Refresh-R

Caitlin Collins, Thibaut Jombart

MRC Centre for Outbreak Analysis and Modelling
Imperial College London

Genetic data analysis using R
30-10-2014

Outline

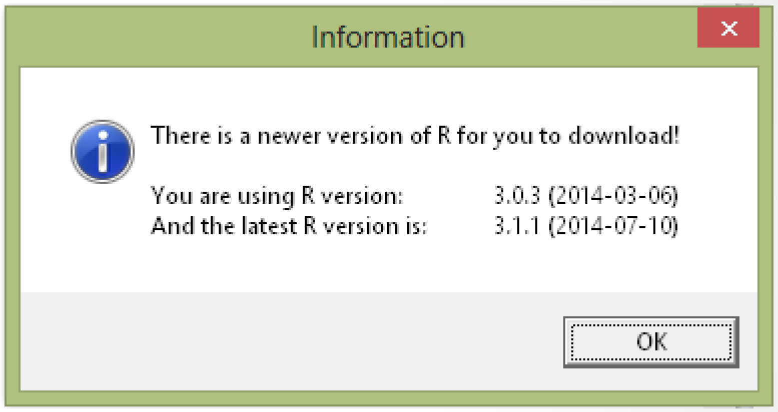
- Getting started
  - Installing R, loading libraries, getting help
- Data
  - Reading in data, data structures in R
- Objects
  - Classes, accessors
- Commands to know
  - Logical operations, subsetting
- Useful links
  - **R intro:** http://cran.r-project.org/doc/manuals/R-intro.pdf
  - ***adegenet* on the web:** http://adegenet.r-forge.r-project.org/

Getting staRted

…

Installing R

- Get the software—*for free!*
  - http://www.r-project.org
- Check your version
  - `> R.version.string`
  - *Are you using version **3.1.1**?*
    - **If not:**
    - `> install.packages("installr")`
    - `> library(installr)`
    - `> updateR()`
    - **In RStudio…**
      - Select menu: Tools > Global options… > R version > Change



Information

There is a newer version of R for you to download!

```
You are using R version:        3.0.3 (2014-03-06)
And the latest R version is:    3.1.1 (2014-07-10)
```

- R editors
  - **RStudio:** http://www.rstudio.org/
  - **Tinn-R:** http://sourceforge.net/projects/tinn-r/
  - **Emacs*:** http://ftp.gnu.org/gnu/emacs/
    - *requires compilation on Windows…

Modula-R

- Core
  - Basic “necessary” packages

- Packages
  - `> library()`
  - `> .libPaths()`

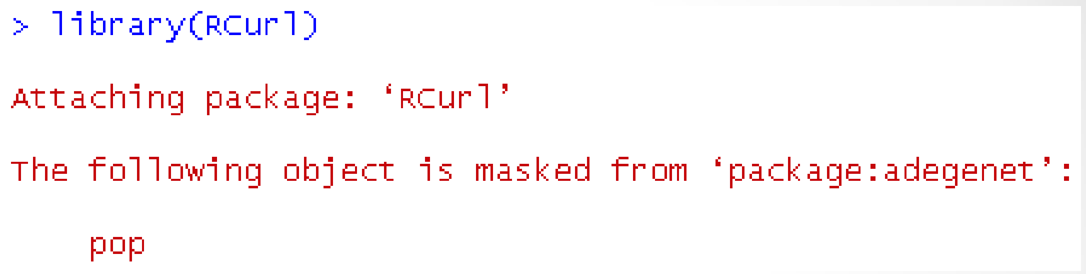
- Libraries must be installed and loaded explicitly
  - `> install.packages(“pkg”, dependencies = TRUE)`
  - `> library(pkg)`
  - `> require(pkg) # used inside functions`

- To unload packages:
  - `> detach("package:pkg", unload = TRUE)`

PRoblem solving

Conflicts

- Example:
  - `> library(adegenet)`
  - `> library(Rcurl)`



```
> library(RCurl)

Attaching package: 'RCurl'

The following object is masked from 'package:adegenet':

    pop
```

- Solution 1:
  - `> detach("package:RCurl", unload = TRUE)`
  - `> detach("package:adegenet", unload = TRUE)`
  - `> library(Rcurl) # load the minor pkg first`
  - `> library(adegenet) # load the more important pkg second`

- Solution 2.1:
  - `> adegenet::pop(x) # one-time solution`

- Solution 2.2:
  - `> pop <- adegenet::pop # “permanent” solution`
  - `> pop(x)`

Help!

- Package-level
  - `> ?pkg`
  - `> vignette(all = FALSE) # list vignettes from all *attached* pkgs`
- Function-level
  - `> help("fn") # search documentation for this package`
  - `> ?fn # same as above`
  - `> ??fn # extensive search (including in unloaded packages)`
  - `> ?plot.fn # searches for a specific version of a common fn (eg. plot)`
  - `> example(fn) # runs the example from the end of fn's documentation`
- More
  - Tutorials
    - ***adegenet* (basics):** http://adegenet.r-forge.r-project.org/files/tutorial-basics.pdf
    - ***adegenet* (DAPC):** http://adegenet.r-forge.r-project.org/files/tutorial-dapc.pdf
    - ***adegenet* (genomics):** http://adegenet.r-forge.r-project.org/files/tutorial-genomics.pdf
    - ***adegenet* (spatial-PCA):** http://adegenet.r-forge.r-project.org/files/tutorial-spca.pdf
  - Online resources
    - ***adegenet*:** http://adegenet.r-forge.r-project.org/
  - Forums & mailing lists
    - ***adegenet*:** adegenet-forum@lists.r-forge.r-project.org
    - ***R-sig-phylo:*** https://stat.ethz.ch/mailman/listinfo/r-sig-phylo
    - ***R-sig-genetics:*** https://stat.ethz.ch/mailman/listinfo/r-sig-genetics
  - Google <3

Errors

- Interpret
  - `> ?fn # check the documentation`
  - `> warnings() # prints errors if multiple errors generated`
  - `> traceback() # prints the calls that led to the error`
  - `> debug(fn(x)) # execute a fn one statement at a time`
  - `> options(error = recover) # switches to browser mode where the error occurs`
- Silence
  - `> suppressWarnings(fn(x))`
  - `> fn(x, silent = TRUE)`
- More:
  - http://www.biostat.jhsph.edu/~rpeng/docs/R-debug-tools.pdf

Objets d'R

(data)

Reading in data

- Working directory
  - `> getwd()`
  - `> setwd("C:/Users/YourName/")`

- Load data
  - `> foo <- get(load("~/PathFromWD.Rdata"))`
  - `> read.csv("~/PathFromWD.csv")`
  - `> read.table("~/PathFromWD.csv", header = TRUE)`

- Check
  - `> head(foo)`
  - `> dim(foo)`
  - `> names(foo)`
  - `> str(foo)`
  - `> summary(foo)`

Object classes

- Structures:
  - Vectors
  - Matrices
  - Data frames
  - Lists

- Useful functions:
  - `> class(x)`
  - Example with matrix:
  - `> matrix(x)`
  - `> is.matrix(x)`
  - `> as.matrix(x)`

Data types

- Types:
  - Character
  - Numeric
  - Integer
  - Logical
  - Factor
- Useful functions:
  - `> class(x) # get class or type`
  - `> length(x)`
  - `> str(x) # get structure of x`
  - `> names(x)`
  - `> c(x1, x2, …) # combine → vector`
  - `> cbind(x1, x2, …) # bind columns`
  - `> rbind(x, x2, …) # bind rows`
  - `> ls() # list all current objects`
  - `> rm(x) # remove object x`

S4s & accessors

- S3
  - The basic R object system
  - Easy to interact with

- S4 objects
  - More formal, rigorous system for objects, classes, methods
  - Use **slots** to compartmentalise interactions
    - Accessed by “accessors” (ie. `@`, `$`)
  - Examples of S4 objects: genind, genpop, genlight

- Useful functions for S4 objects:
  - **Example with genlight object:**
  - `> x <- new(“genlight”, input) # create objects`
  - `> getClassDef(“genlight”) # examine structure of objects of this class`
  - `> showClass(x)# get class`
  - `> slotNames(x)# retrieve all names of slots`
  - `> x@loc.names # access the slot containing loci names`
  - `> y <- as.matrix(x)[,c(1,2)] # make “y” a subset of x containing columns 1 & 2`

Commands to know

Logical operators

- `> !x # NOT x`
- `> x & y # x AND y (element-wise)`
- `> x && y # x AND y (left-to-right, 1st element only)`
- `> x | y # x OR y (element-wise)`
- `> x || y # x OR y (left-to-right, 1st element only)`
- `> x == y # is x EQUAL to y? → TRUE/FALSE`
- `> x %in% y # element-wise: is` x_n `in y? → TRUE/FALSE`
- `> !x %in% y # element-wise: is` x_n `NOT in y? → T/F`
- `> which(x %in% y) # → indices of the x in y`
- `> all(x == 1) # are ALL elements of x EQUAL to 1?`
- `> any(x==1) # are ANY elements of x EQUAL to 1?`
- `> which(x==1) # which elements of x are EQUAL to 1?`

Subsetting

- `> y <- x[1]` # get the 1st element of a vector
- `> y <- x[c(1:10, 100), 7]` # get the first 10 rows and the 100th row of column 7
- `> y <- x[[1]]` # get the 1st element of a list
- `> y <- x[[3]][[2]]` # get the 2nd item from within the 3rd item of a list
- `> y <- x$loc.names` # get the item/ slot of x named "loc.names"
- `> y <- x[["loc.names]]` # same as above
- `> nom <- "loc.names"; y <- x[[nom]]` # same as above
- `> y <- x[!is.na(x)]` # get all elements of x that are *not* NA
- `> y <- x[-which(x >= 20)]` # get all elements of x *except* those greater than or equal to 20
- `> y <- x[which(!x >= 20)]` # same as above
- `> y <- subset(x, age >= 20)` # same as above
- `> y <- sample(x, 50, replace = TRUE)` # get a random sample of 50 values from x, allowing for replacement

Thank you!

Questions?