Données numériques en sciences sociales

Mastère Spécialisé Data Science

Flora ZIADI

15/01/2018

Sommaire

L'objectif de ce projet est d'extraire tous les guides touristiques de France sur le site Decitre. Plus précisément, je souhaiterais recueillir les informations suivantes : le titre, l'auteur, l'édition, la date de parution, la région et le prix. Puis afficher le livre le moins cher et disponible immédiatement (en stock) pour chaque région.

Le lien du site est le suivant : https://www.decitre.fr/livres/loisirs-nature-voyages/guides-de-voyage/guides-france-regions.html

Nous sommes obligés de faire une extraction par région pour pouvoir recueillir l'information région.

Dans un premier temps, nous allons extraire les informations pour la région Alsace. Puis dans un second temps, nous automatiserons notre code pour recueillir l'information pour toutes les régions.

Partie 1: Extraction des informations pour la région Alsace.

1) Accéder à la page web et la parser

Pour l'instant, nous essayons d'extraire la première page.

```
library(RCurl)

library(XML)

# on récupère le code source
page.brute <- getURL("https://www.decitre.fr/livres/loisirs-nature-voyages/guides-de-voyage/guides-france-regions/alsace.html",followlocation =T)
# on restructure l'arbre
page.parsee <-  htmlParse(page.brute)
```

Lorsque la structure précédente ne marche pas, nous pouvons utiliser la structure suivante qui est plus complexe :

```
# Un crawler complet (cookies, followlocation, useragent, timeout) -- Recommande
mycurl <- getCurlHandle()
curlSetOpt(cookiejar= "~/Rcookies", curl = mycurl)
raw.page <- getURL("https://www.decitre.fr/livres/loisirs-nature-voyages/guides-de-voyage/guides-france-regions/alsace.html",
                   curl= mycurl,
                   useragent = "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.120 Safari/537.36",
                   timeout = 60,
                   followlocation= TRUE)
parsed <- htmlParse(raw.page)
```

A présent, nous essayons d'extraire la totalité des pages pour la région alsace. Pour cela, nous utilisons **une boucle for** pour automatiser la collecte.

```
alsace <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guides
-de-voyage/guides-france-regions/alsace.html?p=", 1:10)

page.brute <- list()                    ## On crée un liste vide
for (i in 1:10) {                       ## Pour i= 1, 2, 3 :
  cat("\rPage num.", i)                 ## On affiche où l' on en est
  page.brute[[i]] <- getURL(alsace[i]) ## On enregistre la page dans la liste
  Sys.sleep(2)                          ## On dort deux secondes
}

page.parsee <-  htmlParse(page.brute)
```

2) Extraire les informations suivantes : le titre, l'auteur, l'édition, la date de parution, la région, le prix.

Pour extraire les données d'une page HTML, nous utilisons le langage de requêtes XPATH qui permet de sélectionner les nœuds d'un document (XML).

```
titre0 <- xpathSApply(page.parsee, "//div[contains(@class, 'content-50 catalo
g-product-list-details fiche_page_recherche')]//a[@class='product-title']", x
mlValue)
auteur0<- xpathSApply(page.parsee, "//div[contains(@class, 'content-50 catalo
g-product-list-details fiche_page_recherche')]//div[@class='authors']", xmlVa
lue)
edition0<- xpathSApply(page.parsee, "//div[contains(@class, 'content-50 catal
og-product-list-details fiche_page_recherche')]//ul [@class='extra-infos hide
-on-grid-container hide-on-responsive']//li[1]", xmlValue)
# date_de_parution<- xpathSApply(page.parsee, "//div[contains(@class, 'conten
t-50 catalog-product-list-details fiche_page_recherche')]//ul [@class='extra-
infos hide-on-grid-container hide-on-responsive']//li[3]", xmlValue)
# Pour la date de parution, il y a certains livres, qui n'ont pas de date de
parution. Ainsi, je n'ai pas le même nombre de lignes que les autres vecteurs
. Pour contrer ce problème, j'ai choisi de prendre la ligne entière (édition,
forme du livre et date de parution), puis de supprimer les deux premiers élém
ents à l'aide des regex.
date_de_parution0<- xpathSApply(page.parsee, "//div[contains(@class, 'content
-50 catalog-product-list-details fiche_page_recherche')]//ul [@class='extra-i
nfos hide-on-grid-container hide-on-responsive']", xmlValue)
prix0<- xpathSApply(page.parsee, "//div[contains(@class, 'content-25 catalog-
product-list-actions')]//span[@class='final-price']", xmlValue)
etat0<- xpathSApply(page.parsee, "//div[contains(@class, 'content-25 catalog-
product-list-actions')]//div[@class='stock-info']", xmlValue)
```

3) Nettoyer les données

A présent nous allons nettoyer les données obtenues à partir du langage de requêtes XPATH à l'aide des expressions régulières REGEX.

REGEX est un langage qui permet de manipuler des chaînes de caractères. Nous utiliserons deux fonctions dans R :

- grep pour rechercher et filtrer :

grep(une_regex, un_objet_texte)

- gsub pour rechercher et remplacer :

gsub(une_regex, le_remplacement, un_objet_texte)

```
titre <- gsub("\\n", "", titre0) # on supprime les \\n
titre <- gsub("^\\s+|\\s+$", "", titre) # on supprime les espaces en début et
fin de chaînes

auteur <- gsub("^\\s+|\\s+$", "", auteur0) # on supprime les espaces en début
et fin de chaînes

edition <- gsub("\\n", "", edition0) # on supprime les \\n
edition <- gsub("^\\s+|\\s+$", "", edition) # on supprime les espaces en débu
t et fin de chaînes

date_de_parution<- gsub("\\D", "",date_de_parution0) # on récupère les donnée
s digitales
date_de_parution <- gsub("(\\d{2})(\\d{2})(\\d{4})","\\1/\\2/\\3", date_de_pa
rution) # on met les données digitales sous forme de date

prix<- gsub("\\D", "",prix0) # on remplace par vide tout ce qui n'est pas des
digits
prix <- gsub("(\\d{2}|\\d{1})(\\d{2})", "\\1,\\2",prix) # on met les données
digitales sous la forme d'un ou deux chiffres avant la virgule puis 2 chiffre
s après le point
prix <- gsub(",","\\.",prix) # on transforme la virgule en point
prix <- as.numeric(as.character(prix)) #on transforme la chaine de caractère
en variable numérique. Cette transformation est nécessaire pour la suite pour
rechercher le prix minimal.

etat <-gsub("Informations*.+", "",etat0) #on supprime la ligne "InformationsC
et article.."
etat <- gsub("\\n", "", etat) # on supprime les \\n
etat <- gsub("^\\s+|\\s+$", "", etat) # on supprime les espaces en début et f
in de chaînes
```

4) Mettre les données dans un data.frame

On met à présent les données dans un data.frame.

```
table <- data.frame(titre, auteur, edition, prix, date_de_parution, etat)
head(table)
```

On affiche les premières lignes de table.

	Titre	auteur	edition	Prix	date_de_parution	etat
1	Alsace. Massif des Vosges, Escapade en Forêt-Noire et à Bâle édition 2016	Michelin	Michelin	14.90	18/04/2016	Neuf - En stock
2	Alsace (Grand-Est) édition 2017-2018	Le Routard	Hachette Tourisme	13.20	12/04/2017	Neuf - En stock
3	Alsace2e édition	Claire Angot, Sonia de Araujo, Christophe Corbel, Hugues Derouard	Lonely Planet	14.50	06/04/2017	Neuf - Expédié sous 3 à 6 jours
4	Strasbourg édition 2016	Michelin	Michelin	9.90	15/02/2016	Neuf - Expédié sous 3 à 6 jours
5	Strasbourg4e édition	Nicolas Peyroles, Samuel Teller, Sylvain Moizie	Guides Gallimard	8.90	13/04/2017	Neuf - En stock
6	L'Alsace à vélo	Chamina	Chamina	13.50	20/01/2018	Neuf - Précommande

5) Filtrer le tableau

On souhaite afficher le livre le moins cher et disponible immédiatement (en stock). Pour cela, utilise la fonction grep.

```
# on sélectionne d'abord les livres disponibles immédiatement
table2 <- table[grep("En stock", table$etat), ]

# puis on filtre sur le livre le moins cher
min <-min(table2$prix) # on cherche le minimum
table3 <- table2[grep(paste0("^",min,"$"),table2$prix), ] #on restreint le pr
ix uniquement au min. Par exemple, cette syntaxe évite de retourner 16 ou 6.9
lorsque le minimum vaut 6.
table3
```

Le livre le moins cher pour la région Alsace qui est disponible immédiatement est le suivant :

	Titre	Auteur	Edition	Prix	Date de parution	Etat
5	Strasbourg4e édition	Nicolas Peyroles, Samuel Teller, Sylvain Moizie	Guides Gallimard	8.9	13/04/2017	Neuf - En stock

Partie 2 : Automatisation du code pour afficher toutes les régions

A présent, nous allons automatiser le code que nous venons de faire pour la région Alsace pour recueillir l'information pour toutes les régions.

Pour cela, nous créons la fonction extraction :

```
extraction=function(adresse_region, nb_pages, nom_region)
{
  # 1) Accéder à la page web et la parser
  page.brute <- list()                  ## On crée un liste vide
  for (i in 1:nb_pages) {               ## Pour i= 1, 2, 3 :
    cat("\rPage num.", i)               ## On affiche où l' on en est
    page.brute[[i]] <- getURL(adresse_region[i])## On enregistre la page dans
la liste
    Sys.sleep(2)                        ## On dort deux secondes
  }
  page.parsee <-  htmlParse(page.brute)

  # 2) Extraire les informations suivantes : le titre, l'auteur, l'édition, l
a date de parution, la région, le prix.
  titre0 <- xpathSApply(page.parsee, "//div[contains(@class, 'content-50 cata
log-product-list-details fiche_page_recherche')]//a[@class='product-title']",
xmlValue)
  auteur0<- xpathSApply(page.parsee, "//div[contains(@class, 'content-50 cata
log-product-list-details fiche_page_recherche')]//div[@class='authors']", xml
Value)
  edition0<- xpathSApply(page.parsee, "//div[contains(@class, 'content-50 cat
alog-product-list-details fiche_page_recherche')]//ul [@class='extra-infos hi
de-on-grid-container hide-on-responsive']//li[1]", xmlValue)
  date_de_parution0<- xpathSApply(page.parsee, "//div[contains(@class, 'conte
nt-50 catalog-product-list-details fiche_page_recherche')]//ul [@class='extra
-infos hide-on-grid-container hide-on-responsive']", xmlValue)
  prix0<- xpathSApply(page.parsee, "//div[contains(@class, 'content-25 catalo
g-product-list-actions')]//span[@class='final-price']", xmlValue)
  etat0<- xpathSApply(page.parsee, "//div[contains(@class, 'content-25 catalo
g-product-list-actions')]//div[@class='stock-info']", xmlValue)

  #3) Nettoyer les données
```

```
  titre <- gsub("\\n", "", titre0) # on supprime les \\n
  titre <- gsub("^\\s+|\\s+$", "", titre) # on supprime les espaces en début
et fin de chaînes

  auteur <- gsub("^\\s+|\\s+$", "", auteur0) # on supprime les espaces en déb
ut et fin de chaînes

  edition <- gsub("\\n", "", edition0) # on supprime les \\n
  edition <- gsub("^\\s+|\\s+$", "", edition) # on supprime les espaces en dé
but et fin de chaînes

  date_de_parution<- gsub("\\D", "",date_de_parution0) # on récupère les donn
ées digitales
  date_de_parution <- gsub("(\\d{2})(\\d{2})(\\d{4})", "\\1/\\2/\\3",date_de_
parution) # on met les données digitales sous forme de date

  prix<- gsub("\\D", "",prix0) # on remplace par vide tout ce qui n'est pas d
es digits
  prix <- gsub("(\\d{2}|\\d{1})(\\d{2})", "\\1,\\2",prix) # on met les donnée
s digitales sous la forme d'un ou deux chiffres avant la virgule puis 2 chiff
res après le point
  prix <- gsub(",","\\.",prix) # on transforme la virgule en point
  prix <- as.numeric(as.character(prix)) #on transforme la chaine de caractèr
e en variable numérique

  etat <-gsub("Informations*.+", "",etat0) #on supprime la ligne "Information
sCet article.."
  etat <- gsub("\\n", "", etat) # on supprime les \\n
  etat <- gsub("^\\s+|\\s+$", "", etat) # on supprime les espaces en début et
fin de chaînes

  #4) Mettre les données dans un data.frame
  table <- data.frame(nom_region, titre, auteur, edition, prix, date_de_parut
ion, etat)

  #5) Filtrer le tableau

  # on séléctionne tout d'abord les livres disponibles immédiatement
  table2 <- table[grep("En stock", table$etat), ]

  # puis on filtre sur le livre le moins cher
  min <-min(table2$prix) # on cherche le minimum
  table3 <- table2[grep(paste0("^",min,"$"),table2$prix), ]  # on restreint l
e prix uniquement au min.

  # return(table) # On peut également afficher la liste entière des guides pa
r région
  return(table3)
}
```

Puis nous appelons cette fonction pour chaque région.

```
alsace <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guides
-de-voyage/guides-france-regions/alsace.html?p=", 1:10)
guide_alsace <- extraction(alsace,10, "Alsace")

aquitaine <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/gui
des-de-voyage/guides-france-regions/aquitaine.html?p=", 1:12)
guide_aquitaine <- extraction(aquitaine,12, "Aquitaine")

#auvergne <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/gui
des-de-voyage/guides-france-regions/auvergne.html?p=", 1:9)
#guide_auvergne <- extraction(auvergne,9, "Auvergne")

bourgogne <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/gui
des-de-voyage/guides-france-regions/bourgogne.html?p=", 1:5)
guide_bourgogne <- extraction(bourgogne,5, "Bourgogne")

bretagne <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guid
es-de-voyage/guides-france-regions/bretagne.html?p=", 1:14)
guide_bretagne <- extraction(bretagne,14, "Bretagne")

centre <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guides
-de-voyage/guides-france-regions/centre.html?p=", 1:3)
guide_centre <- extraction(centre,3, "Centre")

champagne <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/gui
des-de-voyage/guides-france-regions/champagne-ardenne.html?p=", 1:3)
guide_champagne <- extraction(champagne,3, "Champagne-Ardenne")

corse <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guides-
de-voyage/guides-france-regions/corse.html?p=", 1:8)
guide_corse <- extraction(corse,8, "Corse")

franche_comte <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages
/guides-de-voyage/guides-france-regions/franche-comte.html?p=", 1:2)
guide_franche_comte<- extraction(franche_comte,2, "Franche-Comté")

languedoc_roussillon <- paste0("https://www.decitre.fr/livres/loisirs-nature-
voyages/guides-de-voyage/guides-france-regions/languedoc-roussillon.html?p=",
1:9)
guide_languedoc_roussillon <- extraction(languedoc_roussillon,9, "Languedoc-R
oussillon")

limousin <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guid
es-de-voyage/guides-france-regions/limousin.html?p=", 1:2)
guide_limousin <- extraction(limousin ,2, "Limousin")

lorraine <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guid
```

```
es-de-voyage/guides-france-regions/lorraine.html?p=", 1:3)
guide_lorraine <- extraction(lorraine ,3, "Lorraine")

midi_pyrenees <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages
/guides-de-voyage/guides-france-regions/midi-pyrenees.html?p=", 1:11)
guide_midi_pyrenees <- extraction(midi_pyrenees ,11, "Midi-Pyrénées")

nord_pas_de_calais <- paste0("https://www.decitre.fr/livres/loisirs-nature-vo
yages/guides-de-voyage/guides-france-regions/nord-pas-de-calais.html?p=", 1:6
)
guide_nord_pas_de_calais <- extraction(nord_pas_de_calais ,6, "Nord-Pas-de-Ca
lais")

normandie <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/gui
des-de-voyage/guides-france-regions/normandie.html?p=", 1:9)
guide_normandie <- extraction(normandie ,9, "Normandie")

# paris <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guide
s-de-voyage/guides-france-regions/paris-ile-de-france.html?p=", 1:39)
# guide_paris <- extraction(paris ,39, "Paris & Ile-de-France")
# il manque un auteur pour le dataframe

loire <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guides-
de-voyage/guides-france-regions/pays-de-loire.html?p=", 1:8)
guide_loire <- extraction(loire ,8, "Pays de la Loire")

picardie <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/guid
es-de-voyage/guides-france-regions/picardie.html?p=", 1:2)
guide_picardie <- extraction(picardie ,2, "Picardie")

poitou_charentes <- paste0("https://www.decitre.fr/livres/loisirs-nature-voya
ges/guides-de-voyage/guides-france-regions/poitou-charentes.html?p=", 1:4)
guide_poitou_charentes<- extraction(poitou_charentes ,4, "Poitou-Charentes")

provence_alpes_cote_azur <- paste0("https://www.decitre.fr/livres/loisirs-nat
ure-voyages/guides-de-voyage/guides-france-regions/provence-alpes-cote-d-azur
.html?p=", 1:27)
guide_provence_alpes_cote_azur<- extraction(provence_alpes_cote_azur ,27, "Pr
ovence-Alpes-Côte-d'Azur")

rhone_alpes <- paste0("https://www.decitre.fr/livres/loisirs-nature-voyages/g
uides-de-voyage/guides-france-regions/rhone-alpes.html?p=", 1:5)
guide_rhone_alpes<- extraction(rhone_alpes ,5, "Rhône-Alpes")

# On met tous les guides retenus dans un seul vecteur
table_region <- rbind(guide_alsace, guide_aquitaine, guide_bourgogne, guide_b
retagne, guide_centre, guide_champagne, guide_corse, guide_franche_comte, gui
de_languedoc_roussillon, guide_limousin, guide_lorraine, guide_midi_pyrenees,
guide_nord_pas_de_calais, guide_normandie, guide_loire, guide_picardie, guide
```

```
_poitou_charentes, guide_provence_alpes_cote_azur, guide_rhone_alpes)
table_region
# au total on a 2670 guides touristiques dans le dataframe table (3594 - 761
pour paris - 163 pour l'auvergne)
```

Pour les régions Auvergne et Paris, je n'ai pas pu afficher les guides touristiques car au niveau de la création du dataframe table (cf. ligne 190 du code R), j'ai un message d'erreur. En effet, le nombre de lignes pour la colonne auteur est différent des autres colonnes, car il y a quelques auteurs qui ne sont pas renseignés sur le site. Ainsi, en XPATH je n'arrive pas à les détecter et je ne trouve pas d'astuce pour les reconnaître comme j'ai pu trouver pour la date de parution.

On obtient ainsi le livre le moins cher et disponible immédiatement (en stock) pour chaque région. Il également possible d'afficher la liste entière des guides touristiques par région en remplaçant "titre" au lieu de "titre3" dans la valeur retournée par la fonction extraction.

Nom region	Titre	Auteur	Edition	prix	Date de parution	Etat
Alsace	Le château du Haut-Koenisbourg	Corinne Albaut, Dorothée Duntze	Patrimoine CMN (Editions du)	7.00	01/06/2005	Neuf - En stock
Aquitaine	The Proumeyssac Cave	Gérard Delorme, Pierre Vidal, Georgette Duret, Bernard Bordier, Francis Guichard	Sud Ouest (Editions)	4.60	19/02/2010	Neuf - En stock
Bourgogne	Un grand week-end en Bourgogne du Sud	Marie-Hélène Chaplain	Hachette Tourisme	8.95	26/04/2017	Neuf - En stock
Bretagne	La Bretagne. Nature, traditions, histoire	Chloé Chamouton	Geste Editions	4.90	01/02/2016	Neuf - En stock
Bretagne	Rennes	Chloé Chamouton	Geste Editions	4.90	01/06/2016	Neuf - En stock
Bretagne	Saint-Malo	Chloé Chamouton	Geste Editions	4.90	01/06/2016	Neuf - En stock
Centre	Berry édition 2010-2011	Philippe Gloaguen	Hachette	12.10	21/04/2010	Neuf - En stock
Champagne -Ardenne	Verdun, Argonne, Saint-Mihiel. Les champs de bataille	Amaury de Valroger	Michelin	12.90	29/10/2011	Neuf - En stock

Corse	Bonifacio	Marie-Hélène Ferrari, Xavier Lorenzi	Clémentine (Editions)	3.50	01/04/2008	Neuf - En stock
Franche-Comté	Franche-Comté édition 2010-2011	Philippe Gloaguen	Hachette Tourisme	12.10	17/03/2010	Neuf - En stock
Languedoc-Roussillon	Pays Pyrénées-Méditerranée	Le Routard	Hachette Tourisme	4.90	18/05/2016	Neuf - En stock
Limousin	Limoges	Laurent Bourdelas	Geste Editions	4.90	13/05/2016	Neuf - En stock
Lorraine	Lorraine édition 2009	Fiona Debrabander	Hachette Tourisme	11.90	03/12/2008	Neuf - En stock
Midi-Pyrénées	Toulouse, 800 promesses de bons moments	Elodie Pages	Bonneton (Christine)	8.00	26/10/2016	Neuf - En stock
Nord-Pas-de-Calais	Lille, 800 promesses de bons moments	Tirloy, Michel Ragot	Bonneton (Christine)	8.00	26/10/2016	Neuf - En stock
Normandie	Villas de Cabourg, Calvados	Carmen Popescu	Cahiers du temps	6.00	01/01/2003	Neuf - En stock
Pays de la Loire	Nantes en quelques jours édition 2017 - avec 1 Plan détachable	Bénédicte Houdré	Lonely Planet	8.99	20/04/2017	Neuf - En stock
Picardie	Picardie édition 2009-2010	Philippe Gloaguen	Hachette Tourisme	12.10	22/04/2009	Neuf - En stock
Poitou-Charentes	Fort Boyard	David Canard	Geste Editions	4.90	01/02/2016	Neuf - En stock
Poitou-Charentes	L'île d'Oléron	Alain Crespin	Geste Editions	4.90	01/03/2016	Neuf - En stock
Provence-Alpes-Côte-d'Azur	La Côte d'Azur	Hervé Champollion	Ouest-France	4.50	19/02/2010	Neuf - En stock
Rhône-Alpes	Drôme des collines	Collectif	CREN	6.00	03/04/2015	Neuf - En stock