Workshop: Statistical Analysis in R

Richard Schuster
Carleton University and University of Northern British Columbia

Matthias-Claudio Loretto
Department of Cognitive Biology, University of Vienna

Schedule

22.2. 9:00 - 12:00: Introduction to R

Topic 1: The basics of R. What is it, how does it work?

Topic 2: Manipulating data.

Topic 3: Key Components of (Generalized) Linear Models

22.2. 13:00 - 16:00: Fitting Models and visualization

Topic 4: Normal linear regression

Topic 5: ANOVA and general linear models

Topic 6: Graphs and visualization

Schedule

23.2. 9:00 - 12:00: Mixed-effects and generalized linear models
Topic 7: Linear mixed-effects models (random effects)

Topic 8: Generalized linear models

Topic 9: Over-dispersion + zero-inflation

23.2. 13:00 - 15:30: GLMMs and advanced topics

Topic 10: Generalized linear mixed effects models

Topic 11: Null hypothesis significance testing (NHST) vs information theoretic (IT) approach and model selection

The basics of R

Background:

- R is a free software package – use with caution
- many R books and (free) online courses/tutorials available

Installing R:

- When you install R, only some package are loaded. You may need to add in packages later on as you need them (more on this later).

The basics of R

Running R:

- To run R, click on the icon. You get a work session window.
- You could type in your commands here and they would run as you enter them.

Exercise: Type in

```
A<- 5
B<- 6
C<- A+B
```

- The output appears in the work session window also.
- You could copy and paste this into WORD as this is just simple text.

The basics of R

Running R:

- Instead of typing the commands into the session window, you can enter your commands into a separate file called *script*.
- The script is just R commands, organized and put into a text file.
- You can run this all at once (only if you are very confident), or in segments (preferred).

  Exercise: Open a new script window, and type in commands:

  plot(cars)

  lines(lowess(cars))

  To run the R commands in the script, simply highlight the parts you want to run and press Ctrl and R at the same time OR, click on Edit and find Run in the list.
- Since a graph is produced, a new window appears.
- You could copy and paste this into WORD.

The basics of R

Useful Things to Know:

- R is case sensitive. i.e., `Trees` versus `trees`.
- R does not like spaces nor special characters. Instead, use a ‘.’ e.g.: `trees.pine` identifies a variable in the dataset `trees`.
- R uses two slashes instead of one to indicate a subfolder. For example, if your data are in: `E:\measurements\trees.txt` then in R you would use `E:\\measurements\\trees.txt` since the single slash has a different meaning in R
- Alternative: `E:/measurements/trees.txt`
- Any R commands that start with # are just *comments* that you can add to explain what the script does.

The basics of R

Working directory

- To avoid entering always the whole path for saving or reading files you can set a folder as current working directory with setwd()
- setwd("D:/blabla/blub")
- If you are not sure which it is ask: getwd()

The basics of R

Saving and cleaning up:

- You can save the work session, the script, and the graph window anytime you wish by using File and Save for the window that is active.
- To save the objects you have created, you can use File and Save Workspace to save all objects (i.e., all the data and outputs you have put into objects).
- You can then later on use File and Load Workspace to bring the objects back in and continue your work.
- You can also cut and paste any of the outputs from your session window into WORD or other files.

The basics of R

Saving and cleaning up:

- If you have created a graph, you can use File and Copy to Clipboard and then As a Metafile (gives a better graph then as a Bitmap).
- You can also save the file as a picture using File and Save As..
- At any time, you can use Edit and Clear Console to clean out the session window. However, the data you brought in, and any variables and objects will still be there.
- To remove all of these, click on Misc and then select Remove all objects, OR type rm(list=ls(all=TRUE)) in the session window.
- When you begin an R session with new data, it is always a good idea to start with no objects.

The basics of R

Types of Objects (more on this later in the course):

- **Factor**: a class variable, often represented as letters but maybe represented as numbers
- **Vector**: a "column" of numbers
- **Matrix**: several columns of numbers
- **Dataframe**: Like a matrix, but can have columns of numbers and columns of letters; names for columns and rows are possible
- **List**: Can be several objects all stored together such as regression outputs, matrices, etc.

The basics of R - help

The R website has a number of manuals that you might find useful, including an introduction: http://cran.r-project.org/doc/manuals/R-intro.pdf and http://cran.r-project.org/doc/contrib/usingR.pdf

At any time, you can also use help( ) where the function is given in the brackets or just **?** + function name

- Hard to follow
- Tells you about the specific options for a function and the syntax
- There are usually a few examples

The basics of R

Expanding the R package:

- When you run R, only some of the functions are brought into the work session automatically to save memory.
- To add others, use require( ) or install.packages("package_name ") where the package is given in brackets.
- Many other parts of R are extra to the main package. To bring these in:
  - Access the website: https://cran.r-project.org/ host currently >12,000 packages
  - Some packages can be found on https://r-forge.r-project.org/
  - Download the package to the R directory to the library sub-folder

  For example, if you installed R in: C:\Program Files\R\R-2.8.0\, then you can add more software into C:\Program Files\R\R-2.8.0\library\
- You can then use library( ) to bring in these other packages for your analysis.

The basics of R

Learning R:

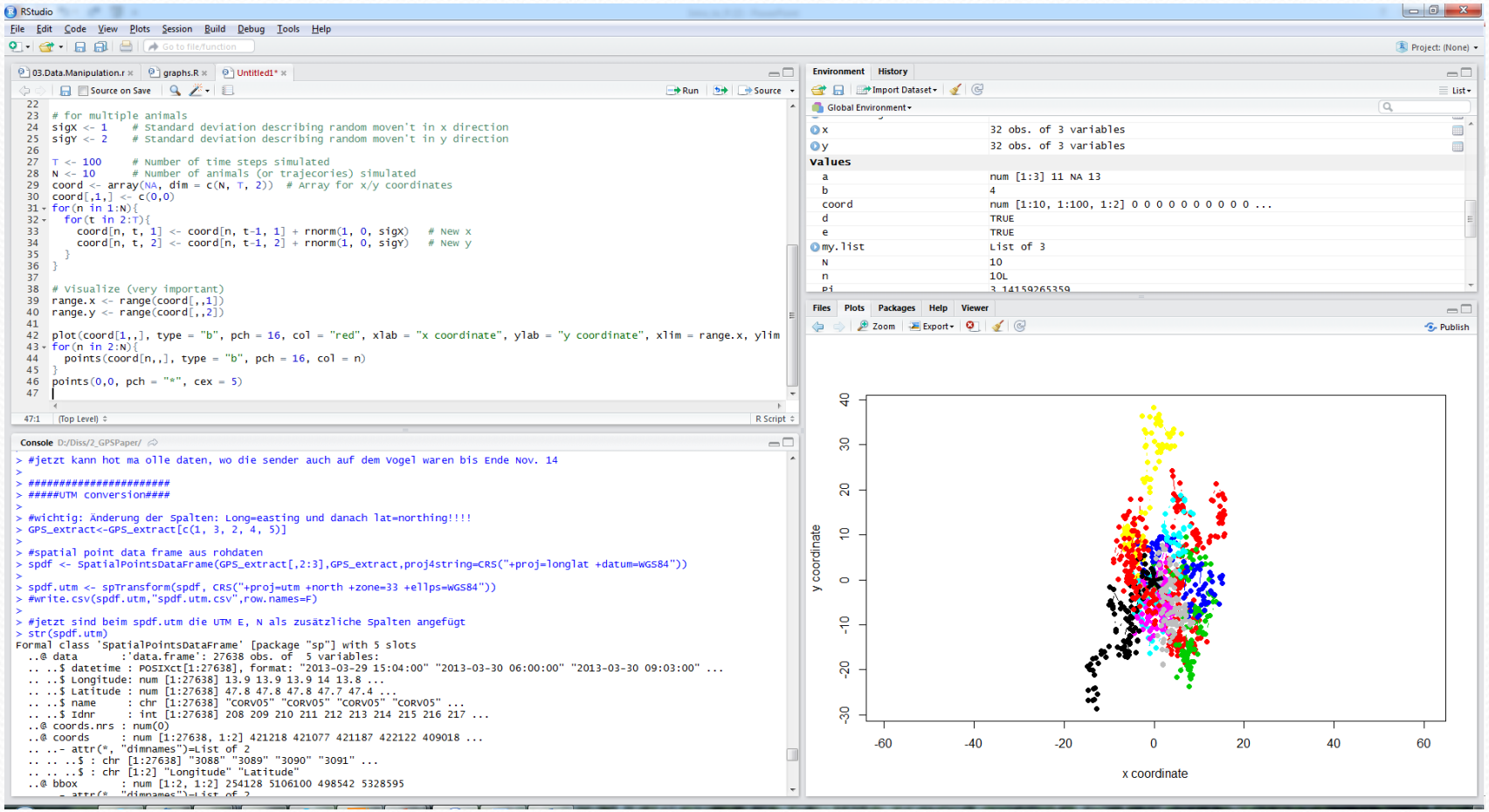
- Documentation and examples using R code or script on the web.
- Examples are very helpful for reducing the time you spend in getting R to do what you would like.
- Practice.
- **Google!!!**
- http://stackoverflow.com/
- R-bloggers, R for Dummies,….

R Studio

Rstudio (IDE, integrated developement environment) divides its window into 4 panels, each of which may have multiple structure tabs. Which tabs are in which panels is configurable.
Some of the important tabs include:

Console: This is where you can execute R commands interactively
History: A record of past commands (can be saved, reloaded, etc.)
Workspace: A listing of the objects available in your R session
Plots: Where plots show up
Help: Where documentation files appear when you ask for them
Files: A file manager for locating, loading, moving, renaming, files.
Packages: Install and load packages here.
Open Files: Open files have a tab labeled with the file name.

R Studio


RStudio
File Edit Code View Plots Session Build Debug Tools Help
03.Data.Manipulation.r
graphs.R
Untitled1*
Source on Save
Run
Source
# for multiple animals
sigX <- 1 # Standard deviation describing random moven't in x direction
sigY <- 2 # Standard deviation describing random moven't in y direction
T <- 100 # Number of time steps simulated
N <- 10 # Number of animals (or trajecories) simulated
coord <- array(NA, dim = c(N, T, 2)) # Array for x/y coordinates
coord[,1,] <- c(0,0)
for(n in 1:N){
for(t in 2:T){
coord[n, t, 1] <- coord[n, t-1, 1] + rnorm(1, 0, sigX) # New x
coord[n, t, 2] <- coord[n, t-1, 2] + rnorm(1, 0, sigY) # New y
}
}
# Visualize (very important)
range.x <- range(coord[,,1])
range.y <- range(coord[,,2])
plot(coord[1,,], type = "b", pch = 16, col = "red", xlab = "x coordinate", ylab = "y coordinate", xlim = range.x, ylim
for(n in 2:N){
points(coord[n,,], type = "b", pch = 16, col = n)
}
points(0,0, pch = "*", cex = 5)
47:1 (Top Level) R Script
Console D:/Diss/2_GPSPaper/
> #jetzt kann hot ma olle daten, wo die sender auch auf dem Vogel waren bis Ende Nov. 14
> ######################
> #####UTM conversion####
> #wichtig: Änderung der Spalten: Long=easting und danach lat=northing!!!!
> GPS_extract<-GPS_extract[c(1, 3, 2, 4, 5)]
> #spatial point data frame aus rohdaten
> spdf <- SpatialPointsDataFrame(GPS_extract[,2:3],GPS_extract,proj4string=CRS("+proj=longlat +datum=WGS84"))
> spdf.utm <- spTransform(spdf, CRS("+proj=utm +north +zone=33 +ellps=WGS84"))
> #write.csv(spdf.utm,"spdf.utm.csv",row.names=F)
> #jetzt sind beim spdf.utm die UTM E, N als zusätzliche Spalten angefügt
> str(spdf.utm)
Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
..@ data : 'data.frame': 27638 obs. of 5 variables:
.. ..$ datetime : POSIXct[1:27638], format: "2013-03-29 15:04:00" "2013-03-30 06:00:00" "2013-03-30 09:03:00" ...
.. ..$ Longitude: num [1:27638] 13.9 13.9 13.9 14 13.8 ...
.. ..$ Latitude : num [1:27638] 47.8 47.8 47.8 47.7 47.4 ...
.. ..$ name : chr [1:27638] "CORV05" "CORV05" "CORV05" "CORV05" ...
.. ..$ Idnr : int [1:27638] 208 209 210 211 212 213 214 215 216 217 ...
..@ coords.nrs : num(0)
..@ coords : num [1:27638, 1:2] 421218 421077 421187 422122 409018 ...
.. ..- attr(*, "dimnames")=List of 2
.. .. ..$: chr [1:27638] "3088" "3089" "3090" "3091" ...
.. .. ..$: chr [1:2] "Longitude" "Latitude"
..@ bbox : num [1:2, 1:2] 254128 5106100 498542 5328595
Environment History
Import Dataset
Global Environment
x 32 obs. of 3 variables
y 32 obs. of 3 variables
values
a num [1:3] 11 NA 13
b 4
coord num [1:10, 1:100, 1:2] 0 0 0 0 0 0 0 0 0 0 ...
d TRUE
e TRUE
my.list List of 3
N 10
n 10L
Files Plots Packages Help Viewer
Zoom Export Publish
y coordinate
x coordinate
40
30
20
10
0
-10
-20
-30
-60
-40
-20
0
20
40
60

Importing data into R (R-Studio)

Data can be imported as csv file (comma separated value), which can be created from e.g. Excel

read.csv("test.csv")

Caution: Depending on your system settings the „list separator“ can be set to „;“ instead of „,“

read.csv("test.csv", sep=";") #specify list separator

For windows this can also be changed in Start/ControlPanel/Region and Language/Additional settings/List separator

Or open data in Rstudio via ImportDataset From Txt file and specify

In (German) Excel often comma (,) is used instead of point(.) for decimals - this can be changed in File/Options/Advanced/Decimal and Thousands separators

-> **Exercise: import data**