Analysing toxicity data using R

Georgia Tsiliki

NTUA

http://www.chemeng.ntua.gr/labs/control_lab

gtsiliki@central.ntua.gr

eNM workshop, 30 September 2016, NTUA

Overview

- Introduction to R
- Basic programming
- Basic Plotting
- Modelling and analysis tools for toxicity data
  - Multiple linear regression
  - Bayesian linear regression
- Statistics
- Application to protein corona data (Liu et al. (2015))

Scope

- The aim of this session is to provide you with a basic fluency in the language and give a few examples on how nanotoxicity data (or similar!) can be analysed. It is suggested that you work through this document at the computer, having started an R session. Type in all of the commands that are printed, and check that you understand how they operate.

- The R project

  https://www.r-project.org/

- The R prompt

  ```
  >
  ```

Setting up your machine

- Binaries for Windows, Mac, Linux
- Rstudio- by far the most popular IDE for R
  - Lots of webinars

RStudio®

- Download packages via:

CRAN

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

GitHub

```
> install.packages(<name of package>, dependencies=TRUE)
> library(<name of package>)
>
```

R: Introduction

- R is a software language for carrying out complicated (or less complidated!) statistical analyses. It includes routines for data summary and exploration, graphical presentation and data modelling.
- When you work in R you **create objects** that are stored in the **current workspace** (sometimes called image). Each object created remains in the image unless you explicitly delete it. At the end of the session the workspace will be lost unless you save it. You can save the workspace at any time by clicking on the disc icon at the top of the control panel.
- Commands written in R are **saved in memory** throughout the session. You can scroll back to previous commands typed by using the `up' arrow key (and `down' to scroll back again). You can also `copy' and `paste' using standard windows editor techniques. If at any point you want to save the transcript of your session, click on `File' and then `Save History’.
- You finish an R session by typing

```
> q()
```

at which point you will also be prompted as to whether or not you want to save the current workspace. If you do not, it will be lost.

R: basic commands

- R stores information and operates on objects. The simplest objects are scalars, vectors and matrices

```
> x<-6
> y<-4
> z<-x+y
> z
[1] 10

> sqrt(16)

> ls()
[1] "x" "y" "z"
> rm(x,y,z)
> rm(list=ls())
```

R: vectors/ arrays

- Vectors can be created in many ways

```
> x<- c(5,7,17)
> y<- 1:10
> z<- seq(1,9,by=2)
[1] 1 3 5 7 9
> z<- seq(8,20,length=6)
[1] 8.0 10.4 12.8 15.2 17.6 20.0

> rep(1:3,6)
[1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
> rep(1:3,c(6,6,6))
[1] 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3
```

ENM
eNanoMapper

R: vectors/ arrays

- Exercises:

```
> x<- c(5,7,17)
> y<- c(1,2,5)
```

What would be the results for:

a. length(x)

b. sum(x^2)

c. x+ (y*2)

Use rep() to define

a. 6,8,6,8,6,8,6,8

b. 6,6,6,6,8,8,8,8

R: vectors/ arrays

- Vectors, matrices, arrays or data frames

```
> x<- c(5,7,17)
> x[1]
[1] 5
> mean(x)
[1] 9.666667

> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.000   6.000   7.000   9.667  12.000  17.000

> z<- matrix(NA,10,3)
> y<- matrix(c(5,9,2,3,4,6,7,0,8,12,2,9),3,4)
> as.data.frame(y)
```

R: vectors/ arrays

- Vectors, matrices, arrays or data frames

```
> y[1,2]
[1] 3
> y[c(2,3),1]
[1] 9 2

> t(y)  # transpose

> solve(y[,1:3])  # inverse

> apply(y,2,mean)
[1] 5.333333 4.333333 5.000000 7.666667

> y1<- y[c(2,3),1]
```

R: known distributions

- Simulate data: generate a sample of 10 from N(0,2^2)

```
>#  set.seed(1253)
> rnorm(10,0,2)
 [1]  0.8685267  3.1123541 -1.1864866  1.3069373 -0.2910654 -1.7497150 -1.3899950
 [8] 0.3707912 2.6354128 -3.1562509


> dnorm(y1,0,2)  #  density function for the Gaussian distribution
[1] 7.991871e-06 1.209854e-01


> ?rpois


>?rexp
```

Importing data into R

Different types of data

Importing data into R requires different approaches depending on file formats

- Flat files

  > read.table()

  > read.csv()
- Excel files

  xlsx, readxl, XLConnect
- XML
- JSON
- Retrieve data from ftp server – API's
- Data from MySQL databases

Data manipulation

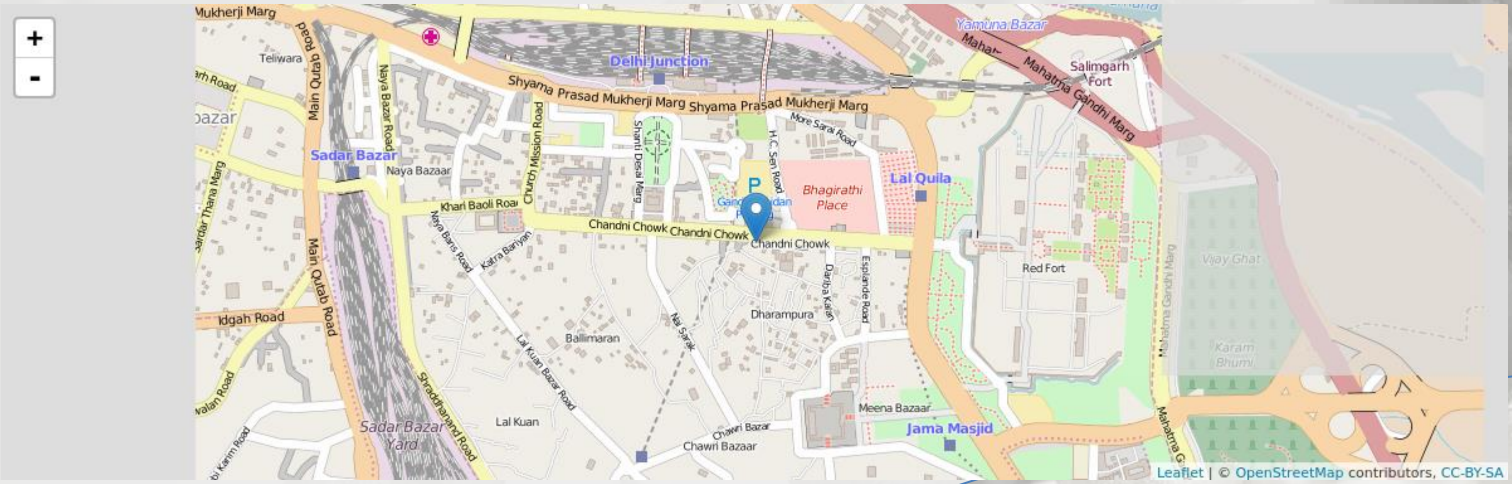
R can help in cleaning, preparing and structuring the data, for data analysis. A cumbersome and time-consuming task!

- Normalisation/ scaling
- Missing value imputation
- Strings manipulation (e.g. tweets)
- Fast aggregation for large datasets
- Handle time-series data
- Example: a standard way to organize values within datasets in a uniform way by **tidy** package

Data vizualisation

A whole lot of data visualization packages: great possibilities with just a few lines of code

- heat map, city map, mosaic map, bar chart, scatter plot, histogram, 3D graph, box plot
- gglot2, hexbin, tableplot, ggmap, leaflet

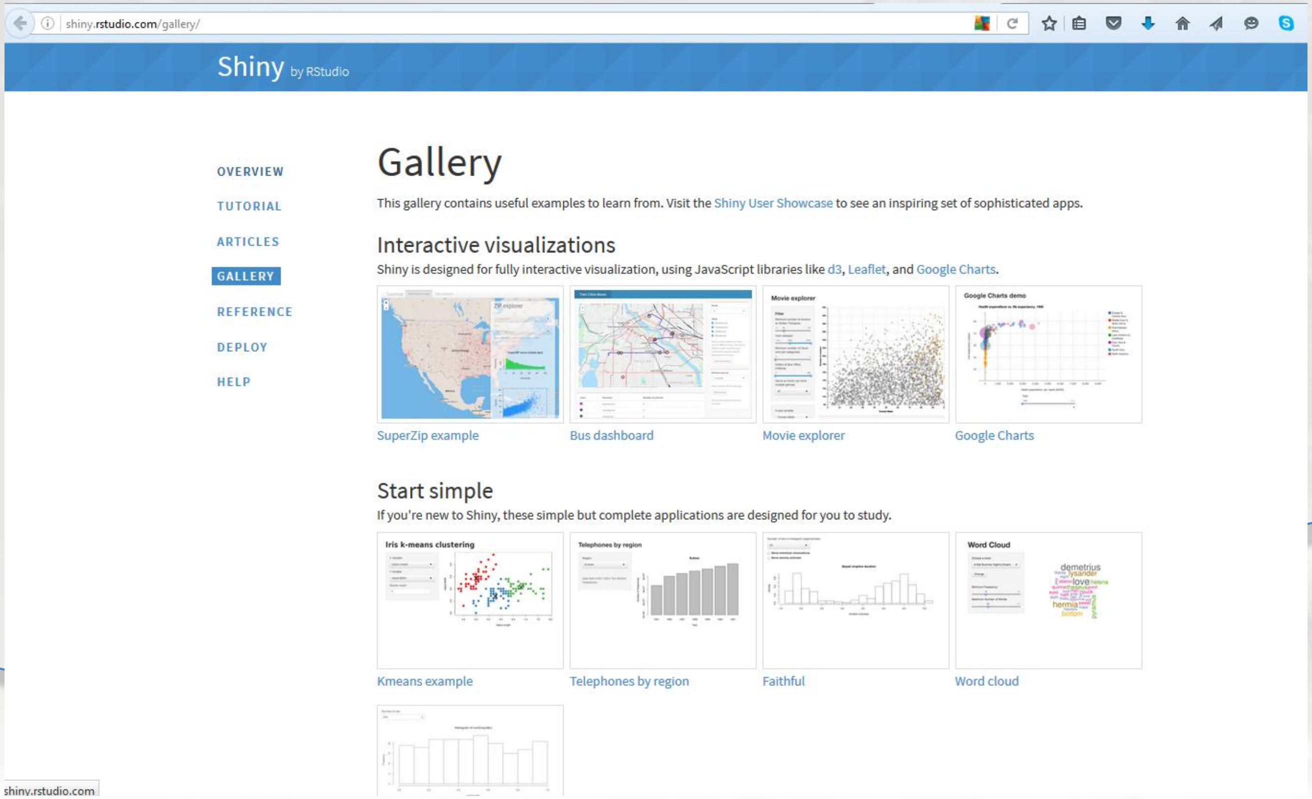


Shiny

Shiny: interactive web applications

Shiny= R + interactive + web

Build interactive web applications without needing to known HTML, CSS or Javascript!



shiny.rstudio.com/gallery/

Shiny by RStudio

Gallery

This gallery contains useful examples to learn from. Visit the Shiny User Showcase to see an inspiring set of sophisticated apps.

Interactive visualizations

Shiny is designed for fully interactive visualization, using JavaScript libraries like d3, Leaflet, and Google Charts.

Start simple

If you're new to Shiny, these simple but complete applications are designed for you to study.

toxFlow

toxFlow is a recently developed shiny app by Varsou DD hosted on the Unit of Process Control & Informatics laboratory server

- Please visit http://147.102.86.129:3838/

Conclusions

Take home message

- **Free online interactive tutorials:**
  - **Learn R in R: take a Swirl course!**

    http://swirlstats.com/

    ```
    > library(swirl)
    > swirl()
    ```
  - **O'Reilly school**

    http://tryr.codeschool.com/
- **News**
  - R-Bloggers.com
  - The R journal (https://journal.r-project.org/)
- **Manuals**
  - Official CRAN website
- **Books**
  - R for everyone by Lander (http://www.jaredlander.com/r-for-everyone/)
  - R in action by Kabacoff (https://www.manning.com/books/r-in-action)
- **Inner workings of R**
  - In the mood for some R package internal work? (http://r-pkgs.had.co.nz/)

ENM
eNanoMapper

- In what follows, you will need
  - the R source code file **‘eNMworkshop_NanoAnalysis usingR- Athens2016.R’**
  - the dataset file **‘liu3_10PCF.csv’** derived from

    Liu et al., Prediction of Nanoparticles-Cell Association based on Corona Proteins and Physicochemical Properties, Nanoscale 2015

Sources used for this tutorial:
https://www.r-bloggers.com/how-to-learn-r-2/
http://www.r-tutor.com/
http://www.rdatamining.com/training
https://rpubs.com/davoodastaraky/mtRegression
https://www.r-bloggers.com/bayesian-linear-regression-analysis-without-tears-r/