

Création de rapport avec le package Oxom

Clément Walter

2017-01-06

Installation préalable

Pour fonctionner, le package `Oxom` utilise le package `r2excel` non disponible sur le CRAN mais seulement sur GitHub. Il faut donc au préalable l'installer *à la main* par cette ligne :

```
install.packages('devtools')
devtools::install_github('kassambara/r2excel')
```

On peut maintenant installer le package `Oxom` qui génère les rapports :

```
devtools::install_github('clemflaflemme/Oxom')
```

Ces opérations ne sont à réaliser qu'une seule fois par ordinateur, avant la première utilisation.

Organisation des fichiers

Le package `Oxom` est bati en supposant une arborescance de fichiers spécifique qu'il ne faudra pas changer. Dans le Google Drive mis en place, le dossier `skaze` est la racine. Il comporte plusieurs dossiers, dont `Clients`. C'est dans ce dossier que va travailler le programme.

Le dossier client

Chaque client doit avoir son dossier propre, et le nom de ce dossier sera repris dans les rapports générés comme le nom du client, c'est-à-dire de l'*Advertiser*.

A l'intérieur, il y a :

- le dossier `Weborama` : pour les données téléchargées depuis l'interface `Weborama`
- le dossier `Mediarithmics` : pour les données téléchargées depuis l'interface `Mediarithmics`
- le fichier `Mediarithmics-goals.xlsx` : pour les goals reportés à la main depuis le site de `Mediarithmics`.
- le fichier `Weborama-goals.xlsx` qui correspond au rapport généré par `Weborama` mais non disponible en téléchargement en ligne

Chargement des données depuis les DSP

Weborama

Les données utilisées sont celles qu'on trouve à l'adresse suivante :

```
Weborama Campaign Manager > Reporting > Campaign
```

Attention il faut exporter chaque campagne en cliquant dessus d'abord. Le tableau qui s'affiche en dessous du graphique doit être un détail des performances par Ad. Pour `Son-video`, il y a deux campagnes : `Prospecting` et `Retargeting`.

Il faut télécharger ces fichiers (autant que de campagne) et les placer simplement dans le dossier `Weborama`.

Mediarithmics

Les données utilisées sont celles que l'on télécharge jour après jour dans `Display campaign`. Il faut bien télécharger un fichier par jour. Ils auront les mêmes noms pour une même campagne avec un numéro ajouté automatiquement par l'ordinateur, inutile de les renommer.

Au bout du compte il y a un fichier par jour et par campagne dans le dossier `Mediarithmics`.

Goals

Les données des goals ne sont pas obtenus par les téléchargement précédent pour l'instant.

Mediarithmics

Il faut créer et remplir à la main un fichier excel à trois colonnes selon le modèle de celui dans `Son-video` :

- Date
- Goals
- Campaign

et le nommer `Mediarithmics-goals.xlsx`. Les données sont obtenues dans l'onglet `Goals` de `Mediarithmics`.

Weborama

Les données viennent d'un fichier reçu par mail. Il faut le demander à chaque actualisation et doit être de la forme de celui déjà fourni pour `Son-vidéo` ; voir le modèle dans cette étude. Il doit être renommé `Weborama-goals.xlsx` et recoupé avec deux onglet "Prospecting" et "Retargeting" au lieu du `Data` initial.

Utilisation du package

On ouvre un terminal R ou `RStudio`. On se place dans le dossier client choisi, par exemple :

```
setwd('/Volumes/Stockage/Google Drive/skaze/Clients/Son-video/')
```

puis on exécute la commande :

```
Oxom::makeReport(save.dir = "Rapport.xlsx",  
                 pic.dir = "sv-logo.png")
```

Les différents paramètres utilisés sont :

- `save.dir` : l'adresse où sera enregistré le rapport. Ce peut être seulement le nom du fichier (comme ci-dessus) et il sera alors enregistré dans le dossier client. On peut également donner une adresse complète, par exemple `r save.dir = "/Users/clementwalter/Desktop/rapport.xlsx"`. Dans ce cas, le fichier sera enregistré dans le dossier spécifié.
- `pic.dir` correspond à l'adresse d'un logo client à afficher sur le rapport. Ce fichier doit être en `.png`.

On a ici généré un rapport avec la dépense réelle effectuée sur les plateformes. Pour imposer un coût total différent (à montrer au client), on peut ajouter l'option `budget` ou `marge` :

- `budget` va normaliser les dépenses pour parvenir à un total égale à `budget`
- `marge` est un coefficient multiplicateur à appliquer à toutes les dépenses. Par exemple, `marge = 1.3` va multiplier toutes les dépenses par 1.3

In fine, cela s'écrit par exemple :

```
Oxom::makeReport(save.dir = "Rapport-normalise.xlsx",  
                pic.dir = "sv-logo.png",  
                budget = 5000)
```

Quand le programme s'arrête, on trouve dans le dossier de l'étude (Son-video ici) les rapports créés, ici Rapport.xlsx et Rapport-normalise.xlsx :

```
list.files(pattern = "(Rapport)")
```

```
## [1] "Rapport-normalise.xlsx" "Rapport.xlsx"  
}
```