

Consolider sa reproductibilité avec le package `prefixer`

Bénédicte Garnier

2023-10-24

Il est très simple d'ajouter, dans un script R, les noms des packages aux appels des fonctions. Sous forme de préfixes, ces indications permettent de repérer les packages utilisés et donc faciliter la reproductibilité de l'analyse. Nous prenons ici l'exemple d'un script générant un nuage de mots à partir du package de statistique textuelle `R.temis`.

Table des matières

Le script d'origine	2
Ajouter le nom des packages aux fonctions avec <code>prefixer</code>	4
Installer le package <i>prefixer</i> à partir de GitHub	4
Prefixer le script	4
Utiliser la fenêtre shiny permettant de choisir le préfixe	5
Résultat: le script est enrichi	5

Table 1: Fonctions utilisées dans la fiche

Packages	Fonctions
<code>prefixer</code>	<code>prefixer</code>
<code>R.temis</code>	<code>import_corpus</code>
<code>R.temis</code>	<code>build_dtm</code>
<code>R.temis</code>	<code>frequent_terms</code>
<code>R.temis</code>	<code>combine_terms</code>
<code>R.temis</code>	<code>word_cloud</code>

Pour illustrer ce propos, nous reprenons le script de la fiche de Coralie Cottet *Générer des graphes de mot avec R.temis*. Nous allons ajouter un prefixe à chaque fonction utilisée pour générer un nuage de mots.

Documentation :

- **RStudio Addins** sont des extensions pour l'environnement de développement intégré (IDE) RStudio
- **prefixer: Shiny Gadget to interactively prefix function in a script**
- Séminaire R à l'Usage des Sciences Sociales, *Esquisse, moins tu sais coder plus tu vas rigoler*

Installation des packages :

```
install.packages("remotes")
install.packages("R.temis")
install.packages("dyplyr")
```

Fichiers à utilisés :

Les [voeux du président Hollande pour les années 2013 à 2017](#).

Chaque discours correspond à un fichier texte et ces 5 discours sont placés dans un dossier unique appelé *dossier_de_textes*

Le script d'origine

```
## Appel des packages
library(R.temis)
library(tidyverse)

## Statistiques textuelles
## Création du tableau lexical associé au corpus des discours

# Importation des textes avec la fonction `import_corpus`
corpusv <- import_corpus("datas/dossier_de_textes", format="txt", language ="fr")

# Création du tableau lexical associé
# On ne garde pas les mots-outils et on garde aussi les mots de 1 seule lettre

dtm <-build_dtm(corpusv, remove_stopwords = T, min_length = 1)
dtm
```

```

# On obtient un *tableau lexical* de 5 unités de texte (les voeux) et 1575 mots différents .

## Afficher les occurrences des mots du corpus
# Calcul des occurrences des mots
frequent_terms(dtm)

# On remarque que les mots `tous`, `tout` et `toutes` sont très fréquents et qu'ils sont emp

#### Personnalisation du lexique

# Création du dictionnaire/lexique associé au tableau lexical
dic <- dictionary(dtm)

# On crée une nouvelle table `dic2` en créant une première colonne appelée `word` avec l'in
dic2 = dic %>%
  rownames_to_column(var="word") %>%
  mutate(Term = word)

row.names(dic2) <- dic2$word

# On peut alors remplacer les mots de la colonne Term par l'intitulé souhaité. Ici tout.tes
dic2$Term[dic2$word == "toutes"] <- "tous.tes"
dic2$Term[dic2$word == "tout"] <- "tous.tes"
dic2$Term[dic2$word == "tous"] <- "tous.tes"

# Lemmatisation à l'aide du lemmatiseur personnalisé
dtmlem <- combine_terms(dtm, dic2)

# Ensemble de mots à retirer
mots_a_retirer <- c("a", "plus")

# Suppression de mots dans le tableau lexical
dtm2 <- dtmlem[, !colnames(dtmlem) %in% mots_a_retirer]

# Nouvelles fréquences des mots associés au corpus
frequent_terms(dtm2)

### Affichage du nuage de mots

cloud <- word_cloud(dtm2, color = 'black', min.freq = 1, n = 50)
title(main = "Mots les plus fréquents dans les discours de F. Hollande entre 2013 et 2017")

```

Ajouter le nom des packages aux fonctions avec prefixer

Installer le package *prefixer* à partir de GitHub

L'addon *prefix* est disponible sur la forge logicielle de *dreamRs*

```
install.packages("remotes")  
install_github("dreamRs/prefixer")
```

Celui-ci permet d'ouvrir une interface qui propose pour chaque fonction du script de lui ajouter un préfixe composé du nom de la fonction suivi de ::

prefixer:: Cancel

Function:
build_dtm

Context:
dtm <-build_dtm(corpusv, remove_stopwords = T, min_length = 1)

Suggested prefix:
R.temis::build_dtm

:: Add prefix ▶ Skip

[▶ Skip all occurrences](#)

If your script use unloaded packages, functions from these packages won't be prefixed. You can load packages by [clicking here](#) .

Prefixer le script

A la fin du script R, on charge le package *prefixer* puis on ouvre l'interface avec la commande `prefixer()`

```
## Appel du package
library(prefixer)
## Ouverture de l'interface RShiny
prefixer()
```

Dans le cas de notre script:

```
#cloud<-word_cloud(dtm2, color= 'black', min.freq=1,n =50)
#title(main = "Mots les plus fréquents dans les discours de F. Hollande entre 2013 et 2017")

library(prefixer)
prefixer()
```

Utiliser la fenêtre shiny permettant de choisir le préfixe

Note

Seulement pour le format html

Résultat: le script est enrichi

```
## Statistiques textuelles
## Création du tableau lexical associé au corpus des discours

# Importation des textes avec la fonction `import_corpus`
corpusv <- R.temis::import_corpus("datas/dossier_de_textes", format="txt", language ="fr")

# Création du tableau lexical associé
# On ne garde pas les mots-outils et on garde aussi les mots de 1 seule lettre

dtm <-R.temis::build_dtm(corpusv, remove_stopwords = T, min_length = 1)
dtm

# On obtient un *tableau lexical* de 5 unités de texte (les voeux) et 1575 mots différents .

## Afficher les occurrences des mots du corpus
# Calcul des occurrences des mots
R.temis::frequent_terms(dtm)
```

```

# On remarque que les mots `tous`, `tout` et `toutes` sont très fréquents et qu'ils sont emp

#### Personnalisation du lexique

# Création du dictionnaire/lexique associé au tableau lexical
dic <- R.temis::dictionary(dtm)

# On crée une nouvelle table `dic2` en créant une première colonne appelée `word` avec l'in
dic2 = dic %>%
  tibble::rownames_to_column(var="word") %>%
  dplyr::mutate(Term = stringr::word)

row.names(dic2) <- dic2$word

# On peut alors remplacer les mots de la colonne Term par l'intitulé souhaité. Ici tout.tes
dic2$Term[dic2$word == "toutes"] <- "tous.tes"
dic2$Term[dic2$word == "tout"] <- "tous.tes"
dic2$Term[dic2$word == "tous"] <- "tous.tes"

# Lemmatisation à l'aide du lemmatiseur personnalisé
dtmlem <- R.temis::combine_terms(dtm, dic2)

# Ensemble de mots à retirer
mots_a_retirer <- c("a", "plus")

# Suppression de mots dans le tableau lexical
dtm2 <- dtmlem[, !colnames(dtmlem) %in% mots_a_retirer]

# Nouvelles fréquences des mots associés au corpus
R.temis::frequent_terms(dtm2)

### Affichage du nuage de mots

cloud <- R.temis::word_cloud(dtm2, color = 'black', min.freq = 1, n = 50)
title(main = "Mots les plus fréquents dans les discours de F. Hollande entre 2013 et 2017")

```