datatools

Package pour R / février 2022

# Codebook : todo list

Empêcher les sauts de page au milieu des tableaux

Réparer « Champ ».

Accélérer la génération du codebook

Traduire en anglais

Rajouter un paramètre « Présentation » dans la fonction.

S’inspirer du package codebook pour améliorer la présentation et la documentation de la fonction

# Les prochains chantiers

* Une première version de anonymize() pour la France à partir des résultats au bac = BAPTISTE
* Une première version de anonymize() pour un autre pays que la France.
* Une première version de educ() ou status() pour la France, à partir des résultats au bac et/ou de l’enquête Emploi.
* Internationaliser la fonction gender() avec le plus de bases de données possibles = CLEMENT ?
* Une première version de country() à partir du scraping de behindthenames.com et du fichier des prénoms = PIERRE
* Une première version de year() pour la France et pour un autre pays que la France, par exemple les USA à partir du fichier de la sécurité sociale.
* N’importe quelle autre fonction parmi la liste ci-dessous.

# Questions

-Ecriture du code avec dplyr ?

\*Précalculer les proportions d’hommes et de femmes dans un df précalculé :

Prénom /Sexe/Proba.

-Problèmes d’encodage dans les fichiers, la documentation et les aides (voir ?unaccent)

-Outils collaboratifs

# Présentation

L’objectif de ce package est de proposer un certain nombre de fonctions permettant d’extraire des informations utiles pour les sciences sociales à partir de données sur les prénoms et les caractéristiques socio-démographiques de leurs porteurs.

Attention : le développement de ce package implique probablement que des données extraites de bases de prénoms soient incluses dans les sources du package. Il faudra donc se demander sous quelles licences il sera possible de distribuer le package.

# Créer le package

Les notes ci-dessous sont établies d’après le livre de Hadley Wichham :

<http://r-pkgs.had.co.nz/>

## Développer le package

On utilise git et github. Pour une première utilisation :

Installer Git :

* Windows: <http://git-scm.com/download/win>.
* OS X: <http://git-scm.com/download/mac>.
* Debian/Ubuntu: sudo apt-get install git-core.
* Other Linux distros: <http://git-scm.com/download/linux>.

Indiquer à Git un nom et une adresse email. Dans le shell, entrer les commandes suivantes :

* git config --global user.name "YOUR FULL NAME"
* git config --global user.email "YOUR EMAIL ADDRESS"

On peut vérifier que l’installation est correcte avec :

* git config --global -- list

3. Créer un compte dans GitHub, en utilisant la même adresse que précédemment.

## Inviter des collaborateurs

S’ils ont déjà des comptes GitHub : aller dans le projet > Collaborators, et inviter.

S’ils n’ont pas de compte Github : lire le livre de Wickham pour installer un environnement R et RStudio + Git + Github, puis faire un clone :

File > New Project > Version Control > Git, puis entrer l’URL du projet et le nom du projet.

## Effectuer une modification du package

Pour effectuer des modifications dans le package en actualisant le dépôt GitHub, il faut procéder de la façon suivante :

Ouvrir le projet RStudio.

Charger les modifications effectuées par d’autres : aller dans l’onglet Git et cliquer sur « Pull ».

Effectuer les modifications souhaitées.

Tester le package : taper CTRL+MAJ+B pour construire le package (build). Si la construction ne génère pas d’erreur, le package modifié en local (dans votre ordinateur) est chargé et peut être utilisé.

Téléverser les modifications : une fois les modifications testées et validées, la version modifiée en local du package doit être téléversée. Aller dans l’onglet « Git » de RStudio, cliquer sur « Commit » pour documenter par une courte description les modifications effectuées, puis cliquer sur « Push » pour téléverser le commit.

## Ajouter une nouvelle fonction

Cette fonction peut être ajoutée dans un des scripts R déjà existants dans le dossier R. Ou elle peut être ajoutée dans un nouveau script R dans ce même dossier. Elle doit être écrite et documentée sur le même modèle que les autres fonctions (voir par exemple dans le script soundex.R, qui peut servir de modèle).

Documenter la fonction : écrire la documentation dans le script R de la fonction, avec le raccourci CTRL+MAJ+D (voir ci-dessous pour plus de détails).

Ajouter dans le fichier DESCRIPTION les packages supplémentaires requis par la nouvelle fonction. Ne pas le faire directement, manuellement. Utiliser :

devtools::use\_package("packagename")

## Les contenus du package

Le package contient trois éléments :

### 1. Un répertoire R/

Le répertoire R/ contient les scripts qui constituent le package.

### 2. Un fichier DESCRIPTION

Le fichier DESCRIPTION contient les méta-données du package et les données qui définissent son environnement, en particulier la liste des packages dont il dépend.

### 3. Un fichier NAMESPACE

## Documenter le package

### Fonctions

Le répertoire man/ contient la documentation du package.

La documentation des fonctions est produite à l’aide de roxygen2 en ajoutant des commentaires précédant les fonctions dans les fichiers R, puis en tapant CTRL+SHIFT+D ou devtools::document().

Des formatages des textes des fonctions sont possibles :

* \emph{italics}: italics.
* \strong{bold}: bold.
* \code{r\_function\_call(with = "arguments")}: r\_function\_call(with = "arguments")

Links

* To other documentation:
  + \code{\link{function}}: Function in this package
  + \code{\link[MASS]{abbey}}: Function in another package
  + \link[=dest]{name}: Link to dest, but show name
  + \linkS4class{abc}: Link to an S4 class
* To the web:
  + \url{http://rstudio.com}: A URL
  + \href{http://rstudio.com}{Rstudio}: A URL with custom link text
  + \email{hadley@@rstudio.com} (note the doubled @): An email address

Lists

* Ordered (numbered) lists:

#' \enumerate{

#' \item First item

#' \item Second item

#' }

* Unordered (bulleted) lists:

#' \itemize{

#' \item First item

#' \item Second item

#' }

* Definition (named) lists:

#' \describe{

#' \item{One}{First item}

#' \item{Two}{Second item}

#' }

Mathematics

You can use standard LaTeX math (with no extensions). Choose between either inline or block display:

* \eqn{a + b}: Inline equation
* \deqn{a + b}: Display (block) equation

Tables

Tables are created with \tabular{}. It has two arguments:

* Column alignment, specified by letter for each column (l = left, r = right, c = center.)
* Table contents, with columns separated by \tab and rows by \cr.

The following function turns an R data frame into the correct format (it ignores column and row names, but should get you started):

Th

### Package

### Vignettes

Les vignettes sont des documents au format RMarkdown rédigés permettant de compléter la documentation « officielle » et standardisée du package. Plusieurs vignettes peuvent être ajoutées à la documentation d’un même package. Il peut s’agir d’une introduction générale au package, d’un mode d’emploi détaillé, de développement plus ou moins détaillés sur des points particuliers qui nécessitent de fournir des précisions aux utilisateurs du package, etc.

On peut créer une nouvelle vignette avec l’instruction suivante :

devtools::use\_vignette("newvignette")

## Inclure des données

Il y a trois façons principales d’inclure des données dans un package :

* Si les données sont mises à la disposition des utilisateurs, il faut les ranger dans le répertoire data/. C’est l’endroit où placer en particulier les données fournies à titre d’exemples.
* Si les données ne sont pas mises à la disposition des utilisateurs, il faut les ranger dans R/sysdata.rda. C’est l’endroit où placer les données utilisées par les fonctions.
* S’il s’agit de stocker des données brutes, il faut les ranger dans inst/extdata.

Les données mises à disposition des utilisateurs ou bien utilisées dans les fonctions sont généralement recodées à partir de données sources. Il est recommandé d’inclure dans les sources du package le script qui constitue les données livrées avec le package. On peut le faire avec la fonction suivante :

devtools::use\_data\_raw()

Pour créer des données qui ne sont pas disponibles pour l’utilisateur, ajouter le paramètre internal = TRUE.

# Fonctions possibles

## age() ou year()

La fonction age() permet d'obtenir l’âge probable d'un individu à partir de son prénom, et éventuellement de quelques informations supplémentaires (date de naissance, pays…).

Remarque : la fonction peut aussi renvoyer une indication de la dispersion.

Question : Que faire si la distribution est multimodale ?

Attention : exclure avant 1945 du calcul parce que les effectifs ne sont pas ceux de la totalité des naissances, mais seulement de celles encore vivante en 1946. Ou alors appliquer une correction tenant compte du taux de survie (qui est différent pour les garçons et les filles).

A faire ensuite :

* probabilité qu’une personne observée telle année avec tel prénom ait plus qu’un certain âge ou moins qu’un certain âge
* Au lieu de prédire une année de naissance, prédire une décennie, avec un paramètre precision = year/decade

Voir aussi : gender(), status(), race().

## anonymize()

Sur le modèle de l’usage de plus en plus répandu de <http://coulmont.com/bac/>, cette fonction remplace le prénom fourni en paramètre par un autre prénom possédant des caractéristiques socio-démographiques (sexe, âge, milieu social, appartenance ethnique) similaires.

Les données du bac sont à partir de 2008, donc marche correctement pour les personnes nées à partir de 1990. Pour les personnes nées avant, utilisation de l’enquête emploi de 1988 = prénom, sexe, profession des parents (extraction fournie par Olivier Godechot à Baptiste Coulmont).

Remarque : il peut arriver que le prénom le plus proche d’un prénom courant soit un prénom rare, autrement dit pas les mêmes propriétés. Exemple quand on change des Marius en Thomas.

Remarque : On peut utiliser le fichier départemental des prénoms pour croiser des fréquences départementales de prénoms et des caractéristiques et scores socio-économiques des départements. Voir le billet de François Guillem sur « Prénoms, régions et analyse en composantes principales », 2011.

Attention : les données de l’enquête emploi ne sont pas diffusables telles quelles dans le package, donc utiliser un summary ad hoc.

Paramètres de la fonction anonymize :

* Year\_min et year\_max pour préciser l’âge de la personne à anonymiser
* Gender pour fixer le sexe.
* Freq pour fixer la fréquence ou le rang du prénom
* Country pour préciser l’origine culturelle
* Region pour préciser le département ou la région.
* Status si tu veux garder du milieu social
* Random = TRUE ou FALSE pour indiquer s’il faut tirer au sort entre plusieurs = pour pas tomber toujours sur le même.

## country(), origin()…

La fonction country() permet d’estimer le pays d’origine, de naissance ou de résidence d'un individu à partir de son prénom, et éventuellement de quelques informations supplémentaires.

Comment faire techniquement ? Baptiste Coulmont et Patrick Simon ont eu recours à une entreprise privée allemande, qui s’appuie sur la présence de n-grams (3-grams en l’occurrence ?) identifiables à des origines culturelles déterminées. Mais payant, pas libre.

Donc on peut trouver des données pour beaucoup de pays, mais elles ne sont pas libres. Un travail de recensement des données disponibles (top one, top 10…) dans le papier de Anne Sofie Beck Knudsen, « Those Who Stayed : Individualism, self-selection and Cultural Change during the Age of Mass Migration », January 2019. Autre source qu’on peut scraper : behindthenames.com. L’idée est d’accumuler une base de prénoms au format long = prénom/zone/rang.

Utiliser aussi <https://adoption.com/baby-names>.

Ou mieux : <https://www.thenamemeaning.com/baby-name-categories>

Il faudra un paramètre distinguant les deux types de sources possibles : d’une part les administrations nationales, qui fournissent des listes de fréquences, et d’autre part les entrepreneurs culturels, qui fournissent des listes de prénoms populaires, ou remarquables, etc.

Pour la zone, utiliser les codes ISO : us, fr, en, nl, etc.

Attention : pose la question d’une détection de l’appartenance ethnique à partir du prénom. La position de la CNIL :

Dans ce document :

<https://www.cnil.fr/sites/default/files/typo/document/RapportdiversiteVD.pdf>

La recommandation n° 4 (p. 31) : « permettre, sous certaines conditions, l’analyse des prénoms et des noms de famille ».

Le texte de la recommandation est très critique sur la fiabilité de la méthode et sur ses risques. Mais il prévoit que le recours aux prénoms et aux patronymes pour étudier des pratiques discriminatoires peut être légitime sous certaines conditions : quand il n’existe pas d’autre source disponible, et quand les informations sont classées non pas en catégories d’appartenance ethnique, mais en catégories ou en grades d’exposition aux discriminations.

Voir aussi : age(), gender(), status().

## distance()

Calcule la distance sociale entre deux prénoms. Possible à partir de :

* Données du bac
* Fichier départemental des prénoms. Pierre : calculer les rangs de chaque prénom dans chaque département, puis calculer la matrice des corrélations entre les rangs ?

## educ()

La fonction educ() permet d’estimer le capital scolaire d’un individu à partir de son prénom, et éventuellement de quelques informations supplémentaires (date de naissance, pays…).

Voir aussi : status()

## extract()

Aussi extract = extrait le prénom d’une chaîne de caractère. Problème : les noms de famille qui sont des prénoms (en France, aux Etats-Unis, etc.).

Par extension : qui trouve tous les prénoms dans un texte.

Voir l’algorithme Soundex de réduction des noms (suppression des voyelles, codage des consonnes phonétiquement proches).

Voir ce qu’OpenEdition appelle les « entités nommées », développé avec l’INRIA pour trouver les noms de lieux et de personnes dans les textes des articles et des ouvrages.

## firstname()

La fonction firstname() renvoie une proposition de prénom à partir d’un certain nombre de paramètres décrivant l’individu à prénommer.

## generate()

Renvoie des listes aléatoires de prénoms, ou répondant à certains critères.

## gender()

Sur le modèle du package américain "gender" (<https://cran.rstudio.com/web/packages/gender/>) et du package danois "namesDK" (<http://www.56n.dk/classify-gender-based-on-danish-first-names/>), la fonction gender() permet d'obtenir le sexe d'un individu à partir de son prénom, et éventuellement de quelques informations supplémentaires (date de naissance, pays…).

Voir aussi : age(), status(), country()

## rank()

Renvoie le rang du prénom dans le classement de l’année considérée ?

## reverse()

La fonction reverse() inverse l’ordre du prénom et du nom de famille dans la chaîne de caractère fournie en paramètre, et ajoute éventuellement un séparateur indiqué en paramètre supplémentaire.

## status()

La fonction status() permet d’estimer le milieu social d’appartenance d'un individu à partir de son prénom, et éventuellement de quelques informations supplémentaires (date de naissance, pays…).

On peut imaginer aussi une fonction educ() qui renvoie une mesure du capital scolaire associé au prénom.

Voir aussi : age(), gender(), country().

# Ressources R

## Packages et scripts

gender : <https://cran.r-project.org/web/packages/gender/index.html>. Le package gender prédit le genre en fonction du prénom et de la date de naissance, en s’appuyant sur des données de recensement et de sécurités sociale. Le package suppose le téléchargement séparé des données, qui peut être réalisé avec la fonction install\_genderdata\_package(). Les pays implémentés sont : c("United States", "Canada", "United Kingdom", "Denmark", "Iceland", "Norway", "Sweden").

genderizeR

NamesDK : <https://github.com/mikkelkrogsholm/namesDK> et <http://www.56n.dk/classify-gender-based-on-danish-first-names/>. Le package propose une fonction gender() qui utilise les listes officielles danoises des prénoms masculins, féminins et unisexes pour classer les noms en masculins, féminins ou unisexes.

gender : Clément Bert-Erboul, post-doctorant, a développé une fonction et une base de données de 300.000 prénoms aspirés à diverses sources pour déterminer le genre. Il a appliqué cette fonction notamment à des données sur le Brésil.

## Problèmes d’accentuation et d’encodage

Une fonction globale :

Old :

Ensuite, il faut nettoyer les problèmes d’accentuation :

Baptiste :

library(foreign)

library(tidyverse)

df <- read.dbf("~/Downloads/nat2015.dbf",[as.is](http://as.is)=T)

# problème d'encodage :

df$preusuel <- iconv(df$preusuel, from="latin1", to="UTF-8")

# il faut ensuite enlever tous les caractères accentués

df$preusuel<-chartr('ÁÀÂÄÃÅÇÉÈÊËÍÏÎÌÑÓÒÔÖÕÚÙÛÜÝ', 'AAAAAACEEEEEIIIINOOOOOUUUUY',df$preusuel)

df$annais <- as.numeric(as.character(df$annais))

Sébastien :

Une petite contribution concernant la désambiguisation des formes accentuées : voici un paramétrage que je trouve bien pratique,

iconv("ÃaéiÍǐǨÁé", "utf-8", "ASCII//TRANSLIT")

L'idée est de convertir, en utilisant les règles de translittération implémentées, la chaîne de caractère (encodée en ce qu'on veut, latin1, utf-8, etc.) en ASCII.

Cela revient à la fonction proposée par Baptiste, mais dans une version plus universelle. (Seront traités tous les cas, par exemple, de prénom étrangers ayant des diacritiques que l'on n'aurait pas pensé à indiquer à la fonction chartr).

# Données sources

Les données sources sont distribuées avec le package.

## France

### Fichier des prénoms

Le fichier des prénoms est désormais en libre accès :

Fichier 2017 :

<https://www.data.gouv.fr/fr/datasets/ficher-des-prenoms-de-1900-a-2017/>

Fichiers précédents :

<https://www.data.gouv.fr/fr/datasets/fichier-prenoms-nd/>

<https://www.data.gouv.fr/fr/datasets/liste-des-prenoms-par-annee-prs/>

La licence : Licence ouverte. Voir :

<https://www.etalab.gouv.fr/licence-ouverte-open-licence>

Utilisation, modification et diffusion libres. Mentionner la source.

Attention : vérifier le nettoyage, et en particulier refaire un summarize après nettoyage des accents pour rassembler les différentes orthographes des prénoms accentués.

### Listes électorales

Accessibles à tous les citoyens qui les demandent. Par rapport au fichier départemental des prénoms, permet d’accéder au niveau communal (et au niveau de l’arrondissement).

### Enquête TRA

Pour les vieux prénoms

Données et modalités d’accès :

<https://tra.site.ined.fr/>

### Données pas forcément libres mais utilisables avec transformation de manière agrégée, ou autres données

Résultats du bac

Enquête emploi 1988

Fichier SIRENE = en accès libre. Un gros fichier d’entrepreneurs, avec des informations : géolocalisation, secteurs d’activité.

Scraping des mesures nominatives dans le journal officiel. L’informaticien CNRS Nathann Cohen <http://steinertriples.fr/ncohen/data>. Par exemple toutes les nominations d’ambassadeurs. Disponible en téléchargement.

## International

Info Clément :

Sur le Harvard Dataverse:

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/YPRQH8>

including 6.2 million names for 182 different countries to disambiguate the gender. (2016-11)

Les sources de cette base sont : *Social Security Administration (US) Alberta government (Canada) Michael (2007) Office for National Statistics of United Kingdom (ONS) Tang et al. (2011) US Census Bureau (2000) Wikipedia WIPO (Assemblies list) Statistics Sweden Instituto Nacional de Estadística (Spain) Institut National de la Statistique (France) Yu et al. (2014) Denmark Statistics*

L’avantage de Dataverse est que la source est légalement utilisable et l’utilisation du dépôt valorise le travail de ceux qui ont réalisé la base.

Il y a un package R qui permet l’accès aux bases de données Dataverse

<https://cran.r-project.org/web/packages/dataverse/index.html>.

Ça peut-être une idée pour réaliser un dépôt et permettre une mise à jour de base de données en local.

Sinon :

Il y a peu de pays qui donnent accès à des informations qui ressemblent à celles du fichier des prénoms (France, USA...). Rien pour l'Allemagne. Données réduites pour l'Angleterre ou la Suisse.

Données non libres pour les Pays bas.

The [North Atlantic Population Project](https://www.nappdata.org/napp/) provides data for Canada, the United Kingdom, Germany, Iceland, Norway, and Sweden for years between 1758 and 1910, based on census microdata from those countries.

Beaucoup de pays donnent les top100 de leurs prénoms. Utile pour nourrir la fonction country().

## Angleterre

<https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/livebirths/datasets/babynamesenglandandwalesbabynamesstatisticsboys>

<http://names.darkgreener.com/>

## Belgique

Ici nous avons la Belgique 2007/2017 en licence un peu libre Open Government Licence v3.0 :

<https://statbel.fgov.be/fr/open-data/prenoms-des-nouveaux-nes-par-commune>

## Brésil

Je connais cette base brésilienne faite en 2010 avec cet api, mais c’est moins évident qu’une liste, mais au moins le script est en licence GNU. Les données sont copyright 2016 IBGE - Instituto Brasileiro de Geografia e Estatística

<https://github.com/turicas/genero-nomes>

Il y a une base spécifiquement Brésilienne sur le Harvard Dataverse (libre) :

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ORH029>

la stratégie est assez maligne pour cette base puisqu’il s’agit de « *74,000 Brazilian first names and how often each name is used for a male or female candidates in the 2000, 2004, 2008, and 2012 municipal elections in Brazil*.” Une bonne manière de constater la récurrence de prénoms chez les élus brésiliens en fonction de la géographie 😊

Je suis en contact également avec les administrateurs du site <https://brasil.io/dataset/genero-nomes/nomes> qui regroupe 100788 lignes issues du recensement brésilien de 2010.

## Etats-Unis

U.S. Census data is provided by [IPUMS USA](https://usa.ipums.org/usa/) from the Minnesota Population Center, University of Minnesota. The IPUMS data includes 1% and 5% samples from the Census returns. The Census, taken decennially, includes respondent’s birth dates and gender. With the gender package, it is possible to use this dataset for years between 1789 and 1930. The dataset includes approximately 339,967 unique names.

U.S. Social Security Administration data was collected from applicants to Social Security. The Social Security Board was created in the New Deal in 1935. Early applicants, however, were people who were nearing retirement age not people who were being born, so the dataset extends further into the past. However, the Social Security Administration did not immediately require all persons born in the United States to register for a Social Security Number. (See Shane Landrum, “The State’s Big Family Bible: Birth Certificates, Personal Identity, and Citizenship in the United States, 1840–1950” [PhD dissertation, Brandeis University, 2014].) A [consequence](https://github.com/ropensci/gender/issues/9) of this—for reasons that are not entirely clear—is that for years before 1918, the SSA dataset is heavily female; after about 1940 it skews slightly male. For this reason this package corrects the prediction to assume a secondary sex ratio that is evenly distributed between males and females. Also, the SSA dataset only includes names that were used more than five times in a given year, so the “long tail” of names is excluded. Even so, the dataset includes 91,320 unique names. The SSA dataset extends from 1880 to 2012, but for years before 1930 you should use the IPUMS method.

## Les données généaologiques

En plus des sites de statistiques nationales, nous nous sommes dit que les sites de généalogies pourraient être une piste.

Werelate propose ses données sur **2,900,000 de personnes,** sous licences libres (CC-BY-SA License et GFDL).

Le cas d’un Clément dans la base

<https://www.werelate.org/wiki/Person:Adrian_Slattery_%282%29>

Un autre sur le modèle du wiki est <https://en.rodovid.org/wk/>  sous CC :BY

<https://en.rodovid.org/wk/Person:3> <- les pages sont numérotées !!!! idéal pour scraper

Ces deux bases concernent plutôt des cohortes anciennes.

Il y a aussi <https://www.geneanet.org/>

Les conditions d’utilisation laissent peut-être une porte pour négocier un usage pro/recherche. Ils peuvent peut-être, être sensibles à notre démarche.

<https://www.geneanet.org/legal/cgu/>

# La structuration des données

Garder toute la base année par année, ou structurer en deux :

* Un fichier des prénoms qui ne changent pas
* Un fichier des prénoms qui changent, avec plutôt comme granularité la décennie que l’année.

Objectif : gagner de la place.

# Pistes bibliographiques

Antoine Mazières et Camille Roth ont publié un article dans le BMS qui s'appuie sur une classification par origine nationale (ils ont scrappé je ne sais plus quelles bases) : il faudrait voir si c'est possible de récupérer une partie de leurs données. Baptiste : demande à Antoine Mazières, qui serait sûrement d’accord.

Didier Chavrier, chargé de mission égalité-diversité à l’Université d’Orléans, croisé dans le bureau de Julien Barnier, s’intéresse aux questions d’appartenances ethniques des prénoms : [didier.chavrier@univ-orleans.fr](mailto:didier.chavrier@univ-orleans.fr)