evropský sociální fond v ČR

EVROPSKÁ UNIE

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OP Vzdělávání pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Tento projekt je spolufinancován Evropským sociálním fondem a Státním rozpočtem ČR
InoBio – CZ.1.07/2.2.00/28.0018

Introduction, R basics

11. 11. 2013

Ing. Daniel Volařík, Ph.D.

european social fund in the czech republic

EUROPEAN UNION

MINISTRY OF EDUCATION, YOUTH AND SPORTS

OP Education for Competitiveness

INVESTMENTS IN EDUCATION DEVELOPMENT

Program of the course

Time	Lecture	Classroom	Lecturer
11. 11. 9:00–11:50	Introduction, R basics	A34	Volařík
11. 11. 13:30–16:20	Data exploration in R	Z13	Chamagne
12. 11. 9:00–11:50	Graphics in R	Z6	Volařík
12. 11. 13:30–16:20	ANOVA/linear models	Z6	Chamagne
13. 11. 9:00–11:50	Linear models/GLM	Z6	Volařík
13. 11. 13:30–16:20	Poisson regression (count data)	Z6	Chamagne
14. 11. 9:00–11:50	Logistic regression	Z6	Volařík
14. 11. 13:30–16:20	Model selection	Z6	Chamagne

Statistical analysis in ecology

- It is challenging
- Non-normal distributions; response variable often can not be negative or it is just presence-absence
- Hierarchical structure of data – e.g. tree, forest stand, mountain range; repeated measures on the same individuals, spatially autocorrelated data
- That‘s why it is sometimes difficult – we have to move further from basic statistical course

R statistical environment

- http://www.r-project.org/
- It is a free software environment for statistical computing and graphics.
- It can be regarded as an implementation of the S language.
- R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand
- Now it is developed by the R Development Core Team

Why to use R?

- It is powerful and it is free
- Wide range of statistical techniques implemented
- New methods are quickly implemented
- Huge amount of resources (books, web pages)
- Modern graphics – could be used for results publishing
- Possible problem is the lack of graphical user interface – to make the most of R, one should learn R programming (writing R scripts)

R basics

- command line interface
- RStudio – one of the integrated development environments (IDEs) for R – organized layout, some extra options
- Scripts – more comfortable, also documentation of the analysis
  - When you want to revise the analysis
  - When adapting the analysis for some new and similar
  - Collaborating with other researchers

R basics

- # for comment
- Calculations in R

```
> 12 * 2
> 12 - 8
> 2*(15+5)
```

R syntax

- elementary commands consist of either expressions or assignments
- they are separated by a semi-colon (‘;’), or by a newline
- Example of expression – command is evaluated and the result is printed

```
> c(1,2,3)
```

- Example of assignments – the result is stored in the variable

```
> x <- c(1,2,3)
```

R objects – vectors

- an ordered collection of numbers (numeric vector), character (character vector), or values TRUE, FALSE, NA (logical vectors)
- To create a vector in R, you can use c() function

```
> c(1,2,3)
```

- Characters have to be in quotes

```
> c("A", "B", "E", "K")
```

R objects – matrices

- a rectangular array of numbers, symbols, or expressions, arranged in rows and columns
- matrix() function

```
> matrix(data = c(1,3,45,6,7,8,9,10,12),
nrow = 3, ncol =3)
```

R objects – lists

- an object consisting of an ordered collection of objects known as its components
- Components could be of different types
- Various functions return an object which is list (lm, glm)
- list() function

```
> list(vector = c(1,3,“k“,6,7,8,9,10,12),
cislo = 3, cislo2 = 45)
```

- Components could be named

R objects – data frame

- Special type of list
- Mostly table with columns and rows

```
> data.frame(column1 = c(1,3,45), column2 =
c(6,7,8), column3 = c(9,10,12))
```

- How to import a table to R?

```
> read.table("file.csv", sep = ";", dec = ",",
header = TRUE)
> ?read.table
```

Subsetting data frames

- Basic operation that is very needed
- In models, in graphs you need to choose particular variable from data
- Also when you want to make graphs or models for some part of your data – e.g. you have data from conifer and deciduous forest in the table, but you want to work only with data from conifer forests
- There are several possibilities how to do this in R – don‘t be confused by that.

How to choose particular variable from a table

- There are at least 3 (4) possibilities:
  - Using symbol $ (table$variable)
  - Using brackets []

  (table[[“variable name“]] or table[, “variable name“]
  - Using subset() function

  subset(table, select = Variable)
- R help works also for the function behind symbols like $ or [. But you have to quote them (e.g. ?“$“)
- Functions attach() and detach() – using function attach() you can make available all variable in particular table under their names
  - Attach(table); variable; all operations with variable; detach(table)

How to subset specific part of samples from the table

- You can use:
  - brackets []

    table[table$variable == “type you want“,]

    table[table$variable == 1,]

  - Using subset() function

    subset(table, Variable == “type you want“)

    subset(table, Variable == 1)

How to subset specific part of samples from the table

- Binary operators:

X < Y

X > Y

X <= Y

X >= Y

X == Y (X is equal to Y)

X != Y (X is not equal to Y)

X %in% Y (if there is match of Y for X)

Example: c(1,2,3) %in% c(1,3,5)

How to subset specific part of samples from the table

- Logical operators:
  - Using logical opetarors you can combine criteria for subseting

X & Y (and – both criteria are true)

X | Y (or – criteria X is true or criteria Y is true or both)

- Example: subset(table, Variable_1 == 1 & Variable_2 > 3)

Some other basic functions

- mean() – for calculating mean
- range() – to find out range of values (minimum and maximum value)
- dim() – to find out dimension of object (e.g. number of elements in vector, number of rows and columns in matrix)

- rep() – when you want to repeat some value N times
- seq() – when you want to make regular sequence specifiing starting and (maximal) end values and increment

Writing your own functions

- “learning to write useful functions is one of the main ways to make your use of R comfortable and productive“ (from R-intro).