

Introduction à OpenRefine

Pour toute question ou commentaire, veuillez m'écrire à julie.faure-lacroix@calculquebec.ca

Vous pouvez le télécharger à cette adresse <http://openrefine.org/> (<http://openrefine.org/>)

Pourquoi utiliser OpenRefine?

1. pour garder une trace de ce qu'on a fait à nos données
2. parce qu'on peut annuler une action facilement
3. parce qu'OpenRefine ne modifie pas le fichier original, il crée une copie
4. parce qu'on peut sauvegarder des routines et le appliquer à d'autres fichiers
5. parce qu'il contient des algorithmes d'aggrégation puissants

Quelques notes sur OpenRefine

1. OpenRefine a été initialement conçu par Google sous le nom de **Google Refine** et lorsque le financement a été épuisé pour le projet, il est devenu un projet ouvert sous le nom d'OpenRefine. Si vous cherchez de l'aide en ligne, les deux noms peuvent vous retourner de l'information utile.
2. OpenRefine est une application Java et **une sorte de Java** a été développé spécialement pour OpenRefine. Il s'agit d'un langage appelé **GREL (General Refine Expression Language)**.
3. Il existe des **groupes en ligne** axés sur l'utilisation d'OpenRefine:

<https://groups.google.com/forum/?hl=en#!forum/openrefine> (<https://groups.google.com/forum/?hl=en#!forum/openrefine>)

LA DOCUMENTATION S'EST BEAUCOUP AMÉLIORÉE mais ça vaut encore la peine de regarder les groupes de discussion

1. OpenRefine est relativement performant jusqu'à une concurrence de **1 million de cellules, 50 Megabytes (MB)s ou 50 colonne par entrée**. Vous pouvez améliorer la performance en allouant plus de mémoire à Openrefine <https://docs.openrefine.org/manual/installing> (<https://docs.openrefine.org/manual/installing>).
2. Comme le projet est *open source*, il existe plusieurs versions modifiées d'OpenRefine qui permette notamment d'utiliser des fichiers de millions de lignes ou de faire les manipulations en parallèle.

Installer OpenRefine

Nous utiliserons OpenRefine sur un serveur distant. Cependant, suivez les instructions suivantes lorsque vous utilisez OpenRefine sur votre ordinateur personnel.

Il n'y a pas vraiment d'application à installer pour utiliser OpenRefine. Cependant, assurez-vous que **Java** est installé et à jour.

Vous devriez avoir téléchargé OpenRefine et extrait les fichiers à un emplacement sur votre ordinateur. Sous **Windows**, vous pouvez vous rendre à l'endroit où les fichiers sont extraits et double-cliquer "openrefine.exe". Sous **Mac et Linux**, utilisez un terminal pour vous rendre à l'endroit où sont placés les fichiers et exécutez

./refine

Sous **Linux et hypothétiquement sous Mac avec Homebrew**, il se pourrait que vous puissiez installer OpenRefine et ensuite l'exécuter dans le terminal avec la commande

openrefine

Lorsqu'OpenRefine est lancé, vous devriez avoir un terminal (peu importe votre système d'exploitation).

```
You have 7863M of free memory.
Your current configuration is set to use 1400M of memory.
OpenRefine can run better when given more memory. Read our FAQ on how to allocate more memory here:
https://github.com/OpenRefine/OpenRefine/wiki/FAQ:-Allocate-More-Memory
Starting OpenRefine at 'http://127.0.0.1:3333/'
09:58:19.125 [      refine_server] Starting Server bound to '127.0.0.1:3333' (0ms)
09:58:19.126 [      refine_server] refine.memory size: 1400M JVM Max heap: 1407188992 (1ms)
09:58:19.139 [      refine_server] Initializing context: '/' from '/opt/openrefine/webapp' (13ms)
09:58:19.519 [      refine] Starting OpenRefine 2.8 [TRUNK]... (380ms)
```

Une page web devrait s'ouvrir. Si ce n'est pas le cas, ouvrez un navigateur et allez à l'adresse <http://127.0.0.1:3333/> (<http://127.0.0.1:3333/>).

Utiliser OpenRefine

Changer la langue

Vous pouvez utiliser OpenRefine en Français (partiellement traduit) en cliquant sur "language settings" sur la page d'accueil et en sélectionnant Français.

Récupération des données

Nous allons maintenant récupérer des données provenant **de la base de données ouvertes de la ville de Montréal**

<http://donnees.ville.montreal.qc.ca/dataset/declarations-exterminations-punaisses-de-lit>
(<http://donnees.ville.montreal.qc.ca/dataset/declarations-exterminations-punaisses-de-lit>)

Ce sont des données de déclarations de punaises de lit sur l'île de Montréal. Notez l'information mentionnée sur concernant le jeu de données:

Déclarations des gestionnaires de parasites. Les formulaires de déclaration ont été soumis depuis le 5 juillet 2011. Les données ont un faible degré de fiabilité car elles sont consignées manuellement par des tiers, soit les gestionnaires de parasites et ne font l'objet d'aucune validation de la part de la Ville de Montréal. [...] Une déclaration unique dans le quartier de référence René-Lévesque durant l'année 2013 indique 901 logements traités, ce qui est vraisemblablement une erreur de frappe, et fait augmenter considérablement le nombre de logements traités

Ce sont donc des données parfaites pour s'exercer! Sur votre ordinateur, vous devriez avoir **extrait le fichier csv sur votre ordinateur**.

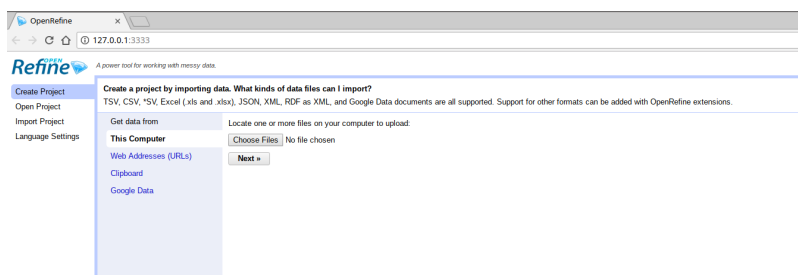
Sur le serveur, nous allons importer les données directement sur site avec l'adresse <https://data.montreal.ca/dataset/49ff9fe4-eb30-4c1a-a30a-fca82d4f5c2f/resource/6173de60-c2da-4d63-bc75-0607cb8dcb74/download/declarations-exterminations-punaisses-de-lit.csv>

(<https://data.montreal.ca/dataset/49ff9fe4-eb30-4c1a-a30a-fca82d4f5c2f/resource/6173de60-c2da-4d63-bc75-0607cb8dcb74/download/declarations-exterminations-punaises-de-lit.csv>).

Création d'un projet

OpenRefine peut importer plusieurs types de fichiers: **tsv (tab separated)**, **csv (comma separated)**, **xls**, **xlsx**, **JSON**, **XML**, **RDF as XML**, **Google Spreadsheets**, etc.

1. Dans l'onglet "**Créer un projet**"
2. Cliquez "**Cet ordinateur**" pour chercher le fichier csv sur votre ordinateur ou "**Adresses web**" pour aller chercher le fichier en ligne.
3. Cliquez "**Suivant**".
4. Vous devriez voir un **aperçu du fichier**.
5. Vous pouvez **changer le format des caractères**, peut être intéressant si vous travaillez dans windows(UTF-8).
6. Jouez dans le bas avec les types de séparation de fichiers pour voir les résultats.
7. Si tout vous semble correct, donnez un nom à **la copie du fichier** et cliquez "**Créer un projet**".
8. Jetez un coup d'oeil aux noms des colonnes.
9. Dans OpenRefine, les modifications de font le plus souvent sur les **colonnes**. On peut aussi voir qu'il y a des **lignes** et chaque information est contenue dans une **cellule**.



Filtres de texte

Chaque colonne a un menu. Vous pouvez commencer par "**Filtrer le texte**" sur la colonne NOM_ARROND. On peut chercher plein de mots.

Format de données

Changer le format des données

Par défaut, OpenRefine considère toutes les cellules comme du texte. Pour faire des manipulations sur les colonnes qui ne sont pas du texte, il faut convertir les cellules au format approprié. Pour ce faire, il vous faut cliquer la flèche à gauche du nom de la colonne, puis: Éditer les cellules/Transformations courantes et sélectionner la transformation désirée. Les cellules apparaîtront en vert lorsqu'elles ne seront plus en format texte.

Facette (facetting)

Cette étape est la plus importante d'OpenRefine. Elle vous permettra d'explorer et de vous familiariser avec les données. C'est particulièrement utile lorsqu'on a d'assez gros jeux de données. Le facetting nous permet d'avoir une **vue d'ensemble de jeux de données complexes** et de commencer à les explorer plus en détail.

Les facettes (**facets**) vous permettent d'agglomérer certaines données et d'effectuer des modifications sur ces groupes de données.

Facette textuelle

Prenons la **première colonne avec le nom des arrondissements**. Si on clique la flèche et **Facette** puis **Facette textuelle**, une boîte s'ouvre avec les différentes écritures des noms d'arrondissements.

Comme les entrées ont été faites par de multiples personnes et avec des accents, c'est un peu le bordel.

Si on **maintient le mot dans la boîte de filtre texte**, on peut avoir une facette juste pour ce mot. Si on enlève la recherche texte, on a toute la colonne arrondissements. On peut avoir le compte pour chaque arrondissement sur le côté et ça nous donne un aperçu rapide.

Par exemple, si on met "Parc" dans la boîte de filtre de NOM_QR, on trouverait qu'il y a des quartiers contenant le mot *Parc* dans Côte-des-neiges-Notre-Dame-de-grâce, Le Plateau-Mont-Royal et Villeray-Saint-Michel-Parc-Extension

Si on changeait un nom directement dans une cellule, on verrait le changement dans facette.

Facette numérique

Dans ce jeu de données, le nombre d'exterminations est le seul réel élément numérique, mais nous pouvons aussi voir comment ça pourrait fonctionner avec le numéro de déclarations ou le numéro du quartier.

Facette chronologique

Il est possible d'organiser les données selon la date plutôt que numériquement. Tout comme les facetts numériques, il faut avoir préalablement changé le format de la colonne pour date pour pouvoir faire la facette.

Attention: Il est possible que la conversion au format date fonctionne de façon aléatoire pour vous. Il semblerait qu'il s'agisse d'un problème avec Java et le fuseau horaire de votre ordinateur. Toujours vérifier que la conversion a produit le résultat escompté!

Autres Facettes

On peut utiliser d'autres facettes comme :

1. **Nuage de points** (montre un graphique)
2. Personnalisées. Exemples par défaut:
 - 2.1. "Mot" éclate un texte et compte les occurrences de mots
 - 2.2. "Doublons" identifie les doublons

2.3. "Longueur du texte" compte le nombre de caractères dans une cellule. Peut être utile pour dénicher des commentaires entrés dans une cellule Oui/Non

2.4. "Par valeur nulle" Peut être intéressant s'il y a énormément d'options et qu'on ne veut se concentrer que sur les valeurs nulles.

Édition

On peut **cliquer directement dans une cellule pour changer manuellement du texte**. Changeons un seul mot et regardons ce qui arrive dans notre facette. On peut aussi cliquer "edit" sur une facette et changer toutes les cellules qui font partie de la facette.

Toujours faire attention de bien respecter le type de cellule lorsqu'on fait une édition manuelle!

On peut aussi le faire par **ligne de code**, surtout lorsque ça implique plusieurs modifications similaires sur un grand nombre de cellules. Par exemple, il est courant de travailler avec des jeux de données sans accent. Ça permet de partager les données avec des gens pour lesquels la locale ne permettrait pas l'interprétation de certains caractères spéciaux.

Pour éditer les cellules à l'aide du code, on doit cliquer sur la flèche à gauche du nom de colonne, puis **Éditer les cellules/Transformer**.

Exercice

Par exemple, avec GREL (le langage d'OpenRefine), l'expression **value.replace("é", "e")** permet de remplacer tous les "é" par des "e" dans toutes les cellules de la colonne sélectionnée. On peut enregistrer la modification et voir ce que ça donne.

On pourrait aussi écrire `replace(value, "é", "e")` et c'est certainement plus intuitif pour quelqu'un qui utilise Excel ou R, mais ce ne serait pas la méthode standard de l'écrire dans GREL, qui parle plus à des utilisateurs de Python par exemple.

On peut ajouter des séries de modifications en ajoutant un point et une autre modification, par exemple:

```
value.replace("é", "e").replace("Ville-Marie", "Ville-Mario").replace("mot_orig",
"nouveau_mot").modification(arguments)
```

Un exemple que j'apprécie personnellement

```
value.toLowerCase().replace("à", "a").replace("â", "a").replace("á",
"a").replace("ä", "a").replace("è", "e").replace("ê", "e").replace("ë", "e").replace("é", "e").re
-","-").replace(" ", "").replace(".", "").replace("-", "-").replace("/", "-").replace("-",
"-").replace("'", "'')
```

Notez la différence entre "-" et "-". Si vous n'arrivez pas à trouver la combinaison de touches pour créer certains caractères spéciaux, vous pouvez copier-coller le texte dans l'aperçu.

Vous pouvez sélectionner **"retransformer X fois maximum"** pour créer des boucles qui vont tenter de retransformer un nombre X de fois. Ex: utile si plusieurs "- - -", "- -", "-"...

Réutiliser des commandes passées

On peut reprendre des commandes déjà exécutées dans l'onglet *historique* de l'interface de GREL. C'est très pratique pour réutiliser des commandes dont on prévoit se servir fréquemment.

Exercice

Réutilisez la commande passée avec la colonne NOM_QR.

Transformations courantes

Supprimer les espaces de début et de fin

Quand on travaille avec du texte, il y a souvent des cellules avec des espaces blancs au début et à la fin. Toujours, toujours le faire!

C'est autre exemple de modification qui peut être faite avec du code et qui coûte rien:

```
value.trim()
```

Annuler/Répéter

Cliquer sur le **Défaire/Refaire** et cliquer sur **l'étape** ou on a commis une erreur. On peut remonter dans le temps jusqu'au point qu'on veut enlever. On peut aussi avancer dans le temps si on est finalement d'accord avec notre choix. On ne peut pas enlever seulement une étape dans le milieu parce qu'il se pourrait que ça ait des répercussions sur les étapes suivantes.

Groupements (*clustering*)

Il est possible dans Openrefine d'identifier les erreurs de frappe et les mots écrits au son. Des algorithmes automatiques sont à votre disposition dans le menu **flèche/grouper et éditer**.

Différentes options s'offrent à vous, qui sont décrites ici

<https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth>

(<https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth>)

1. Collision de clés

Groupe de méthodes les plus **rapides** car la complexité est linéaire. Méthodes qui créent une représentation alternative d'une valeur (ou une *clé*) qui contient seulement le coeur important d'une chaîne de caractères. **Compare ensuite les clés** entre elles (*collision*) pour trouver celles qui sont identiques.

1.1. empreinte

Produit le **moins de faux positifs**. Comprend une série d'étapes qui sont décrites ici:

<https://github.com/OpenRefine/OpenRefine/blob/master/main/src/com/google/refine/clustering/binning/Fingerp>
(<https://github.com/OpenRefine/OpenRefine/blob/master/main/src/com/google/refine/clustering/binning/Fingerp>)

1.2. empreinte N-grammes (Taille des N-grammes)

Similaire à empreinte. Choisir une taille de n-gramme très élevée n'a pas beaucoup d'avantage, mais une valeur de 1 ou 2 peut trouver des combinaisons qu'empreinte ne trouve pas, tout en produisant **plus de faux positifs**.

1.3. metaphone3

Pour l'**anglais**. Utilise la prononciation des mots.

1.4. phonétique de Cologne

Pour l'**allemand**. Utilise la prononciation des mots.

1. Plus proche voisin (kNN)

Permet de définir une valeur de **distance entre les paires de chaînes de caractères**. Si deux chaînes de caractères tombent à l'intérieur de cette distance, elle seront agglomérées. Peut prendre énormément de temps!

2.1. levenshtein (rayon, bloc de caractères)

Mesure le **nombre d'opérations de modification nécessaires pour passer d'une chaîne de caractères à l'autre**. Bon pour les typos, mais mettre une longue distance maximale peut ralentir le calcul considérablement.

2.2. ppm (rayon, bloc de caractères)

Implémentation d'un code permettant de comparer des **chaînes d'ADN**. Méthode produisant généralement beaucoup de faux positifs.

Si vous identifiez un regroupement probable, vous pouvez accepter le remplacement par défaut ou cliquer sur une des façon d'écrire si le remplacement par défaut de correspond pas à celle que vous préférez. S'il y a plusieurs merges similaires possibles, on peut tous les faire d'un coup.

Le visuel sur le côté peut être utile s'il y a beaucoup beaucoup de clusters et permet de naviguer selon les tailles de cluster.

Exercice

Dans la colonne NOM_QR, modifiez 3 cellules "Beaurivage" de cette façon: Borivage Baurivage Borrivage

Toujours dans la boîte des noms d'arrondissements, cliquez sur "**Cluster**". Cluster utilise différents algorithmes de clustering de texte.

Jouez avec les différents modèles et les différentes options. Fingerprint = précis, Phonetica = pas précis, mais magique.

Vous pouvez ensuite cocher **"merge"** et cliquer **"Merge selected and recluster"** pour voir si le changement influence quelque chose, ou bien **"Merge and close"** si vous êtes satisfait des modifications.

Créer/Renommer/Diviser des colonnes

Lorsqu'on fait des modifications drastiques à une colonne, c'est possible de vouloir garder la colonne d'origine intacte pour pouvoir évaluer si on a fait les bonnes modifications.

edit column/add column based on this column.

Il faut changer de nom. On peut même modifier les cellules directement.

Exercice

Sur la colonne `_NOMQR` éditer la colonne/ajouter une colonne en fonction de cette colonne Nommer la nouvelle colonne `_NOMQR2`, cliquer **"Ok"** Sur la colonne `_NOMQR2` éditer la colonne/diviser en plusieurs colonnes, séarateur -, diviser en 2 colonnes

Si on veut remettre les colonnes ensemble on peut utiliser l'option **join columns** ou bien avec du code en faisant éditer la colonne/ajouter une colonne en fonction de cette colonne `_NOMQR2 1` et taper **value + "-" + cells["NOM_QR2 2"].value**

Vous pouvez constater que les lignes contenant des cellules vides sont mal interprétées. Vous pourriez modifier ces cellules et les remplacer par un espace

Si après coup vous vouliez renommer la colonne... **Changer les noms** des colonnes: **flèche/edit column/rename column**

Exercice

Remplacer un "1" par la lettre "I" dans la colonne `_DATEFINTRAIT` Transformer la colonne en date. Identifier la cellule demeurée en texte dans la colonne. Modifier la cellule et sélectionner le format *date*

Une solution plus stable est d'utiliser `value.toDate("format")` pour convertir manuellement si la méthode par défaut ne donne pas les résultats escomptés. Par exemple, `value.toDate("y")` extraira l'année seulement.

On peut diviser les colonnes pour extraire le début de la date de déclaration et enlever l'heure.

edit column/split into several columns

normalement l'idéal c'est d'avoir un diviseur. Ici on peut prendre le "T" et 2 colonnes maximum.

facetter et cliquer seulement sur date

autre facet sur la colonne avec les noms

clusters pour corriger les erreurs

Si on veut remettre les colonnes ensemble

facet by blank sur la colonne 2 et sur la colonne 3

On va voir qu'il y a plein de blanks dans la colonne 3.

sélectionner "False" dans le facet by blank.

On va seulement avoir ceux qui n'ont pas de blank.

value + " " + cells["col2 3"].value dans la colonne 1

Split [0] élément 1, [1] élément 2, [-1] dernier élément

Changer les noms des colonnes: flèche/edit column/rename column

On peut également extraire la date en texte à partir d'une colonne de date en écrivant le code suivant:

```
value.toString("YYYY-MM-dd")
```

Le format peut être variable

Organiser par plusieurs colonnes

Sort: Essayer avec les coordonnées pour montrer que des fois les données sont trop nombreuses pour faire une facette et on veut juste les blanks

Transpose

Vous pouvez transposer les lignes en colonnes et les colonnes en lignes. La principale limitation est que vous ne pouvez pas agréger les données comme les tableaux croisés dynamiques peuvent le faire (MS Excel).

Transposer les cellules au travers des colonnes en lignes: lorsque vous avez plusieurs colonnes qui contiennent des valeurs que vous voulez regrouper en une seule colonne. Ex: regrouper LONGITUDE et LATITUDE en une colonne COORDONNÉE et VALEUR. Transforme de "Wide" en "Long"

Transposer les cellules en colonnes séparées: lorsque vous avez une colonne qui a un patron répété ex: ABCABCABC. Vous sélectionnez le nombre de cellules à étendre en colonnes.

Convertir en liste les colonnes de clé/valeur: Lorsque vous avez une colonne qui contient des noms de catégories et une colonne qui contient les valeurs. Transforme de "Long" en "Wide".

Investiguer/enlever des lignes

Lorsque nous nettoyons les données, il peut arriver que des lignes contiennent des données que nous ne pouvons utiliser. Nous devons nous débarrasser de ses lignes, mais il est possible de le faire en une seule fois en utilisant des étoiles et les drapeaux dans la colonne **"Toutes"**.

Inspectons la colonne NBR_EXTERMINATIONS.

Enlever les cellules vides

Un problème récurrent dans les jeux de données sales est la présence de cellules vides. Sont-elles des erreurs de saisie? Des données manquantes? Une absence de résultat? À vous de décider. Pour ce qui est d'OpenRefine, il prend généralement mal en charge les cellules vides et les NA. Idéalement, il faudrait avoir fait ce traitement avant la manipulation des données dans OpenRefine. Cependant, quelques options s'offrent à vous si ce n'est pas le cas.

Ex: La Côte Saint-Luc, c'est la Côte Saint-Luc. On pourrait si on veut remplacer les cases vides par Côte Saint-Luc pour fins d'analyse si on le désire. Également, il existe des lignes sans début/fin de traitement. À nous de choisir ce que ça veut dire. Finalement il n'y avait pas de contamination? Ils n'ont jamais été traités? L'inspecteur était trop lâche? Allez savoir...

Solution 1: les enlever

Cliquer Facette par valeur nulle

Sélectionner true

Toutes/éditer les lignes/Marquer les lignes

Ces lignes sont maintenant marquées pour déletion

Ou bien

Cliquer Facette par valeur nulle

Sélectionner true

Toutes/editer les lignes/supprimer les lignes correspondantes

On a maintenant un plus petit jeu de données

Solution: remplacer par NA

Les cellules vides qui sont en fait des **NA** sont assez problématiques dans OpenRefine. Idéalement, vous voudriez avoir fait cette étape avant d'utiliser OpenRefine.

Facette textuelle par colonne

editer la facette (blank) pour mettre des NA

La description mentionne une entrée unique en 2013 dans René-Lévesques notant 901 exterminations. Essayez de trouver cette cellule. Qu'en pensez-vous? Qu'est-ce que ça vous indique sur les autres cellules vides? Dans ce contexte, que choisiriez-vous de faire avec les cellules vides?

Inclure/exclure des entrées

Lorsqu'on inclue/exclue des données dans les facettes, ces données ne sont pas enlevées du fichier, mais ne seront pas exportées ou considérées dans les modifications subséquentes.

cliquer sur Ville-Marie pour le faire apparaître en orange

cliquer sur include pour Saint-Léonard et Villeray

On a maintenant un subset des données qu'on pourra exporter quand on aura finit.

Changer l'ordre des colonnes, enlever les colonnes

Flèche à côté de Toutes, éditer les colonnes/Retrier/Supprimer les colonnes

On peut réorganiser ou enlever des colonnes maintenant inutiles.

Bonification des données

fetch URLs

On peut bonifier le jeu de données en ajoutant des colonnes contenant des informations trouvées automatiquement en ligne. Par exemple, nous pourrions vouloir utiliser le nom de l'arrondissement pour trouver une géolocalisation ou une géolocalisation pour trouver le type de bâtiment ou le nom de la rue. Une des options les plus connues pour accéder à des informations de géolocalisation est l'API de Google maps

<https://cloud.google.com/maps-platform/> (<https://cloud.google.com/maps-platform/>)

Cependant, l'API est maintenant gratuite de façon limitée (avec un nombre de crédits offerts à l'inscription), mais il peut en devenir coûteux d'interroger leur API. Une option gratuite est OpenStreetMap

<https://wiki.openstreetmap.org/wiki/API> (<https://wiki.openstreetmap.org/wiki/API>)

Pour l'exercice, on peut vouloir utiliser les données de géolocalisation des inspections pour déterminer l'élévation de l'endroit où l'inspection a eu lieu. Pour ce faire, nous utiliseront une API qui est uniquement dédiée à l'élévation selon la géolocalisation:

<https://elevation-api.io/> (<https://elevation-api.io/>)

cette api nous permet de faire un nombre illimité de requêtes à 1km de précision et ensuite il y a une facturation pour des précisions plus élevées.

Que nous allons interroger à l'aide des données de latitude et longitude dont nous disposons déjà.

il serait peut-être judicieux de faire d'abord un facette numérique sur les coordonnées pour s'assurer qu'il n'y a pas eu d'erreur de frappe et des "points dans l'océan".

sur "latitude", ajouter une colonne en fonction de cette colonne

nommer "query_api"

remplacer "value" par "https://elevation-api.io/api/elevation?points=(" + value + "," + cells.LONGITUDE.value + ")"

ceci nous donne un URL qui nous permettra d'aller chercher les informations que l'API nous permet d'obtenir sur les coordonnées que nous avons sélectionnées, c'est-à-dire l'élévation dans ce cas.

Lorsque c'est fait, nous avons une colonne nommée query_api qui sera la base pour interroger l'API. Pour ce faire,

Sur la colonne "query_api: éditer la colonne/ ajouter une colonne en moissonnant les données depuis les URL d'une colonne

nommer query_result

Délais de récupération: 500 millisecondes

pour faire une pause. Sans contrôle de votre part, Openrefine va se connecter frénétiquement sur l'API pour obtenir des données d'élévation. Il est donc important de limiter le nombre de requêtes par unité de temps.

On laisse ensuite la magie se faire et quelques secondes/minutes/heures plus tard on devrait avoir une nouvelle colonne contenant du texte (presque) indéchiffrable:

```
{"elevations":
[{"lat":45.5563940269788,"lon":-73.6459326267949,"elevation":34.0}], "resolution":"1000m"}
```

Comme nous ne voulons garder que la valeur pour "elevation", soit dans ce cas 34.0, nous devons indiquer à OpenRefine ce qu'il doit faire avec la colonne `_queryresult`.

Sur la colonne "query_result": éditer la colonne/ajouter une colonne en fonction de cette colonne

```
value.parseJson().elevations[0]['elevation']
```

Réconcilier

La réconciliation est utile lorsqu'on sait que certaines bases de données existantes permettent de bonifier les données. Il suffit de choisir une colonne qui contiendrait des identifiants uniques qui pourraient être croisés avec les bases de données et choisir: `réconcilier/Démarrer la réconciliation`

choisir Wikidata reconciliation for openrefine

si vous ne trouvez pas les bases de données que vous désirez, vous pouvez les ajouter à wikidata

choisir le type nécessaire, ici Q578521

Reconcile column "arrond" » Access Service API

Reconcile each cell to an entity of one of these types:

- ☒ borough of Montreal Q578521
- ☐ federal electoral district of Canada Q17202187
- ☐ commune of France Q484170
- ☐ neighborhood Q123705
- ☐ city or town Q27676416
- ☐ Wikinews article Q17633526
- ☐ railway station Q55488
- ☐ federal electoral district in Quebec Q3248048
- ☐ metropolis

Also use relevant details from other columns:

Column	Include? As Property
numero	<input type="checkbox"/>
nom_projet	<input type="checkbox"/>
nom_rue	<input type="checkbox"/>
nb_log	<input checked="" type="checkbox"/>
hlm_familles	<input checked="" type="checkbox"/>
hlm_pa	<input checked="" type="checkbox"/>
hlm_autres	<input checked="" type="checkbox"/>
projetype	<input checked="" type="checkbox"/>
type_prog	<input checked="" type="checkbox"/>
an_orig	<input type="checkbox"/>
an_effect	<input type="checkbox"/>

Reconcile against type:

☐ Reconcile against no particular type

☒ Auto-match candidates with high confidence

Maximum number of candidates to return

Le processus est semi automatisé parce que dans certains cas OpenRefine ne réussira pas à réconcilier les cellules avec la base de données et il faudra le faire à la main. Normalement, c'est assez rapide parce que wikidata permet environ 3 requêtes par seconde. Une fois les données réconciliées, on peut aller chercher les données supplémentaires disponibles via la base de données.

edit column/add columns from reconciled values

et faire des tests. Dans notre cas, on a seulement que "Area" et "country" à aller chercher. Mais on pourrait choisir une base de données différente pour avoir des résultats différents. Il faut aussi considérer qu'on a pas particulièrement un jeu de données adapté à la réconciliation.

Add columns from reconciled column arrond

Add Property

Suggested Properties

- [area](#)
- [China administrative division code](#)
- [contains settlement](#)
- [country](#)
- [dissolved, abolished or demolished](#)
- [employment by economic sector](#)
- [Facebook Places ID](#)
- [GSS code \(2011\)](#)
- [household wealth](#)
- [inception](#)
- [Latvian National Address Register ID](#)
- [located in the administrative territorial entity](#)
- [money supply](#)
- [number of households](#)
- [official name](#)

Preview

arrond	country
Ahuntsic-Cartierville	remove configure
Lachine	Canada
Le Sud-Ouest	Canada
Montréal-Nord	Canada
Rosemont-La Petite-Patrie	Canada
Saint-Laurent	Canada
Saint-Laurent	Canada
Villeray-Saint-Michel-Parc-Extension	Canada
Villeray-Saint-Michel-Parc-Extension	Canada
Villeray-Saint-Michel-Parc-Extension	Canada

exporter

L'icône **Exporter** permet d'accéder à plein de formats, mais un format intéressant est **"Patrons"**

Patrons

Ça crée un **patron en format JSON** qui peut être utilisé pour l'exportation dans des formats qui ne sont pas encore supportés par OpenRefine. Le prefix et suffix sont utiles lorsqu'on veut conserver le template JSON. Sinon, on peut se limiter à mettre du texte. La partie *jsonize* est aussi inutile à moins de vouloir utiliser JSON. Ce template peut être utilisé pour partager les données dans un contexte qui ne requiert pas le jeu de données complet, mais plutôt un format agréable pour la lecture.

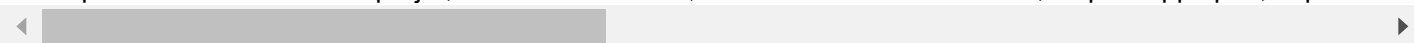
OpenRefine offre un exemple pour exporter en YAML: <https://github.com/OpenRefine/OpenRefine/wiki/Export-As-YAML> (<https://github.com/OpenRefine/OpenRefine/wiki/Export-As-YAML>)

Réutiliser les routines dans le futur

Dans l'onglet **Défaire/Refaire**, on peut cliquer extract et ça va nous donner un fichier JSON qui pourra servir à la documentation ou à la réutilisation. On peut sélectionner les parties qu'on veut garder de notre routine.

copier coller dans un fichier texte (pas de Word!)

et on peut ouvrir un nouveau projet, un nouveau fichier, retourner à Défaire/Refaire, cliquer Appliquer, copier-



In []:

In []:

In []: