ASSELIN Anouk
BATTAGLIERI Ornella
SOUNAC Nicolas

Projet R

ANALYSE DU WEB

12/10/2015

Introduction

- R n'est pas qu'un outil statistique
- R et analyse du Web
  - accès aux données
  - extraction de données
- Développement de packages
- Domaines divers et variés
  - Météorologie
  - Agriculture
  - Politique
  - Sport
  - …
- **Objectif** : analyser les données du Web

Sommaire

- Introduction
- 1. Extraction de données avec rvest
- 2. Analyse des réseaux sociaux
  - Présentation
  - Package Rfacebook
  - Démonstration
- 3. Analyse de sites Web (RGoogleAnalytics)
  - Présentation
  - Package RGoogleAnalytics
  - Avantages/Inconvénients
- Conclusion

1. Extraction de données avec rvest

- Package dans la famille des packages de « Web scraping »
  - XML & httr: packages fournissant les outils basiques pour manipuler les documents et protocoles web (inclus dans rvest)
  - scrapeR: permet la vectorisation mais package « pauvre »
  - RCurl: complet et performant mais très complexe et nécessite une maîtrise préalable de libcurl

Rvest se place comme une alternative simple d'accès à RCurl

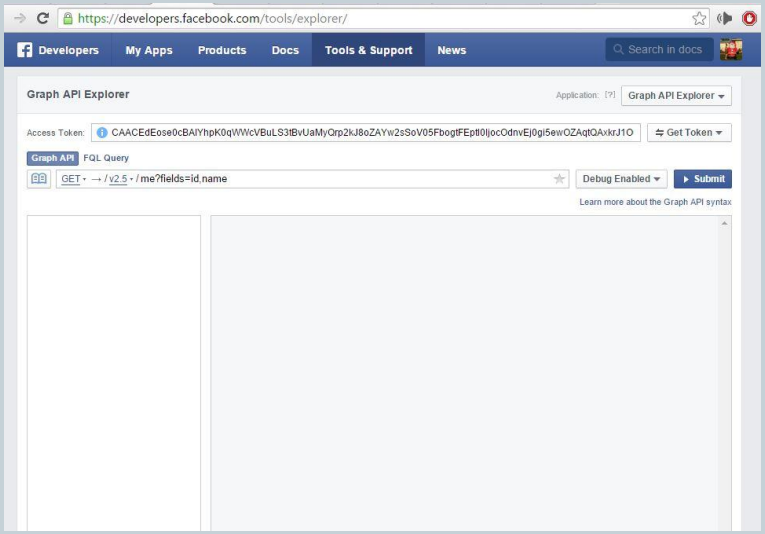
1. Extraction de données avec rvest

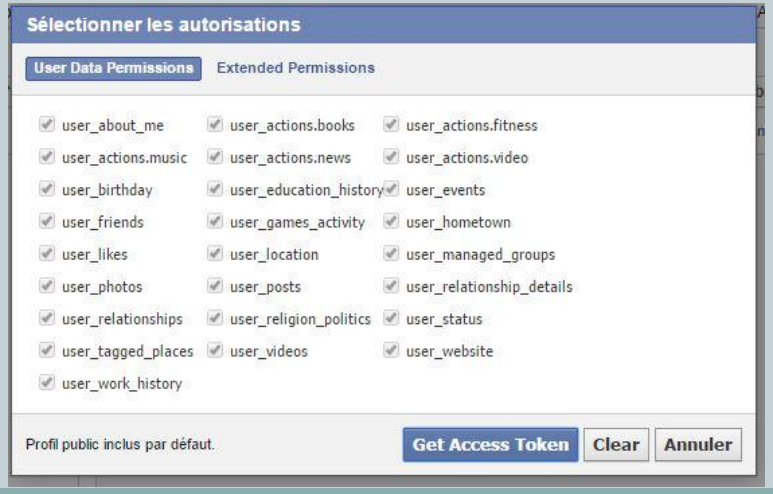
- Deux types principaux de fonctionnalités:

Extraction de données à partir du code HTML/XML d'une page Web	Remplissage et soumission de requête sur le Web
• Récupération du code source d'une page Web à partir de son adresse URL • Extraction des données par sélecteurs • Nettoyage des données extraites	• Simulation d'une session Internet • Navigation dans la session • Récupération, remplissage et soumission des requêtes de la session

2. Analyse des réseaux sociaux

- Accès et utilisation des données de réseaux sociaux
- Exemples : Facebook, Twitter, LinkedIn
- Accès obligatoire via API
  - Authentification
  - Création d'une application
  - Autorisation via access_token
  - https://developers.facebook.com/tools/explorer/





2. Analyse des réseaux sociaux

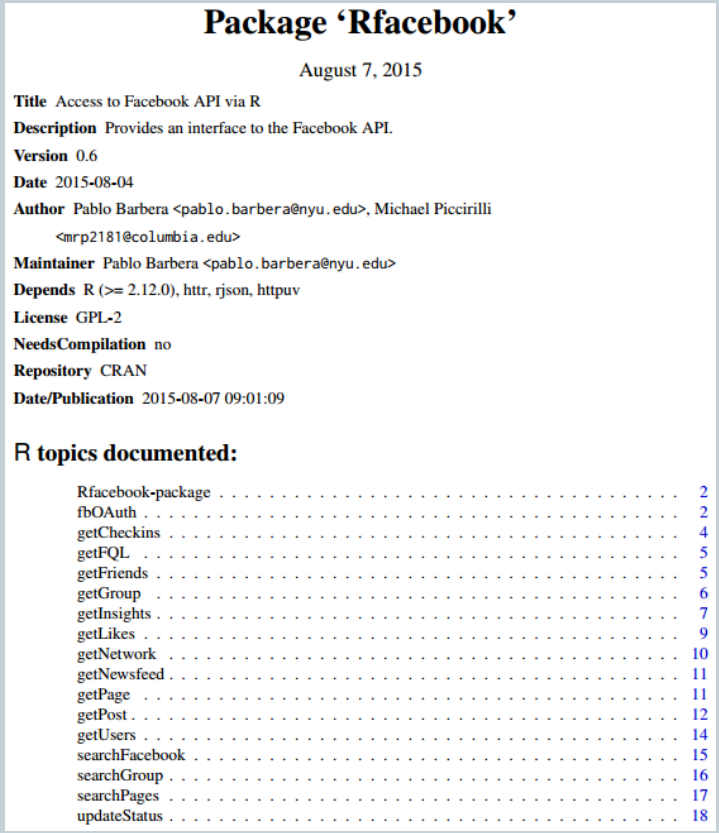
- Difficulté : confidentialité et protection des données
  - Exemple : version 2.0 Graph API de Facebook
  - Amélioration protection des données

Details

This function requires the use of a OAuth token with extended permissions. After the introduction of version 2.0 of the Graph API, only friends who are using the application that you used to generate the token to query the API will be returned.

2. Analyse des réseaux sociaux

- Présentation du package Rfacebook
  - MAJ récentes
  - Fonctions obsolètes (API 2.0)
  - Utilisation des fonctions
    - fbOAuth
    - getFriends
    - getLikes
    - getNetwork
    - getPage
    - getPost
    - getUsers
    - searchPages
    - updateStatus



Package 'Rfacebook'

August 7, 2015

Title Access to Facebook API via R

Description Provides an interface to the Facebook API.

Version 0.6

Date 2015-08-04

Author Pablo Barbera <pablo.barbera@nyu.edu>, Michael Piccirilli <mrp2181@columbia.edu>

Maintainer Pablo Barbera <pablo.barbera@nyu.edu>

Depends R (>= 2.12.0), httr, rjson, httpuv

License GPL-2

NeedsCompilation no

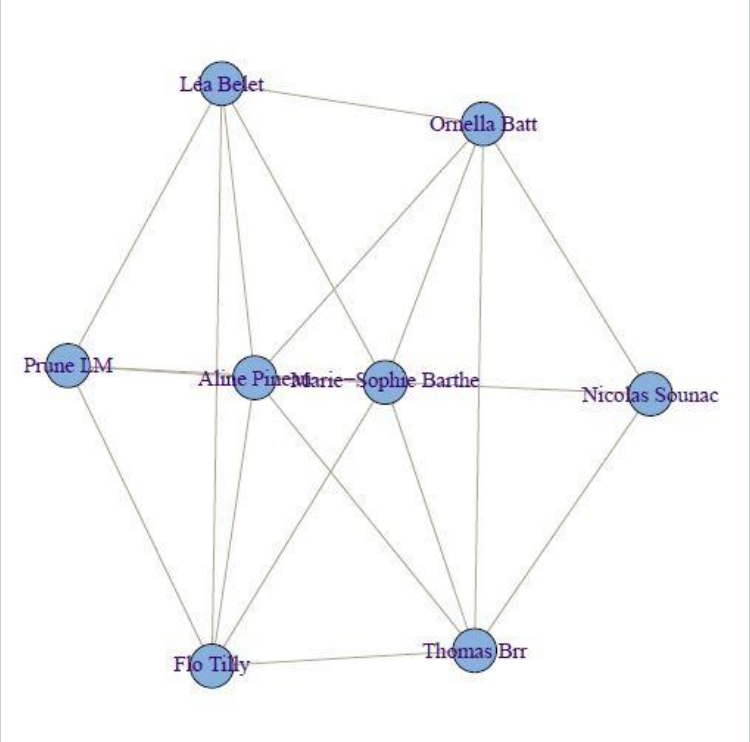
Repository CRAN

Date/Publication 2015-08-07 09:01:09

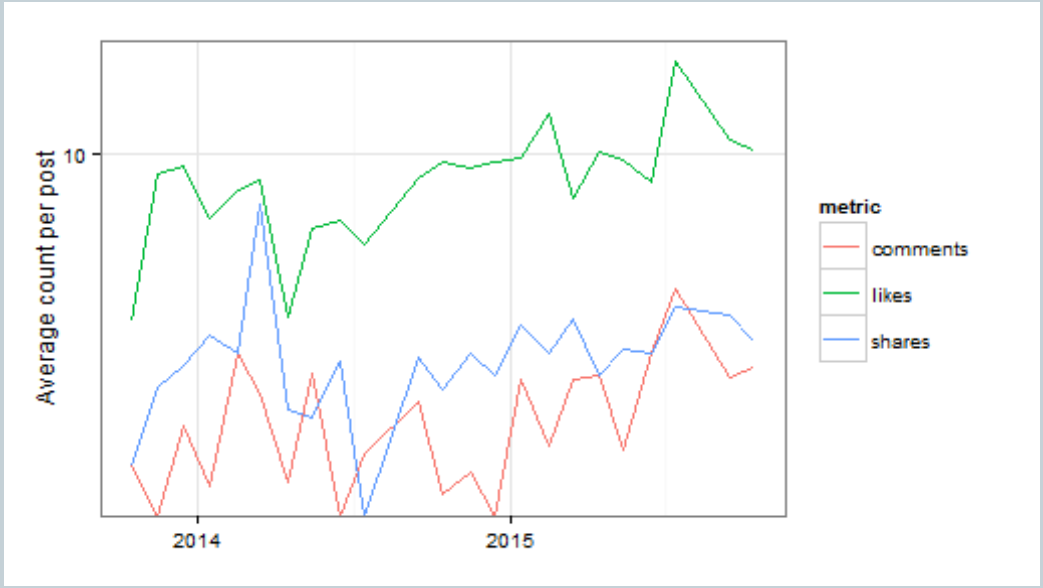
R topics documented:

2. Analyse des réseaux sociaux

- Démonstration



Utilisation de la fonction getNetwork

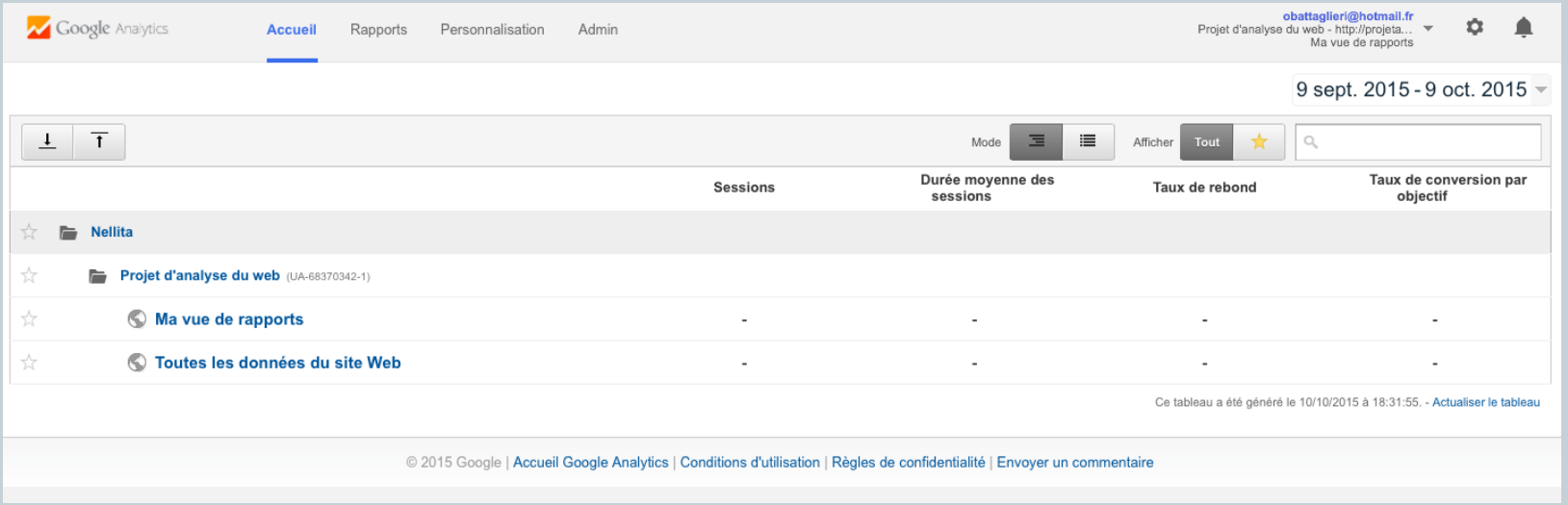


Nombre moyen de commentaires, de mentions « j'aime »et de partages par post de la page facebook « Agrocampus Ouest »

3. RGoogleAnalytics

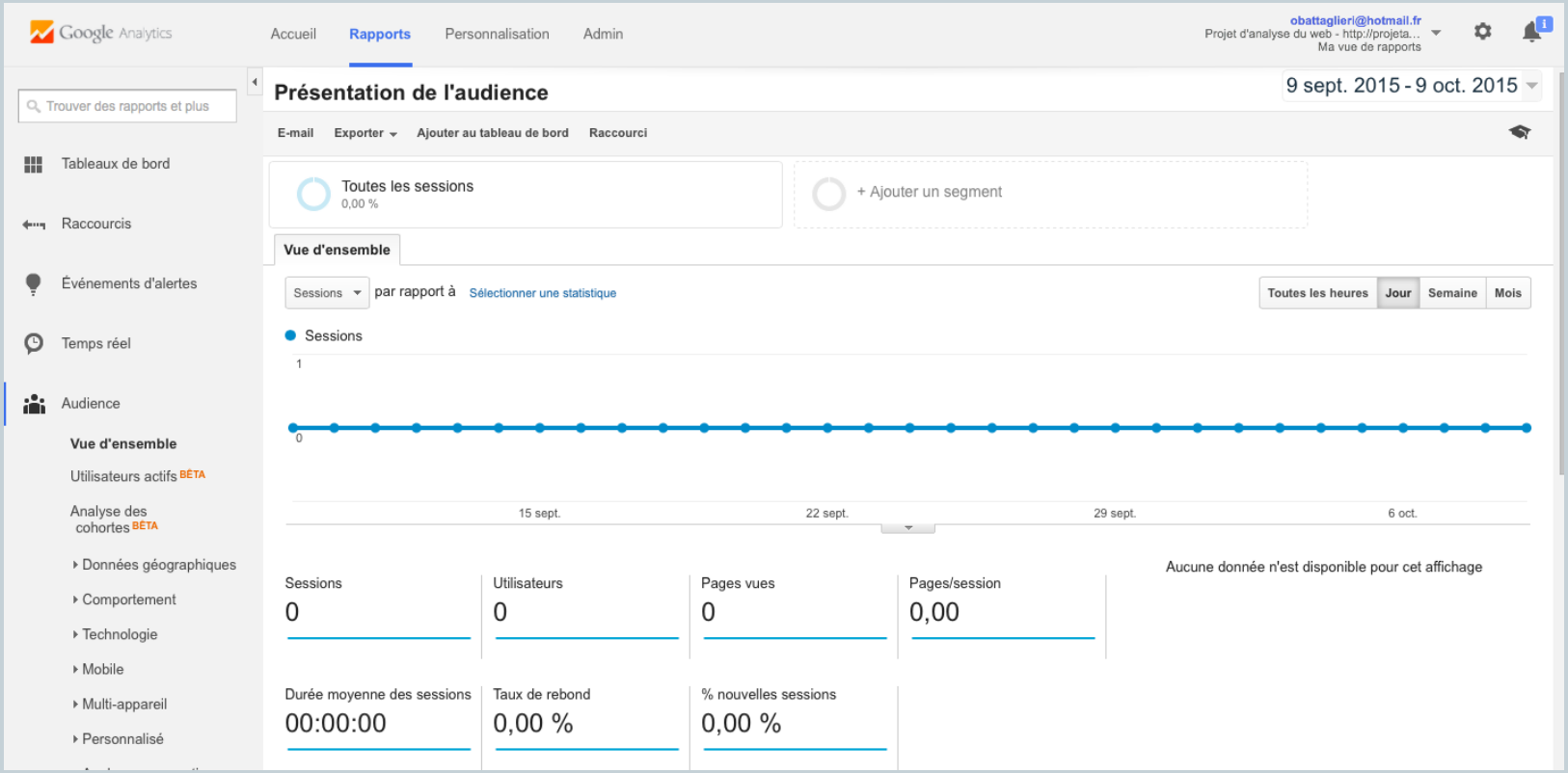
3. RGoogleAnalytics

- GoogleAnalytics
  - analyser les informations sur l'utilisation d'un site Web



- Exemples d'utilisation
  - Évaluer le nombre de ventes et de conversions
  - Déterminer utilisation et attractivité du site
  - Identifier les parties les plus performantes et les pages les plus consultées
  - Évaluer le succès du site sur les réseaux sociaux

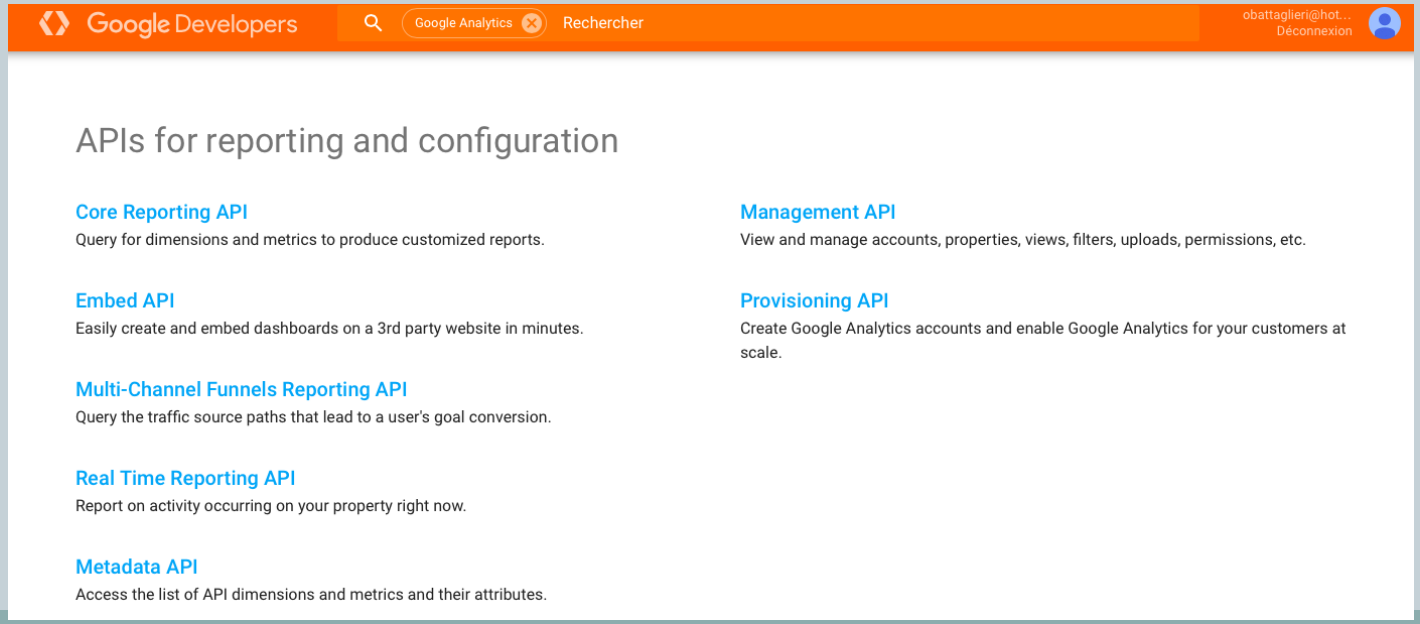
3. RGoogleAnalytics



3. RGoogleAnalytics

- RGoogleAnalytics : accès et extraction de données de GoogleAnalytics API

- API = ensemble de méthodes, classes et fonctions aidant les utilisateurs à se servir des fonctionnalités d'un logiciel



3. RGoogleAnalytics

```
require(RGoogleAnalytics)

##############
# Premiere etape : Autoriser l'accès au compte Google Analytics

token <- Auth(client.id,client.secret)

##############
# 2e etape: sauvegarder le "token" pour les sessions futres

save(token,file="./token_file")

#
#  Une fois sauvegardé, le "token" peut être téléchargé dans les sessions futures par simple:
#  load("./token_file")

##############
# 3e etape: Valider le "token"

ValidateToken(token)


##############
# 4e etape: Construire une liste constituée de tous les paramètres de requête

query.list <- Init(start.date = "2013-11-28",
                   end.date = "2013-12-04",
                   dimensions = "ga:date,ga:pagePath,ga:hour,ga:medium",
                   metrics = "ga:sessions,ga:pageviews",
                   max.results = 10000,
                   sort = "-ga:date",
                   filters = "ga:medium==referral",
                   table.id = "ga:33093633")

##############
# 5e etape: construire la requête pour valider ses paramètres

ga.query <- QueryBuilder(query.list)

##############
#  Extraire les données et les exposer dans un jeu de données

ga.data <- GetReportData(ga.query, token)
```

3. RGoogleAnalytics

- Fonction Auth
  - Auth: pour autoriser l'accès au compte de l'utilisateur
  - Création d'un « token object » = gage d'autorisation
  - Ce « token » peut être enregistré comme document sur l'ordinateur pour utilisations ultérieures

```
oauth_token <- Auth(client.id =
 "1089746702867-1urguiif4qcsphjl6d635668n0pvvear.apps.googleusercontent.com
", client.secret = "FKVwIrM-h0Tu0TuZAmr6hy10")
```

- Fonction GetReportData
  - Exécuter la requête : ce que l'on veut regarder sur notre compte, de quand à quand, le nombre de visites, etc.

https://developers.google.com/analytics/devguides/reporting/core/dimsmets

3. RGoogleAnalytics

- Fonction QueryBuilder
  - Initialise une liste des paramètres d'intérêt = argument de GetReportData
- Fonction Init
  - Transforme une ensemble de requêtes en une liste simple utilisée en argument par QueryBuilder
  - Par exemple :
    - Intervalle de temps d'intérêt
    - Organisation des données
    - Noms des utilisateurs
    - Fréquence et durée des visites
    - Rapidité de réponse du site

3. RGoogleAnalytics

Dimensions

- User
- Session
- Traffic Sources
- Adwords
- Goal Conversions
- Platform or Device
- Geo Network
- System
- Social Activities
- Page Tracking
- Content Grouping
- Internal Search
- Site Speed
- App Tracking
- Event Tracking
- Ecommerce

Metrics

- Social Interactions
- User Timings
- Exceptions
- Content Experiments
- Custom Variables or Columns
- Time
- DoubleClick Campaign Manager
- Audience
- Adsense
- Ad Exchange
- Channel Grouping
- Related Products

3. RGoogleAnalytics

```
query.list <- Init(start.date = "2013-11-28",
                   end.date = "2013-12-04",
                   dimensions = "ga:date,ga:pagePath,ga:hour,ga:medium",
                   metrics = "ga:sessions,ga:pageviews",
                   max.results = 10000,
                   sort = "-ga:date",
                   filters = "ga:medium==referral",
                   table.id = "ga:33093633")
```

3. RGoogleAnalytics

```
require(RGoogleAnalytics)

################
# Premiere etape : Autoriser l'accès au compte Google Analytics

token <- Auth(client.id,client.secret)

################
# 2e etape: sauvegarder le "token" pour les sessions futres

save(token,file="./token_file")

#
#  Une fois sauvegardé, le "token" peut être téléchargé dans les sessions futures par simple:
#  load("./token_file")

################
# 3e etape: Valider le "token"

ValidateToken(token)
```

3. RGoogleAnalytics

```
################
# 4e etape: Construire une liste constituée de tous les paramètres de requête

query.list <- Init(start.date = "2013-11-28",
                   end.date = "2013-12-04",
                   dimensions = "ga:date,ga:pagePath,ga:hour,ga:medium",
                   metrics = "ga:sessions,ga:pageviews",
                   max.results = 10000,
                   sort = "-ga:date",
                   filters = "ga:medium==referral",
                   table.id = "ga:33093633")

################
# 5e etape: construire la requête pour valider ses paramètres

ga.query <- QueryBuilder(query.list)

###############
#  Extraire les données et les exposer dans un jeu de données

ga.data <- GetReportData(ga.query, token)
```

3. RGoogleAnalytics

- Avantages:
  - Résumé clair de l'information
  - Fonctions assez simples à comprendre
- Inconvénients
  - Difficulté de prise en main
  - Règles de confidentialité complexes gênant le fonctionnement de certaines fonctions
  - Erreurs d'accès fréquentes
  - Intérêt *a priori* faible par rapport à une utilisation classique de Google Analytics

Conclusion

- Quelles sont les fonctionnalités du logiciel R en matière d'analyse du Web ?
  - Variées, utiles, mais complexes
  - Lien entre Rvest et RFacebook

Bibliographie

- RGoogleAnalytics: R Wrapper for the Google Analytics API, CRAN
  https://cran.r-project.org/web/packages/RGoogleAnalytics/index.html
- Access to Facebook API via R , Pablo Barbera , Michael Piccirilli
  https://cran.r-project.org/web/packages/Rfacebook/Rfacebook.pdf
- Tools for Parsing and Generating XML Within R and S-Plus, Duncan Temple Lang and the CRAN Team
  https://cran.r-project.org/web/packages/XML/XML.pdf
- Tools for working with URL and HTTP, Hadley Wickham
  https://cran.r-project.org/web/packages/httr/httr.pdf
- Tools for scraping data from HTML and XML documents , Ryan M. Acton
  https://cran.r-project.org/web/packages/scrapeR/scrapeR.pdf
- Easily harvest (scrape) Web pages , Hadley Wickham
  https://cran.r-project.org/web/packages/rvest/rvest.pdf