Challenge: Help the R-Project

The way R users are looking for new packages could and should be dramatically improved. Right now they use command line tools like apt / rpm and the package installer of R-Studio to do that job, but there is not a good web interface for package searching. You will help them. The goal of this challenge is to build an interface to help the R community. Why?

What is R?

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies). It's a cool language, have a look to it if you don't know it yet.

What is CRAN?

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. The use these CRAN Servers to store the R packages.

A CRAN server looks like: http://cran.r-project.org/src/contrib/. It is just a simple Apache Dir with a bunch of tar.gz files.

PACKAGES file

Every CRAN server contains a plain file listing all the packages in that server. You can access to it using this URL:

http://cran.r-project.org/src/contrib/PACKAGES

Format of PACKAGES file

```
# Package example

  [...]

  Package: adehabitatHR
  Version: 0.4.2
  Depends: R (>= 2.10.0), sp, methods, deldir, ade4, adehabitatMA,
  adehabitatLT
  Suggests: maptools, tkrplot, MASS, rgeos, gpclib
  License: GPL (>= 2)

  [...]

# Package example
```

Package URL format

You can build the URL of every R package as:

http://cran.rproject.org/src/contrib/[PACKAGE_NAME]_[PACKAGE_VERSION].tar.gz

Example Package URL: http://cran.r-project.org/src/contrib/shape_1.4.1.tar.gz
Inside every package, after you uncompress it, there is a file called DESCRIPTION where you can get some extra information about the package:

DESCRIPTION

```
# Description example

  Package: abc
  Version: 1.6
  Date: 2012-16-02
  Title: Tools for Approximate Bayesian Computation (ABC)
  Author: Katalin Csillery, Michael Blum and Olivier Francois
  Maintainer: Michael Blum <michael.blum@imag.fr>
  Depends: R (>= 2.10), MASS, nnet, quantreg, locfit
  Description: The package implements several ABC algorithms for
               performing parameter estimation and model selection.
               Cross-validation tools are also available for measuring the
               accuracy of ABC estimates, and to calculate the
               misclassification probabilities of different models.
  Repository: CRAN
  License: GPL (>= 3)
  Packaged: 2012-08-14 15:10:43 UTC; mblum
  Date/Publication: 2012-08-14 16:27:09

# Description example
```

What do we want to do?

We want that you create a Ruby application to index all the packages in a CRAN server. For that we want that you do:

1. Extract some information regarding every package and store it (You will need to get some info from PACKAGES file and some other info from DESCRIPTION)
2. Design the business logic needed for storing all the information (models, libs, DB structure...)
3. A task that will run every day at 12pm to index any new package that appeared during the day (we want to store all the versions of a given package. It means that the package abc_1.2.1.tar.gz could be tomorrow abc_1.3.0.tar.gz, and we want to store version 1.2.1 and 1.3.0)
4. A simple view listing all the packages you have indexed
5. Tests, of course
6. Push the code to github and send us the URL.

Which information do we want to store about a package?

- Package name
- Version
- Date/Publication
- Title
- Description
- Authors
- Maintainers

Info required about authors/maintainers

- Name
- Email

Tips

- Ruby application (Ruby, Sinatra, Rails, Goliath... up to you)
- Use the DB you feel more comfortable with.
- You don't need to index all the packages, but at least 50 of them
- To read the information, and convert it into hashes, from the files PACKAGES and DESCRIPTION you can use this gem: https://github.com/bmaland/treetop-dcf
- Don't invest more than 4-5 hours.
- Overengineering could consume your time. We want to see good code, but you don't need to show off for the sake of showing off. Write honest code and be pragmatic.