.3 192.3
```

Subsetting

```
myDataFrame[1, 1:2]
```

```
##   heights firstNames
## 1   188.2       jeff
```

```
myDataFrame$firstNames
```

```
## [1] jeff   roger  andrew brian
## Levels: andrew brian jeff roger
```

Logical subsetting

```
myDataFrame[firstNames == "jeff", ]
```

```
##   heights firstNames
## 1   188.2       jeff
```

```
myDataFrame[heights < 190, ]
```

```
##   heights firstNames
## 1   188.2       jeff
## 2   181.3      roger
## 4   192.3      brian
```

Variable naming conventions

Variable names should be short, but descriptive. Here are some common styles

Camel caps

```
myHeightCM = 188
```

Underscore

```
my_height_cm = 188
```

Dot separated

```
my.height.cm = 188
```

Style guides

- http://4dpiecharts.com/r-code-style-guide/
- http://google-styleguide.googlecode.com/svn/trunk/google-r-style.html
- http://wiki.fhcrc.org/bioc/Coding_Standards