Google Analytics Sample - Data Extraction

Tung Anh

2025-07-16

Introduction

Ce carnet montre comment extraire et traiter les données **d'exemple de Google Analytics** à partir de l'API BigQuery. Le jeu de données contient 12 mois de données anonymisées (d'août 2016 à août 2017) provenant du Google Merchandise Store, un véritable site de commerce électronique vendant des articles de marque Google.

Configuration

```
# Charger les bibliothèques nécessaires
library(bigrquery)
library(DBI)
library(dplyr)
library(tidyr)
library(readr)
library(lubridate)
library(arrow)
library(glue)
library(here)
```

Authentication

Pour utiliser directement l'API BigQuery depuis un compte Google, nous utilisons l'authentification via le navigateur web :

```
bq_auth() # Sign in to Google Account with active Google Cloud console
```

Les données originales

Les données originales peuvent être accessibles en utilisant le code ci-dessous :

```
project_id <- "bigdata-382912"
sql <- "
SELECT *
```

```
FROM `bigquery-public-data.google_analytics_sample.ga_sessions_20170801`
LIMIT 100"

tb <- bq_project_query(project_id, sql)
org_df <- bq_table_download(tb)

head(org_df)
```

```
## # A tibble: 6 x 16
##   visitorId visitNumber  visitId visitStartTime date  totals       trafficSource
##       <int>       <int>    <int>          <int> <chr> <list>       <list>
## 1        NA           1   1.50e9     1501591568 2017~ <named list> <named list>
## 2        NA           2   1.50e9     1501589647 2017~ <named list> <named list>
## 3        NA           1   1.50e9     1501616621 2017~ <named list> <named list>
## 4        NA           1   1.50e9     1501601200 2017~ <named list> <named list>
## 5        NA           1   1.50e9     1501615525 2017~ <named list> <named list>
## 6        NA           1   1.50e9     1501610896 2017~ <named list> <named list>
## # i 9 more variables: device <list>, geoNetwork <list>,
## #   customDimensions <list>, hits <list>, fullVisitorId <chr>, userId <chr>,
## #   clientId <chr>, channelGrouping <chr>, socialEngagementType <chr>
```

Les données originales contiennent 16 colonnes, dont 6 sont imbriquées (`totals`, `trafficSource`, `device`, `geoNetwork`, `customDimensions`, `hits`) et 3 sont obscurcies (`visitorId`, `userId` et `clientId`).

Extraire une journée de colonnes aplaties

Nous accédons à une journée de données provenant du jeu de données d'exemple de Google Analytics, en aplatissant une sélection de colonnes non imbriquées et en négligeant les colonnes obscurcies :

```
# Query SQL
main_query <- "
SELECT
  -- single columns
  date,
  fullVisitorId,
  visitId,
  visitNumber,
  visitStartTime,
  channelGrouping,
  -- total unnested columns
  totals.newVisits AS total_newVisits,
  totals.pageviews AS total_pageviews,
  totals.hits AS total_hits,
  totals.bounces AS total_bounces,
  totals.timeOnsite AS total_timeOnsite,
  totals.transactions AS total_transactions,
  totals.transactionRevenue/1000000 AS total_transactionRevenue,
  totals.totalTransactionRevenue/1000000 AS total_totalTransactionRevenue,
  -- trafficSource unnested columns
  trafficSource.source AS trafficSource_source,
  trafficSource.campaign AS trafficSource_campaign,
```

```
  -- device unnested columns
  device.browser AS device_browser,
  device.deviceCategory As device_deviceCategory,
  -- geoNetwork unnested columns
  geoNetwork.continent AS geoNetwork_continent,
  geoNetwork.country AS geoNetwork_country

FROM `bigquery-public-data.google_analytics_sample.ga_sessions_20170801`
"

# Charger l'output de query sur un df
df <- bq_project_query(project_id, main_query) %>%
      bq_table_download()

# Vérifier le résultat
head(df)
```

```
## # A tibble: 6 x 20
##   date     fullVisitorId      visitId visitNumber visitStartTime channelGrouping
##   <chr>    <chr>                <int>       <int>          <int> <chr>
## 1 20170801 34183340117798720~  1.50e9           1     1501591568 Organic Search
## 2 20170801 24743978550413224~  1.50e9           2     1501589647 Referral
## 3 20170801 58704628207131101~  1.50e9           1     1501616621 Referral
## 4 20170801 93978091713494803~  1.50e9           1     1501601200 Referral
## 5 20170801 60899029431845783~  1.50e9           1     1501615525 Referral
## 6 20170801 97748531789345392~  1.50e9           1     1501610896 Paid Search
## # i 14 more variables: total_newVisits <int>, total_pageviews <int>,
## #   total_hits <int>, total_bounces <int>, total_timeOnsite <int>,
## #   total_transactions <int>, total_transactionRevenue <dbl>,
## #   total_totalTransactionRevenue <dbl>, trafficSource_source <chr>,
## #   trafficSource_campaign <chr>, device_browser <chr>,
## #   device_deviceCategory <chr>, geoNetwork_continent <chr>,
## #   geoNetwork_country <chr>
```

Description des données

- `date` : Date de la session au format AAAAMMJJ.
- `fullVisitorId` : Identifiant unique d'un utilisateur (visiteur) sur plusieurs sessions.
- `visitId` : Identifiant unique d'une session spécifique (visite) parmi les visites d'un utilisateur.
- `visitNumber` : Nombre de visites qu'un utilisateur a effectuées sur le site jusqu'à et incluant cette session.
- `visitStartTime` : Horodatage (en format UNIX) du début de la session.
- `channelGrouping` : Catégorie du canal de trafic ayant amené le visiteur sur le site (ex. Recherche organique, Direct, Référent).
- `total_newVisits` : Indicateur si cette session est la première visite de l'utilisateur (1 si nouveau visiteur, sinon 0).

- `total_pageviews` : Nombre total de pages vues pendant la session.
- `total_hits` : Nombre total d'interactions enregistrées pendant la session.
- `total_bounces` : Nombre de sessions à une seule page (rebonds) durant cette visite.
- `total_timeOnsite` : Temps total passé sur le site pendant la session (en secondes).
- `total_transactions` : Nombre de transactions réalisées pendant la session.
- `total_transactionRevenue` : Revenu total généré par les transactions pendant la session (en micros, c'est-à-dire multiplié par 1 000 000).
- `total_totalTransactionRevenue` : Revenu total des transactions agrégé sur toutes les transactions durant la session (en micros).
- `trafficSource_source` : Source du trafic pour la session (ex. google, direct, newsletter).
- `trafficSource_campaign` : Nom de la campagne marketing associée au trafic.
- `device_browser` : Navigateur utilisé par le visiteur (ex. Chrome, Firefox).
- `device_deviceCategory` : Catégorie de l'appareil utilisé pour accéder au site (ex. ordinateur, mobile, tablette).
- `geoNetwork_continent` : Continent d'origine de la session.
- `geoNetwork_country` : Pays d'origine de la session.

Fonction généralisée pour extraire les données selon une plage de dates

Cette fonction télécharge les données quotidiennes pour une plage de dates donnée et les sauvegarde au format Parquet organisées par année et mois :

```
download_ga_data <- function(project_id, start_date, end_date, saved_path, cardinality_threshold = 10) {
  # Transformer date au bon format
  start_date <- ymd(start_date)
  end_date <- ymd(end_date)

  current_date <- start_date

  while (current_date <= end_date) {
    date_str <- format(current_date, "%Y%m%d")
    year_str <- format(current_date, "%Y")
    month_str <- format(current_date, "%m")

    # Build query with glue
    main_query <- glue("
      SELECT
        date,
        fullVisitorId,
        visitId,
```

```
        visitNumber,
        visitStartTime,
        channelGrouping,
        totals.newVisits AS total_newVisits,
        totals.pageviews AS total_pageviews,
        totals.hits AS total_hits,
        totals.bounces AS total_bounces,
        totals.timeOnsite AS total_timeOnsite,
        totals.transactions AS total_transactions,
        totals.transactionRevenue/1000000 AS total_transactionRevenue,
        totals.totalTransactionRevenue/1000000 AS total_totalTransactionRevenue,
        trafficSource.source AS trafficSource_source,
        trafficSource.campaign AS trafficSource_campaign,
        device.browser AS device_browser,
        device.deviceCategory AS device_deviceCategory,
        geoNetwork.continent AS geoNetwork_continent,
        geoNetwork.country AS geoNetwork_country
      FROM `bigquery-public-data.google_analytics_sample.ga_sessions_{date_str}`
    ")

    # Exécuter la requête et télécharger les données
    df <- bq_project_query(project_id, main_query) %>% bq_table_download()

    # Convertir les colonnes numériques à faible cardinalité en facteurs
    df <- df %>%
      mutate(across(
        .cols = where(is.numeric),
        .fns = ~ if (n_distinct(.) <= cardinality_threshold) factor(.) else .
      ))

    # Créer les dossiers
    dir_path <- file.path(saved_path, year_str, month_str)
    if (!dir.exists(dir_path)) dir.create(dir_path, recursive = TRUE)

    # Sauvegarder en format parquet
    save_path <- file.path(dir_path, paste0("ga_sessions_", date_str, ".parquet"))
    arrow::write_parquet(df, save_path)

    message(glue("Données sauvegardées pour {date_str} dans {save_path}"))

    # Passer au jour suivant
    current_date <- current_date + days(1)
  }
}
```

Téléchargement complete

Pour télécharger l'intégralité des données (d'août 2016 à août 2017), nous exécutons le code ci-dessous :

```
download_ga_data(
  project_id="bigdata-382912",
  start_date="20160801",
```

```
  end_date="20170731",
  saved_path =  here("data")
  )
```