concepts

Code

R basic concept

1. indexing
   all indexing in R is from 1 not 0
   the slicing (eg 1:3): start and end are included
2. R language is case sensitive.
3. R operations are vector based, performs on each element of vector.

Variable naming Conventions

Or period `.`

- A variable name must start with a letter and can be a combination of letters, digits, period (.) and underscore (_). If it starts with period (.), it cannot be followed by a digit.
- A variable name cannot start with a number or underscore (_)
- Variable names are case-sensitive
- Reserved words cannot be used as variable name (TRUE, FALSE, NULL, `if`...)

datatype

1. data type intro

common data type: numeric, logical, character

1. numeric
   Includes both both int and float (they have different classes like `integer` or `numeric` (which is double), but the mode is all `numeric`) Note: to explicitly name integer, use suffix `L` , eg `1L` means integer 1
2. character
   String or character (like python), denoted by double quote `"` or single quote `'` no diff
3. logical
   `TRUE` and `FALSE` , can also `T` and `F`

2. Inspect data type

1. `mode()`
   describe the data type used for storage.
   Only(?) have: `numeric, logical, character` , etc.
2. `class()`
   describe the object class of the input variable
   e.g., `numeric` (here numeric is "double", differen), `integer` , `list` , `matrix` , `factor` , etc.
3. egs

```
> mode(1L); class(1L)
[1] "numeric"
[1] "integer"

> mode(factor(c("a","b"))); class(factor(c("a","b")))
[1] "numeric"
[1] "factor"

> mode(matrix(0,2,2)); class(matrix(0,2,2))
[1] "numeric"
[1] "matrix"

# the class of basic vectors (only the numeric has a difference)
> class(1)
[1] "numeric"
> class(1L)
[1] "integer"
> class(T)
[1] "logical"
> class("a")
[1] "character"

# the return class of both mode and class is "character"
> class(mode(1))
[1] "character"
> class(class(1))
[1] "character"
```

Special values

Missing value `NA`

1. Intro Missing values often arise in real data problems, but they can also arise because of the way calculations are performed.
2. Inspect: `is.na()`

Technical: storage of numbers

lec1 49-50