

goo.gl/qik6dD

Web scraping et **APIs** avec **R**

à l'*Usage*

des *Sciences*

Sociales

François Briatte

Séminaire RUSS

Mai 2017

Web scraping

Feature	Statistics	Machine Learning
Theoretical basis	Probability theory	“Hey, I wonder if this will work??”
Measurement	Should be careful and consistent	Whatever
Foundational result	Central limit theorem	Web scraping
Distribution naming conventions	Long dead European males	Distributions?

Source: [Phil Schrod](#)

API (Application programming interface)

FOAAS

Fuck Off As A Service

v1.1.0

FOAAS (Fuck Off As A Service) provides a modern, RESTful, scalable solution to the common problem of telling people to fuck off.

Hors d'oeuvre

Récupération de données d'enquêtes

Deux projets d'Anthony Damico à connaître :

- [Analyze Survey Data for Free](#) (avec R + [MonetDBLite](#))
- [Iodown](#) · *locally download and prepare publicly-available microdata*

Un des meilleurs exemples
d'utilisation de R pour accéder
à des données d'enquêtes.
Tout est déjà codé :-)



Menu

Merci de poser vos questions
au fil de la présentation

La deuxième partie requiert **R**
+ une connexion Internet

Contexte

Stratégies

Outils

Syntaxes

25 slides

Interfaces

Scraping

Références

25 slides
€ *exemples*

Contexte

- **Augmentation de la disponibilité des données** dans des formats facilement téléchargeables et manipulables
- **Volontés institutionnelles de faciliter la réutilisation** de certaines données :  / gov. / science – ex. **OPEN** (Open, Public, Electronic and Necessary) **Government Data Act, USA**
- **Avancées dans les formats, licences et standards** d'exploitation de fichiers – ex. **CSVY, NDJSON, CC, ODbL**
- **Montée en capacité des infrastructures et outils Web** – (connexions, serveurs, interfaces interactives, etc.)

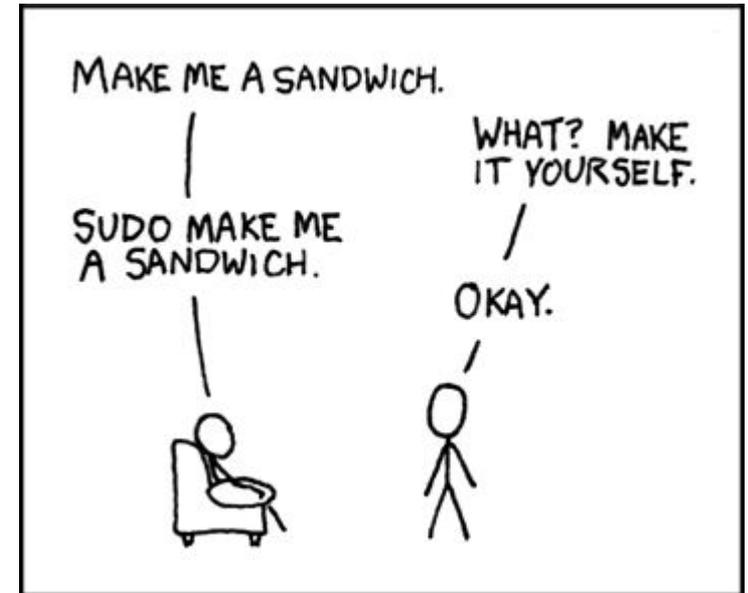
Contexte

Approximately 10% of [the 206 new packages that arrived on CRAN in January] have to do with **providing access to data** in by some means or another. Some packages contain the data sets, some provide **wrappers to APIs**, and at least one package provides **code to scrape** data from a site.

Joseph Rickert, January 2017

Contexte

- **Faible compétence collective**
en matière de collecte de données sur Internet
- **Avantage comparatif immédiat**
très rapidement rentabilisable dans les milieux scientifiques
- **Problème connexe** : dépendance aux personnels compétents
- **Problème principal** : forte volatilité des données numériques



[source](#) · [trope](#)

Contexte

Scraping : applications **scientifiques**, applications **militantes**

- **Secteur public et corruption** : en Roumanie, l'Agence Nationale de l'Intégrité publie « 7 millions » de déclarations d'intérêts – mais **les publie-t-elle vraiment toutes ?**
- **Secteur privé et illégalité** : aux États-Unis, **AirBnb** publie ses bases utilisateurs – mais sont-elles complètes ?

Murray Cox and **Tom Slee**, who had independently been scraping **Airbnb** data before the company publicly released what they said was all of their data, has evidence that **the home-sharing company tried to cover up potentially illegal postings**. Be wary of (data) sharing that seems too good to be true.

Contexte

UCLA iSchool faculty **scraped** **climate science data** from federal websites, fearing it would disappear under Trump's administration. A similar event is taking place at **NYU** on February 4th and both were inspired by a hackathon at the **University of Toronto** to preserve US environmental data last December.

NYU Data Science Community Newsletter #87, Jan 27, 2017

Los Alamos National Lab **released** 16 years of GPS solar weather data thanks to the Obama administration. Let's agree to **scrape** this from data.gov (search "GPS energetic particles" and store it locally.

NYU Data Science Community Newsletter #88, Feb 3, 2017

John Rozsa, a graduate student at **Eastern Michigan University**, built **EPA Data Dump** to **house** all of the EPA data he **scraped** from the federal website to ensure researchers will have access, even if the agency is terminated. Thank you, John. Let us know if you need help.

NYU Data Science Community Newsletter #91, Feb 24, 2017

Stratégies

Entrée

Sortie

Données structurées
offline · CSV, SQL

Interfaces graphiques
offline · Excel, R, SAS

Données structurées
online · HTML, JSON

Interfaces graphiques
online · JS, PHP, [Shiny](#)

Stratégies

Entrée

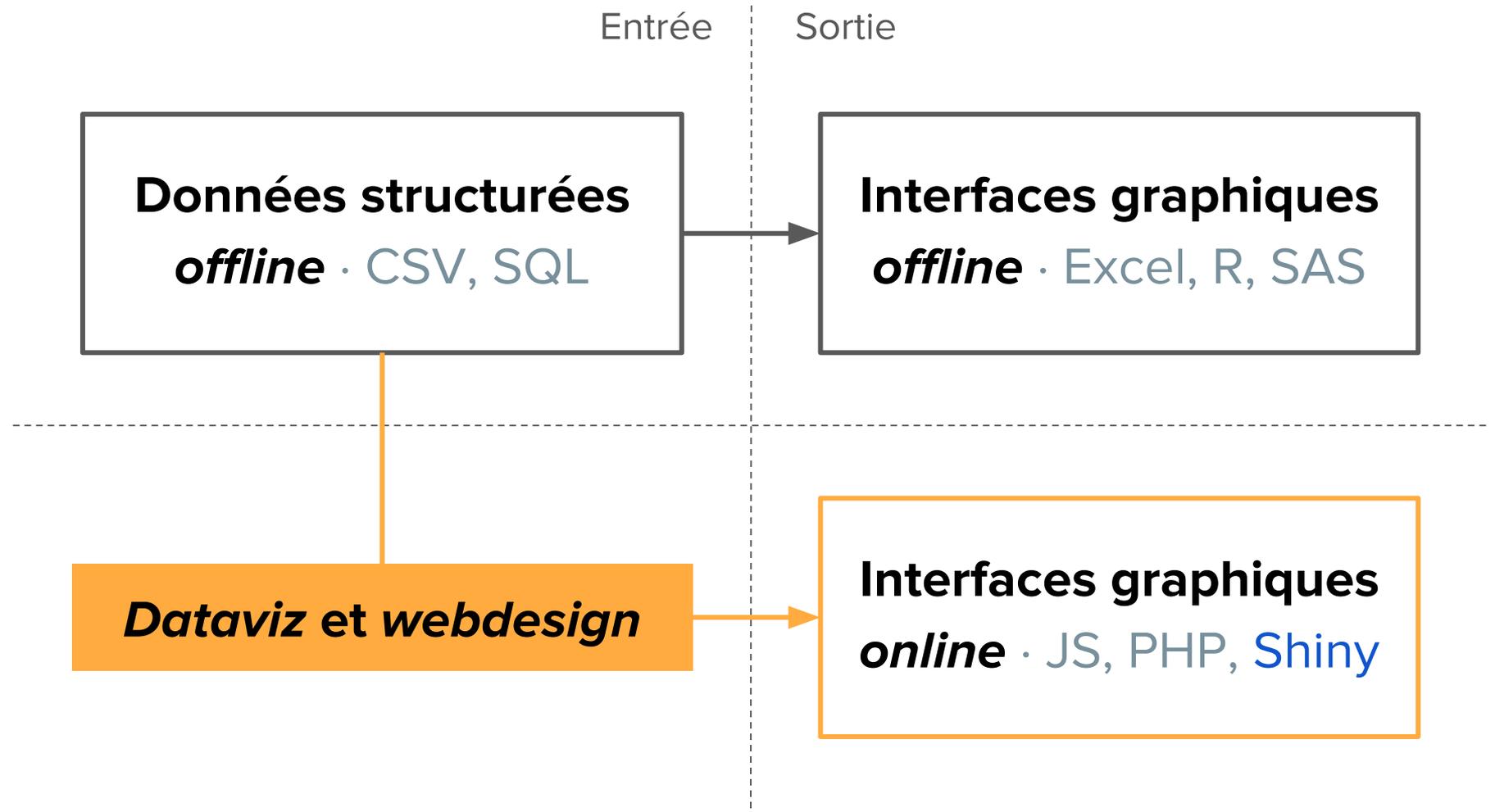
Sortie

Données structurées
offline · CSV, SQL

Interfaces graphiques
offline · Excel, R, SAS

Dataviz et webdesign

Interfaces graphiques
online · JS, PHP, *Shiny*



Stratégies

Entrée

Sortie

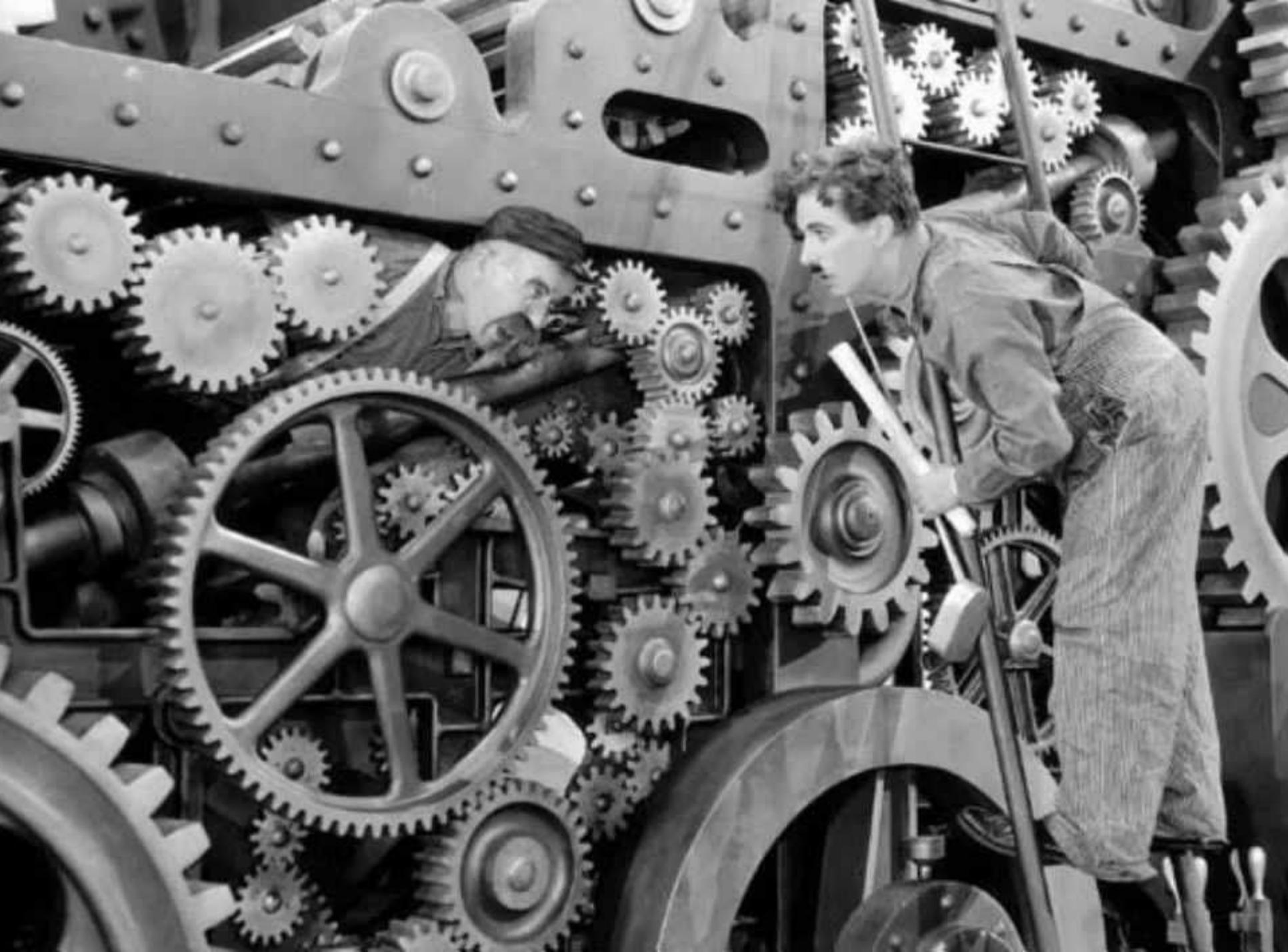
**Données structurées
offline** · CSV, SQL

**Interfaces graphiques
offline** · Excel, R, SAS

**Données structurées
online** · HTML, JSON

Web scraping et APIs

*scraping ≠ crawling
scraping ≠ “scrapping”*



Outils

- **Navigateurs Web** : **Developer Tools** – source HTML et éléments, accès au **DOM**, console JavaScript
- **Utilitaires** : **curl** + **wget** – téléchargement, tests de connexion sous différents protocoles ou **user agents**
- **Syntaxes** : **HTML** + **CSS**, **XPath** (XML), **CSSSelect**, expressions régulières, formats de fichiers, langages *server-side*, langages de requête (*cf.* section suivante)
- **Packages R** : **httr**, **rvest** + **xml2**, **stringr**, lecteurs de formats (ex. **jsonlite**, **readr**), *wrappers* d'APIs

Developer Tools (Chrome, Firefox, etc.)

The image shows a web browser window with the URL `https://statapp.site.ined.fr/fr/les-rencontres-passees/vendredi-14-fevrier-2014`. The page content includes the INED logo, navigation tabs for 'LES RENCONTRES EN 2016', 'LES RENCONTRES PASSES', and 'INFOS PRATIQUES'. The main heading is 'Traitement des données manquantes' with a sub-heading 'à l'Ined, salle Sauvy de 14h à 17h'. A 'TÉLÉCHARGEMENT' section offers PDF downloads for a 'Programme' (194,2 Ko), 'Résumés' (197,7 Ko), and 'Bibliographie' (280,1 Ko). A list of speakers is shown, with 'Guillaume CHAUVET' selected. The browser's developer tools are open, displaying the DOM tree with the selected `strong` element highlighted. The 'Styles' pane shows the default `strong` styling, and the 'Elements' pane shows the corresponding HTML structure.

Vendredi 14 février 2014 - In X

Secure `https://statapp.site.ined.fr/fr/les-rencontres-passees/vendredi-14-fevrier-2014`

ined INSTITUT NATIONAL D'ÉTUDES DÉMOGRAPHIQUES

LES RENCONTRES DE STATISTIQUE APPLIQUÉE FR | EN

PRÉSENTATION LES RENCONTRES EN 2016 LES RENCONTRES PASSES INFOS PRATIQUES

Les rencontres passées Vendredi 14 février 2014

Traitement des données manquantes

à l'Ined, salle Sauvy de 14h à 17h

Imputation, MIVQUE et préservation des relations entre variables Les données manquantes représentent souvent un cauchemar pour tout statisticien confronté à une analyse de ses données. La non-réponse dans les enquêtes peut en effet affecter significativement les estimateurs calculés. Après avoir défini différents types de non-réponse rencontrés dans les fichiers d'enquêtes, cette séance se propose de présenter dans un premier exposé quelques solutions à mettre en œuvre pour la correction de non-réponse. Le cas des données manquantes en analyse de données sera ensuite présenté dans le cas de variables catégorielles, par adaptation de l'algorithme NIPALS. Enfin, un dernier exposé présentera la mise en œuvre d'une méthode d'imputation jointe, par régression aléatoire, visant à préserver les relations entre variables tout en évitant la variance dite d'in

TÉLÉCHARGEMENT

- Programme [PDF | 194,2 Ko]
- Résumés [PDF | 197,7 Ko]
- Bibliographie [PDF | 280,1 Ko]

Guillaume CHAUVET (Ensaï (Crest))Exposé introductif : Méthodes de correction de la non-réponse dans les enquêtes

Christian DERQUENNE (EDF R&D - Département OSIRIS) : Données manquantes et algorithme NIPALS : le cas des variables catégorielles

Brigitte GELEIN (Ensaï) : Imputation, MIVQUE et préservation des relations entre variables

© INED 2017

Elements Console Sources Network Timeline Profiles Application

```
<!DOCTYPE html>
<html lang="fr" class=" js canvas no-touch geolocation hashchange history cssanimations
cssstransitions fontface video audio svg" prefix="og: http://ogp.me/ns#">
  #shadow-root (open)
  <head>...</head>
  <body class="minisite msvertc desktop">
    <div id="largeur" class="wrapper">
      <header id="top" class="row">...</header>
      <nav id="tools">...</nav>
      <nav id="mainnav">...</nav>
      <nav id="mobilemainnav">...</nav>
      <section id="road">...</section>
      <nav id="submain">...</nav>
    <div id="main" class="row">
      ::before
      <div class="row">...</div>
      <div class="row">
        ::before
        <div id="centre" class="ten columns">
          <div id="bloc_centre-haut">...</div>
          <div id="paragraphes" class="paragraphe">
            <a class="cacher" name="para_a-l-ined-salle-sauvy-de-14h-a-17h">...</a>
            <div id="para_nb_1" class="paragraphe_simple">
              <div class="para-titre para-texte">
                <h2>à l'Ined, salle Sauvy de 14h à 17h</h2>
                <div class="rte">
                  <p>...</p>
                  <ul>
                    <li>
                      <strong>Guillaume CHAUVET</strong> == $0
                      " (Ensaï (Crest))Exposé introductif : Méthodes de correction de la non-
                      réponse dans les enquêtes"
                    </li>
                    <li>...</li>
                  </ul>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

html body #largeur #main div #centre #paragraphes #para_nb_1 div div.rte ul li strong

chauvet 1 of 1 Cancel

Styles Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

```
element.style {
}
strong {
  font-weight: 700;
  line-height: inherit;
}
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr, acronym,
address, big, cite, code, del, dfn, em, img, ins, kbd, q, s,
samp, small, strike, strong, sub, sup, tt, var, b, u, i,
```

Console

Syntaxes HTML / CSS / XPath / CSSSelect

```
143 <div id="main" class="row">
144   <div class="row">
145     <div id="bloc_avant-main" ><div class="bloc " id="bloc_avant-main_titre"><div id
class="clearfix"><h1 id="h1" class="misob ">Traitement des données manquantes</h1></di
</div>
146   <div class=
147
148     <di
class="cacher" name
class="paragraphe_s
149     <h2>à l'Ined, s
150     <p>Impu
représentent souver
réponse dans les en
avoir défini diffé
propose de présente
non-réponse. Le cas
variables catégorie
mise en œuvre d'une
relations entre variables tout en évitant la variance dite d'imputation.</p>
151 <ul>
152 <li><strong>Guillaume CHAUVET</strong> (Ensaï (Crest))Exposé introductif : Méthodes de corre
non-réponse dans les enquêtes</li>
153 <li><strong>Christian DERQUENNE</strong> (EDF R&D - Département OSIRIS) : Données manquantes
algorithme NIPALS : le cas des variables catégorielles</li>
154 <li><strong>Brigitte GELEIN</strong> (Ensaï) : Imputation, MIVQUE et préservation des relati
variables</li>
155 </ul>
156   </div>
</div>
```

```
1 library(rvest)
2 library(stringr)
3
4 b <- "https://statapp.site.ined.fr/fr/"
5 p <- str_c(b, "les-rencontres-passees/vendredi-14-fevrier-2014")
6 read_html(p) %>%
7   html_nodes("#main strong") %>%
8   html_text

[1] "Guillaume CHAUVET" "Christian DERQUENNE" "Brigitte GELEIN"
```

Wget pour récupérer un cours complet

```
wget \  
  --directory-prefix=/Users/fr/Desktop \  
  -c \  
  --proxy=off \  
  -Q0 \  
  --passive-ftp \  
  -r \  
  -l6 \  
  --no-parent \  
http://www.stat.cmu.edu/~cshalizi/uADA/16/
```

GET ALL



THE DATA

Outils

Autres aspects à considérer

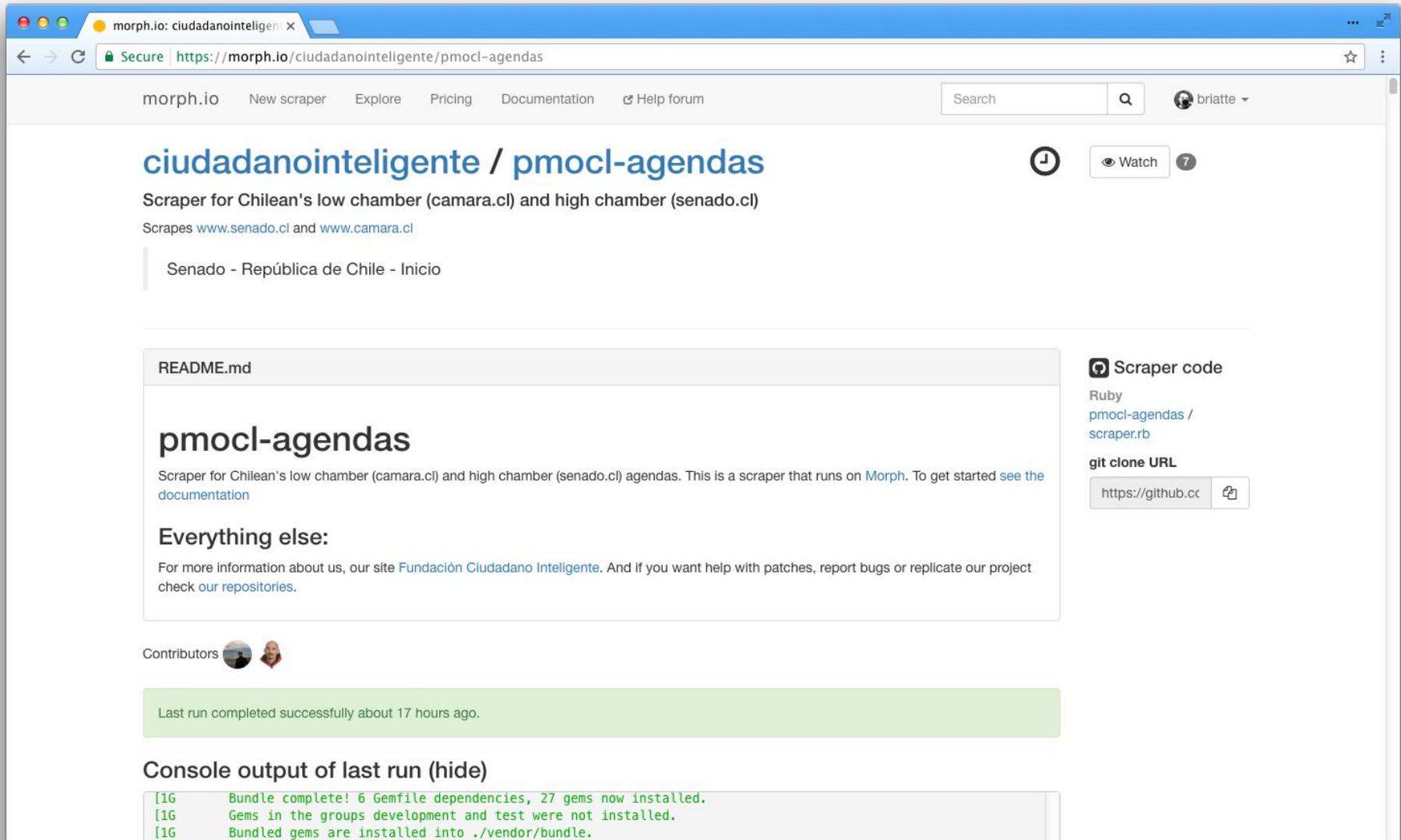
Alternatives

- JavaScript ([CasperJS](#), [PhantomJS](#), [SlimerJS](#), [headless Chrome](#))
- Python ([BeautifulSoup](#), [lxml](#), [Scrapy](#), [Facebook Page Post Scraper](#))
- Ruby ([Nokogiri](#)), Perl ([WWW::Mechanize](#)), PHP, ...

Automation

- [Cron jobs](#)
- [Morph.io](#) · ex-ScraperWiki

Morph.io (Python, Ruby, PHP, Perl, Node.js)



The screenshot shows a web browser window with the URL `https://morph.io/ciudadanointeligente/pmocl-agendas`. The page title is `ciudadanointeligente / pmocl-agendas`. The navigation bar includes links for `morph.io`, `New scraper`, `Explore`, `Pricing`, `Documentation`, and `Help forum`. A search bar and a user profile for `briatte` are also visible.

The main content area displays the scraper's name and description: `Scraper for Chilean's low chamber (camara.cl) and high chamber (senado.cl)`. It lists the scraped URLs: `www.senado.cl` and `www.camara.cl`. Below this, there is a snippet of the scraped data: `Senado - República de Chile - Inicio`.

The `README.md` section contains the following text:

```
pmocl-agendas

Scraper for Chilean's low chamber (camara.cl) and high chamber (senado.cl) agendas. This is a scraper that runs on Morph. To get started see the documentation

Everything else:

For more information about us, our site Fundación Ciudadano Inteligente. And if you want help with patches, report bugs or replicate our project check our repositories.
```

On the right side, there is a `Scraper code` section with the following information:

- Language: `Ruby`
- Repository: `pmocl-agendas / scraper.rb`
- Git clone URL: `https://github.cc`

Below the README, there is a `Contributors` section with two profile pictures. A green notification bar states: `Last run completed successfully about 17 hours ago.`

The `Console output of last run (hide)` section shows the following output:

```
[1G Bundle complete! 6 Gemfile dependencies, 27 gems now installed.
[1G Gems in the groups development and test were not installed.
[1G Bundled gems are installed into ./vendor/bundle.
```

Outils

Autres aspects à considérer

Hardware

- Connexion Internet robuste + [VPN](#) (parfois utile...)
- Mémoires (cache, disque) rapides + laisser du *lag* serveur

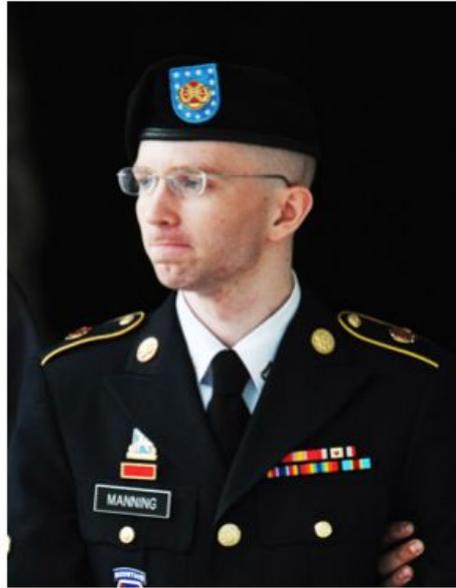
Légalité

- Conditions d'utilisation des sites Internet
- Conditions de redistribution des données
- Données personnelles (ex. adresses email, photos)

WHEN PROGRAMMERS SCRAPE BY

By Rusty Foster February 10, 2014

“Scraping” is not a word with a lot of positive connotations. In “A Christmas Carol,” Charles Dickens describes Ebenezer Scrooge as “a squeezing, wrenching, grasping, scraping, clutching, covetous old sinner.” We feel the impropriety of Scrooge’s scraping all the more keenly for his wealth and power, since scraping is most often an act of the weak or desperate; we talk of “scraping by” or “scraping the bottom of the barrel,” and of “bowing and scraping” to someone more powerful. It’s also recently become news: Aaron Swartz, Chelsea Manning, and Edward Snowden have all found themselves in trouble during the past few years for something called “scraping.”



Unnamed intelligence officials, describing how Edward Snowden collected his archive of N.S.A. files, told the *Times*, “We do not believe this was an individual sitting at a machine and downloading this much material in sequence” and that the process was “quite automated.” The officials didn’t specify what software Snowden used, but “said it functioned like Googlebot, a widely used web crawler that Google developed to find and index new pages on the web.” The difference between a “crawler” and a “scraper” is subtle, but typically a crawler is smarter about the links it follows, what it downloads, and what it leaves uncopied. For the most part, though, “crawling” is just scraping with a fancier name, and Google created one of the world’s most valuable companies in part by being better at scraping than anyone else. Google was incorporated in 1998, and by 2002 its Web-scraping “Googlebots” were so ubiquitous and voracious that, in a short story titled “Robot Exclusion Protocol,” the programmer and writer Paul Ford imagined one trying to index his bathroom. Some have suggested that Google’s recent acquisition of the smart-device maker Nest Labs is effectively an effort to scrape real-world data about our homes and lives, to add to the company’s trove of information about us, which now includes information about the Web pages we visit, our e-mails, the books we read, our shopping habits, and more.

Someone scraped 40,000 Tinder selfies to make a facial dataset for AI experiments

40 000 photos de profil prélevées sur Tinder publiées en ligne

Un utilisateur a automatiquement « aspiré » ces photos afin d'en faire une base de données servant à développer des programmes d'intelligence artificielle.

 scoliann / [TinderFaceScraper](#)

 Watch ▾ 21

 Code

 Issues 4

 Pull requests 2

 Projects 0

 Wiki

 Pulse

 Graphs

Open class-action lawsuit against this person #9

 Open

vilu85 opened this issue 5 days ago · 1 comment

Syntaxes

- **HTML** + **CSS** : référentiels **W3C** et **DOM**
- **XPath** : exploration d'arborescences **XML**
- **CSSSelect** : conversion de sélecteurs CSS vers XPath

Et aussi

- **Expressions régulières** : sélection de chaînes de texte
- **Formats de fichiers** (ex. **JSON**, RDF, XML)
- **Langages *server-side*** : générateurs DHTML (ex. **ASP**, PHP)
- **Langages de requêtes** (ex. SQL, **SPARQL**)

Questions ?



Scraping : récupérer les données

Approche manuelle

- **Lecture des sources HTML** (avec Developer Tools)
- **Requêtes HTTP GET** (via rvest ou via httr si [cookies](#))
- **Boucles de téléchargement** (avec gestion des erreurs)

Autres approches

- *Scrapers* interactifs (ex. [Scraper](#) pour Google Chrome)
- *Headless browsers* (ex. [PhantomJS](#), [Selenium](#), [Chrome](#))

Spécification W3C (draft) : [WebDriver](#)

Packages R : [seleniumPipes](#), [RSelenium](#), [webdriver](#) (draft)

↑
(en cours)

Scraping : reconstituer les données

Lecture

- **Parsing** – réencodage (UTF-8), extraction des champs
- **Nettoyage** – conversions, corrections, standardisations

Stockage

- **CSV** (fichiers intermédiaires, interopérables)
- **RDS** (objets finalisés, compressés)
- **Base de données**



Drivers *monetdb* : [MonetDB.R](#), [MonetDBLite](#) (présentation)

Drivers **SQL** : RMySQL, RPostgreSQL, RSQLite · interfaçables avec [dplyr](#)

Scraping : exemple pour séminaires Ined

Récupérer les noms des intervenants des séminaires Ined 'RUSS' et 'STATAPP' (rencontres de statistique appliquée)

- **Code R** : [Gist](#) (200 lignes, quelques minutes d'exécution)
- **Dépendances** : [dplyr](#), [genderizeR](#), [readr](#), [rvest](#), [stringr](#)

Protocole

- Récupérer la liste des séances (*seminars*)
- Récupérer la liste des intervenants (*speakers*)
- Imputer le sexe des intervenants et estimer la proportion d'intervenants féminins

Scraping : exemple pour vulcanologues

```
# Récupération d'éruptions volcaniques

# Source : Bob Rudis, http://stackoverflow.com/a/42100715

library(httr) # GET, POST
library(rvest) # html_*
library(purrr) # map
library(dplyr) # select

# émulation du formulaire de la page...

x <- POST("http://volcano.si.edu/search\_eruption\_results.cfm",
  body = list(bp = "", `eruption_category[]` = "",
             `country[]` = "", polygon = "", cp = "1"),
  encode = "form")
```

Scraping : exemple pour vulcanologues

```
# Récupération d'éruptions volcaniques
# Source : Bob Rudis, http://stackoverflow.com/a/42100715
# ... et extraction de la table de résultats :
content(x, as = "parsed") %>%
  html_nodes("div.DivTableSearch") %>%
  html_nodes("div.tr") %>%
  map(html_children) %>%
  map(html_text) %>%
  map(as.list) %>%
  map_df(setNames, c("volcano_name", "subregion", "eruption_type",
                    "start_date", "max_vei", "X1")) %>%
  select(-X1)
```

Interfaces

Définition

- **API** : *Application Programming Interface* (cf. *Awesome API*)
- **Entrée** (ex. URL, SPARQL) → **Sortie** (ex. CSV, JSON, XML)
- Avec ou sans **authentification** (ex. *OAuth* · cf. *OAuth Bible*)

Exemples

- *Banque Mondiale* (ouverte ; sorties JSON et XML)
- *Regards Citoyens* (ouverte ; sorties CSV, JSON et XML)
- *Twitter* (via *OAuth* ; **REST** et Streaming ; sorties JSON)

Interfaces

Catalogues

- News API

voir aussi les packages R de journaux
ex. [GuardianR](#) / [rdian](#), [nytimes](#)

- Public APIs

exemple bizarre : [nazihuntR](#) – *lolwat*

Packages R

- Web Technologies and Services (CRAN Task View)

Ex. [oecdR](#), [WDI](#) (World Development Indicators)

Ex. [Rfacebook](#), [Rlinkedin](#), [twitterR](#) / [RTwitterAPI](#) / [streamR](#) / [rtweet](#)

Ex. [rwikidata](#), [WikidataQueryServiceR](#), [WikiDataR](#), [WikipediR](#)

- SPARQL (pour communiquer avec un *endpoint* SPARQL)

Exemple de conditions d'utilisation

1. Limitation of Liability for API Use

2. No Abuse or Overuse of the API

Abuse or excessively frequent requests to GitHub via the API may result in the temporary or permanent suspension of your account's access to the API. GitHub, in our sole discretion, will determine abuse or excessive usage of the API. We will make a reasonable attempt to warn you via email prior to suspension.

It is prohibited to use the API to download data or Content from GitHub for spamming purposes, including for the purposes of selling GitHub users' personal information, such as to recruiters, headhunters, and job boards.

All use of the GitHub API is subject to these Terms of Service and the [GitHub Privacy Statement](#).

GitHub may offer subscription-based access to our API for those Users who require high-throughput access or access that would result in resale of GitHub's Service.

3. GitHub May Terminate Your Use of the API



RTFM

Interfaces : exemple GET / R

```
# Imputation du sexe à partir du prénom
# API : genderize.io

# Avec genderizeR
library(genderizeR) # imports magrittr
findGivenNames("benoit")

# Avec httr
library(httr)
GET("https://api.genderize.io/?name=benoit") %>%
  content() # formatted from JSON
```

Interfaces : exemple OAuth / R

Récupération de tweets à partir de mots-clés · [détails](#)

Étape 1 : [créer une application](#) sur Twitter Apps

[Details](#)

[Settings](#)

Keys and Access Tokens

[Permissions](#)

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)

Consumer Secret (API Secret)

Attention à ne pas les publier sur GitHub !

Access Level

Read-only ([modify app permissions](#))

Owner

[REDACTED]

Owner ID

[REDACTED]

Interfaces : exemple OAuth / R

Étape 2 : créer un objet d'authentification

```
library(ROAuth)

requestURL      <- "https://api.twitter.com/oauth/request_token"
accessURL      <- "https://api.twitter.com/oauth/access_token"
authURL        <- "https://api.twitter.com/oauth/authorize"
consumerKey    <- "#####" # add your consumer key here!
consumerSecret <- "#####" # add your consumer secret here!

oauth <- OAuthFactory$new(consumerKey      = consumerKey,
                          consumerSecret  = consumerSecret,
                          requestURL      = requestURL,
                          accessURL      = accessURL,
                          authURL        = authURL)
```

Interfaces : exemple OAuth / R

Étape 3 : lancer le processus d'authentification

```
# locate CAcert.org trusted certificates
cacert <- system.file("Cur1SSL", "cacert.pem", package = "RCurl")

# executing this line will open a browser window
oauth$handshake(cainfo = cacert)
```

You've granted access to [redacted]!

Next, return to [redacted] and enter this PIN to complete the authorization process:

Code PIN à copier dans R

```
# save the 'certified' OAuth object
saveRDS(oauth, file = "oauth.rds")
```

Interfaces : exemple OAuth / R

Étape 4 : récupérer le [Public Stream](#) en morceaux (*chunks*)

```
library(streamR)

a <- readRDS("oauth.rds") # credentials
q <- c("#hashtags", "and", "other", "#search", "keywords")

for (i in (15 * 24 * 2):1) { # run for 15 days
  f <- paste0("tweets", sprintf("%03.0f", i), ".json")
  # capture matching public tweets for 25 minutes
  try(filterStream(f, track = q, timeout = 60 * 25, oauth = a))
  # sleep for 5 minutes to avoid choking the API
  Sys.sleep(60 * 5)
}
```

Interfaces vs. Scraping

Pour récupérer des données Wikipédia :

Approche faiblement structurée – récupération du code HTML (souvent irrégulier) des pages-source :

Ex. [Extracting notable deaths from Wikipedia](#)

Ex. [The U.S. has been at war 222 out of 239 years](#)

Inconvénient

$\Pr (\text{code breaks}) \gg 0$

Interfaces vs. Scraping

Pour récupérer des données Wikipédia :

Approche fortement structurée – utiliser [Wikidata](#) et son *endpoint* [SPARQL](#) pour récupérer toutes les entités :

Ex. [Importation de données Wikidata](#) dans  Gephi

Ex. [Tableaux de Vermeer montrant des cartes](#) (source)

Inconvénient | Syntaxe SPARQL

Interfaces + Scraping

Certains types de données requièrent de pouvoir **combiner *scraping* et appels d'API** :

Ex. Analyse de sentiment sur **données musicales**

- Albums [Last.fm](#) · REST ou XML-RPC, i.e. HTTP + XML
- Albums [Spotify](#) · OAuth2
- Paroles [Genius](#) · OAuth2

Applications (avec R) : [Radiohead](#), [Kanye West](#)

Interfaces + *Scraping*

```
# Extraction du texte principal d'une page HTML
# API : mercury.postlight.com/web-parser

library(httr)      # GET
library(rvest)    # read_html, html_text
library(stringr)  # str_*

j <- GET(str_c("https://mercury.postlight.com/parser?url=",
              "https://www.monde-diplomatique.fr/",
              "2015/12/GOMBIN/54357")),
        add_headers("x-api-key" = "#####")) # API key here!

content(j)$content %>%
  read_html %>%
  html_text %>%
  str_replace_all("\\s", " ") %>%
  writeLines("body.txt")
```

Références

- **Guides introductifs et tutoriels**

[[A Short Introduction to HTML](#)
[Scraping for Craft Beers](#),
[Utilisation d'APIs natives...](#)]

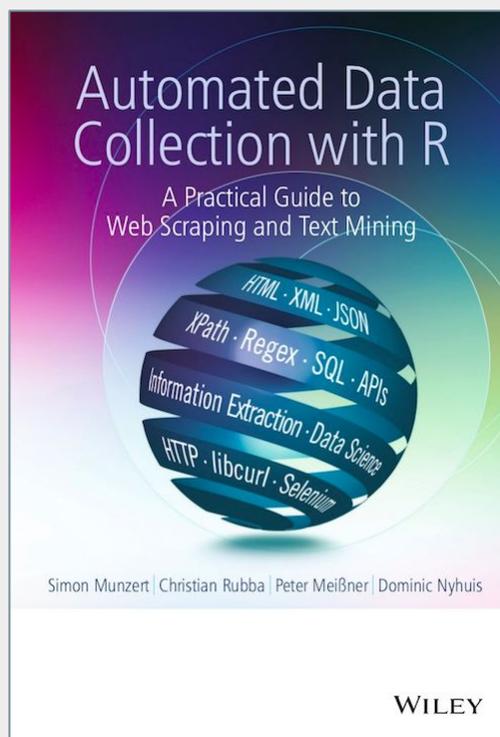
- **Cas pratiques**

- **Manuels**

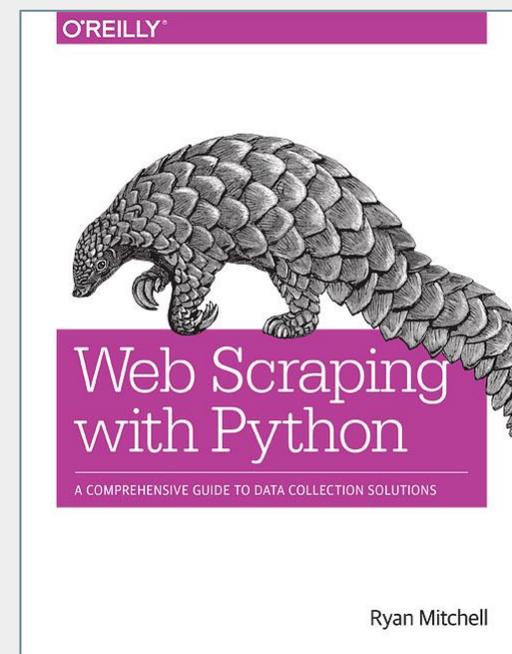
- **Référentiels**

- **StackOverflow**

[[css](#), [httr](#), [phantomjs](#),
[regex](#), [rvest](#), [xpath...](#)]



... et aussi



Lectures supplémentaires

- **Closing APIs and the public scrutiny of very large online platforms** – Bernhard Rieder
- **Global Open Data Index** – Open Knowledge Foundation
- **On Missing Data Sets** – Mimi Onuoha
- **rOpenSci** – “Transforming science through open data”
 - CRAN Task View: **Open Data**
 - CRAN Task View: **Web Technologies and Services**
- **Scraping Web sources: Two illustrations**

À vos heures perdues...

Utilisez les fichiers **RDF/XML** et/ou l'*endpoint* SPARQL de Persée pour réutiliser leurs données bibliographiques :



data.persee.fr/sparql (docs, schemas)

Sparql endpoint

Accédez via ce lien, en pleine page, au SPARQL endpoint standard de Persée, ou utilisez le formulaire simplifié ci-dessous :

Default Data Set Name (Graph IRI)

Query Text

À vos heures perdues...

Demandez à [Baptiste Coulmont](#) s'il a besoin d'un coup de main pour récupérer l'une de ses sources de prénoms :

memoiredeshommes.sga.defense.gouv.fr

PREMIÈRE GUERRE MONDIALE



Base des Morts pour la France de la Première Guerre mondiale

La recherche s'effectue sur un ou plusieurs critères. Aucun champ n'est **obligatoire**.

Vous pouvez accéder à d'autres critères en cliquant sur "**Afficher plus d'options de recherche**" (dans ce cas, réponse non exhaustive, basée sur l'**indexation collaborative**).

[➔ Aide à la recherche](#)

Recherche

Nom	<input type="text"/>		Commence par ▾			
Prénom(s)	<input type="text"/>		Commence par ▾			
Date de naissance	<input type="text"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>	

À vos heures perdues...

Aidez à récolter les investitures aux élections législatives
(code en bash + Python) : [données](#), [script PS](#), [scripts FI et EM](#)

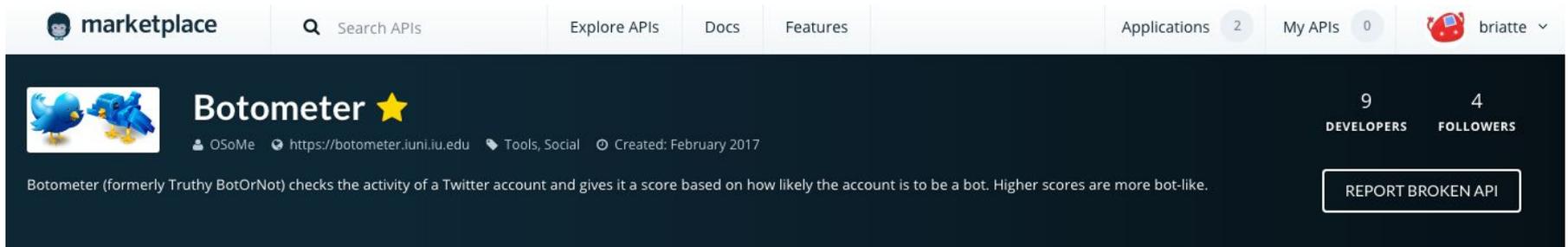
(avec bash : autoupdate ; avec Python : Morph.io)

```
1  #!/bin/bash
2
3  cd $(dirname $0)/..
4
5  bin/download.sh
6
7  if git status | grep "data/" > /dev/null; then
8      if ! grep "[0-9]," data/investitures-deputes-PS-2017.csv > /dev/null ; then
9          echo "WARNING: no result from http://www.parti-socialiste.fr/liste-candidats-aux-legislatives-
10         exit 1
11     fi
12     git commit data -m "autoupdate"
13     git push
14 fi
```

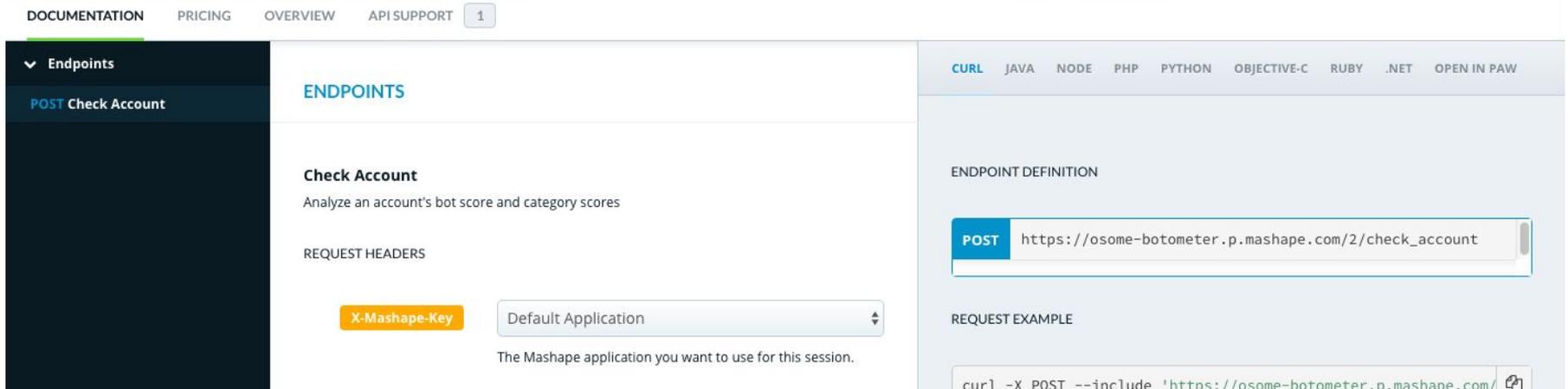
À vos heures perdues...

Apprenez à utiliser [Mashape](#) pour utiliser tout un tas d'APIs (comme ce [détecteur de bots sur Twitter](#))

market.mashape.com/OSoMe/botometer



The screenshot shows the Mashape marketplace interface. At the top, there's a navigation bar with 'marketplace', a search bar, and links for 'Explore APIs', 'Docs', 'Features', 'Applications' (2), and 'My APIs' (0). The user profile 'briatte' is visible in the top right. Below the navigation, the API card for 'Botometer' is displayed. It features a blue bird icon, the name 'Botometer' with a star, and a description: 'Botometer (formerly Truthy BotOrNot) checks the activity of a Twitter account and gives it a score based on how likely the account is to be a bot. Higher scores are more bot-like.' The card also shows '9 DEVELOPERS' and '4 FOLLOWERS', and a 'REPORT BROKEN API' button.



The screenshot shows the 'Endpoints' section of the Botometer API page. The 'Check Account' endpoint is selected, showing its definition: 'POST https://osome-botometer.p.mashape.com/2/check_account'. The page includes a 'REQUEST EXAMPLE' section with a curl command: 'curl -X POST --include 'https://osome-botometer.p.mashape.com/'. The interface also shows a sidebar with 'Endpoints' and 'Check Account' selected, and a top navigation bar with 'DOCUMENTATION', 'PRICING', 'OVERVIEW', and 'API SUPPORT' (1).