epitweetr: documentation à l'usage des utilisateurs

French translation of the following document: **epitweetr: user documentation - Vignette**

This document is a translation provided by ECDC under the EU Initiative on Health Security. The original document was drafted in English and is available here https://www.ecdc.europa.eu/en/publications-data/epitweetr-tool ECDC is not responsible for the accuracy of the translation

Description

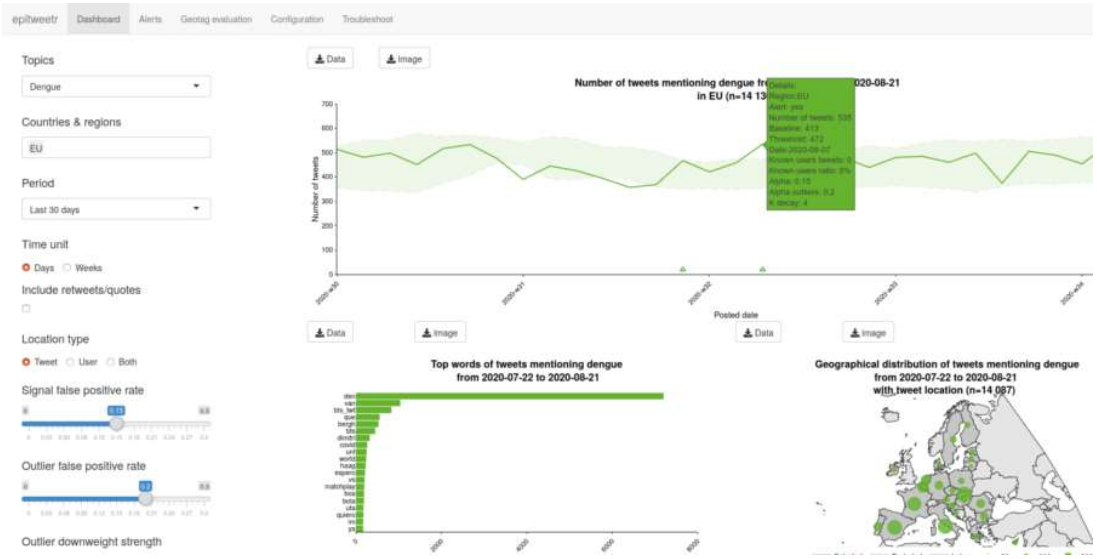
Le paquet `epitweetr` vous permet de suivre automatiquement les tendances des tweets par date, lieu et sujet. Cette surveillance automatisée vise à détecter précocement les menaces pour la santé publique, grâce à certains signaux (augmentation inhabituelle du nombre de tweets à un moment, dans un lieu et sur un sujet précis). Le paquet `epitweetr` a été conçu pour cibler les maladies infectieuses et peut être élargi à tous les dangers ou autres domaines d'étude en modifiant les sujets et les mots-clés.

Le principe général d'`epitweetr` consiste à collecter des tweets et des métadonnées connexes au moyen de l'API de recherche standard sur Twitter (version 1.1) (https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/overview/standard) selon certains sujets et à enregistrer ces tweets dans un format compressé sur votre ordinateur. `epitweetr` géolocalise les tweets et recueille des informations sur les mots importants qui s'y trouvent. Les tweets sont agrégés en fonction du sujet et de la localisation géographique. Ensuite, un algorithme de détection de signaux détermine le nombre de tweets (par sujet et localisation géographique) qui dépasse ce qui est attendu pour une journée donnée. `epitweetr` envoie alors des alertes par courrier électronique aux personnes qui doivent examiner ces signaux de plus près en suivant les processus de veille sanitaire (filtrage, validation, analyse et évaluation préliminaire).

Le paquet se compose d'une application web interactive (Shiny app), qui comporte cinq pages: le «dashboard», tableau de bord où un utilisateur peut visualiser et explorer les tweets (Fig. 1), la page «alerts», où vous pouvez visualiser les alertes en cours et des informations connexes (Fig. 2), la page «geotag evaluation», où vous pouvez évaluer l'algorithme de géolocalisation dans différents champs pour choisir manuellement le seuil de géolocalisation (Fig. 3), la page «configuration», où vous pouvez modifier les paramètres et vérifier le statut des processus sous-jacents (Fig. 4), ainsi que la page «troubleshoot», avec des contrôles automatiques et des suggestions pour utiliser `epitweetr` et toutes ses fonctionnalités (Fig. 5). Sur le «dashboard», l'utilisateur peut voir le nombre de tweets agrégés dans le temps, leur localisation sur une carte et les mots qui y apparaissent le plus fréquemment. Ces visualisations peuvent être filtrées en fonction du sujet, du lieu et de la période qui vous intéressent. D'autres filtres sont disponibles et permettent notamment de régler l'unité utilisée pour la ligne de temps, d'inclure ou non les retweets et les citations, de déterminer les types de géolocalisation qui vous intéressent, la sensibilité de l'intervalle de prévision pour la détection des signaux et le nombre de jours servant à calculer le seuil des signaux. Ces informations peuvent aussi être téléchargées directement à partir de cette interface sous la forme de données, d'images ou de rapports.

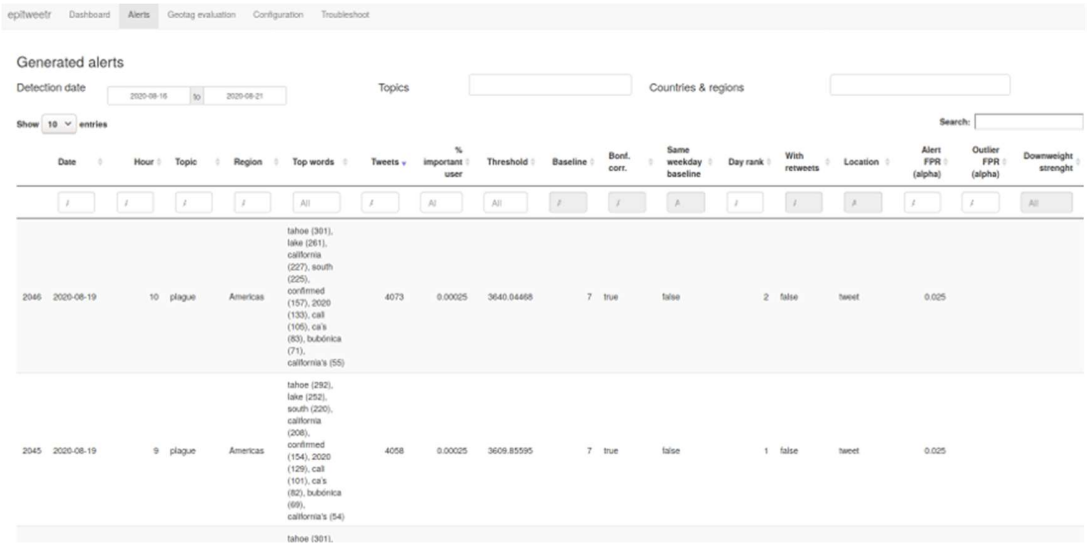
«Dashboard» de Shiny app:

Fig. 1: Capture d'écran du «dashboard» de Shiny app



Page «alerts» de Shiny app:

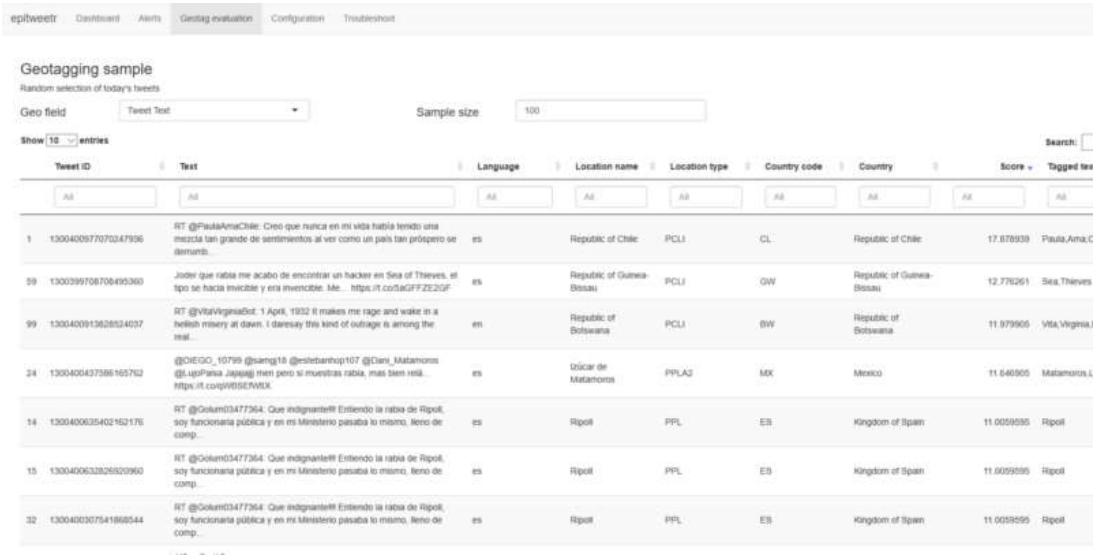
Fig. 2: Page «alerts» de Shiny app



	Date	Hour	Topic	Region	Top words	Tweets	% important user	Threshold	Baseline	Bonf. corr.	Same weekday baseline	Day rank	With retweets	Location	Alert FPR (alpha)	Outlier FPR (alpha)	Downweight strenght
2046	2020-08-19	10	plague	Americas	tahoe (301), lake (261), california (227), south (225), confirmed (157), 2020 (133), cali (105), ca's (83), bubónica (71), california's (55)	4073	0.00025	3640.04468	7	true	false	2	false	tweet	0.025		
2045	2020-08-19	9	plague	Americas	tahoe (292), lake (252), south (220), california (208), confirmed (154), 2020 (129), cali (101), ca's (82), bubónica (69), california's (54)	4058	0.00025	3609.85595	7	true	false	1	false	tweet	0.025		
					tahoe (301),												

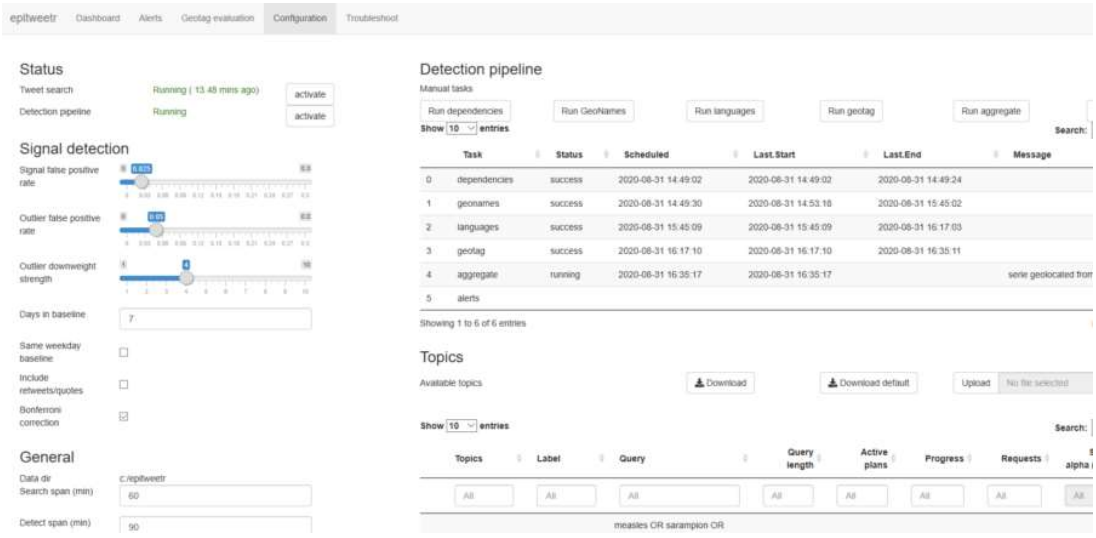
Page «geotag evaluation» de Shiny app:

Fig. 3: Page «geotag evaluation» de Shiny app



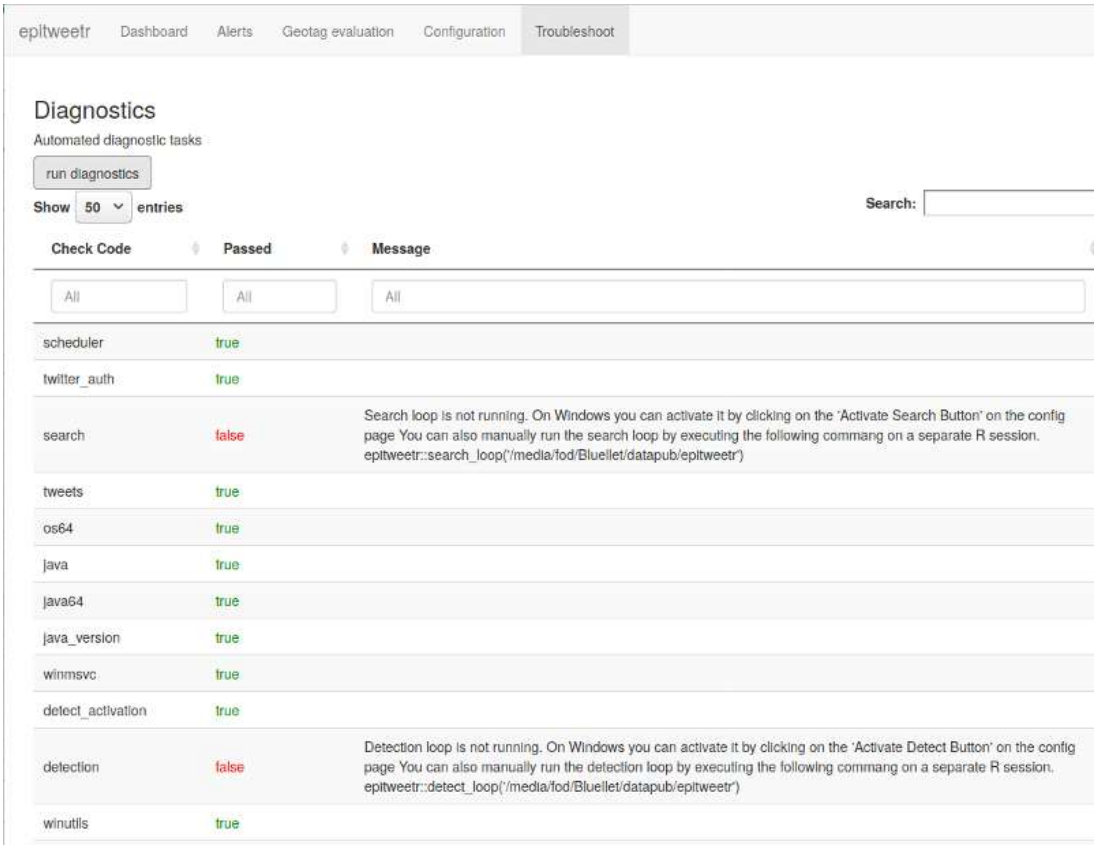
Page «configuration» de Shiny app:

Fig. 4: Page «configuration» de Shiny app



Page «troubleshoot» de Shiny app:

Fig. 5: Page «troubleshoot» de Shiny app



Contexte

Veille sanitaire à l'ECDC

L'article 3 du règlement fondateur du Centre européen de prévention et de contrôle des maladies (ECDC) et la décision nº 1082/2013/UE relative aux menaces transfrontières graves sur la santé font de la détection de menaces pour la santé publique une activité de base de l'ECDC.

L'ECDC mène des activités de veille sanitaire visant à détecter et à évaluer rapidement les menaces pour la santé publique, en concentrant son attention sur les maladies infectieuses, afin d'assurer la sécurité sanitaire de l'UE. Il se sert notamment des médias sociaux comme sources pour détecter précocement les signaux de menaces pour la santé publique. Jusqu'en 2020, la surveillance des médias sociaux passait principalement par le suivi et l'analyse des messages publiés par des organisations ou des experts présélectionnés, essentiellement sur Twitter et Facebook.

Des informations complémentaires et un tutoriel sont disponibles en ligne:

Sources de la veille sanitaire

Suivi des tendances dans les médias sociaux

Certains signaux ne sont pas détectés assez rapidement par les méthodes expliquées ci-dessus, voire ne le sont pas du tout. La surveillance automatisée des métadonnées des médias sociaux (par exemple, l'analyse des tendances) permet de repérer des signaux qui peuvent échapper au suivi de comptes présélectionnés et d'améliorer le temps de détection.

L'analyse des tendances des médias sociaux par sujet, par date et par lieu génère des signaux pertinents pour une détection précoce.

En 2019, l'ECDC a mis au point un prototype d'outil manuel en langage R pour la détection précoce des menaces pour la santé publique sur la base des données de Twitter. Le paquet `epitweetr` est une extension de ce prototype, qui permet une géolocalisation plus large des tweets et une automatisation renforcée.

Objectifs d'epitweetr

L'objectif principal d'`epitweetr` est d'utiliser l'API de recherche standard sur Twitter (version 1.1) pour détecter les signaux précoces de menaces potentielles par sujet et par unité géographique.

Son objectif secondaire est de permettre à l'utilisateur, au moyen d'une interface Shiny interactive, d'explorer les tendances des tweets par date, par localisation géographique et par sujet, en recueillant notamment des informations sur les mots les plus souvent employés et les tweets émanant de sources fiables, à l'aide de graphiques et de tableaux.

Matériel informatique nécessaire

Les configurations matérielles minimales et recommandées pour l'ordinateur sont présentées dans le tableau ci-dessous:

Matériel informatique nécessaire	Minimum	Recommandé
RAM nécessaire	8 Go	16 Go recommandés
CPU nécessaire	4 cœurs	12 cœurs
Espace nécessaire pour 3 ans de stockage	3 To	5 To

L'utilisation du CPU et celle de la RAM peuvent être réglées sur la page «configuration» de Shiny app (voir la section *L'application utilisateur interactive (Shiny app)>La page «configuration»*). Les besoins concernant la RAM, le CPU et l'espace de stockage peuvent dépendre du nombre et de l'étendue des sujets que vous demandez dans le processus de collecte.

Installation

`epitweetr` est conçu comme une plate-forme indépendante, qui tourne sous Windows, Linux et Mac. Nous recommandons d'utiliser `epitweetr` sur un ordinateur qui peut rester allumé constamment. Vous pouvez éteindre l'ordinateur, mais vous risquez de manquer certains tweets si la durée d'indisponibilité est suffisamment longue, ce qui aura des implications pour la détection des alertes. Avant d'utiliser `epitweetr`, les éléments suivants doivent être installés:

Conditions préalables pour faire tourner epitweetr

- R version 3.6.3 ou supérieure
- Java 1.8 eg. OpenJDK version «1.8» *https://www.java.com/download/*. La version 64-bit est préférable à la version 32-bit, pour des raisons de limitations de la mémoire. Sous Mac, il faut également le Java Development Kit *https://docs.oracle.com/javase/9/install/installation-jdk-and-jre-macos.htm*]
- Si vous comptez le faire tourner sous Windows, vous aurez aussi besoin de Microsoft Visual C++ qui; dans la plupart des cas, sera probablement déjà préinstallé:
  - Microsoft Visual C + + 2010 package redistribuable (x64) *https://www.microsoft.com/fr-FR/download/details.aspx?id=14632*

Conditions préalables pour certaines fonctionnalités d'epitweetr

- Pandoc, pour l'exportation de PDF et Markdown
  - *https://pandoc.org/installing.html*
- Installation TeX (TinyTeX ou MiKTeX) (ou une autre installation TeX) pour l'exportation de PDF
  - Le plus facile: *https://yihui.org/tinytex/* – installer à partir de R, déconnexion/reconnexion nécessaire après l'installation
  - *https://miktex.org/download* – installation complète requise, déconnexion/reconnexion nécessaire après l'installation
- Optimisation de l'apprentissage machine (uniquement pour les utilisateurs avancés)
  - Open Blas (optimiseur BLAS), pour accélérer certains des processus de géolocalisation: *https://www.openblas.net/*. Instructions d'installation: *https://github.com/fommil/netlib-Java*
  - **or** Intel MKL (*https://software.intel.com/content/www/us/en/develop/tools/math-kernel-library/choose-download.html*)
- Un planificateur
  - Si vous utilisez Windows, vous devez installer le paquet R: taskscheduleR

- Si vous utilisez Linux, vous devez planifier les tâches manuellement
- Si vous utilisez un Mac, vous devez installer le paquet R: cronR

Autres conditions préalables pour les développeurs R

Si vous souhaitez développer davantage `epitweetr`, les outils de développement suivants sont nécessaires:

- Git (contrôle du code source) *https://git-scm.com/downloads*
- Sbt (compilation du code Scala) *https://www.scala-sbt.org/download.html*
- Si vous utilisez Windows, vous aurez en outre besoin de Rtools: *https://cran.r-project.org/bin/windows/Rtools/*

Dépendances externes

`epitweetr` devra télécharger certaines dépendances pour fonctionner. L'outil le fera automatiquement lors du premier lancement du processus de détection d'alerte. La page «configuration» de Shiny app vous permettra de modifier les URL cibles de ces dépendances, qui sont les suivantes:

- Archives *JAR CRAN*: Dépendances transitives pour faire tourner Spark, Lucene et le code Scala embarqué. *[https://repo1.maven.org/maven2]*
- *Winutils.exe* (Windows uniquement) – Il s'agit d'un binaire Hadoop nécessaire pour exécuter SPARK localement sous Windows *[http://public-repo-1.hortonworks.com/hdp-win-alpha/winutils.exe]*.

Installer `epitweetr` à partir de CRAN

Après avoir installé toutes les dépendances requises énumérées dans la section «Conditions préalables pour faire tourner `epitweetr`», vous pouvez installer `epitweetr`:

```
install.packages(epitweetr)
```

Variables d'environnement

En outre, l'environnement R doit connaître le chemin d'installation de Java. Pour le vérifier, tapez dans la console R:

```
Sys.getenv("JAVA_HOME")
```

Si la valeur renvoyée par la commande est nulle ou vide, vous devrez définir la variable d'environnement Java Home pour votre système d'exploitation: reportez-vous aux instructions propres à votre système d'exploitation. Dans certains cas, `epitweetr` peut fonctionner sans définir la variable d'environnement Java Home.

La première fois que vous exécutez l'application, si l'outil ne trouve pas un gestionnaire de mots de passe sécurisés fourni par le système d'exploitation, vous verrez apparaître une

fenêtre de dialogue demandant un mot de passe de trousseau (Linux et Mac). Il s'agit d'un mot de passe nécessaire pour stocker les identifiants Twitter cryptés. Choisissez un mot de passe fort et mémorisez-le. Il vous sera demandé à chaque fois que vous lancerez l'outil. Vous pouvez l'éviter en définissant une variable d'environnement système nommée ecdc_twitter_tool_kr_password contenant le mot de passe choisi.

Lancer l'application Shiny app d'epitweetr

Vous pouvez lancer l'application Shiny app d'`epitweetr` à partir de la session R en tapant la commande ci-dessous dans la console R. Remplacez «data_dir» par le répertoire de données désigné, qui correspond à un dossier local où vous choisissez de stocker les tweets, les séries chronologiques et les fichiers de configuration:

```
library(epitweetr)
epitweetr_app("data_dir")
```

Veuillez noter que le répertoire de données saisi dans R doit indiquer «/» au lieu de «» (un exemple de chemin correct serait «C:/user/name/Documents»). Cela vaut en particulier pour Windows si vous copiez le chemin d'accès à partir de l'explorateur de fichiers.

Sinon, vous pouvez aussi utiliser un lanceur: dans un fichier .bat exécutable ou un script shell, tapez ce qui suit (en remplaçant «data_dir» par le répertoire de données désigné)

R –vanilla -e epitweetr::epitweetr_app("data_dir")

Vous pouvez vérifier si toutes les éléments requis sont correctement installés sur la page «troubleshoot». De plus amples informations sont disponibles dans la section *L'application utilisateur interactive (Shiny app)>Le «dashboard»: L'interface utilisateur interactive de visualisation>La page «troubleshoot»*

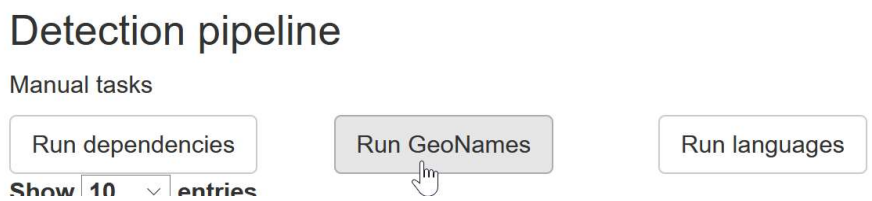
Mettre en place la collecte de tweets et la boucle de détection des alertes

Pour utiliser `epitweetr`, vous devrez collecter les tweets et exécuter la boucle de détection des alertes (geonames, languages, geotag, aggregate et alerts). De plus amples détails sont également disponibles dans les prochaines sections de la documentation. En résumé, les étapes à suivre sont:

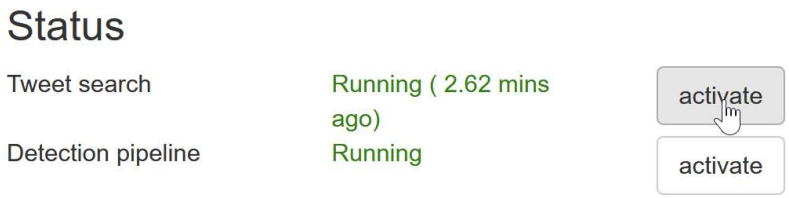
- Lancez Shiny app (à partir de la console R)

```
library(epitweetr)
epitweetr_app("data_dir")
```

- Sur la page «configuration» de Shiny app, dans les tâches manuelles du «Detection pipeline», cliquez sur «Run dependencies», «Run geonames» et «Run languages» (leur statut se changera en «pending»). Cela permet au pipeline de détection de télécharger les éléments nécessaires. Tant qu'aucune langue n'est ajoutée et qu'aucune mise à jour n'est disponible sur geonames.org, ces tâches ne doivent être exécutées que la première fois que vous installez `epitweetr`.



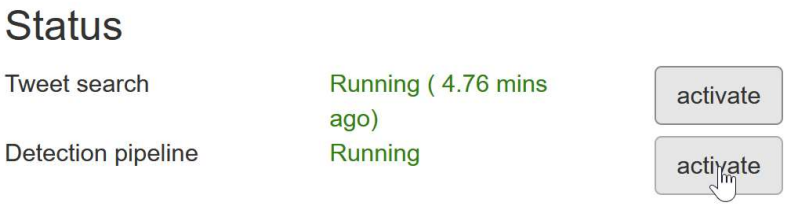
- Configurez l'authentification sur Twitter au moyen d'un compte Twitter ou d'une application de développeur Twitter – voir la section *Collecte de tweets > Authentification sur Twitter* pour plus de précisions.
- Activez la collecte de tweets
  - Windows: Cliquez sur le bouton «activate» en face de «Tweet search»



  - Autres plateformes: Dans une nouvelle session R exécutez la commande suivante

```
library(epitweetr)
search_loop("data_dir")
```

- Vous pouvez avoir la confirmation que la collecte de tweets fonctionne si le statut de «Tweet search» est «Running» sur la page «configuration» de Shiny app (en vert sur la capture d'écran ci-dessus) et «true» sur la page «troubleshoot» de Shiny app.
- Activez le pipeline de détection:
  - Windows: Cliquez sur le bouton «activate» en face de «Detection pipeline»



  - Autres plateformes: Dans une nouvelle session R exécutez la commande suivante

```
library(epitweetr)
detect_loop("data_dir")
```

- Vous pouvez avoir la confirmation que le pipeline de détection fonctionne si le statut de «Detection pipeline» est «Running» sur la page «configuration» de Shiny app et «true» sur la page «troubleshoot» de Shiny app.

- Vous pourrez visualiser les tweets après l'étape d'agrégation dans le tableau «Detection pipeline» sur la page de «configuration» de Shiny app, si «Tweet search» est activé.
- Vous pouvez commencer à travailler sur les signaux générés. **Bonne détection de signaux!**

Pour plus de précisions, consultez la section *Comment ça marche? L'architecture générale sur laquelle repose* `epitweetr`, qui décrit les processus sous-jacents de la collecte de tweets et de la détection de signaux. La section *L'application utilisateur interactive (Shiny app) > La page «configuration»* décrit les différents paramètres de la page «configuration».

Comment ça marche? L'architecture générale sur laquelle repose `epitweetr`

Les sections suivantes décrivent en détail les principes généraux susmentionnés. Les paramètres de bon nombre de ces éléments peuvent être réglés sur la page «configuration» de Shiny app, dont les explications figurent à la section *L'application utilisateur interactive (Shiny app) > La page «configuration»*.

Collecte de tweets

Utilisation de l'API de recherche standard sur Twitter (version 1.1)

`epitweetr` utilise l'API de recherche standard sur Twitter (version 1.1). L'avantage de cette API est qu'il s'agit d'un service fourni par Twitter permettant aux utilisateurs d'`epitweetr` d'accéder gratuitement aux tweets. L'API de recherche n'est pas censée constituer une source exhaustive de tweets. Elle porte sur un échantillon de tweets récemment publiés au cours des 7 derniers jours et privilégie la pertinence plutôt que l'exhaustivité. Il s'ensuit que certains tweets et certains utilisateurs peuvent ne pas apparaître dans les résultats des recherches.

Bien qu'il puisse s'agir d'une limitation dans d'autres domaines de la santé publique ou de la recherche, l'équipe chargée du développement d'`epitweetr` estime qu'aux fins de la détection des signaux, un échantillon de tweets, en combinaison avec d'autres types de sources, est suffisant pour détecter les menaces potentielles importantes.

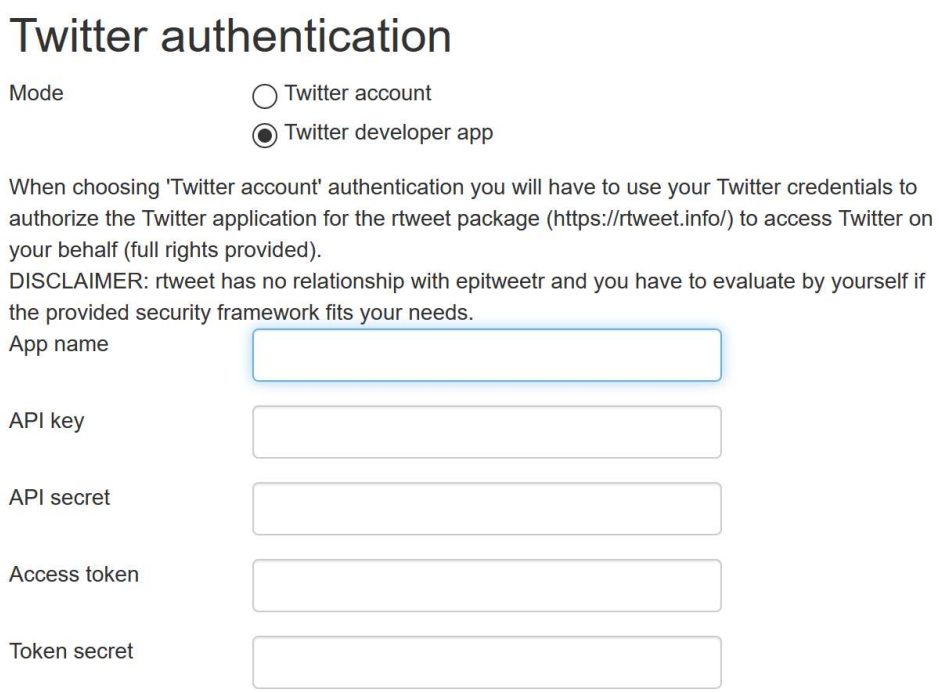
L'API de recherche standard sur Twitter (version 1.1) a d'autres particularités, notamment:

- Seuls les tweets des 5 à 8 derniers jours sont indexés par Twitter
- Elle permet d'effectuer un maximum de 180 recherches toutes les 15 minutes (450 si vous utilisez les identifiants de l'application de développer Twitter; voir section suivante)
- Chaque recherche renvoie au maximum 100 tweets et/ou retweets

Authentification sur Twitter

Vous pouvez authentifier la collecte de tweets au moyen d'un **compte Twitter** (en utilisant le paquet rweet) ou d'une **application Twitter**. Pour cette dernière, vous aurez besoin d'un compte de développeur Twitter, dont l'obtention peut prendre un certain temps, du fait des procédures de vérification. Nous recommandons de recourir à un compte Twitter via le paquet rtweet à des fins de test et à court terme, et d'opter pour l'application de développeur Twitter en cas d'utilisation à long terme.

- **Utilisation d'un compte Twitter:** en passant par rtweet (authentification d'utilisateur)
  - Vous aurez besoin d'un compte Twitter (nom d'utilisateur et mot de passe)
  - Le paquet rtweet enverra une demande à Twitter pour accéder en votre nom à votre compte Twitter.
  - Une fenêtre de dialogue apparaît où vous pouvez saisir votre nom d'utilisateur et votre mot de passe pour confirmer que l'application peut accéder à Twitter en votre nom. Vous enverrez cet identifiant chaque fois que vous accéderez aux tweets.
- **Utilisation d'une application de développeur Twitter:** en passant par `epitweetr` (authentification de l'application)
  - Si vous ne l'avez pas déjà fait, vous devrez créer un compte de développeur Twitter: *https://developer.twitter.com/en/apply-for-access*
  - Créez une application
  - Assurez-vous d'avoir l'accès en lecture et en écriture.
  - Notez vos paramètres OAuth
    - Ajoutez-les à la page «configuration» de Shiny app (voir l'image ci-dessous)
    - Avec ces informations, `epitweetr` peut demander directement un identifiant à Twitter à tout moment. L'avantage de cette méthode est que l'identifiant n'est rattaché à aucune information d'utilisateur et que les tweets sont renvoyés indépendamment de tout contexte d'utilisateur.
    - Cette application vous permet d'effectuer 450 recherches toutes les 15 minutes au lieu des 180 autorisées pour un compte Twitter.

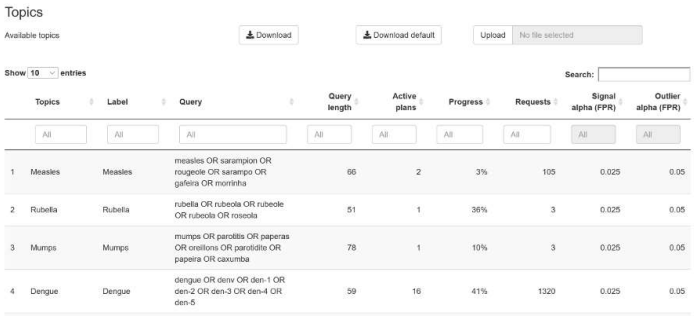


Sujets et requêtes pour la collecte de tweets

Après l'authentification sur Twitter, vous devez spécifier une liste de sujets dans `epitweetr` pour indiquer les tweets à collecter. Chaque sujet correspond à une ou plusieurs requêtes qu'`epitweetr` utilise pour collecter les tweets pertinents (par exemple, plusieurs requêtes utilisant une terminologie et/ou des langues différentes pour un même sujet).

Une requête consiste dans des mots-clés et des opérateurs qui servent à trouver des correspondances avec les attributs des tweets. Un espace séparant des mots-clés suppose une condition AND. Vous pouvez également utiliser l'opérateur OR. Un signe moins devant le mot-clé (sans espace) indique que le mot-clé ne doit pas figurer dans les attributs des tweets. Bien que les requêtes puissent être longues de 512 caractères au maximum, le mieux est d'éviter que votre requête soit trop complexe en la limitant à 10 mots-clés et opérateurs, ce qui veut dire que vous aurez parfois besoin de plusieurs recherches par sujet.

`epitweetr` est fourni avec une liste de sujets par défaut, qui sont ceux utilisés par l'équipe de veille sanitaire de l'ECDC à la date de création du paquet (1er septembre 2020). Vous pouvez voir en détail la liste de sujets sur la page «configuration» de Shiny app (voir la capture d'écran ci-dessous).


Topics

Available topics

Download | Download default | Upload | No file selected

Show 10 entries | Search:

	Topics	Label	Query	Query length	Active plans	Progress	Requests	Signal alpha (FPR)	Outlier alpha (FPR)
	All	All	All	All	All	All	All	All	All
1	Measles	Measles	measles OR sarampion OR rougeole OR sarampo OR gafeira OR morrinha	66	2	3%	105	0.025	0.05
2	Rubella	Rubella	rubella OR rubeola OR rubeole OR rubeola OR roseola	51	1	36%	3	0.025	0.05
3	Mumps	Mumps	mumps OR parotitis OR paperas OR oreillons OR parotidite OR papeira OR caxumba	78	1	10%	3	0.025	0.05
4	Dengue	Dengue	dengue OR denv OR den-1 OR den-2 OR den-3 OR den-4 OR den-5	59	16	41%	1320	0.025	0.05

Sur la page «configuration», vous pouvez aussi télécharger la liste des sujets, la modifier et la charger dans `epitweetr`. La nouvelle liste de sujets sera alors utilisée pour la collecte de tweets et visible dans Shiny app. La liste des sujets est un fichier Excel (*.xlsx), qui permet de traiter des paramètres régionaux propres à l'utilisateur (par exemple, des délimiteurs), ainsi que des caractères spéciaux. Vous pouvez également créer votre propre liste de sujets et la charger, en tenant compte du fait que sa structure doit inclure au moins:

- Le nom du sujet, avec l'en-tête «Topic» dans la feuille de calcul Excel. Ce nom ne peut être composé que par des caractères alphanumériques, des espaces, des tirets et des marques de soulignement. Notez qu'il doit commencer par une lettre.

- La requête, avec l'en-tête «Query» dans la feuille de calcul Excel. Il s'agit de la requête utilisée par `epitweetr` pour collecter des tweets au moyen de l'API de recherche standard sur Twitter. Voir la syntaxe et les contraintes à respecter ci-dessus.

Le fichier topics.xlsx comprend en outre les champs suivants:

- Un ID, avec l'en-tête «#» dans la feuille de calcul Excel, qui indique un entier courant destiné à identifier le sujet.

- Une étiquette, avec l'en-tête «Label» dans la feuille de calcul Excel, qui correspond à ce qu'affiche le menu déroulant des onglets de Shiny app.

- Un paramètre alpha, avec l'en-tête «Signal alpha (FPR)» dans la feuille de calcul Excel. FPR signifie «false positive rate» (taux de faux positifs). Le fait d'augmenter ce paramètre abaissera le seuil de détection des signaux, ce qui se traduira par une sensibilité accrue et éventuellement par la collecte d'un plus grand nombre de signaux. Le réglage peut s'effectuer de manière empirique et en fonction de l'importance et de la nature du sujet.

- La valeur «Length_charact» est un champ généré automatiquement qui calcule le total des caractères utilisés dans la requête. Ce champ est utile dans la mesure où une recherche ne doit pas dépasser 500 caractères.

- La valeur «Length_word» indique le nombre de mots utilisés dans une recherche, opérateurs compris. Le mieux est de limiter à 10 le nombre de vos mots-clés.

- Un paramètre alpha, avec l'en-tête «Outlier alpha (FPR)» dans la feuille de calcul Excel. FPR signifie «false positive rate» (taux de faux positifs). Ce paramètre définit le taux de

faux positifs qui détermine à quoi correspond une valeur aberrante en cas d'ajustement de la pondération des valeurs aberrantes/signaux précédents. Plus il est faible, moins il inclura de valeurs aberrantes précédentes. Un paramètre plus élevée inclura potentiellement davantage de valeurs aberrantes précédentes.

- La valeur «Rank» correspond au nombre de requêtes par sujet.

	A	B	C	D	E	F	G	H	I
1	#	Topic	Label	Alpha	Outliers Alpha	Query	Length_charact	Length_word	rank
2	1	Measles	Measles	0.025	0.05	measles OR sarampion OR rougeole OR sarampo OR gafeira OR morrinha	66	11	1
3	2	Rubella	Rubella	0.025	0.05	rubella OR rubeola OR rubeole OR rubeola OR roseola	51	9	1
4	3	Mumps	Mumps	0.025	0.05	mumps OR parotitis OR paperas OR oreillons OR parotidite OR papeira OR	78	13	1
5	4	Dengue	Dengue	0.025	0.05	dengue OR denv OR den-1 OR den-2 OR den-3 OR den-4 OR den-5	59	13	1
6	5	Haemorrhagic fever	Haemorrhagic fever	0.025	0.05	"hemorrhagic fever" OR "haemorrhagic fever" OR vhf OR "fiebre	129	18	1

Lorsque vous chargez votre propre fichier, vous pouvez modifier les champs des sujets et des requêtes, mais ne modifiez pas les intitulés de colonne.

Plans de collecte des tweets planifiés

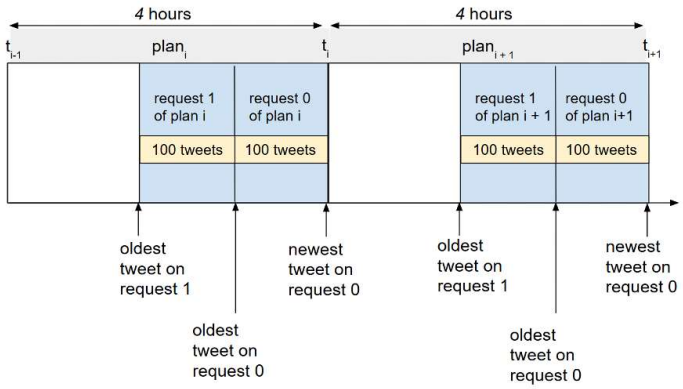
Pour rappel, `epitweetr` est planifié pour effectuer 180 recherches (requêtes) sur Twitter toutes les 15 minutes (ou 450 si vous utilisez une application de développeur Twitter). Chaque recherche peut renvoyer 100 tweets. Les recherches renvoient des tweets et des retweets, au format JSON, qui est un format de données léger.

Afin de collecter un maximum de tweets, compte tenu des limitations de l'API de recherche standard, et d'éviter que les sujets populaires n'empêchent la collecte adéquate d'autres sujets, `epitweetr` utilise des «plans de recherche» pour chaque requête.

Le premier «plan de recherche» collectera les tweets à partir de la date et de l'heure courantes en remontant jusqu'à 7 jours (du fait des limitations de l'API de recherche standard) avant l'application du «plan de recherche». Ce premier «plan de recherche» est le plus vaste, étant donné qu'aucun tweet n'a encore été collecté.

Tous les «plans de recherche» suivants sont planifiés à des intervalles définis dans la page «configuration» de l'interface Shiny app d'`epitweetr` (voir la section *L'application utilisateur interactive (Shiny app) > La page «configuration» > General*). À titre d'exemple, prenons des plans de recherche planifiés à quatre heures d'intervalle. Les plans collectent les tweets correspondant à une requête donnée à partir de la date et de l'heure courantes en remontant jusqu'à quatre heures avant l'application du «plan de recherche» (voir l'image ci-dessous). `epitweetr` effectuera autant de recherches que nécessaire (chacune renvoyant jusqu'à 100 tweets) pour collecter tous les tweets créés dans cet intervalle.

Par exemple, si le «plan de recherche» commence à 4 heures du matin le 10 septembre 2020, `epitweetr` recherchera les tweets correspondant à ses requêtes au cours de la période de quatre heures qui va de minuit à 4 heures du matin, le 10 septembre 2020, en commençant par les plus récents (ceux de 4 heures du matin), puis en remontant en arrière. Si, sur la période de quatre heures entre minuit et 4 heures du matin, l'API ne renvoie plus de résultats, le «plan de recherche» pour cette requête est considéré comme achevé.



Toutefois, si certains sujets sont très populaires (par exemple, la COVID-19 en 2020), il se peut que le «plan de recherche» pour une requête au cours d'une période donnée de quatre heures ne soit pas complet. Dans ce cas, `epitweetr` passera aux «plans de recherche» pour la période suivante de quatre heures et placera tous les «plans de recherche» précédents incomplets dans une file d'attente à exécuter lorsque les «plans de recherche» pour cette nouvelle période de quatre heures seront achevés.

Chaque «plan de recherche» contient les informations suivantes:

Champ	Type	Description
expected_end	Timestamp	Date et heure de fin de la période de recherche courante
scheduled_for	Timestamp	Date et heure planifiées pour la prochaine recherche. Lors de la création du plan, il s'agira de la date et de l'heure courantes et, après chaque recherche, cette valeur sera définie à une date et une heure ultérieures. Pour déterminer la date et l'heure ultérieures, l'application estimera le nombre de recherches nécessaires pour achever le plan. Si elle estime que N recherches sont nécessaires, la suivante sera planifiée dans 1/N du temps restant.
start_on	Timestamp	Date et heure d'achèvement de la première recherche du plan.
end_on	Timestamp	Date et heure d'achèvement de la dernière recherche du plan, si cette recherche a atteint une progression de 100 % du plan.
max_id	Long	L'ID Twitter maximum ciblée par ce plan, qui sera définie après la première recherche.
since_id	Long	L'ID du dernier tweet renvoyé par la dernière recherche de ce plan. La prochaine recherche commencera à collecter les tweets qui précèdent. Cette valeur est mise à jour après chaque recherche et permet à l'API Twitter de renvoyer les tweets qui se situent avant min_time(p_i)

since_target	Long	S'il existe un plan précédent, cette valeur enregistre l'ID du premier tweet qui a été téléchargé pour ce plan. Le plan courant ne collectera pas les tweets avant cette ID. Cette valeur permet à l'API Twitter de renvoyer les tweets qui se situent après p_i-time_back
requests	Int	Nombre de recherches effectuées dans le cadre du plan
progress	Double	Progression du plan courant en pourcentage. Elle est calculée comme suit: (current$max_id - current$since_id)/(current$max_id - current$since_target). Si l'API Twitter ne renvoie aucun tweet, la progression est fixée à 100%. Cela ne s'applique qu'aux réponses sans erreur contenant une liste de tweets vide.

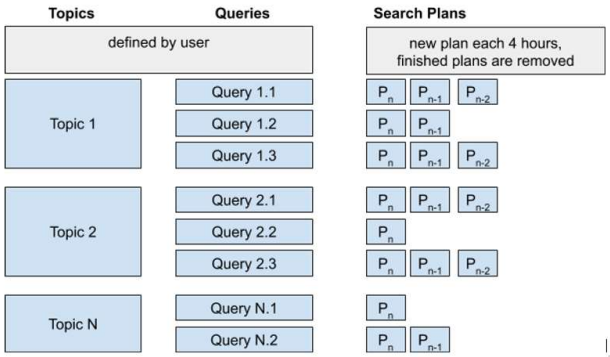
`epitweetr` exécutera les plans conformément aux règles suivantes:

- `epitweetr` détectera le plan inachevé le plus récent pour chaque requête de recherche dont la variable scheduled_for est située dans le passé.
- `epitweetr` exécutera les plans avec le nombre minimum de recherches déjà effectuées. Cela permet de garantir que tous les plans programmés exécutent le même nombre de recherches.
- Il résulte des deux règles précédentes que les recherches portant sur des sujets qui restent sous la limite de 180 de l'API de recherche standard de Twitter (ou 450 si vous utilisez l'authentification de l'application de développer Twitter) seront exécutées en premier et produiront une progression supérieure à celles portant sur des sujets qui dépassent la limite.

Ce choix se justifie par le fait que les sujets qui donnent lieu à des tweets si nombreux que la période de recherche de 4 heures ne suffit pas à les collecter sont probablement déjà connus. Il convient donc de donner la priorité à des sujets plus restreints et peut-être moins connus.

La pandémie de COVID-19 en 2020 en est un exemple. Au début 2020, les informations disponibles concernant la COVID-19 étaient limitées, ce qui permettait de détecter des signaux correspondant à des informations ou des mises à jour pertinentes (par exemple, de nouveaux pays signalant des cas ou confirmant qu'ils étaient causés par un coronavirus). Cependant, à mesure qu'il devenait plus populaire au fil de la pandémie, le vaste sujet de la COVID-19 a perdu de son efficacité pour la détection des signaux et s'est mis à consommer beaucoup de temps et de recherches dans `epitweetr`. Dans un tel cas, il est préférable de donner la priorité à la collecte de sujets plus restreints, tels que des sous-sujets liés à la COVID-19 (par exemple, vaccin AND COVID-19), ou de veiller à ne pas passer à côté d'autres événements moins présents dans les médias sociaux.

Si certains plans de recherche ne peuvent pas être achevé, plusieurs plans de recherche par requête peuvent se trouver dans une file d'attente:



Géolocalisation

Parallèlement à la collecte des tweets, `epitweetr` tente de géolocaliser tous les tweets collectés à l'aide d'un processus d'apprentissage machine non supervisé. Ce processus s'exécute au cours de la période planifiée définie par la propriété «Detect span» dans la page «configuration» sous la rubrique «General settings» (par exemple, si une période de quatre heures est définie, il démarrera toutes les quatre heures et géolocalisera tous les tweets collectés depuis la dernière fois qu'il a été exécuté avec succès).

`epitweetr` enregistre deux types de géolocalisation pour un tweet: la localisation du tweet, qui est une information de géolocalisation se trouvant dans le texte d'un tweet (ou d'un tweet retweeté ou cité), et la localisation de l'utilisateur obtenue à partir des métadonnées disponibles. Pour la détection des signaux, la meilleure localisation est utilisée tandis que les deux types peuvent être visualisés dans le «dashboard».

Géolocalisation basée sur la localisation du tweet

La localisation du tweet est extraite et enregistrée par `epitweetr` sur la base des informations de géolocalisation trouvées dans le texte d'un tweet. Dans le cas d'un tweet retweeté ou cité, les informations de géolocalisation sont extraites du texte du tweet original qui a été retweeté ou cité. Si celles-ci ne sont pas disponibles non plus, aucune localisation n'est enregistrée sur la base du texte du tweet.

`epitweetr` détermine si le texte d'un tweet contient une référence à un lieu particulier en le décomposant en ensembles de mots et en analysant ceux qui sont les plus susceptibles de désigner un lieu à l'aide d'un modèle d'apprentissage machine. L'algorithme ajoute aussi d'autres mots (un par un) à l'ensemble et si le score augmente avec plus de mots, il essaie de trouver un maximum local du score en utilisant un texte plus long. Il compare ensuite ces mots avec le contenu d'une base de données de référence, à savoir geonames.org. Il s'agit d'une base de données géographique disponible sous licence Creative Commons Attribution et accessible au moyen de divers services web. La base de données GeoNames.org contient plus de 25 000 000 de noms géographiques. Par défaut, `epitweetr` se limite à ceux qui existent actuellement et qui correspondent à une population connue (soit un peu plus de 500 000 noms). Vous pouvez passer outre ce paramètre par défaut sur la page

«configuration» de Shiny app, en décochant «Simplified geonames». La base de données contient également les attributs de longitude et de latitude des localités et les variantes orthographiques (références croisées), utiles à des fins de recherche, ainsi que les graphies de beaucoup de ces noms dans d'autres caractères que ceux de l'alphabet latin.

Les correspondances peuvent être trouvées à n'importe quel niveau de la hiérarchie administrative. Le processus de comparaison est pris en charge par Apache Lucene, une bibliothèque open source à hautes performances de moteur de recherche en texte intégral.

Il peut arriver que ce processus trouve plusieurs localisations pour certains textes, mais il ne retiendra que celle dont le score est le plus élevé.

Un score plus élevé est associé à une plus grande probabilité qu'une correspondance soit correcte. Il sera:

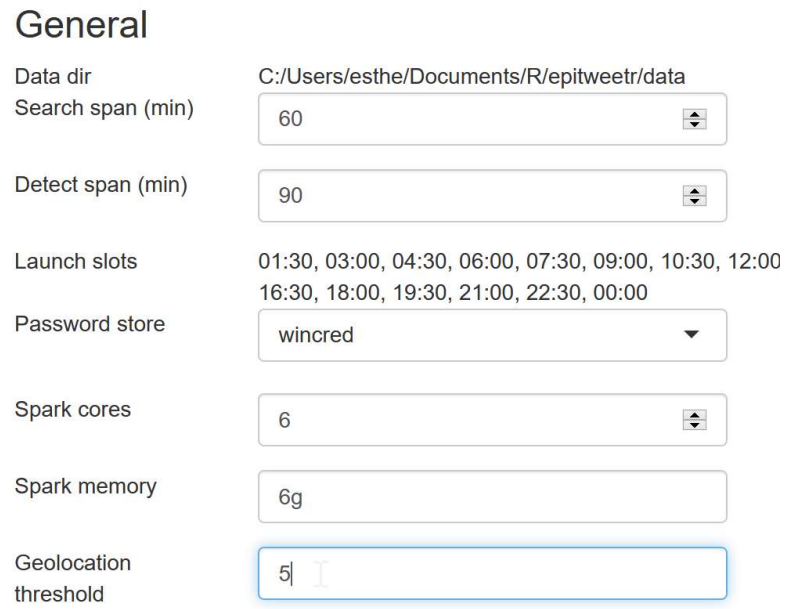
- Plus élevé si des parties inhabituelles du nom correspondent
- Plus élevé si plusieurs niveaux administratifs correspondent
- Plus élevé si la population du lieu concerné est plus nombreuse
- Plus élevé pour les pays et les villes par rapport aux subdivisions administratives
- Plus élevé pour les abréviations en majuscules comme NY
- Moins élevé pour les mots qui risquent davantage d'être d'autres types de noms (non géographiques). Par exemple, la ville de «Fair Play» dans le Colorado. Pour ce faire, le système utilise des modèles de langue fournis par fasttext.cc.

Vous pouvez choisir les langues dans lesquelles vous souhaitez vérifier d'autres types de mots, en sélectionnant la langue active souhaitée dans la page «configuration» de Shiny app et en cliquant sur l'icône «+»:

L'icône «-» permet de désélectionner une langue: ADD IMAGE HERE

Il est possible de définir globalement un score minimum (le «geolocation threshold») dans les paramètres généraux de la page «configuration» pour réduire le nombre de faux positifs (voir image). Toutes les géolocalisations dont le score est inférieur à ce seuil seront rejetées par l'algorithme comme localisation de tweet. S'il y a plus d'une correspondance dépassant le score minimum, celle dont le score est le plus élevé sera retenue.

Le seuil est choisi de manière empirique et peut être évalué par rapport à une lecture humaine des tweets et de leurs localisations, sur la page «geotag evaluation».



Géolocalisation basée sur la localisation de l'utilisateur

Différents types de localisation des utilisateurs sont disponibles sur la base des métadonnées fournies par l'API de recherche standard sur Twitter. Pour les fichiers agrégés, `epitweetr` sélectionne la meilleure dans l'ordre suivant:

- la localisation exacte ou approximative de l'utilisateur au moment du tweet (fournie par l'API)
- si la localisation de l'utilisateur n'est pas disponible et que le tweet est un retweet ou une citation de tweet, la localisation exacte ou approximative de l'utilisateur au moment du retweet ou de la citation (fournie par l'API)
- à défaut, la localisation déclarée par l'utilisateur
- à défaut, le «home» du profil public.

Pour les localisations exactes, la longitude et la latitude sont indiquées. S'il s'agit d'une localisation estimée, `epitweetr` calcule la longitude et la latitude en se basant sur GeoNames.org.

Si les informations relatives à la localisation de l'utilisateur fournies par l'API ne sont pas disponibles, `epitweetr` calculera la longitude et la latitude d'après la localisation déclarée par l'utilisateur ou le nom de lieu indiqué dans son «profil public», en se basant sur GeoNames.org.

Enregistrement des informations sur les tweet géolocalisés

La géolocalisation de la correspondance trouvée est stockée sous la forme d'un code pays (selon la norme ISO 3166) et d'une longitude et d'une latitude associées à la géolocalisation exacte dans les données agrégées.

Mots les plus fréquents dans les tweets

Comme le nombre de tweets et de mots peut être très volumineux, les tweets sont analysés en fonction de la fréquence des mots employés par tranche de 10 000 tweets afin d'obtenir les 500 mots les plus fréquents de chaque tranche pour la même langue, le même jour et le même sujet.

Pour garantir des performances raisonnables d'`epitweetr`, ces 500 mots sont d'abord déterminés au niveau mondial, puis ils servent à créer des sous-ensembles par pays, afin d'obtenir les mots les plus fréquents par pays, par jour et par sujet. Dans les très petites localités où le nombre de tweets géolocalisés est faible, il se peut qu'aucun mot fréquent ne soit trouvé.

Il est à noter que, contrairement aux autres données de visualisation, les mots les plus fréquents dans les tweets se basent toujours sur la géolocalisation liée à la «localisation du tweet» et non à la «localisation de l'utilisateur», quel que soit le filtre sélectionné dans le «dashboard».

Agrégation des données

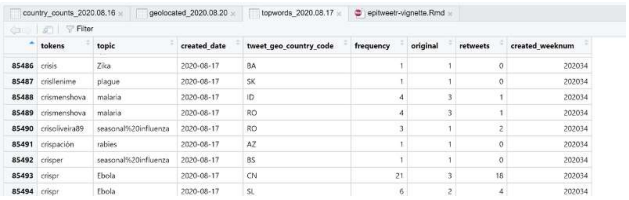
Le processus d'agrégation produit trois fichiers .Rds (un format R natif): geolocated, country_counts et topwords.

Dans le fichier geolocated.Rds, le nombre de tweets ou de retweets est enregistré par sujet, date, longitude et latitude correspondant à la géolocalisation du texte du tweet et longitude et latitude correspondant à la géolocalisation de l'utilisateur. Chacune de ces entrées indique également le pays associé à la géolocalisation du texte du tweet et celui associé à la géolocalisation de l'utilisateur (voir la capture d'écran partielle ci-dessous). Il est à noter que les tweets sans informations de géolocalisation sont également inclus.

Le fichier country_counts.Rds sert à créer la courbe de tendance dans Shiny app. Il s'agit d'un fichier .Rds plus petit, sans les informations de longitude et de latitude, qui mentionne le nombre de tweets par heure dans une journée, par pays (selon la localisation du tweet ou de l'utilisateur), par sujet (voir la capture d'écran), et qui indique si un tweet était un retweet ou non. Les champs known_retweets et known_original donnent le nombre de tweets ou de retweets provenant d'une liste d'«important users». Dans ce fichier, les tweets sans géolocalisation sont également inclus, ce qui vous permet d'afficher tous les tweets lorsque vous sélectionnez «world» comme région, indépendamment du succès de la géolocalisation.

	topic	created_date	created_hour	tweet_geo_country_code	user_geo_country_code	retweets	tweets	known_retweets
33	COVID-19	2020-08-16	19	AU	US	71	13	0
34	COVID-19	2020-08-16	19	GH	PK	5	3	0
35	rabies	2020-08-16	21	PK	ES	20	0	0
36	gonorrhea	2020-08-16	01	VE	NA	1	0	0
37	COVID-19	2020-08-16	04	NA	PE	88	22	0

L'agrégation par mots les plus fréquents est enregistrée dans le fichier topwords.Rds, qui affiche le nombre de tweets ou de retweets (ou les deux) par sujet, mot le plus fréquent, date, pays de localisation du tweet et indique si un tweet était un retweet ou non (voir la capture d'écran).



	tokens	topic	created_date	tweet_geo_country_code	frequency	original	retweets	created_weeknum
85486	crisis	Zika	2020-08-17	BA	1	1	0	202034
85487	crisilenime	plague	2020-08-17	SK	1	1	0	202034
85488	crismenshova	malaria	2020-08-17	ID	4	3	1	202034
85489	crismenshova	malaria	2020-08-17	RO	4	3	1	202034
85490	crisolveira89	seasonal%20influenza	2020-08-17	RO	3	1	2	202034
85491	crispación	rabies	2020-08-17	AZ	1	1	0	202034
85492	crisper	seasonal%20influenza	2020-08-17	BS	1	1	0	202034
85493	crispr	Ebola	2020-08-17	CN	21	3	18	202034
85494	crispr	Ebola	2020-08-17	SL	6	2	4	202034

Détection des signaux

L'objectif principal d'`epitweetr` est de détecter des signaux dans les flux de données observés, c'est-à-dire des comptages qui dépassent les valeurs attendues dans les séries chronologiques agrégées. Pour la détection des signaux, `epitweetr` utilise une version étendue de l'algorithme EARS (Early Aberration Reporting System) (Fricker, Hegler et Dunfee, 2008), qui est désignée par `eears` (extended EARS) dans ce qui suit. Cet algorithme fait partie du paquet R `surveillance` (Salmon, Schumacher et Höhle, 2016).

Par défaut, il utilise une fenêtre glissante correspondant aux sept derniers jours pour calculer un seuil. Si le comptage pour le jour courant dépasse ce seuil, un signal est généré.

Détails de l'algorithme sur lequel s'appuie la détection des signaux

L'algorithme `eears` est appliqué aux comptages des sept derniers blocs de 24 heures précédant le bloc de 24 heures courant dans la détection des signaux. La moyenne glissante et l'écart-type glissant sont calculés:

$$\overline{y}_0 = \frac{1}{7}\sum_{t=-7}^{-1} y_t \quad \text{et} \quad s_0^2 = \frac{1}{7-1}\sum_{t=-7}^{-1}(y_t - \overline{y}_0)^2,$$

où $y_t, t = \cdots, -2, -1{,}0$ désigne la série chronologique de données de comptage observées, le bloc courrant correspondant à l'indice de temps 0. L'indice de temps $-7, \ldots, -1$ désigne les sept blocs précédant le bloc actuel.

Dans l'hypothèse nulle d'absence de pointes, il est supposé que les y_t sont distribués $N(\mu, \sigma^2)$ de manière identique et indépendante, avec une moyenne μ inconnue et une variance σ^2 inconnue. Par conséquent, la limite supérieure d'un simple intervalle de prédiction *plug-in* à 100 % $(1-\alpha) \times$ unilatéral pour y_0 basé sur $y_{-7}, \ldots, y_{-1}$ est donnée comme suit

$$U_0 = \overline{y}_0 + z_{1-a} \times s_0,$$

où z_{1-a} est le quantile (1 − α)- de la distribution normale standard. Une alerte est déclenchée si $y_0 > U_0$. Si l'on utilise α=0,025, cela revient à examiner si y_0 dépasse l'estimation de la moyenne plus 1,96 fois l'écart-type. Cependant, comme le soulignent Allévius et Höhle (2017), la méthode correcte serait de comparer l'observation à la limite supérieure d'un intervalle de prédiction à 95 % bilatéral pour y_0, car cela respecte à la fois la variation d'échantillonnage d'une nouvelle observation *et* l'incertitude provenant de l'estimation des paramètres de la moyenne et de la variance. Par conséquent, la forme statistique appropriée consiste à calculer la limite supérieure par

$$U_0 = \overline{y_0} + t_{1-a}(7-1) \times s_0 \times \sqrt{1+\frac{1}{7}},$$

où $t_{1-a}(k-1)$ désigne le quantile 1 − α de la distribution t avec k − 1 degrés de liberté.

Ajustement de la pondération des signaux précédents

Si les signaux précédents sont inclus sans modification dans les valeurs historiques lors du calcul de la moyenne et de l'écart type glissants pour la détection des signaux, il se peut que la moyenne et l'écart type estimés deviennent trop grands. Dans ce cas, d'importants signaux courants risquent de ne pas être détectés. Pour résoudre ce problème, `epitweetr` ajuste la pondération des signaux précédents pour tenir compte de ces valeurs aberrantes dans l'estimation de la moyenne et de l'écart-type, en recourant à une approche comparable à celle utilisée dans l'étude de Farrington et al. (1996). Les valeurs historiques qui ne sont pas identifiées comme des signaux précédents reçoivent une pondération de «1». De même, les valeurs historiques identifiées comme des signaux reçoivent une pondération inférieure et un nouvel ajustement est effectué en utilisant ces pondérations (avec mise à l'échelle de telle sorte que la somme des observations soit de nouveau de 7). L'annexe I de cette documentation à l'usage des utilisateurs contient des précisions sur la procédure d'ajustement de la pondération.

Planification de la détection des signaux

La détection des signaux s'effectue sur la base de «jours», qui sont des fenêtres glissantes de 24 heures, se déplaçant en fonction de l'intervalle de détection (voir aussi la section *L'application utilisateur interactive (Shiny app) > La page «configuration» > General*). La base de référence est calculée sur les «jours» -1 à -8 (si le «jour» courant est zéro).

Les signaux sont générés selon l'intervalle de détection (voir la section *L'application utilisateur interactive (Shiny app) > La page «configuration» > General*) et donnent lieu à

- des alertes générales par courrier électronique après cet intervalle de détection (par exemple, si l'intervalle de détection est de quatre heures, les alertes par courrier électronique seront envoyées toutes les quatre heures)
- des alertes par courrier électronique en temps réel. Dans ces alertes, les signaux générés précédemment seront omis.

Les différents types d'alertes par courrier électronique pour chaque utilisateur peuvent être spécifiés dans la page «configuration» (voir la section *L'application utilisateur interactive (Shiny app) > La page «configuration» > General*).

Le paramètre alpha: le taux de faux positifs de la détection du signal.

Un attributs déterminant de la détection des signaux tient à la capacité d'un algorithme de détecter les véritables menaces ou événements sans que les enquêteurs soient submergés par un trop grand nombre de faux positifs. À cet effet, le paramètre alpha détermine le seuil de l'intervalle de détection. S'il est élevé, les signaux potentiels sont générés en plus grand nombre et s'il est faible, les signaux potentiels générés sont moins nombreux (mais il se peut que des menaces ou événements potentiels passent inaperçus). Le réglage de ce paramètre se fait souvent de manière empirique, et dépend également des ressources de ceux qui étudient les signaux et de l'importance accordée au risque de passer à côté de menaces ou événements potentiels.

Il existe un paramètre alpha global, qui peut être défini/modifié dans la page «configuration» d'`epitweetr` sous la rubrique «Signal false positive rate» (voir la section *L'application utilisateur interactive (Shiny app) > La page «configuration» > General*)). En outre, il est possible de changer la valeur par défaut dans la liste des sujets. Si vous le souhaitez, vous pouvez associer à chaque sujet son propre paramètre alpha, en fonction de l'importance estimée du sujet pour la santé publique ou des menaces ou événements potentiels concernés.

Correction de Bonferroni

Afin de tenir compte des tests multiples, pour la détection de signaux propres à chaque pays, le paramètre alpha est divisé par défaut par le nombre de pays. Pour la détection de signaux propres à un continent, il est divisé par le nombre de continents. Il s'agit d'une correction de Bonferroni pour les tests multiples.

Pour passer outre, vous pouvez décocher la case «Bonferroni correction» dans la partie «Signal detection» de la page «configuration» de Shiny app.

Utilisation des mêmes jours de la semaine comme base de référence

Il est possible qu'il y ait un «effet jour de la semaine», c'est-à-dire que les tweets soient plus nombreux un jour donné de la semaine (par exemple le lundi) que les autres jours. Pour éviter cela, vous pouvez aussi choisir de calculer la base de référence non pas sur des jours consécutifs, mais sur les N derniers jours qui correspondent à la même fenêtre de 24 heures que N jours auparavant. Ainsi, si N = 7, la base de référence est calculée en utilisant les «jours» -7, -14, -21, -28, -35, -42, -49 et -56 (si le «jour» courant est zéro).

Cette option se trouve sur la page «configuration» de Shiny app, sous la rubrique «Default same weekday baseline».

Envoi d'alertes par courrier électronique

Les courriers électroniques contenant une liste de signaux détectés sont envoyés automatiquement par `epitweetr` selon l'intervalle de détection et la liste des abonnés. Du fait du temps nécessaire à la collecte, à la géolocalisation et à l'agrégation des tweets, les alertes par courrier électronique manqueront les tweets les plus récents qui n'ont pas encore été soumis à ces processus. Le décalage entre les tweets et les alertes devrait être inférieur à (2 * (collect_span) + detect_span), soit 3 h 30 en utilisant les valeurs par défaut.

Les alertes électroniques comprendront pour chaque sujet les informations suivantes sur les signaux:

- La date et l'heure à laquelle le signal a été détecté
- La (les) localisation(s) géographique(s) où le signal a été détecté
- Les mots les plus fréquents dans les tweets
- Le nombre de tweets et le seuil
- Le pourcentage de tweets provenant d'utilisateurs importants
- Des informations sur les paramètres, telles que: l'utilisation ou non de la correction de Bonferroni, le choix ou non du même jour de la semaine comme base de référence, l'inclusion ou non des retweets, etc.

Ces informations sont également disponibles dans la page «alerts» de Shiny app.

Les abonnés peuvent recevoir des alertes en temps réel (c'est-à-dire dès que la boucle de détection est finalisée) ou des alertes planifiées (par exemple, une ou deux fois par jour). La liste des abonnés peut être modifiée dans la page «configuration» en téléchargeant la feuille de calcul Excel. Ce fichier comporte les variables suivantes:

- «User»: nom de l'abonné (par exemple, Jane Doe).
- «Email»: adresse électronique de l'abonné (par exemple, jane.doe@email.com).
- «Topics»: liste des sujets pour lesquels l'abonné recevra des alertes planifiées. Les noms utilisés doivent correspondre à la colonne «Topic» de la liste des sujets.
- «Excluded»: sujet pour lequel les abonnés ne recevront pas d'alertes planifiées.
- «Real time Topics»: liste des sujets pour lesquels l'abonné recevra des alertes en temps réel.
- «Regions»: liste des régions pour lesquelles l'abonné recevra des alertes planifiées.
- «Real time Regions»: liste des régions pour lesquelles l'abonné recevra des alertes en temps réel.

- «Alert Slots»: il s'agit des créneaux de la boucle de détection après lesquels l'abonné recevra l'alerte planifiée. Les créneaux disponibles peuvent être choisis parmi les «Launch slots» dans la section «General» de la page «configuration». Si aucune valeur n'est indiquée, l'abonné recevra des alertes en temps réel pour tous les sujets et toutes les régions, même si des sujets ou des régions sont spécifiés pour les alertes en temps réel dans la feuille de calcul Excel.

En cas d'inclusion de plusieurs sujets et/ou régions dans la liste d'abonnés, ceux-ci doivent être séparés par un point-virgule (;) sans espace (par exemple: Ebola;infectious diseases;dengue). Les noms doivent correspondre à la colonne «Topics» dans la liste des sujets et à la colonne «Name» dans la liste des pays/régions de la page «configuration».

	B	C	D	E	F	G	H	I
	User	**Email**	**Topics**	**Excluded Topics**	**Real time Topics**	**Regions**	**Real time Regions**	**Alert Slots**
	Jane Doe	jane.doe@email.com			infectious diseases;zoonoses		Southern Europe;EU+EEA	8;20

Structure des dossiers

`epitweetr` conserve les tweets, les tweets agrégés et la configuration dans le dossier «data» que vous devez désigner lors du lancement de l'application.

Le dossier **data** contient 3 fichiers JSON:

- properties.json, généré à partir des informations des propriétés générales de Shiny app.
- topics.json géré par la boucle de recherche: il garde une trace des plans de collecte de tweets et de leur progression
- tasks.json géré par la boucle de détection: il conserve les informations et le statut des différentes tâches effectuées par ce processus.

Il comprend aussi les sous-dossiers suivants:

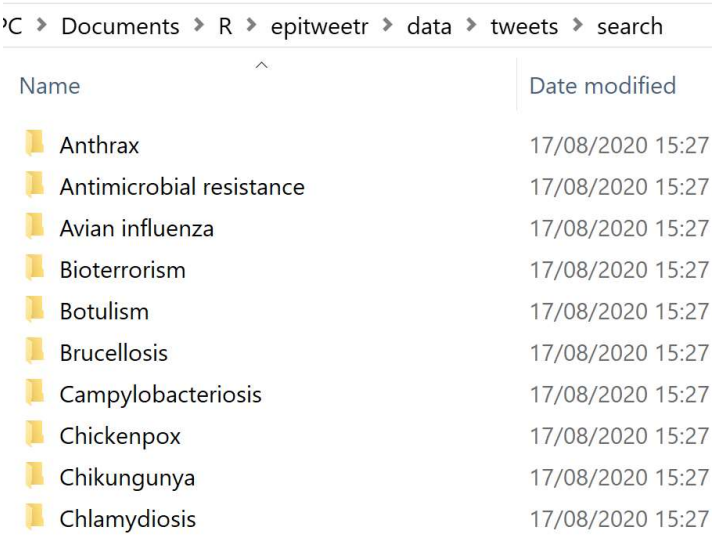
- «geo», où sont stockées les données GeoNames sous forme de fichiers texte et d'index
- «hadoop», où sont stockées les dépendances Spark pour les systèmes d'exploitation Windows
- «jars», où sont stockées les collections de dépendances Java nécessaires aux processus de géolocalisation et d'agrégation
- «languages», où sont stockés les index de fichiers et les modèles fasttext qui servent à effectuer la géolocalisation dans le texte du tweet
- «stats», où sont stockés des fichiers json présentant les statistiques utilisées pour optimiser le processus d'agrégation en reliant les fichiers de tweets et les dates de publication des tweets
- «alerts», où sont stockés les fichiers json des alertes détectées par la boucle de détection

- «tweets» et «séries» qui sont expliqués plus en détail ci-dessous.

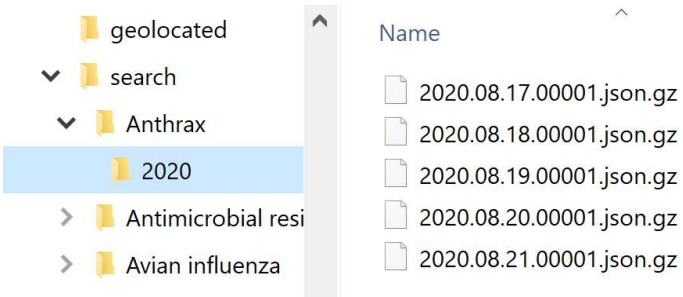
Dossier data > tweets

Dans le dossier data, le sous-dossier tweets se compose de deux autres sous-dossiers: «**search**» et «**geolocated**»

Le dossier *search* contient des sous-dossiers pour chaque sujet figurant dans la liste des sujets:



Dans chacun de ces sujets se trouve une année (par exemple 2020), puis un fichier json compressé contenant les tweets par jour de chaque année. Les dates se rapportent à celles auxquelles le tweet a été collecté (et non publié). Il peut y avoir plus d'un fichier pour un jour si la taille dépasse 100 Mo.



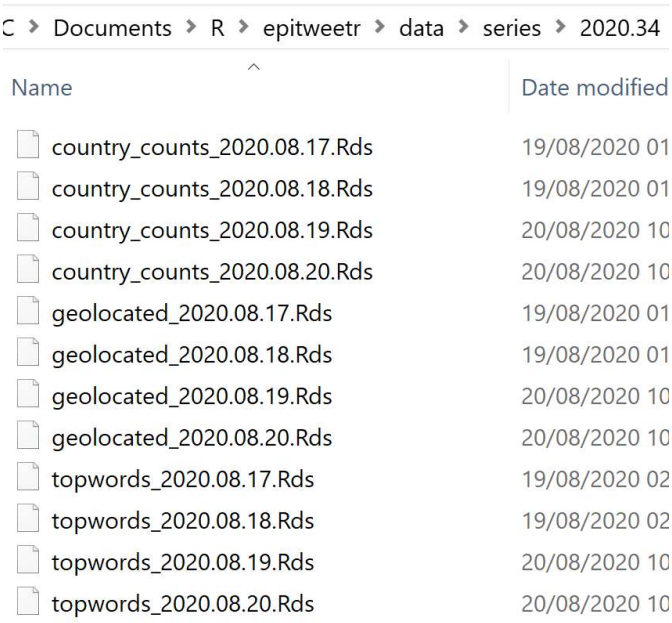
Le dossier *geolocated* contient des fichiers json compressés où sont conservée les informations produites par l'algorithme de géolocalisation.

Dossier data > series

Dans le dossier *series*, `epitweetr` conserve les données agrégées des tweets géolocalisés ainsi que les mots les plus fréquents.

Il y a un dossier pour chaque semaine ISO de dates de collecte contenant des fichiers Rds (un format R natif) pour chaque jour et chaque série:

- geolocated_YYYY.MM.DD.Rds contient le comptage de tweets quotidiens au niveau de localisation le plus fin possible.
- topwords_YYYY.MM.DD.Rds contient le comptage quotidien des mots les plus fréquemment employés dans les tweet au niveau national.
- country_counts_YYYY.MM.DD.Rds contient le nombre de tweets par heure au niveau national.



Il s'agit des informations agrégées décrites dans la section «Comment ça marche? L'architecture générale sur laquelle repose `epitweetr` > Agrégation».

L'application utilisateur interactive (Shiny app)

Vous pouvez lancer l'application utilisateur interactive d'`epitweetr` (Shiny app) à partir de la session R en tapant dans la console R (remplacer «data_dir» par le répertoire de données souhaité):

```
epitweetr_app("data_dir")
```

Sinon, vous pouvez aussi utiliser un lanceur: Dans un fichier bat exécutable ou un script shell, placez le contenu suivant (en remplaçant «data_dir» par le répertoire de données prévu)

R –vanilla -e epitweetr::epitweetr_app('data_dir')

L'application utilisateur interactive d'`epitweetr` comporte cinq pages:

- Le «dashboard», où l'utilisateur peut visualiser et explorer les tweets
- La page «configuration», où vous pouvez modifier les paramètres et vérifier le statut des processus sous-jacents
- La page «alerts», où vous pouvez voir les alertes en cours et les informations associées
- La page «geotag evaluation», où vous pouvez évaluer l'algorithme de géolocalisation dans différents champs pour choisir manuellement le seuil de géolocalisation
- La page «troubleshoot», avec des contrôles automatiques et des conseils pour utiliser `epitweetr` et toutes ses fonctionnalités

Le «dashboard»: L'interface utilisateur interactive de visualisation

Le «dashboard» est l'endroit où vous pouvez explorer de manière interactive les visualisations des tweets. Il comprend un graphique linéaire (courbe de tendance) avec des alertes, une carte et les mots les plus fréquemment employés dans les tweets pour un sujet donné. Veuillez noter que la première fois qu'une période est sélectionnée, vous devrez **attendre 1 à 2 minutes avant de voir les résultats**. De même, pour toute nouvelle sélection (ajout d'une région, changement de thème, etc.), `epitweetr` commencera à lire les données correspondantes; si plusieurs sélections sont effectuées, vous devrez donc peut-être patienter 1 à 2 minutes jusqu'à ce que la dernière sélection apparaisse dans le «dashboard». Quand `epitweetr` lit de nouvelles données, les résultats s'affichent de manière moins intenses, ce qui vous indique que de nouvelles données sont en cours de lecture et de représentation graphique.

Pour explorer les données de manière interactive, vous pouvez choisir parmi plusieurs filtres, tels que les sujets, les pays et les régions, la période de temps, l'unité de temps, la confiance dans le signal et les jours dans la base de référence.

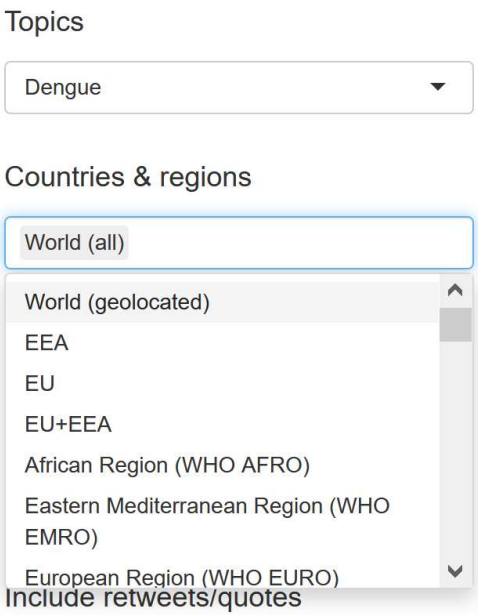
Il est à noter que les options/réglages que vous sélectionnez dans le «dashboard» n'auront aucun effet sur la détection des alertes. Les paramètres de détection d'alertes sont tous sélectionnés sur la page «configuration» de Shiny app.

Filtres

Topics

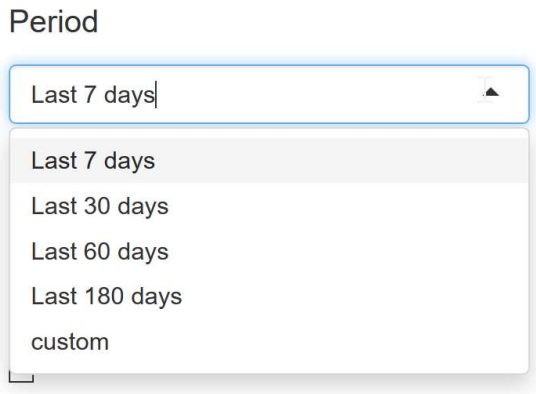
Vous pouvez sélectionner un élément dans la liste déroulante des sujets, qui comprend ce qui est spécifié dans les sujets de la page «configuration». Vous pouvez également taper les premières lettres de votre recherche dans le champ de texte et sélectionner les sujets dans la liste déroulante ainsi filtrée.

Countries & regions



Si vous sélectionnez **World (all)**, tous les tweets sont affichés, quelle que soit leur géolocalisation. Vous pouvez choisir un pays en particulier, des régions et des sous-régions, et vous pouvez sélectionner plusieurs éléments en même temps. Vous pouvez également taper les premières lettres de votre recherche dans le champ de texte et sélectionner l'élément géographique dans la liste déroulante.

Period



Vous pouvez choisir parmi les 7 (par défaut), 30, 60 ou 180 derniers jours. Vous pouvez également sélectionner «custom» et un calendrier apparaîtra pour vous permettre de choisir la période à examiner. Ces périodes seront celles qui seront incluses dans les visualisations. Lorsque vous sélectionnez une période personnalisée, veillez à ce que la première date se situe au moins un jour avant la seconde.

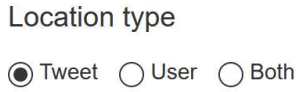
Time unit

Vous pouvez afficher la ligne de temps représentant le nombre de tweets en prenant comme unités de temps des semaines ou des jours. Par défaut, la ligne s'affiche en jours.

Include Retweets/quotes

Par défaut, les retweets ne sont pas inclus dans les visualisations. Si la case «include retweets/quotes» est cochée, les visualisations affichent les résultats des tweets et des retweets/citations. Sinon, les visualisations n'affichent que les tweets (sans les retweets/citations).

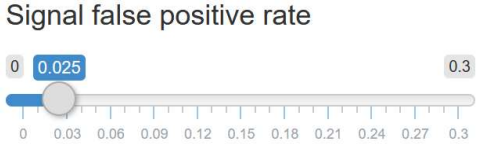
Location type



Les tweets sont géolocalisés en régions, sous-régions et pays. Le «Location type» indique le type de géolocalisation à utiliser:

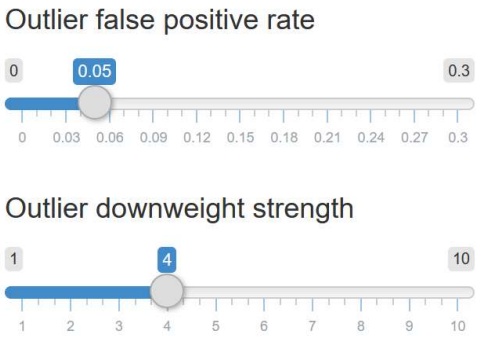
- Tweet: il s'agit des informations géographiques contenues dans le texte du tweet ou, à défaut, des informations géographiques contenues dans le texte retweeté/cité, le cas échéant.
- User: il s'agit d'informations géographiques obtenues sur la base de la localisation de l'utilisateur. Par ordre de priorité, les informations utilisées sont celles correspondant à la localisation de l'utilisateur au moment du tweet, à la localisation de l'API de l'utilisateur ou au champ «home» dans le profil public si aucun des éléments susmentionnés n'est disponible.
- Both: les informations géographiques utilisées pour un tweet correspondront, par ordre de priorité, à la localisation sur la base du texte du tweet, mais si celle-ci n'est pas disponible, à la localisation de l'utilisateur.

Signal detection false positive rate



À l'aide du curseur, vous pouvez tester les variations dans les signaux générés lorsque le paramètre alpha est modifié pour le taux de faux positifs. Il est à noter que cela ne modifiera pas le taux de faux positifs pour les alertes électroniques. Il s'agit simplement d'un outil permettant à l'utilisateur d'explorer ce paramètre. La valeur par défaut est de 0,025. Un taux de faux positifs plus élevé augmentera la sensibilité et éventuellement le nombre de signaux détectés, et vice versa.

Outlier false positive rate and outlier downweight strength



Le taux de faux positifs des valeurs aberrantes se rapporte au taux qui détermine à quoi correspond une valeur aberrante en cas d'ajustement de la pondération des valeurs aberrantes/signaux précédents. Plus il est faible, moins il inclura de valeurs aberrantes précédentes. Un paramètre plus élevée inclura potentiellement davantage de valeurs aberrantes précédentes.

La force d'ajustement de la pondération des valeurs aberrantes détermine dans quelle mesure la pondération d'une valeur aberrante sera ajustée. Plus la valeur est élevée, plus l'ajustement est important. Pour plus d'informations, voir l'annexe I.

Bonferroni correction

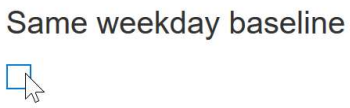
La correction de Bonferroni est sélectionnée par défaut. Elle tient compte de la détection de signaux qui se révèlent des faux positifs dans le cadre des tests multiples. Pour la détection de signaux propres à chaque pays, le paramètre alpha est divisé par le nombre de pays. Pour la détection de signaux propres à un continent, il est divisé par le nombre de continents.

Si vous ne souhaitez pas utiliser cette correction, vous pouvez décocher la case.

Days in baseline

Le nombre de jours par défaut dans la base de référence est de 7. L'utilisateur peut tester l'effet de nombre de jours différents dans la base de référence. Ce réglage ne s'applique qu'à la visualisation, toute modification des alertes par courrier électronique doit être effectuée dans la page «configuration».

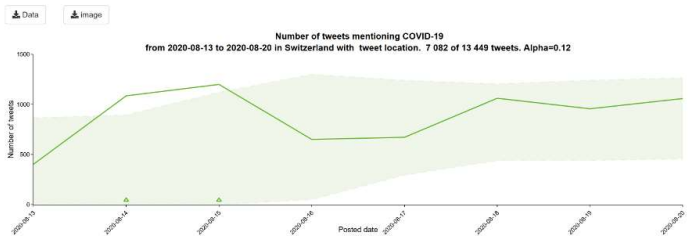
Same weekday baseline



Il est possible qu'il y ait un «effet jour de la semaine», c'est-à-dire que les tweets soient plus nombreux un jour donné de la semaine (par exemple le lundi) que les autres jours. Vous pouvez aussi choisir de calculer la base de référence non pas sur des jours consécutifs, mais sur les N derniers jours qui correspondent à la même fenêtre de 24 heures que N jours auparavant. Ainsi, si N = 7, la base de référence est calculée en utilisant les «jours» -7, -14, -21, -28, -35, -42, -49 et -56 (si le «jour» courant est zéro).

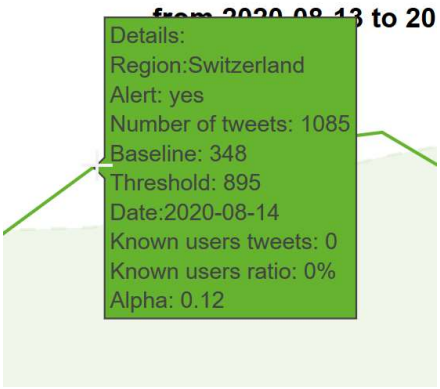
La ligne de temps

Le graphique de la ligne de temps présente une série chronologique, où vous pouvez voir le nombre de tweets pour un sujet, une unité géographique et une période donnés. Les signaux sont indiqués sous la forme de triangles sur le graphique, selon les spécifications du paramètre alpha et des jours de la base de référence dans les filtres. La zone se trouvant sous le seuil est représentée en vert grisé. Il est à noter que les signaux se rapportent au paramètre alpha et aux jours de la base de référence sélectionnés dans les filtres du tableau de bord, plutôt qu'aux spécifications utilisées pour les alertes par courrier électronique. De cette façon, vous pouvez tester l'effet d'une modification de ces paramètres et adapter au besoins les paramètres des alertes électronique.

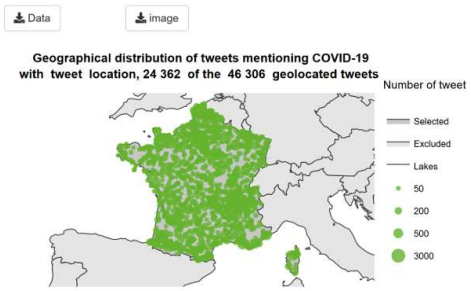


En plaçant le pointeur de la souris sur le graphique, vous obtenez des informations supplémentaires sur le pays, la date, le nombre total de tweets et le nombre de tweets émanant de la liste des utilisateurs connus, le ratio entre les utilisateurs connus et

inconnus, le déclenchement ou non d'un signal par le nombre de tweets, ainsi que le seuil défini et le paramètre alpha.

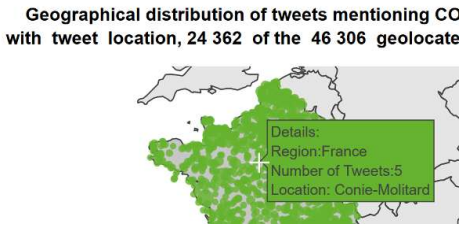


Carte



La carte présente la distribution des tweets sous la forme de symboles proportionnels par pays et par sujet pour la période considérée. Plus le cercle est large, plus le nombre de tweets est important.

Les informations géographiques apparaissant sur la carte sont basées sur les filtres choisis: pays/région/sous-région et type de localisation (tweet, utilisateur ou les deux).

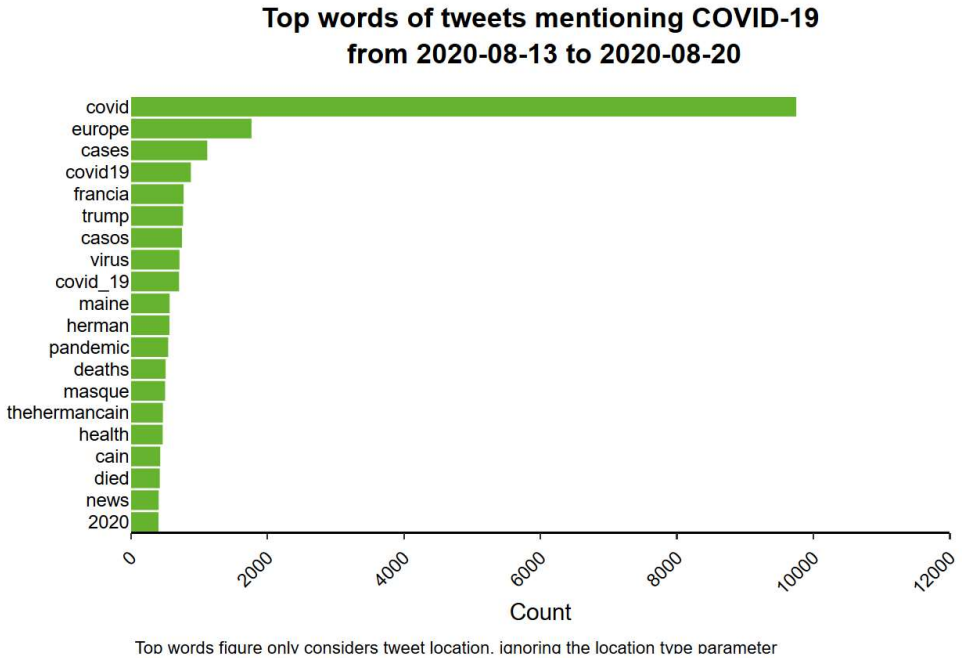


En plaçant le pointeur de la souris sur la carte, vous obtenez des informations sur le nombre de tweets et les noms des unités géographiques auxquels se rapportent les cercles.

Lorsque vous sélectionnez un pays, les symboles présentent la distribution géographique des tweets au niveau infranational. Lorsque vous sélectionnez deux ou plusieurs pays ou toute autre entité géographique (par exemple, des régions ou des continents), les symboles indiquent la distribution géographique des tweets au niveau national. Il est à noter que, si

un tweet est géolocalisé au niveau du pays (par exemple, la France), il ne sera pas affiché lorsque vous sélectionnez uniquement ce pays, puisqu'aucune géolocalisation infranationale n'est disponible.

Mots les plus fréquents dans les tweets



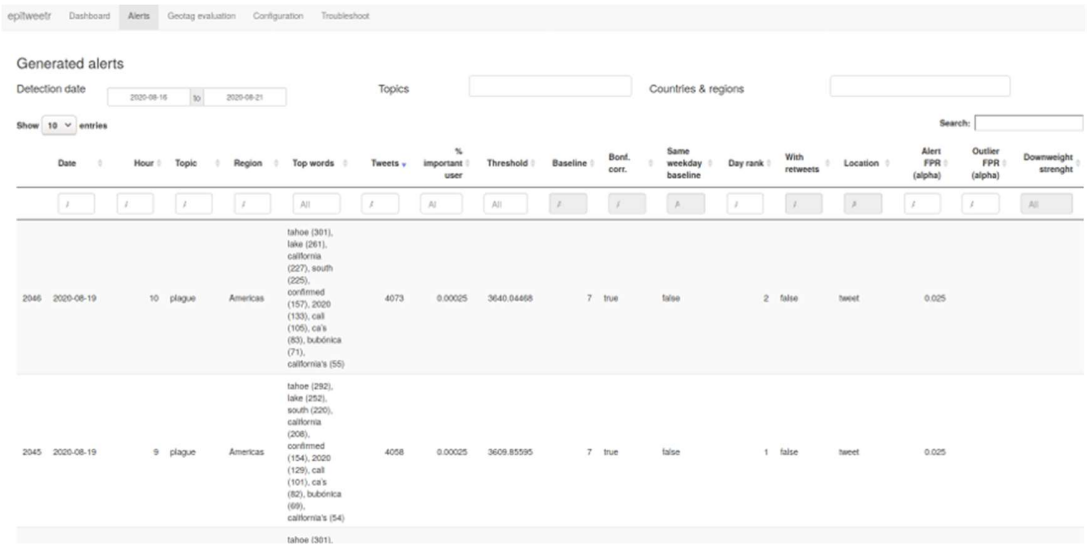
Les tweets sont analysés au niveau mondial pour obtenir les 500 mots les plus fréquents par tranche de 10 000 tweets pour la même langue, le même jour et le même sujet. Ces 500 mots sont ensuite présentés en sous-ensembles par pays.

Le graphique affiche les mots les plus fréquemment employés dans les tweets par sujet pour la période considérée selon les unités géographiques sélectionnées et le filtre appliqué concernant les tweet/retweet.

Il est à noter que, contrairement aux autres données de visualisation, les mots les plus fréquents dans les tweets se basent toujours sur la «localisation du tweet» et ne sont pas influencés par l'option de localisation choisie (localisation de l'utilisateur ou du tweet).

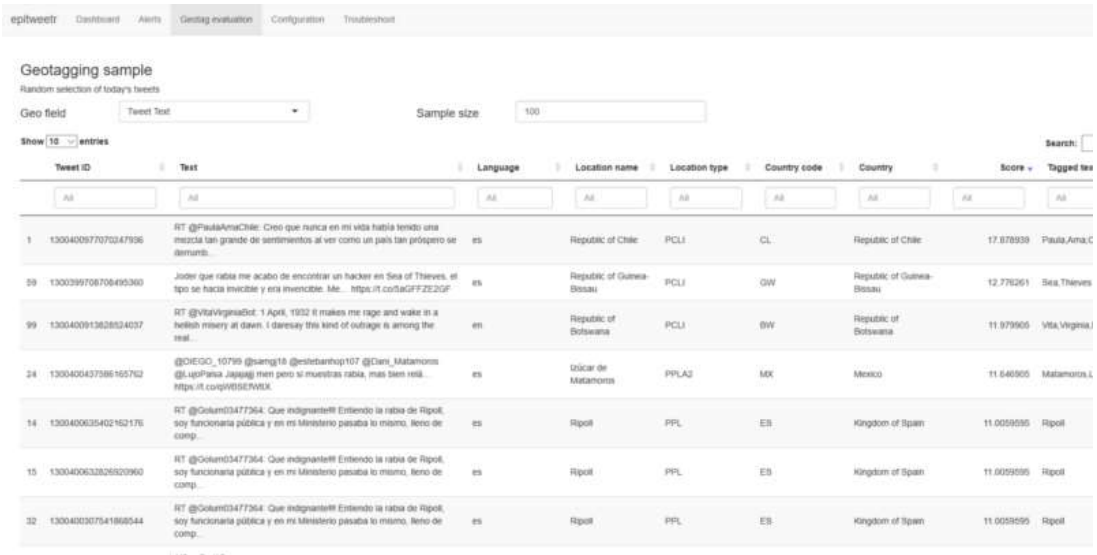
La page «alerts»

La page «alerts» présente, sous une forme sommaire, les signaux détectés pendant la période spécifiée: date, heure, sujet et unité géographique du signal, mots les plus fréquemment employés, nombre total de tweets, nombre de tweets émanant d'utilisateurs importants et seuil. Elle mentionne en outre bon nombre des paramètres définis dans la page «configuration», sur lesquels se fonde la détection des signaux. Cela correspond également au résultat présenté dans les alertes envoyées par courrier électronique.



La page «geotag evaluation»

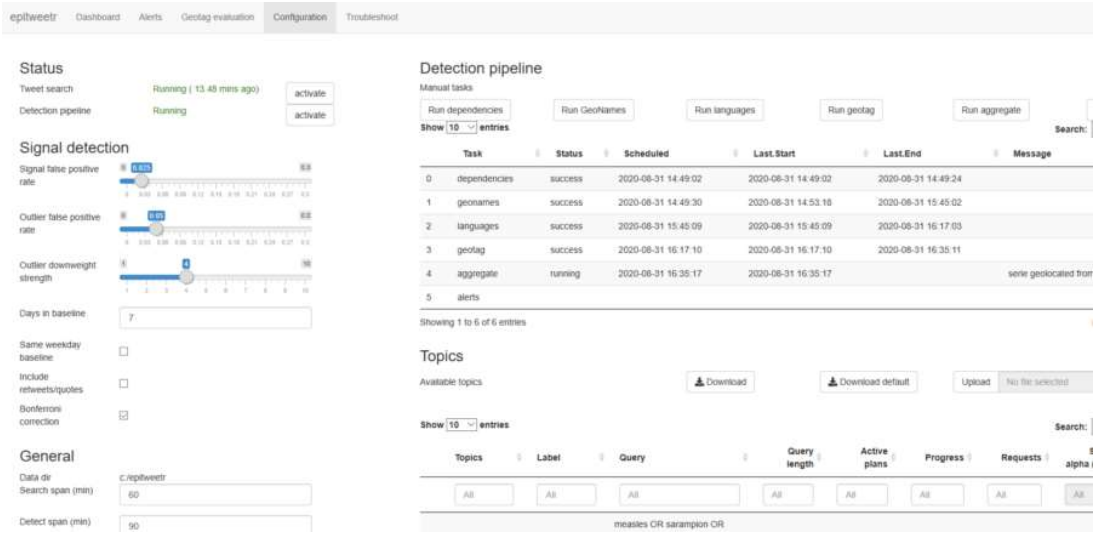
Cette page aide l'utilisateur à définir le seuil de géolocalisation dans la page «configuration». L'utilisateur peut choisir le champ des tweets à tester et le nombre de tweets à échantillonner. Cette page est uniquement destinée à la visualisation et aucune modification de la géolocalisation détectée par `epitweetr` n'est autorisée.



La page «configuration»

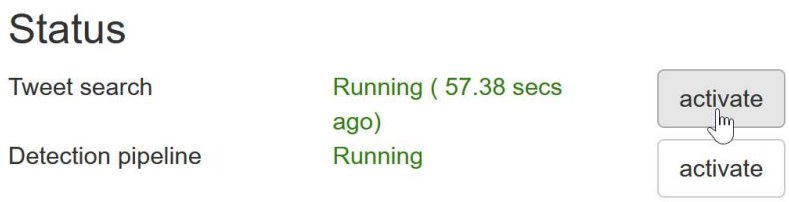
Dans la page «configuration», vous pouvez changer les paramètres de l'outil, vous pouvez vérifier le statut de ses différents processus/pipelines et vous pouvez ajouter, supprimer et modifier les sujets et les requêtes correspondantes, les langues de géolocalisation et la liste des «utilisateurs importants» et des abonnés aux alertes par courrier électronique. Si vous modifiez quoi que ce soit dans les sections «Signal detection» ou «General», n'oubliez pas

de cliquer sur le bouton «Update Properties» à la fin de la section «General». Une description plus détaillée de la page «configuration» est fournie ci-après.



Status

La section «status» vous permet d'évaluer rapidement le dernier point dans le temps et/ou le statut des processus de collecte de tweets (Tweet Search), de géolocalisation, d'agrégation et de détection de signaux (Detection pipeline).



Dans la section «status», vous pouvez vérifier si les processus du pipeline de recherche et du pipeline de détection sont en cours d'exécution. Sous Windows, vous pouvez cliquer sur «activate» pour enregistrer ces processus en tant que tâches planifiées et les exécuter manuellement à partir du planificateur de tâches de Windows.

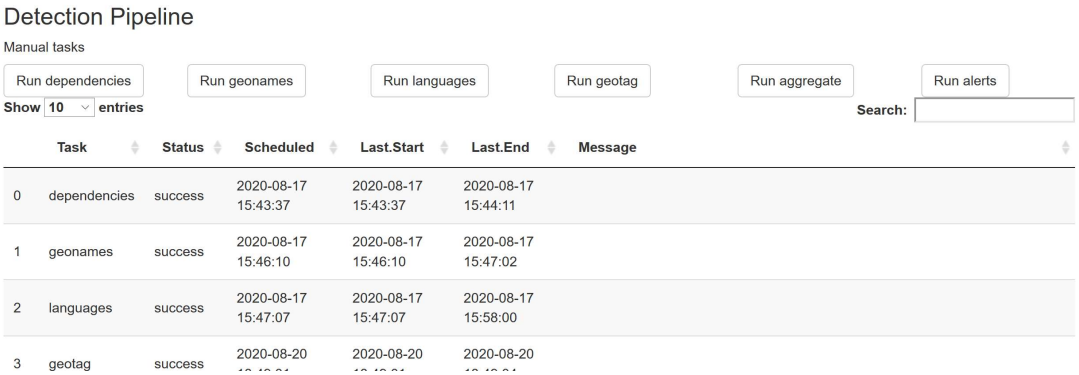
Detection pipeline

Dans les tâches manuelles du «Detection pipeline», lors de la première utilisation d'`epitweetr`, vous devez exécuter manuellement les tâches relatives aux dépendances, aux noms géographique et aux langues en cliquant sur les boutons «Run dependencies», «Run geonames» et «Run languages». Ce n'est à refaire ensuite que si vous téléchargez de nouvelles versions. Les tâches relatives aux noms géographique et aux langues concernent les modèles de géolocalisation et de langue utilisés par `epitweetr`. Si vous souhaitez les mettre à jour (il n'est pas utile de le faire régulièrement, plutôt une fois par an environ), vous pouvez cliquer sur «Run».

Les boutons «Run geotag», «Run aggregate» et «Run alerts» peuvent être utilisés pour forcer le lancement de ces tâches en cas d'erreur ou de problème. Vous pouvez vérifier leur statut dans le tableau «Detection Pipeline».

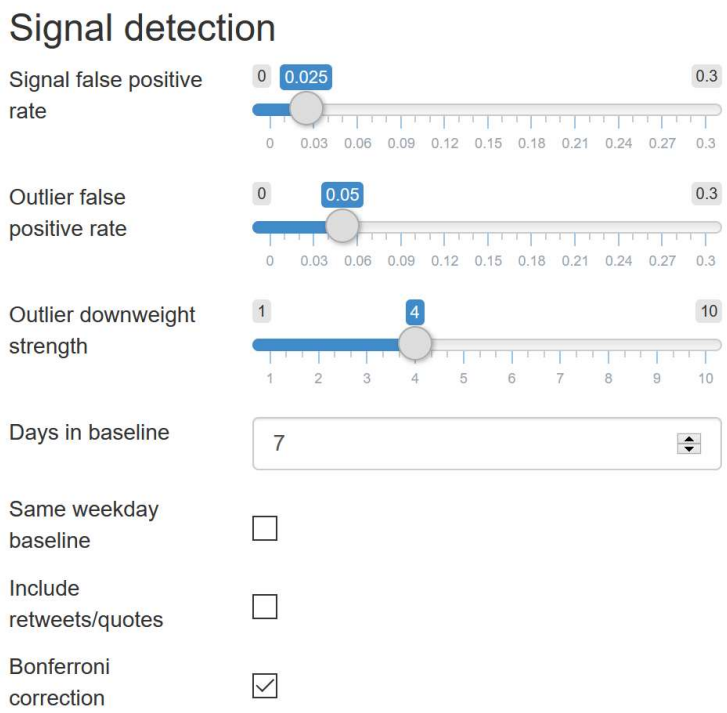
Le pipeline de détection donne plus d'informations sur le statut des processus d'`epitweetr`. C'est utile pour résoudre les problèmes éventuels et suivre le déroulement des cinq tâches qui s'exécutent en arrière-plan. Les tâches relatives aux noms géographique et aux langues servent à télécharger et mettre à jour leurs copies locales. Elles ne se déclencheront que si vous ajoutez une langue ou si vous mettez à jour les noms géographiques. Les dates de début et de fin seront généralement beaucoup plus anciennes que celles des tâches relatives aux balises géographiques, aux agrégations et aux alertes.

Si les pipelines de recherche et de détection sont actifs et fonctionnent, les dates des balises géographiques, des agrégations et des alertes devraient être plus récentes. Ces tâches sont planifiées en fonction de l'intervalle de détection. Leur statut peut être «running», «scheduled», «pending», «failed» ou «aborted» (s'il a échoué plus de trois fois).



Signal detection

Dans la section «signal detection» de la page «configuration», vous pouvez définir le paramètre alpha du taux de faux positifs du signal, qui (s'il est plus grand) augmente l'intervalle de détection (plus de signaux sont détectés), ou (s'il est plus petit) le réduit (moins de signaux sont détectés).



Le taux de faux positifs des valeurs aberrantes se rapporte au taux qui détermine à quoi correspond une valeur aberrante en cas d'ajustement de la pondération des valeurs aberrantes/signaux précédents. Plus il est faible, moins il inclura de valeurs aberrantes précédentes. Un paramètre plus élevée inclura potentiellement davantage de valeurs aberrantes précédentes.

La force d'ajustement de la pondération des valeurs aberrantes détermine dans quelle mesure la pondération d'une valeur aberrante sera ajustée. Plus la valeur est élevée, plus l'ajustement est important. Pour plus d'informations, voir l'annexe I.

`epitweetr` calcule un seuil pour déterminer si le nombre courant de tweets pour une période donnée de 24 heures dépasse ce qui est prévu (voir la section *«Comment ça marche? L'architecture générale sur laquelle repose `epitweetr` > Détection des signaux»*). Ce seuil se fonde sur une valeur par défaut des 7 jours précédents. Vous pouvez modifier le nombre de jours dans le champ «default days in baseline».

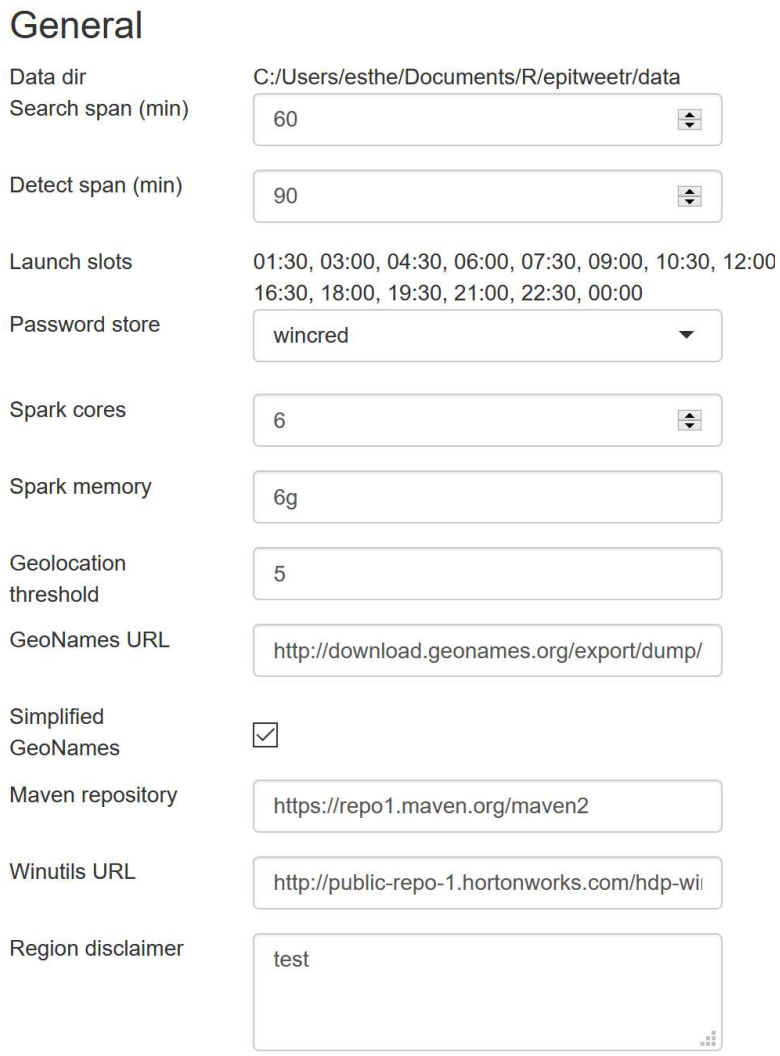
Pour éviter un «effet jour de la semaine» (il se peut que les tweets portant un sujet soient toujours plus nombreux le lundi, par exemple, ce qui pourrait affecter la détection du signal), vous pouvez aussi modifier la valeur par défaut qui consiste à utiliser les 7 jours précédents pour calculer une base de référence et la remplacer par les 7 mêmes jours de la semaine précédents.

Vous pouvez également préciser si la détection des signaux s'applique uniquement au texte des tweets, ou si elle comprend les retweets/citations (cochez la case «Default with retweets/quotes»).

La dernière case à cocher «Default with Bonferroni correction», prend en compte les tests multiples, qui peuvent donner lieu à des faux positifs. Si cette case est cochée, le paramètre alpha de détection des signaux est divisé par le nombre de localisations géographiques où s'effectue la détection des signaux. Par exemple, au niveau national, le paramètre alpha est divisé par le nombre total de pays. Au niveau continental, il est divisé par le nombre total de continents.

Si vous modifiez quoi que ce soit dans la section «Signal detection», n'oubliez pas de cliquer sur le bouton «Update Properties» à la fin de la section «General».

General



- Dans **Data directory**, vous pouvez voir le répertoire utilisé par `epitweetr` pour enregistrer les tweets collectés et les données correspondantes. C'est aussi ce répertoire que le «dashboard» utilise pour obtenir les ensembles de données servant à

l'affichage des visualisations. Vous devez choisir ce dossier lors du lancement d'`epitweetr` ou de la définition de la variable d'environnement «EPI_HOME».

- **Search span** correspond à la durée d'exécution d'un plan de recherche. La valeur par défaut est de 60 minutes. Cette valeur contrôle l'ampleur de la période de recherche des tweets. Si vous la réduisez, vous obtiendrez des tweets plus rapidement, mais vous risquez de «gaspiller» des requêtes sur les sujets ayant très peu de tweets. Si vous l'augmentez, il vous faudra plus de temps pour collecter les tweets mais vous disposerez de plus de requêtes pour les tweets populaires, ce qui augmentera vos chances d'obtenir des résultats exhaustifs. Si vous avez plus d'un plan actif pour certains sujets, vous pouvez voir sur la page «configuration» de Shiny app quand vous n'êtes pas en mesure de collecter les tweets.

- **Detect span** concerne la fréquence à laquelle les processus du pipeline de détection (géolocalisation, agrégation et détection des alertes) sont exécutés. La valeur par défaut est de 90 minutes. Les alertes sont envoyées par courrier électronique à la fin de la boucle de détection. Cette valeur est traitée comme une limite inférieure: la boucle de détection pourrait prendre plus de temps pour se terminer en fonction du volume de tweets et des spécifications de votre système.

- Les **Launch slots** (créneaux de lancement) des processus du pipeline de détection seront espacés en fonction du «Detect span», le premier commençant à minuit. Ces valeurs peuvent être utilisées dans le fichier des abonnés de la page «configuration».

- Pour éviter d'enregistrer les identifiants de Twitter dans des fichiers ordinaires, `epitweetr` utilise une fonctionnalité de stockage de mots de passe dépendant du système, qui se trouve dans le **Password store**. Selon votre système, vous pouvez choisir le mécanisme adapté à l'environnement dans lequel `epitweetr` fonctionne. Pour plus de détails sur chaque implémentation, voir *https://CRAN.R-project.org/package=keyring*

  - wincred: (Windows uniquement) utilise le gestionnaire d'identification de Windows.
  - macos: (MAC uniquement) utilise les services du trousseau de Mac OS
  - file: utilise des fichiers cryptés protégés par un mot de passe
  - secret service: (Linux seulement) utilise le service secret de Linux
  - environment: utilise des variables d'environnement (configuration supplémentaire nécessaire, *https://CRAN.R-project.org/package=keyring*)

- **Spark cores et spark memory:** L'allocation de capacités CPU (Spark cores) et RAM (Spark Memory) pour `epitweetr` est également définie dans la section «general». La valeur par défaut est de 6 cœurs et 6 Go de RAM. Cette valeur dépendra des capacités CPU et RAM de votre machine, auxquelles elle doit être égale ou inférieure.

- **Geolocation threshold:** Au cours du processus de géolocalisation, des ensembles de mots sont traités afin de déterminer les correspondances potentielles avec des lieux existants et de leur attribuer un score. Plus le score est élevé, plus la probabilité que la géolocalisation soit correcte est grande. Un seuil sous lequel toute correspondance est considérée comme insuffisante pour la géolocalisation est fixé dans `epitweetr`. L'échelle va de 1 à 10, et la valeur par défaut est de 5.
- **Geonames URL:** L'URL utilisée pour télécharger la base de données GeoNames (servant à générer des localisation) se trouve dans la section «general». Si cette URL venait à changer, vous pouvez effectuer la modification ici.
- **Simplified geonames:** Comme le fichier GeoNames est très volumineux, le système utilise par défaut une version simplifiée qui ne comprend que des lieux géographiques existants, où la présence d'une population est connue. Vous pouvez décocher cette option si vous souhaitez utiliser l'ensemble de la base de données GeoNames.
- **Maven repository:** Il s'agit de l'URL du dépôt Maven qui sera utilisé pour télécharger les dépendances JAR pour la boucle de détection, principalement Spark et Lucene.
- **Winutils URL:** Il s'agit de l'URL qui sera utilisée pour télécharger winutils.exe, un binaire nécessaire pour exécuter Spark localement sous Windows. Si vous ne souhaitez pas utiliser cette version, vous pouvez la produire vous-même en téléchargeant Hadoop 2.8.4 ou supérieur, que vous compilerez sur une machine Windows.
- **Region disclaimer:** Si vous souhaitez ajouter un avertissement concernant la carte que vous utilisez. Cet avertissement est ajouté aux images de la carte ainsi qu'aux PDF exportés à partir du «dashboard».

Twitter authentication

Vous disposez de deux options d'authentification pour collecter les tweets: au moyen d'un compte Twitter (en utilisant le paquet rtweet) et au moyen d'une application de développeur Twitter. Vous pouvez sélectionner l'option que vous utiliserez dans la section **Twitter authentication**. Voir la section *«Comment ça marche? L'architecture générale sur laquelle repose `epitweetr` > Collecte de tweets > Authentification sur Twitter»* pour plus de précisions sur la manière de procéder.

Email authentication (SMTP)

Dans cette section, vous devez indiquer les informations d'authentification (SMTP) pour l'adresse de courrier électronique qui enverra les alertes.

Si la case **Unsafe certificates** est cochée, `epitweetr` utilisera votre serveur SMTP même si celui-ci envoie un certificat non valide.

Si vous modifiez quoi que ce soit dans la section «General», n'oubliez pas de cliquer sur le bouton «Update Properties».

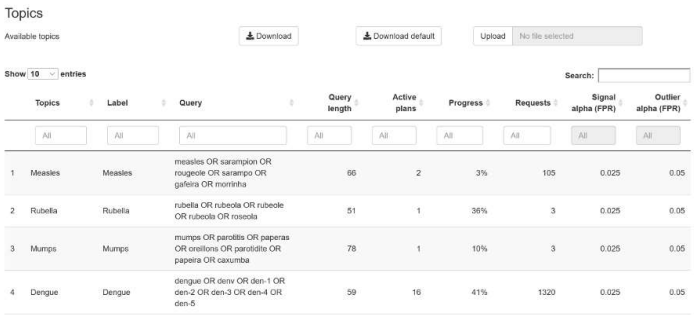
Topics

Ce sont les sujets qui déterminent quels sont les tweets collectés À cet effet, `epitweetr` utilise une feuille de calcul Excel qui contient les sujets et les recherches associées pour interroger l'API de Twitter.

Une requête consiste dans des mots-clés et des opérateurs qui servent à trouver des correspondances avec les attributs des tweets. Voir la section *«Comment ça marche? L'architecture générale sur laquelle repose `epitweetr` > Collecte de tweets > Sujets et requêtes pour la collecte de tweets»* pour plus de précisions sur les requêtes.

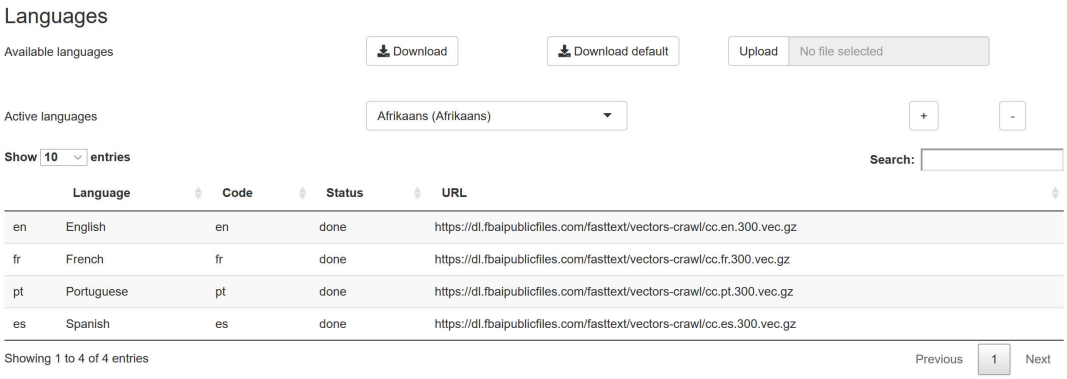
`epitweetr` est fourni avec une liste de sujets par défaut, qui sont ceux utilisés par l'équipe de veille sanitaire de l'ECDC à la date de création du paquet (1er septembre 2020). Vous pouvez télécharger cette liste et charger la vôtre dans la section «Available Topics» de la page «configuration». Voir la section *«Comment ça marche? L'architecture générale sur laquelle repose `epitweetr` > Collecte de tweets > Sujets et requêtes pour la collecte de tweets»* pour plus de précisions sur la structure à respecter dans la liste de sujets.

Dans la section «topics» de la page «configuration», vous pouvez vérifier les sujets, les requêtes associées, la longueur des requêtes et le nombre de plans de recherche actifs associés aux requêtes. Si plusieurs plans de recherche sont actifs, cela signifie qu'`epitweetr` n'a pas pu collecter tous les tweets possibles lors de la dernière session. En outre, vous pouvez voir la progression et le nombre de recherches effectuées depuis le dernier plan.



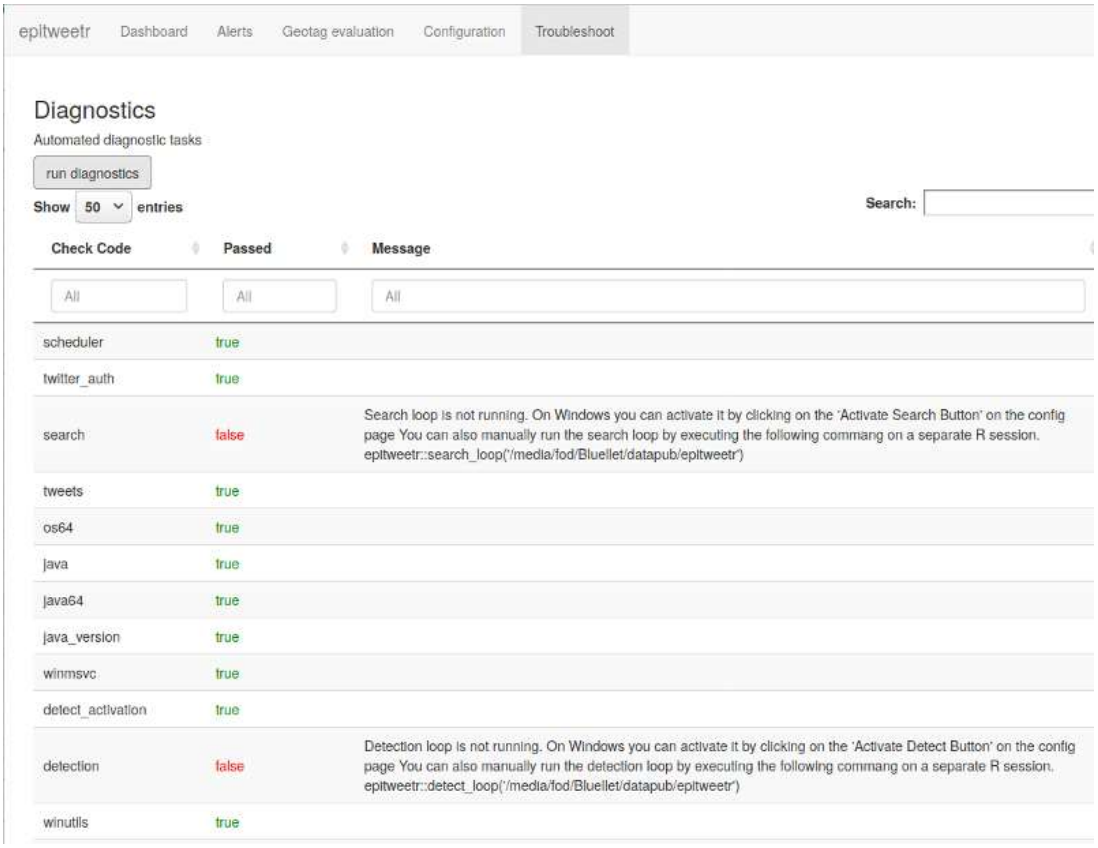
Languages

Dans la section des langues, vous pouvez déterminer quels modèles de langue sont utilisés pour identifier le texte au cours du processus de géolocalisation. Les langues par défaut sont le français, l'anglais, le portugais et l'espagnol. Vous pouvez télécharger des modèles de langues et les charger dans la section «Available Languages» et ajouter des langues à utiliser par `epitweetr` ou en supprimer dans la section «Active Languages». Veuillez tenir compte de la charge de calcul liée à l'ajout d'un trop grand nombre de langues, en fonction des capacités de votre machine.



La page «troubleshoot»

La page «troubleshoot» contient une liste de contrôles automatiques et des suggestions pour utiliser `epitweetr` et toutes ses fonctionnalités. Cliquez sur «Run diagnostics» pour voir la liste des contrôles, vérifier si le processus correspondant est actif («true») ou non («false»), et consulter les suggestions en cas d'erreur. Des informations plus détaillées sont disponibles dans l'annexe II de cette documentation à l'usage des utilisateurs.



Téléchargement des résultats à partir de l'interface utilisateur interactive (Shiny app)

Chaque visualisation du «dashboard» de Shiny app peut être téléchargée sous forme d'image, en utilisant le bouton «image». Un .png (portable network graphic) est un fichier au format polyvalent qui convient pour des images n'ayant pas besoin d'une très haute résolution (comme c'est le cas pour les graphiques destinés à une impression professionnelle).

Il est à noter que le format .png n'est pas supporté par le navigateur Internet Explorer (mais vous pouvez télécharger un fichier .svg à la place).

Vous pouvez également télécharger les données de chaque visualisation en cliquant sur le bouton «data». Vous obtiendrez ainsi un fichier .csv contenant les données sous-jacentes que vous pourrez utiliser pour une analyse plus approfondie ou pour créer vos propres graphiques.

Sinon, vous pouvez utiliser les boutons «PDF» ou «Md» au bas des filtres pour télécharger une copie du «dashboard» au format PDF ou HTML. Pour cela, vous devez avoir installé MiKTeX ou TinyTeX.

Annexe I: Ajustement de la pondération des signaux précédents

Introduction

Dans cette annexe, nous proposons une méthode d'ajustement de la pondération intégrée à l'algorithme `eears` utilisé dans le paquet `epitweetr` qui a été décrite ci-dessus.

Soit le vecteur des valeurs historiques $\mathbf{y}$ qui est de longueur n. Une partie du calcul de l'intervalle de prédiction au temps 0 consiste dans le calcul de la moyenne et de l'écart-type de ces valeurs historiques, c'est à dire

$$\overline{y}_0 = \frac{1}{n}\sum_{t=-n}^{-1} y_t \quad \text{et} \quad s_0^2 = \frac{1}{n-1}\sum_{t=-n}^{-1}(y_t - \overline{y}_0)^2$$

La limite supérieure de l'intervalle de prédiction unilatéral $(1-\alpha)\times 100\%$ pour l'observation y_0 dans un modèle $y_t \overset{\text{iid}}{\sim} N(\mu, \sigma^2), t = -n, \ldots, 0$ est alors calculée comme suit

$$U_0 = \overline{y}_0 + t_{1-\alpha}(n-1) \times s_0 \times \sqrt{1+\frac{1}{n}},$$

où $t_{1-\alpha}(n-1)$ désigne le quantile $1-\alpha$ de la distribution t avec $n-1$ degrés de liberté. Ce calcul du seuil correspond à un calcul statistiquement fiable du seuil (Allévius et Höhle, 2017).

Une extension souhaitée de l'algorithme ci-dessus consiste à traiter les signaux précédents dans les valeurs historiques. Ce problème a déjà été abordé dans le cadre quasi-Poisson de Farrington et al. (1996), en effectuant d'abord un ajustement MLG, puis en réajustant le

MLG avec des pondérations basées sur les valeurs résiduelles d'Anscombe. Nous suivons la même idée générale, mais nous l'adaptons à la réponse gaussienne utilisée dans l'algorithme EARS et aux valeurs résiduelles correspondantes du modèle linéaire.

EARS comme modèle linéaire

Nous observons d'abord que l'estimation ci-dessus de μ et σ^2 par $\overline{y}_0$ et s_0^2 au temps 0 peut être intégrée dans un modèle de régression linéaire, c'est-à-dire que pour $i = 1, \dots, n$, nous prenons le modèle

$$y_i = \mu + \epsilon_i, \quad \text{où} \quad \epsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2).$$

Il est à noter que, par souci de compatibilité avec l'exposition standard de la théorie des modèles linéaires, nous avons indexé les valeurs y de telle sorte que: y_{-n} corresponde à y_1 et y_{-1} correspond à y_n. En termes de matrice, soit $\mathbf{y} = (y_1, \dots, y_n)'$ et pour le modèle à interception seule, la matrice de conception est $\mathbf{X} = (1, \dots, 1)'$, qui a un rang $k = 1$. Donc, d'après la théorie standard des MCO:

$$\hat{\mu} \quad = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \frac{1}{n}\sum_{i=1}^{n} y_i\,,$$

ce qui correspond à $\overline{y}_0$. En outre, définissons les valeurs résiduelles brutes comme $e_i = y_i - \hat{\mu}$ pour $i = 1, \dots, n$ et désignons par $\mathbf{e} = (e_1, \dots, e_n)'$ le vecteur de valeurs résiduelles correspondant. Donc

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{P}\mathbf{y} = (\mathbf{I} - \mathbf{P})\mathbf{y}$$

où $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ est la matrice «chapeau», connue d'après la modélisation linéaire. Avec cette notation, nous pouvons écrire l'estimation pour σ^2 comme dans Chatterjee et Hadi (1988):

$$\hat{\sigma}^2 = \frac{\mathbf{e}'\mathbf{e}}{n-k} = \frac{\mathbf{y}'(\mathbf{I}-\mathbf{P})\mathbf{y}}{n-k} = \frac{1}{n-1}\sum_{t=-7}^{1}(y_t - \hat{\mu})^2,$$

ce qui correspond à l'expression utilisée ci-dessus pour s_0^2.

Ajustement de la pondération

Nous calculons maintenant les **«valeurs résiduelles studentisées de manière externe»** (Chatterjee et Hadi, 1988).

$$r_i^* = \frac{e_i}{\hat{\sigma}_{(i)}\sqrt{1-p_{ii}}}, \quad i = 1, \dots, n,$$

où p_{ii} est le i-ième élément diagonal de la matrice chapeau $\mathbf{P}$ du modèle linéaire correspondant utilisé ci-dessus. En outre,

$$\hat{\sigma}^2_{(i)} = \frac{\mathbf{y}_{(i)}{}'(\mathbf{I} - \mathbf{P}_{(i)})\mathbf{y}_{(i)}}{n-k-1}$$

est l'estimation de la variance obtenue à partir d'une régression linéaire, dont on retire la i-ième observation. La théorie de la modélisation linéaire (Chatterjee et Hadi 1988) indique maintenant que

$$r_i^* \overset{\text{identique}}{\sim} t(n-k-1).$$

Il est à noter que les valeurs résiduelles sont distribuées uniquement de manière identique, car elles ne sont pas indépendantes [voir la section 4.2.1. de Chatterjee et Hadi (1988) pour plus de détails]. Toutefois, la forme de distribution ci-dessus nous permet d'évaluer, pour chaque valeur historique, si elle peut être considérée comme une valeur aberrante. À cet effet, définissons r_{seuil} comme le quantile $1 - \alpha_{\text{valeur aberrante}}$ de la distribution t avec $n-k-1$ degrés de liberté. Une valeur historique est une valeur aberrante (dont une explication possible est qu'elle résulte d'une réelle augmentation du nombre des tweets, par exemple une situation d'épidémie), si $r_i^* > r_{\text{seuil}}$. Nous nous en servirons pour formuler un mécanisme de pondération des valeurs historiques:

Ajustement de la pondération des valeurs aberrantes:

$$\begin{aligned} w_i^{(\text{dw})} &= \begin{cases} 1 & \text{si } r_i^* < r_{\text{seuil}} \\ \left(\frac{r_{\text{seuil}}}{(r_i^*)}\right)^k & \text{sinon} \end{cases} \\ &= \min\left\{1, \left(\frac{r_{\text{seuil}}}{r_i^*}\right)^k\right\}, \end{aligned}$$

où le paramètre decay $k > 0$ est une quantité connue. Dans l'algorithme original de Farrthe et al. (1996), $k = 2$ était utilisé. En outre, une valeur seuil de 1 était utilisée. Ultérieurement, cependant, dans l'article de Noufaily et al. (2013), une valeur seuil de 2,58 a été recommandée. Note: les deux valeurs correspondent aux valeurs résiduelles normalisées d'Anscombe, qui suivent une distribution normale standard. Si nous prenons les quantiles correspondants pour la distribution t avec 6 degrés de liberté, les valeurs seraient de 1,09 et 3,72. Il est aussi à noter que le terme $(r_{\text{seuil}}/r_i^*)^k$ est une légère adaptation de Farrington et al. (1996), qui utilise plutôt $1/(r_i^*)^2$. L'avantage de notre proposition est qu'elle assure un traitement stable des valeurs autour du seuil si celui-ci n'est pas égal à 1. Il pourrait être utile d'envisager une puissance supérieure à 2 pour garantir un ajustement encore plus fort de la pondération des valeurs aberrantes les plus flagrantes. La valeur par défaut courante du paramètre decay dans `epitweetr` est de 4.

Enfin, comme dans Farrington et al. (1996), nous normalisons les pondérations de telle sorte qu'ils donnent une somme de n au moyen de

$$w_i^* = n \times \frac{w_i}{\sum_{i=1}^n w_i}$$

et nous réajustons ensuite le modèle linéaire avec ces pondérations. À cet effet, définissons la matrice de pondération comme $\mathbf{W} = \text{diag}(w_1^*, \ldots, w_n^*)$. Nous pouvons utiliser ultérieurement une méthode des moindres carrés pondérés pour trouver

$$\hat{\mu}_W \quad = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{y} = \frac{1}{n}\sum_{i=1}^{n} w_i^* \, y_i,$$

où le deuxième signe égal s'explique parce que $(\mathbf{X}'\mathbf{W}\mathbf{X}) = \sum_{i=1}^{n} w_i = n$ et $\mathbf{X}'\mathbf{W}\mathbf{y} = \sum_{i=1}^{n} w_i^* \, y_i$. En outre,

$$s_W^2 = \frac{\mathbf{y}'(\mathbf{I} - \mathbf{P}_W)\mathbf{y}}{n-k} = \frac{\sum_{i=1}^{n} w_i^* \, (y_i - \mu_W)^2}{n-1},$$

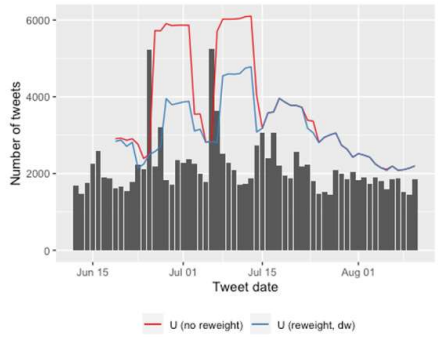
où $P_W = \mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}\mathbf{W}$ est la matrice chapeau des moindres carrés pondérés.

La procédure à la pondération ajustée fonctionne donc avec μ_W et s_W^2 au lieu de $\overline{y}_0$ et s_0^2, respectivement, lors du calcul de la limite supérieure U_0 à l'aide de la formule susmentionnée.

Exemple de la méthode d'ajustement de la pondération utilisant les données d'Ebola

La figure 5 ci-dessous montre la limite supérieure du seuil de détection des signaux pour les données d'Ebola dans `epitweetr`, à la fois le seuil original sans ajustement de la pondération (en rouge) et le seuil supérieur avec ajustement de la pondération (en bleu) après prise en compte des signaux précédents dans les valeurs historiques. Il est à noter que le seuil supérieur avec ajustement de la pondération détecte trois signaux supplémentaires, par rapport au seuil original.

Fig. 5: Limite supérieure avec et sans ajustement de la pondération pour les données d'Ebola dans `epitweetr`



Annexe II: Résolution des problèmes et conseils

La présente annexe contient quelques suggestions et des solutions courantes aux erreurs ou aux problèmes que les utilisateurs d'`epitweetr` peuvent rencontrer, y compris des explications sur les contrôles figurant dans la page «troubleshoot».

La page «troubleshoot»

Après avoir exécuté les diagnostics dans la page «troubleshoot», vous pouvez voir les contrôles effectués et le statut relatifs aux aspects suivants:

- **scheduler**: le paquet R `taskscheduleR` est installé. Applicable uniquement sous Windows
- **twitter_auth**: L'identifiant Twitter a été créé après l'authentification avec le compte Twitter ou l'application de développeur Twitter.
- **search_running**: la tâche de recherche est en cours d'exécution
- **tweets**: les tweets ont été collectés
- **os64**: R est en 64 bits
- **java**: Java est installé et accessible à `epitweetr`
- **java64**: Java 64bits est installé et accessible à `epitweetr`
- **java_version**: la version de Java installée est compatible avec `epitweetr`
- **winmsvc**: Microsoft Visual C++ 2010 SP1 Redistributable Package est installé. Applicable uniquement sous Windows
- **detect_activation**: la boucle de détection a été activée
- **detection_running**: la tâche de détection est en cours d'exécution
- **winutils**: winutils est installé. Si ce n'est pas le cas, il peut être téléchargé en exécutant la tâche de mise à jour des dépendances. Applicable uniquement sous Windows
- **java_deps**: les dépendances de java sont installées
- **move_from_temp**: `epitweetr` peut déplacer de manière autonome des fichiers du dossier temporaire au répertoire de données
- **geonames**: la base de données de Geonames.org est téléchargée et indexée
- **languages**: les vecteurs de langues sont téléchargés et indexés
- **geotag**: la tâche geotag a été exécutée avec succès
- **aggregate**: la tâche agregate a été exécutée avec succès
- **alerts**: des alertes ont été créées
- **pandoc**: Pandoc est installé et accessible à `epitweetr`. Nécessaire pour la création de PDF.
- **tex**: une distribution tex est installée et accessible à `epitweetr`. Nécessaire pour la création de PDF.

Gestion des boucles de recherche et de détection (Windows)

Après avoir activé le pipeline de recherche et de détection à partir de la page «configuration» d'`epitweetr` (Windows), deux tâches seront créées dans le planificateur de tâches et deux fenêtres de terminal Windows s'ouvriront. Veuillez noter que si l'ordinateur est déconnecté/éteint ou si les fenêtres du terminal sont fermées, le pipeline de recherche et de détection s'arrête.

Si vous activez à nouveau ces tâches à partir de la page «configuration» d'`epitweetr`, le système écrasera les tâches créées dans le planificateur de tâches. En revanche, après la première activation réussie de ces tâches à partir d'`epitweetr`, vous pouvez facilement les

gérer à partir du planificateur de tâches. Vous pouvez arrêter ces tâches en les terminant et en les désactivant dans le planificateur de tâches, et vous pouvez les redémarrer en les activant et en les exécutant dans le planificateur de tâches.

Dans le planificateur de tâches, vous pouvez cocher l'option «exécuter même si aucun utilisateur n'a ouvert de session» pour éviter que les tâches de recherche et de détection ne s'arrêtent lorsque vous vous déconnectez ou redémarrez l'ordinateur.

Gestion des boucles de recherche et de détection (Linux et Mac)

Étant donné que le pipeline de recherche et de détection sous Linux ou Mac doit être exécuté manuellement, si l'ordinateur est déconnecté/éteint ou si les fenêtres du terminal sont fermées, le pipeline de recherche et de détection s'arrête. N'oubliez pas de suivre les étapes de la section *Mettre en place la collecte de tweets et la boucle de détection des alertes* pour exécuter à nouveau ces tâches.

Exécuter le pipeline de recherche et de détection

«Cannot execute task #####: the task is already running»

La boucle de détection et de recherche crée deux fichiers contenant leurs ID de processus situés dans le dossier «data» d'`epitweetr`: search.PID et detect.PID. Cette erreur se produit si `epitweetr` trouve un autre processus R en cours d'exécution avec la même ID. Pour y remédier, vous devez d'abord vérifier si la boucle de recherche/détection n'est pas déjà en cours d'exécution dans une autre session R. Si c'est le cas, vous ne devez pas essayer de lancer la tâche car `epitweetr` ne prend en charge qu'une seule instance de la même tâche dans le même dossier de données. Si le processus en cours d'exécution n'est pas associé à la tâche, vous pouvez supprimer manuellement le fichier PID et essayer de le relancer.

Erreur lors de la tentative d'agrégation de fichiers

Cette erreur peut être due à deux raisons: - Pas assez d'espace sur le disque pour les fichiers temporaires. L'agrégation crée des fichiers temporaires avant de les enregistrer dans le dossier d'`epitweetr` correspondant. Si c'est le cas, modifiez la variable d'environnement de votre compte pour mettre TMP et TEMP dans un autre emplacement disposant de plus d'espace. - Si l'erreur apparaît lors de la création d'un fichier en particulier, il se peut qu'il y ait un fichier de série corrompu pour cette date. Supprimez les fichiers «country_counts», «geolocated» et «topwords» pour cette date et relancez manuellement la tâche en cliquant sur le bouton correspondant dans la page «configuration».

Changer l'utilisateur de l'authentification sur Twitter avec un compte Twitter

1. Terminez et désactivez la boucle/tâche de recherche dans le planificateur de tâches (Windows) ou fermez la fenêtre R/terminal contenant la boucle/tâche de recherche (Linux et Mac).

2. Recherchez un fichier appelé «.rtweet_token» dans les fichiers cachés. Il est généralement enregistré dans le dossier Documents.

3. Supprimez ce fichier.

4. Cliquez sur «Update properties» dans la page «configuration» d'`epitweetr`.

5. Activez et exécutez la boucle/tâche de recherche dans le planificateur de tâches (Windows) ou lancez la commande dans une nouvelle fenêtre R/terminal pour y exécuter la boucle/tâche de recherche (Linux et Mac). Pour plus de précisions, voir la section *Mettre en place la collecte de tweets et la boucle de détection des alertes*.

Télécharger GeoNames et/ou des langues

«The specified size exceeds the maximum representable size. Error: Could not create the Java Virtual Machine»

Si cette erreur apparaît lors de l'exécution de GeoNames, cela signifie que la machine tourne avec Java 32bits. Vous devez installer Java 64bits et le rendre accessible à `epitweetr` en définissant la variable d'environnement «JAVA_HOME» ou en indiquant le bon binaire java dans le PATH du système.

«Max number of retried reached failed while processing languages. Error in get_geolocated_period(dataset): To aggregate, or calculate alerts geolocation must have been successfully executed, but no geolocation files were found»

Si vous voyez cette erreur dans la page «configuration», cela signifie qu'`epitweetr` n'a pas pu géolocaliser les tweets collectés. Il est fortement recommandé d'exécuter à nouveau les tâches relatives à GeoNames et aux langues, car il se peut que les fichiers correspondant n'aient pas été correctement téléchargés. Lorsque vous exécutez ces tâches, vous devez veiller à ce que la machine ne se déconnecte/s'éteigne pas ou à ce qu'elle ne passe pas en mode veille.

Les «Launch slots» de la page de configuration affichent «NA» au lieu des créneaux horaires.

Si c'est la première fois que vous installez et lancez `epitweetr`, la tâche «geotag» du pipeline de détection doit être exécutée au moins une fois pour que des créneaux horaires apparaissent dans les «Launch slots» de la page «configuration».

Télécharger une copie du «dashboard» au format PDF

«Error in: LaTeX failed to compile C:\Users\name~1\...\file######.tex.»

Cette erreur apparaît sous Windows lorsque vous cliquez sur «PDF» dans le «dashboard» et qu'aucun PDF n'est enregistré. Cela signifie que le chemin vers les variables d'environnement TEMP et TMP de l'utilisateur est trop long: Windows raccourcit le chemin et `epitweetr` ne peut pas trouver ce nouveau chemin. Suivez les étapes suivantes pour résoudre ce problème:

1. Ouvrez «environment variable for your account»

2. Remplacez le chemin d'accès à TEMP et TMP par un chemin plus court (par exemple, «C:\Temp»). Le même chemin doit être utilisé pour les deux variables d'environnement.

3. Déconnectez-vous et reconnectez-vous

4. Vous pouvez maintenant télécharger et enregistrer le PDF à partir du «dashboard».

«Error: pandoc document conversion failed with error 6»

1. Téléchargez ce script (https://raw.githubusercontent.com/jgm/pandoc/master/macos/uninstall-pandoc.pl)

2. Désinstallez pandoc (https://pandoc.org/installing.html) en exécutant la commande perl uninstall-pandoc.pl

Totaux différents dans les résultats du «dashboard»

Lors du comptage du nombre total de tweets dans le «dashboard» de Shiny app ou dans les données téléchargeables, vous pouvez obtenir des différences dans les nombres totaux de tweets entre les trois résultats. Cela peut être dû aux raisons suivantes:

1. World (all) et World (geolocated)
    - L'option par défaut pour les régions est World (all), ce qui signifie que les tweets non géolocalisés sont inclus dans la courbe de tendance, mais que seuls les tweets géolocalisés peuvent être visualisés dans les cartes et le nombre des mots les plus fréquemment utilisés, donc le total des tweets peut différer entre ces résultats selon que l'option World (all) ou l'option vide par défaut est sélectionnée.
2. Analyse spécifique à un pays
    - Si vous sélectionnez un seul pays dans les filtres, la courbe de tendance affichera tous les tweets pour ce pays, mais la carte affichera les tweets à un niveau infranational. Il se peut que certains tweets aient été géolocalisés comme provenant d'un certain pays, sans autres données infranationales. Ces tweets seront alors visibles dans le total de la courbe de tendance, mais pas dans les bulles infranationales de la carte.
3. Mots les plus fréquents
    - À la différence des autres résultats du «dashboard», le nombre des mots les plus fréquents est toujours basé sur la localisation du tweet, quel que soit le filtre (pour des raisons de capacités de la mémoire). Par conséquent, si la localisation de l'utilisateur ou les deux types de localisation sont sélectionnés dans le filtre, ce nombre peut afficher un total différent des deux autres résultats.

Réception des alertes en temps réel uniquement

Cela concerne les utilisateurs qui ont sélectionné des sujets et/ou des régions pour recevoir les alertes correspondantes en temps réel ou à intervalle planifié. Si, dans ces cas, vous ne recevez que les alertes en temps réel avec tous les sujets et toutes les régions, il se peut qu'aucun créneau horaire n'ait été inclus dans le fichier des abonnés à partir de la page «configuration». Ces créneaux horaires servent pour les alertes planifiées et si aucun créneau n'est indiqué dans le fichier, les alertes concernant tous les sujets et régions sont envoyées en tant qu'alertes en temps réel.

Non-réception des alertes par courrier électronique

Si vous ne recevez pas d'alertes par courrier électronique et que vous voyez une erreur dans `epitweetr` mentionnant une connexion refusée, cela signifie que le système n'a pas pu se connecter au compte de courrier électronique indiqué dans la page «configuration». Il peut y avoir plusieurs raisons à cela:

- Le serveur ou le port indiqués dans la page «configuration» sont incorrects.
- La tentative de connexion d'`epitweetr` au compte de courrier électronique est bloquée par le serveur. Cela peut se produire avec certains comptes de courrier électronique. Dans ce cas, veuillez contacter le département informatique de votre organisation.
- Si vous utilisez un compte Gmail, vous devez autoriser les applications moins sécurisées dans les paramètres de votre compte.

Références

Allévius, Benjamin, et Michael Höhle, 2017, «Prospective Detection of Outbreaks», *arXiv:1711.08960 [Stat]*, novembre, https://arxiv.org/abs/1711.08960.

Chatterjee, Samprit, et Ali S. Hadi, 1988, *Sensitivity Analysis in Linear Regression*, Wiley Series in Probability and Mathematical Statistics, New York, Wiley.

Farrington, C. P., N. J. Andrews, A. D. Beale, et M. A. Catchpole, 1996, «A Statistical Algorithm for the Early Detection of Outbreaks of Infectious Disease», *Journal of the Royal Statistical Society,. Series A (Statistics in Society)* 159 (3), p. 547, https://doi.org/10.2307/2983331.

Fricker, Ronald D., Benjamin L. Hegler, et David A. Dunfee, 2008, «Comparing Syndromic Surveillance Detection Methods: EARS' Versus a CUSUM-Based Methodology», *Statistics in Medicine* 27 (17), p. 3407-3429. https://doi.org/10.1002/sim.3197.

Noufaily, Angela, Doyo Enki, Paddy Farrington, Paul Garthwaite, Nick Andrews, et Andre Charlett, 2013, «An Improved Algorithm for Outbreak Detection in Multiple Surveillance Systems», *Online Journal of Public Health Informatics* 5 (1): e148, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3692796/.

Salmon, Maëlle, Dirk Schumacher, et Michael Höhle, 2016, «Monitoring Count Time Series in *R* : Aberration Detection in Public Health Surveillance», *Journal of Statistical Software* 70 (10). https://doi.org/10.18637/jss.v070.i10.