Biometry. Lecture 8

Alexey Shipunov

Minot State University

February 18, 2015

Outline

> setwd("<working folder>")

or

"Change dir"

in menu!

On Mac, be sure that startup option is working: getwd()
(getwd() checks if R is in working folder, dir() checks the folder content)

M

```
> i <- read.table("http://ashipunov.info/data/islets.txt",
+ sep="\t", h=T)
> square <- i$width * i$length * .6 * .6
> scatter.smooth(log10(square), i$species)
> identify(log10(square), i$species)
```

How to create ranked data

In R, ranked data is normally represented by the same numerical vector or *ordered factor*. Command cut() will break continuous data into ranks:

```
> height <- trees[,2]
> cut(height, 3, labels=c(1:3), ordered=T)
> cut(height, 3, ordered=T)
```

Types of data
Categorical data

Just observations

- Some data cannot be ordered at all
- Sex, color, absence/presence are good examples
- If even we label red color as "1" and green color as "2" the "1.5" is a nonsense.
- Therefore, if we use numbers for categorical data, they are only *labels*.

Binary data

- Absence/presence is a specific subset of categorical data which only two possible values
- One of the easiest representation is with numbers 0 and 1
- Computers normally prefer binary data over non-binary

Logical data

Practically, it is just a kind of binary data:

```
> height < 72
> height >= 72
> height == 72 # not "="!
> presence <- c(F, T, T, F, F)
> presence
> presence * 1 # convert to 1/0
> (presence * 1) == 1 # convert back
```

"==" is a logical test: "Is equal?". In R, "=" has a different meaning, it is
a replacement for "<–".

Categorical data in R

Character and logical vectors may be used for categorical data:

```
> sex <- c("male", "female", "male", "male",
+ "female", "male", "male")
> is.character(sex)
> is.vector(sex)
> str(sex)
> str(presence)
```

Squeezing numbers from the categorical data

```
> sex <- c("male", "female", "male", "male",
+ "female", "male", "male")
> presence <- c(F, T, T, F, F)
> table(sex)
> table(presence)
```

The `table()` command will let us to have some numbers even from categorical data!

Character to factor

```
> plot(sex) # error!
> sex.f <- as.factor(sex)
> plot(sex.f) # makes bar plot
```

Features of factors

```
> is.factor(sex.f)
> is.character(sex.f)
> str(sex.f)
> levels(sex.f)
> sex.f[6:7] # two levels!
> sex.f[6:7, drop=TRUE] # one level
```

Factor has levels which will not automatically drop with a sub-setting.

Factors to numbers

```
> as.numeric(sex.f)
> w <- c(69, 68, 93, 87, 59, 82, 72)
> x <- c(174, 162, 188, 192, 165, 168, 172)
> plot(x, w, pch=as.numeric(sex.f), col=as.numeric(sex.f))
> legend("topleft", pch=1:2, col=1:2, legend=levels(sex.f))
```

Objects `x`, `sex` and `w` could be height, gender and weight of seven people in small office, respectively.

Factors to ranks

```
> m <- c("L", "S", "XL", "XXL", "S", "M", "L") # t-shirts
> m.f <- factor(m)
> levels(m) # Wrong order, alphabetical
> m.o <- ordered(m.f, levels=c("S", "M", "L", "XL", "XXL"))
> levels(m.o)
```

The danger of factors

```
> a <- factor(3:5)
> a
> as.numeric(a) # wrong!!!
> as.numeric(as.character(a)) # correct
```

Some rules about vectors

- For every type of R object, there are functions
  is.<something>() and as.<something>() (e.g.,
  as.vector() and as.numeric() will convert to vector and to
  numeric vector, respectively).
- Object names must not start with a number
- R is case-sensitive
- Please avoid to use names of popular functions (like c()) and
  reserved keywords: T (TRUE), F (FALSE), NA (missing data), NaN
  (not a number), Inf (result of dividing by zero), also constants like
  pi, letters and LETTERS

If you want *e* constant, use exp(1)

Finishing...

Save your commands!

(savehistory(<todaysdate>.r) or File -> Save as... on Mac)

Summary: most important commands

- `as.<something>()` —converts objects
- `table()` —summarizes categorical data

M

For Further Reading

A. Shipunov.
Biometry [Electronic resource].
2012—onwards.
Mode of access:
`http://ashipunov.info/shipunov/school/biol_240`

A. Shipunov, and others.
Visual statistics. Use R!
Ongoing translation from Russian.