Data analysis with *tricot*

Kauê de Sousa

2020-09-17

Content

Section 1, Thursday 17

- Set up
- Short introduction to Git
- Short introduction to R
- Quick recap on the tricot approach
- Introduction to rank-base models (Bradley-Terry and Plackett-Luce)

Section 2, Friday 18

- Plackett-Luce rankings
- Visualization of tricot results
- Linking ranks with covariates

Section 3, Tuesday 22

- Model selection
- Common issues (and how to avoid it) in analyzing incomplete rankings
- Short introduction to report production using rmarkdown

Section 4, Wednesday 23

- Case groups (each country works with their own data)

Aim

Learn the principles to analyse the tricot data and how to interpret the results

Set up

- Create a free GitHub account
- Install Git in your machine. Here is a tutorial depending on your OS.
- Install or update R, preferably v4.0.2
- Install R Studio Desktop
- Install GitHub Desktop

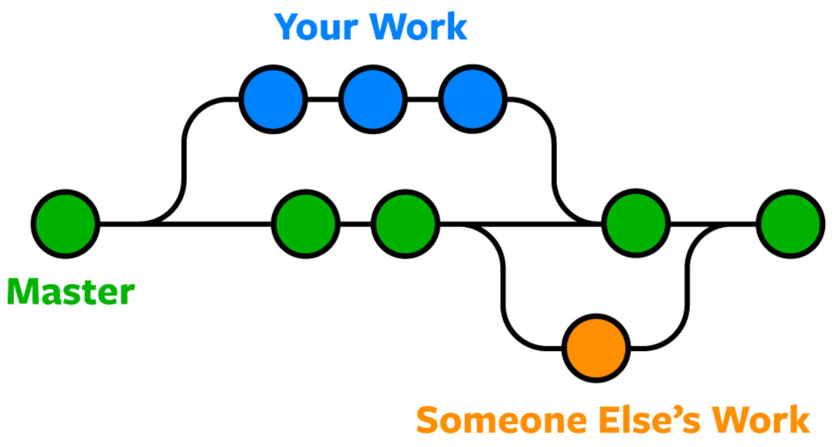
A short introduction to Git

What is Git

- Git is a version control system: can record snapshots and track the content of a folder as it changes over time.
- Every time we commit a snapshot, Git records a snapshot of the entire project, saves it, and assigns it a version.
- These snapshots are kept inside a sub-folder called `.git`.
- If we remove `.git`, we remove the repository and history (but keep the working directory).

Why Git (motivation)

- Version control
- Code can became a disaster without version control
- Roll-back functionality (if something wrong happens, we can go back to the latest good version)
- Branching
- Reproducibility



Common Git commands

copy your Git repo locally

```
git clone
```

check the status of your local repo compared to the Git repo

```
git status
```

add the files from local to Git

```
git add .
```

tells Git what are you doing with the previous command git add

```
git commit -m "something"
```

tells Git to which branch you want to send the update

```
git push origin master
```

We are going to work more on that when we start with R

Short introduction to R

Why R

Free and open source.

Software for data science:

- experiment/survey design
- data retrieval
- data wrangling
- data analysis
- reporting

A programming language, so we can

- use existing functions to code up our data science tasks
- write new functions for customised/novel tasks

Companies that use R

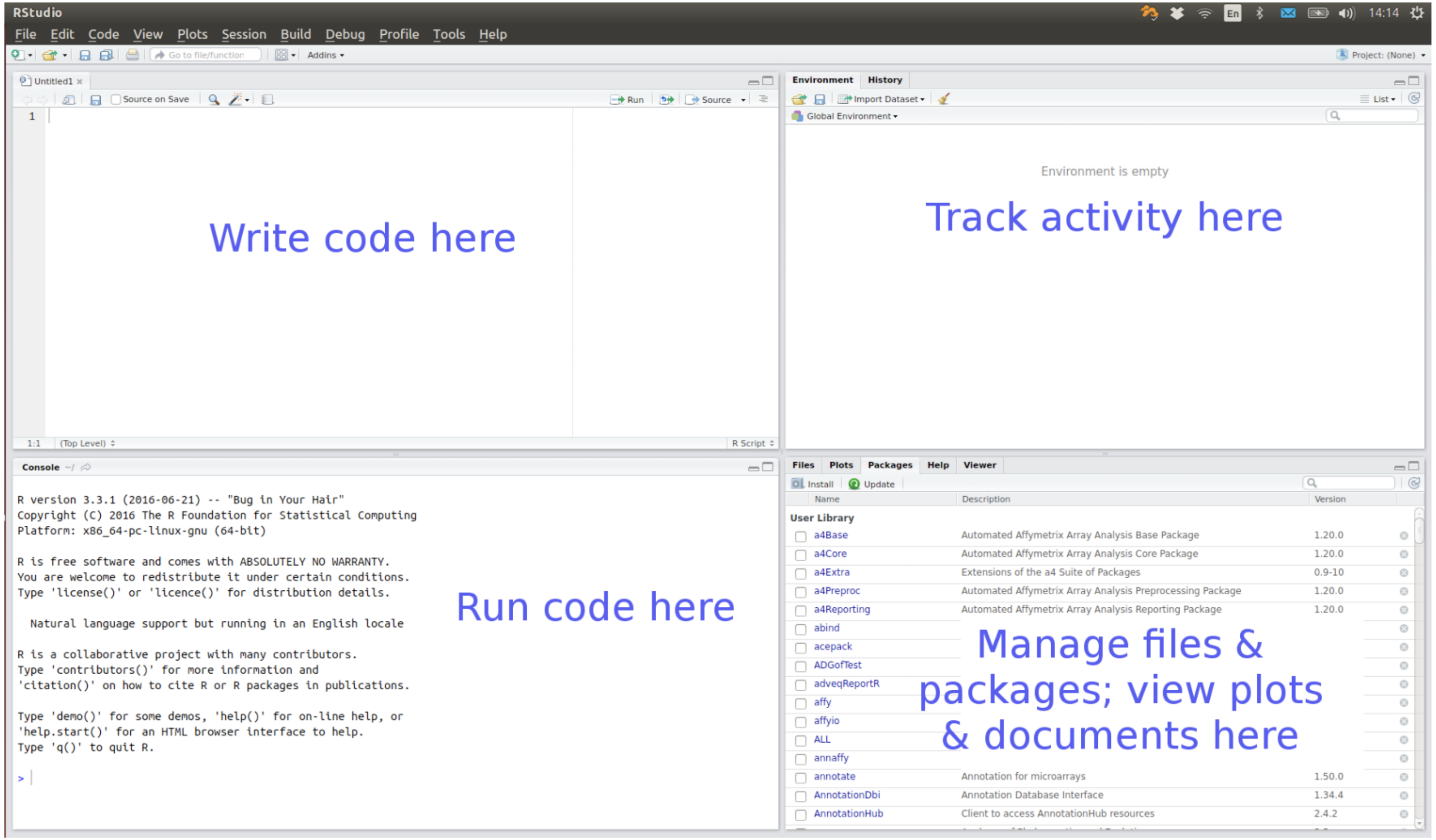
- AirBnB
- Amazon
- BBC
- The Economist
- Facebook

R code-along

We can type commands directly into the R console

```r
3 + 4
?"+" #look up help for "+"
x <- 3 + 4
y <- log(x)
ls() # list of objects in the current workspace
data() # find out what standard data sets there are
plot(iris) # plot Fisher's iris data
```

RStudio



Write code here

Track activity here

```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Run code here

Name	Description	Version
User Library		
□ a4Base	Automated Affymetrix Array Analysis Base Package	1.20.0
□ a4Core	Automated Affymetrix Array Analysis Core Package	1.20.0
□ a4Extra	Extensions of the a4 Suite of Packages	0.9-10
□ a4Preproc	Automated Affymetrix Array Analysis Preprocessing Package	1.20.0
□ a4Reporting	Automated Affymetrix Array Analysis Reporting Package	1.20.0
□ abind		
□ acepack		
□ ADGofTest		
□ adveqReportR		
□ affy		
□ affyio		
□ ALL		
□ annaffy		
□ annotate	Annotation for microarrays	1.50.0
□ AnnotationDbi	Annotation Database Interface	1.34.4
□ AnnotationHub	Client to access AnnotationHub resources	2.4.2

Manage files & packages; view plots & documents here

R packages

A collection of R functions, complied code and sample data. They are stored under a directory called "library" in the R environment

Most day-to-day work will require at least one contributed package.

The Comprehensive R Archive Network (CRAN) is where most of the packages are

To install a package from CRAN we use the command

```
install.packages("ggplot2")
```

Install the following packages

- climatrends
- tidyverse
- PlackettLuce
- patchwork
- ggparty

Install packages

```r
install.packages(c("climatrends", "tidyverse", "PlackettLuce", "patchwork", "ggparty"))
```

Using a R package

```r
library("climatrends")
library("tidyverse")
library("PlackettLuce")
library("patchwork")
library("ggparty")
```

Data structures

R is a vector based language

Data structures are the building blocks of code. In R there are four main types of structure:

- vectors and factors
- matrices and arrays
- lists
- data frames

Vectors

A single number is a special case of a numeric vector. Vectors of length greater than one can be created using the concatenate function, c.

```
x <- c(1, 3, 6)
```

The elements of the vector must be of the same type: common types are numeric, character and logical

```
x <- 1:3
x
# [1] 1 2 3
y <- c("red", "yellow", "green")
y
# [1] "red" "yellow" "green"

z <- c(TRUE, FALSE)
```

Missing values (of any type) are represented by the symbol NA.

Data frames

Data Frames Data sets are stored in R as data frames. These are structured as a list of objects, typically vectors, of the same length

```
str(iris)

> 'data.frame': 150 obs. of 5 variables:
> $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
> $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
> $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
> $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
> $ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Here Species is a factor, a special data structure for categorical variables.

Creating a Data Frame and Extracting Values

```r
x <- 1:3
y <- c("red", "yellow", "green")
dt <- data.frame(x, y)

dt

  x      y
1 1    red
2 2 yellow
3 3  green

dt$x
dt[[1]] # or dt[["x"]]
dt[1, 2:3] # or dt[1, c("x", "y")]
```

RStudio projects

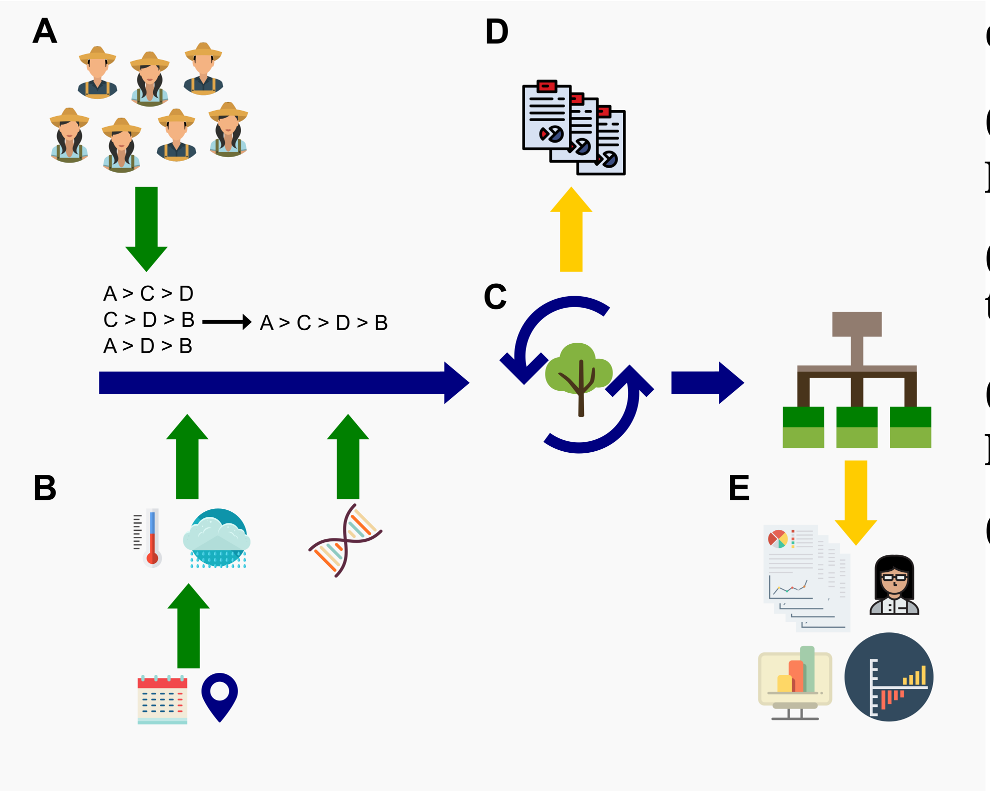
An Rstudio project is a context for work on a specific project

- automatically sets working directory to project folder
- as separate workspace and command history

Project-oriented workflow

https://www.tidyverse.org/blog/2017/12/workflow-vs-script/

Our workflow



(A) Several participants contribute with small tasks. All data is combined using rankings.

(B) Explanatory variables are added (e.g. using lonlat and planting dates, or even DNA markers)

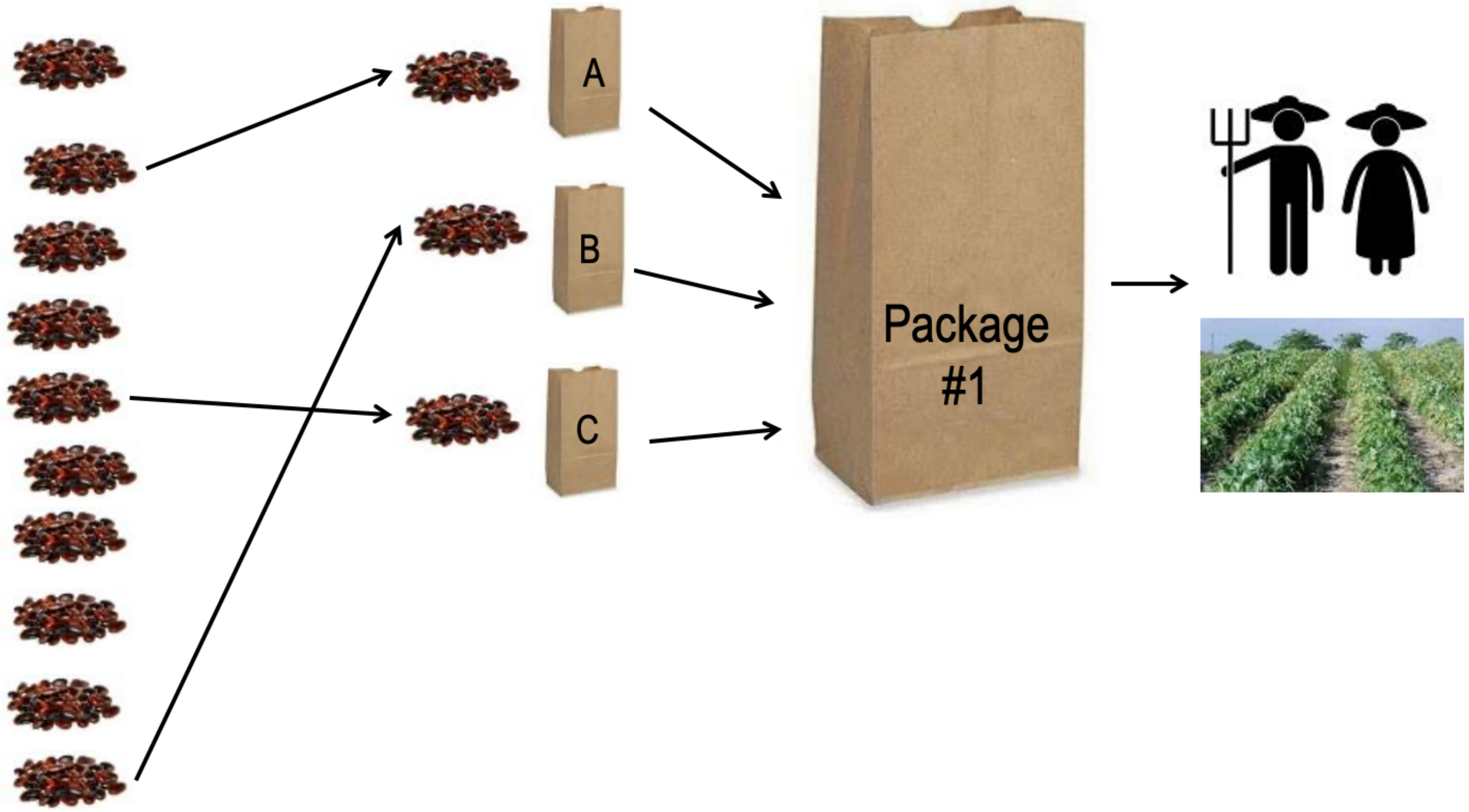
(C) Model selection to find the variables that best explain the data forward with cross-validation

(D) Automated reports can be generated and feedback to participants in **(A)** is given

(E) A stable *tree* is used for further analysis

It starts with tricot

Triadic comparison of technologies



Rank-based models

Rankings data arise in a range of applications, such as sport tournaments and consumer studies. In rankings data, each observation is an ordering of a set of items.

Classic models are Bradley-Terry and Plackett-Luce

The first works with pairwise comparisons and the last with rankings with > 3 items

It measures the odds that one option is chosen over a set of options