1 Objectif

Utilisation de l'API de Google Analytics pour analyser les données de fréquentation des sites web sous R. Le package RGoogleAnalytics.

Google Analytics est un service de mesure d'audience des sites Web. On parle de *Web Analytics*. L'objectif est d'apprécier la fréquentation. Mais au-delà du simple comptage, et c'est ce qui fait réellement sa force, l'outil propose de nombreuses possibilités d'études qui permettent de mieux comprendre le comportement des visiteurs. Ainsi, en identifiant les pages les plus regardées, les pages d'entrées, en analysant les trajectoires, en ventilant les accès selon les pays, etc., il nous permet de mieux organiser notre site de manière à répondre plus précisément aux attentes des internautes.

RGoogleAnalytics est un package R permettant d'accéder aux données Google Analytics via une API (interface de programmation applicative). La librairie nous donne accès aux informations stockées par le service, tant en termes de fréquentation brute qu'en termes d'analyses préprogrammées. Concrètement, nous spécifions les axes d'étude (les « dimensions » ou les « contextes » si l'on se réfère à la terminologie *Business Intelligence*) et les mesures (les « faits »), l'API fournit les tableaux correspondants. On pourrait par exemple extraire facilement le nombre de sessions (mesure) par pays (dimension) (section 2.2).

Cette utilisation de type reporting est déjà intéressante en soi. Nous recourrons à R pour produire des tableaux et graphiques avenants à partir des données fournies par Google Analytics. L'intérêt du dispositif est décuplé lorsqu'on se rend compte qu'en rapatriant les données sous R, un outil statistique puissant comme nous le savons tous, il est possible d'initier des études approfondies. Ainsi, dans ce tutoriel, nous réaliserons une analyse factorielle des correspondances (AFC) à partir d'un tableau croisé de fréquentation obtenu lors d'une requête en ligne via l'API.

Ce document décrit l'utilisation sous R des fonctionnalités du package RGoogleAnalytics à partir des données du site Tanagra au sens large (pages du logiciel, mais aussi les pages relatives aux supports de cours, tutoriels, etc.) sur le mois de novembre 2017.

2 Google Analytics – Le site Tanagra

Pour bien appréhender son intérêt, prenons le temps de regarder un peu en quoi consiste le service Google Analytics. Pour ma part, je l'utilise depuis le 1er février 2008. Il me sert à

suivre l'activité et les préoccupations des internautes par rapport à mes pages web. Je me suis rendu compte par exemple que 3 pages concentraient l'intérêt des visiteurs ces dernières années (plus de 30% des accès). C'est toujours bon à savoir.

Schématiquement (en simplifiant beaucoup), après avoir créé un compte sur le site de Google Analytics, nous disposons d'un code javascript que nous insérons dans nos différentes pages. Des informations sont transmises aux serveurs de Google à chaque accès déclenchant l'exécution du code. Elles sont stockées au fur et à mesure.

2.1 Fréquentation brute – Sessions (et utilisateurs)

Lorsque nous nous connectons à notre compte, nous disposons d'une application proposant plusieurs indicateurs présentés sous différents angles. Pour les utilisateurs avertis, il est possible de créer ses propres tableaux de bord et rapports personnalisés, précisant directement les mesures et axes d'analyse qui répondent à nos préoccupations.

Sur la période allant du 1er au 30 novembre 2017, voici ce que nous observons dans la « Vue d'ensemble » de l'Audience. Il s'agit des chiffres bruts de fréquentation.

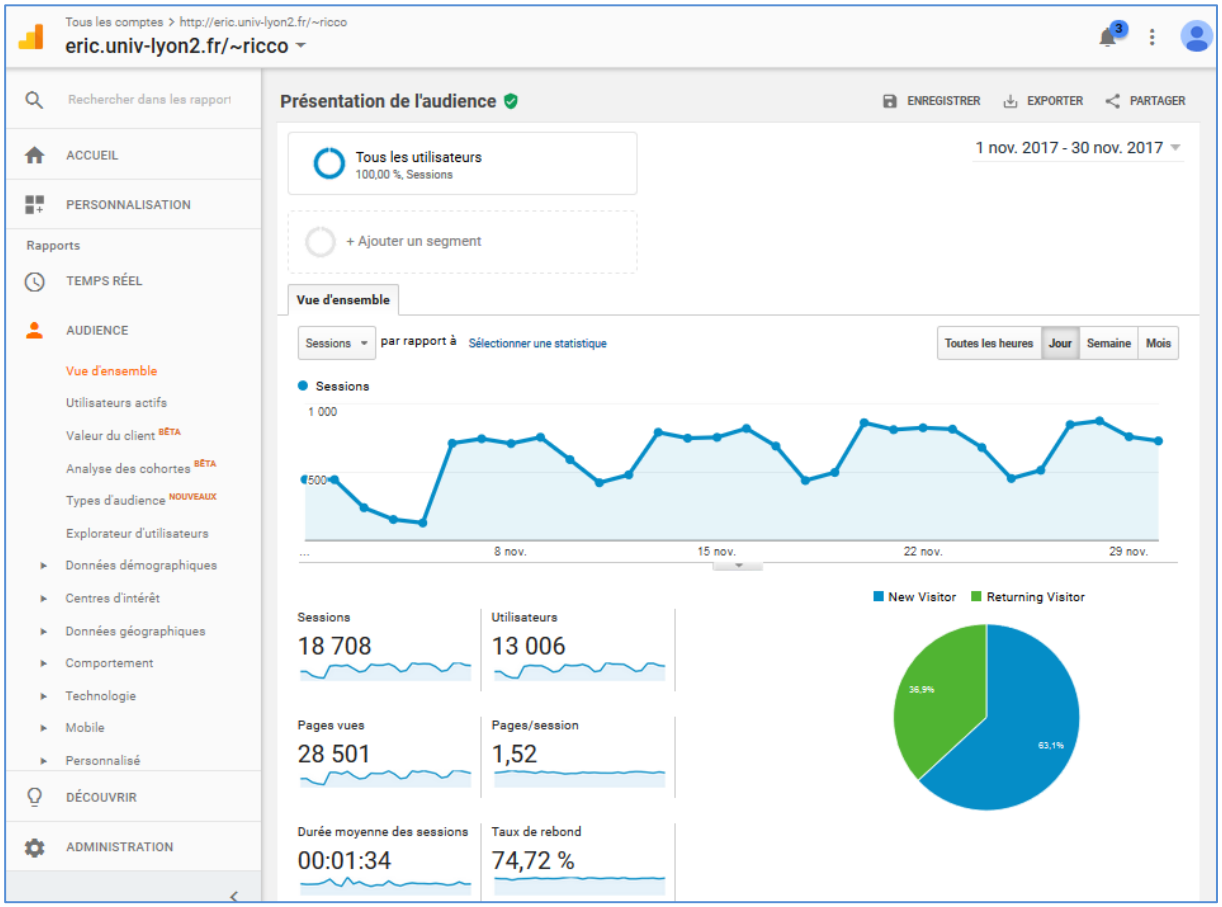
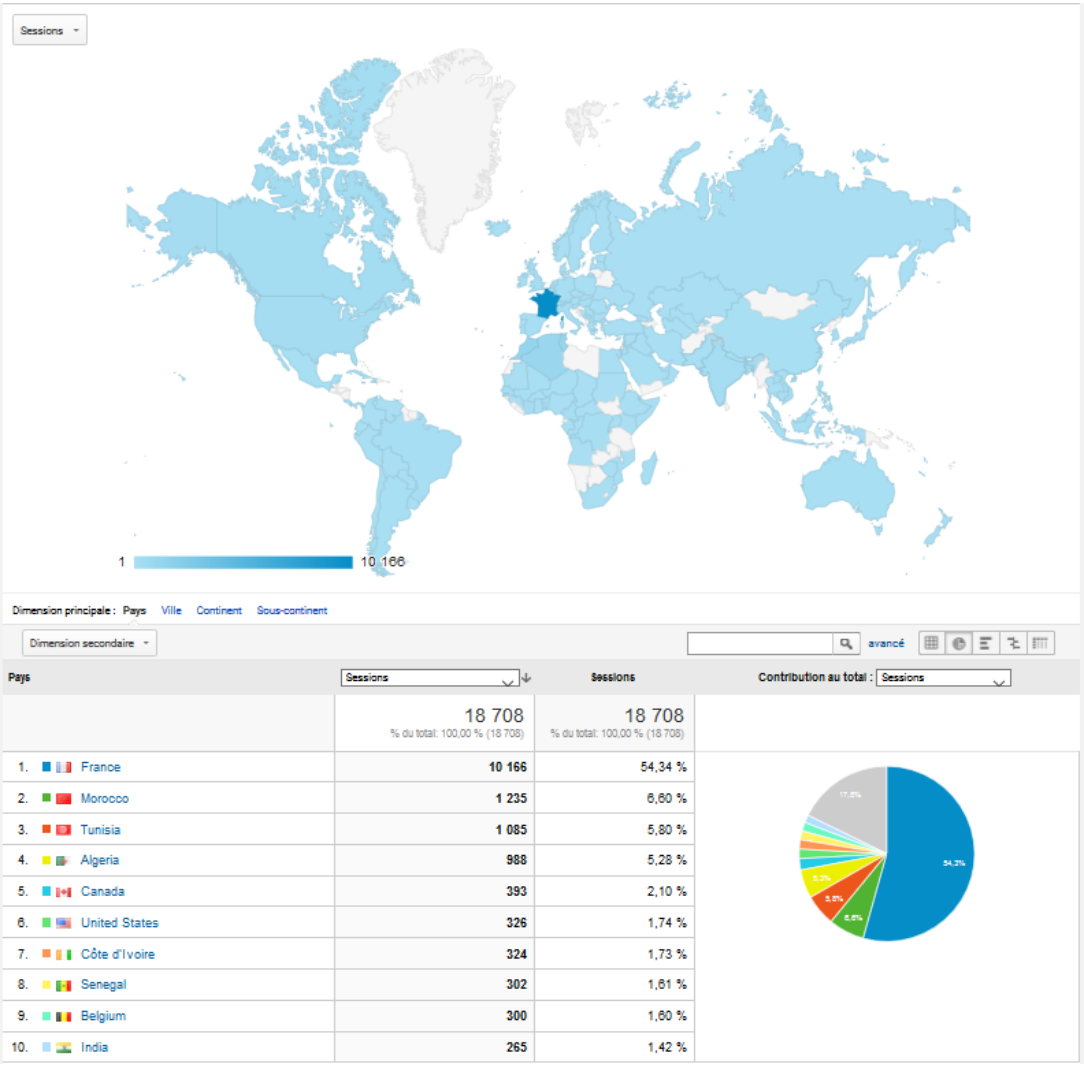


Figure 1 - Fréquentation journalière durant le mois de novembre 2017

Nous constatons qu'il y a eu 18.708 sessions pour 13.006 utilisateurs sur le mois analysé[1]. Il y a naturellement les décrochages liés aux week-ends. Une baisse anormale lors du premier week-end de novembre 2017 attire notre attention. Elle est consécutive à une panne du serveur du Laboratoire ERIC qui héberge une grande partie des pages. De fait, cette courbe, pourtant relativement fruste, se révèle être très instructive. Elle a permis d'identifier en un coup d'œil une anomalie de fonctionnement.

2.2 Analyse – Les accès par pays

La puissance de Google Analytics s'exprime à travers ses potentialités d'analyse. Ci-dessous, nous calculons le nombre de sessions par pays. Nous constatons que la France en représente plus de la moitié (54.34%) sur la période considérée.



Pays	Sessions	Sessions
	18 708 % du total: 100,00 % (18 708)	18 708 % du total: 100,00 % (18 708)
1. France	10 166	54,34 %
2. Morocco	1 235	6,60 %
3. Tunisia	1 085	5,80 %
4. Algeria	988	5,28 %
5. Canada	393	2,10 %
6. United States	326	1,74 %
7. Côte d'Ivoire	324	1,73 %
8. Senegal	302	1,61 %
9. Belgium	300	1,60 %
10. India	265	1,42 %

Figure 2 - Fréquentation par pays durant le mois de novembre 2017

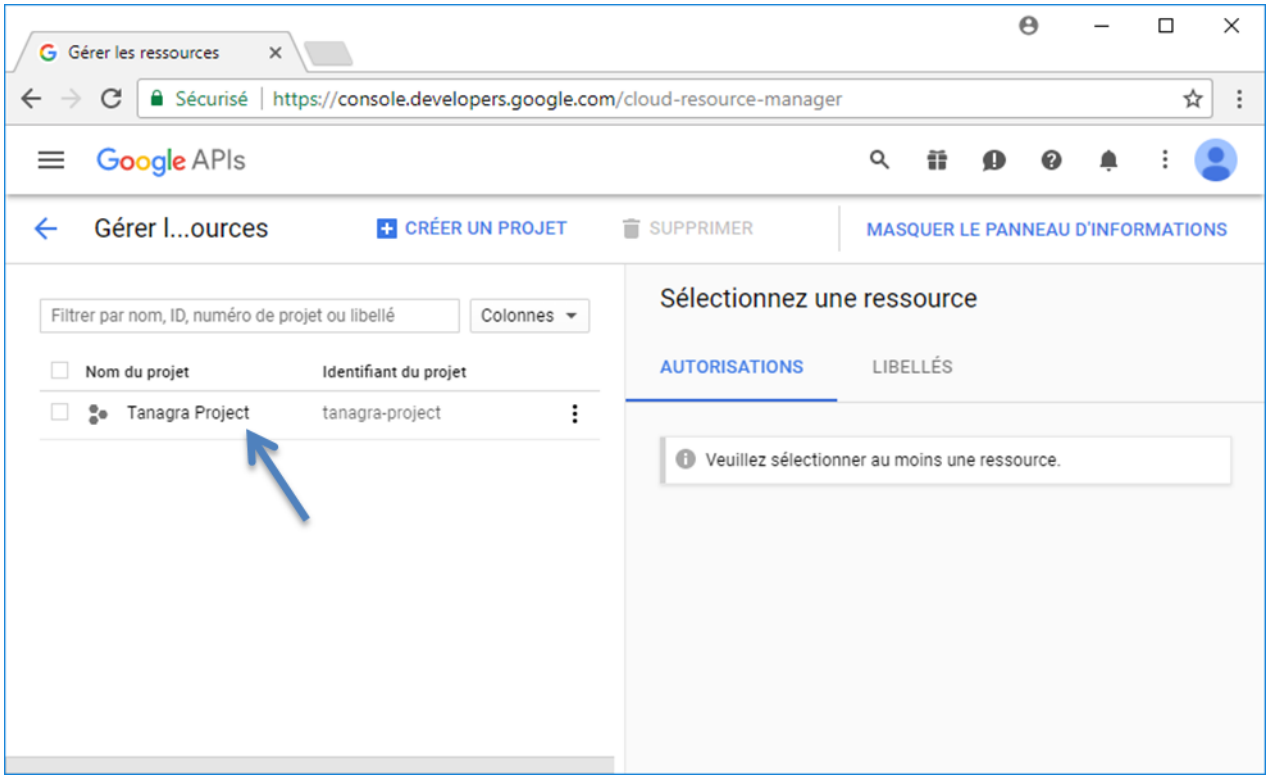
[1] Voir le support de Google Analytics concernant la distinction entre sessions et utilisateurs.

On pourrait ainsi multiplier les analyses avec l'interface en ligne. L'enjeu pour nous dans ce tutoriel est d'accéder à ces mêmes informations sous R pour, d'une part, pouvoir les présenter de manière personnalisée, d'autre part, réaliser par la suite des études plus consistantes.

3 Activation de l'API Google Analytics

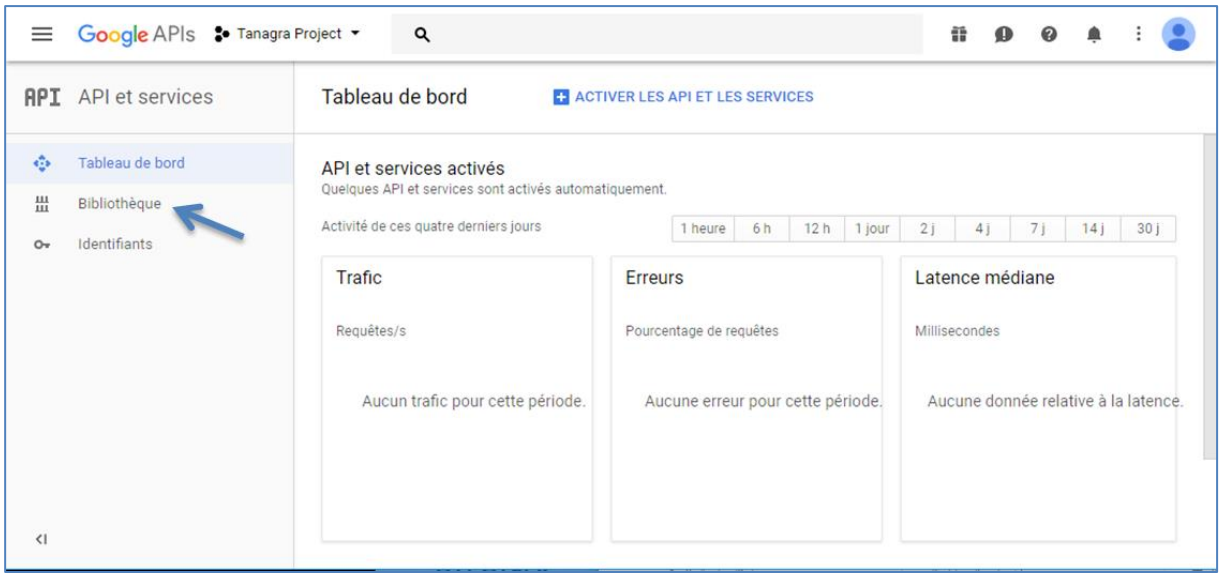
Il faut bien sûr au préalable créer un compte Google Analytics et paramétrer au moins quelques pages. Puis, laisser passer un certain temps pour que des données suffisamment substantielles soient récoltées.

Pour activer l'API Google Analytics, nous allons sur la page de gestion des API « Google Developers » (https://console.developers.google.com/project). Pour ma part, j'ai créé le projet « Tanagra Project » comme on peut le voir ci-dessous.

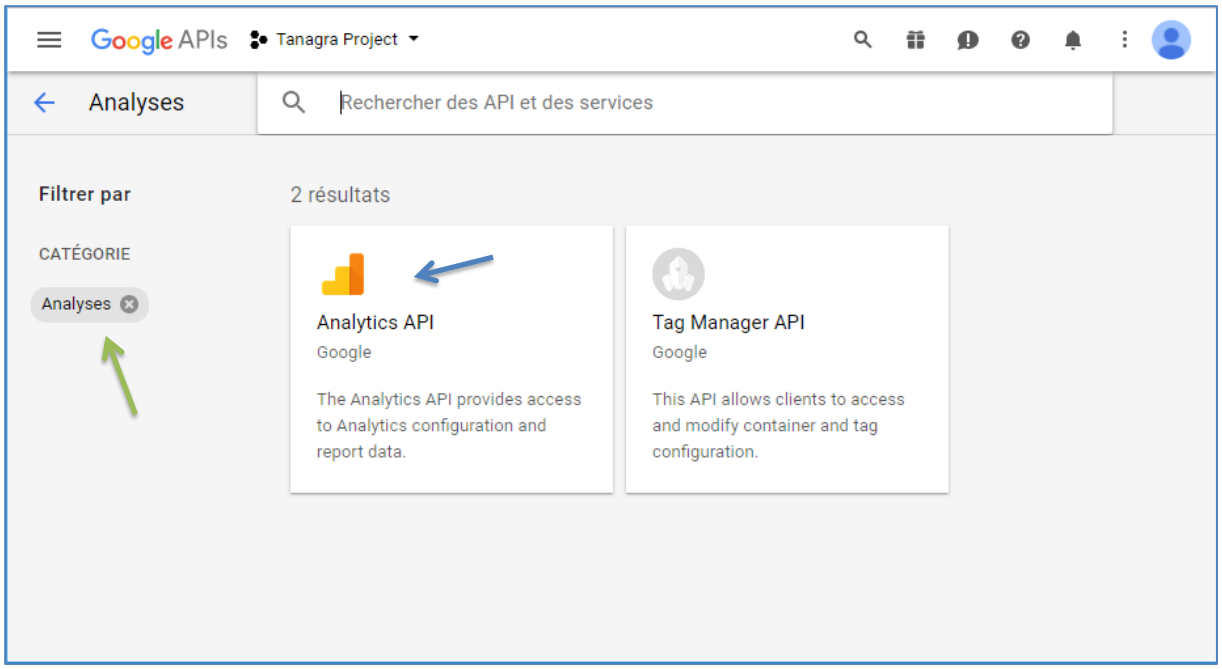


Si c'est votre première connexion, il faut donc créer un projet.

Nous l'ouvrons par la suite. Dans la page « API – API et services », nous cherchons l'API qui nous convient en cliquant sur l'item « Bibliothèque ».

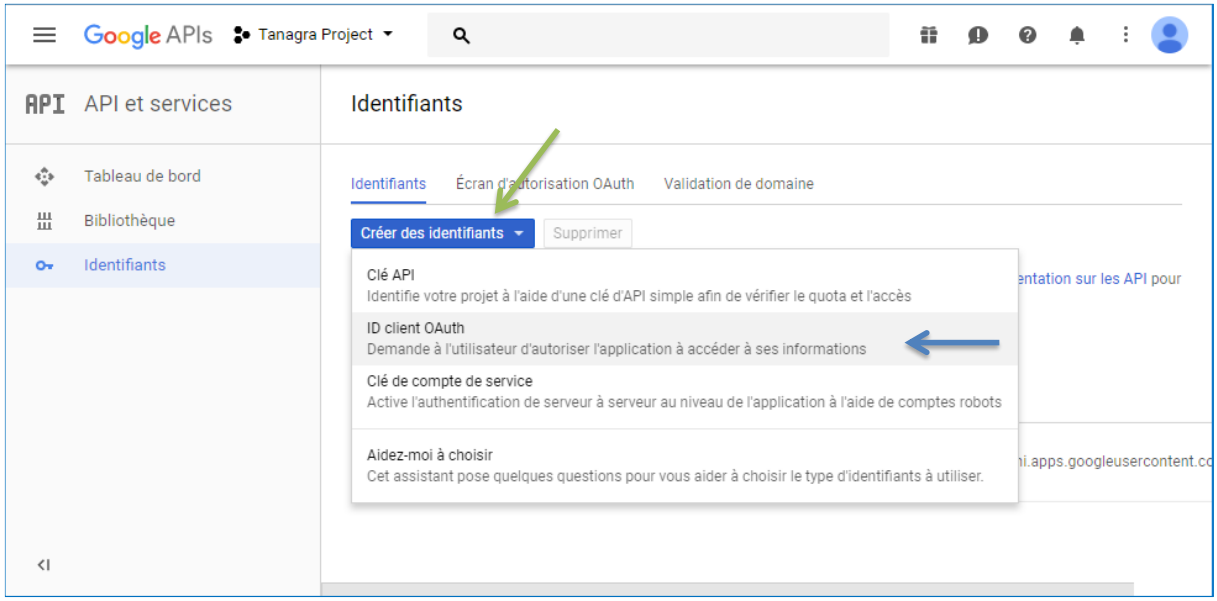


La liste des API est longue. Nous filtrons sur « Analyses » pour trouver facilement « Analytics API ».

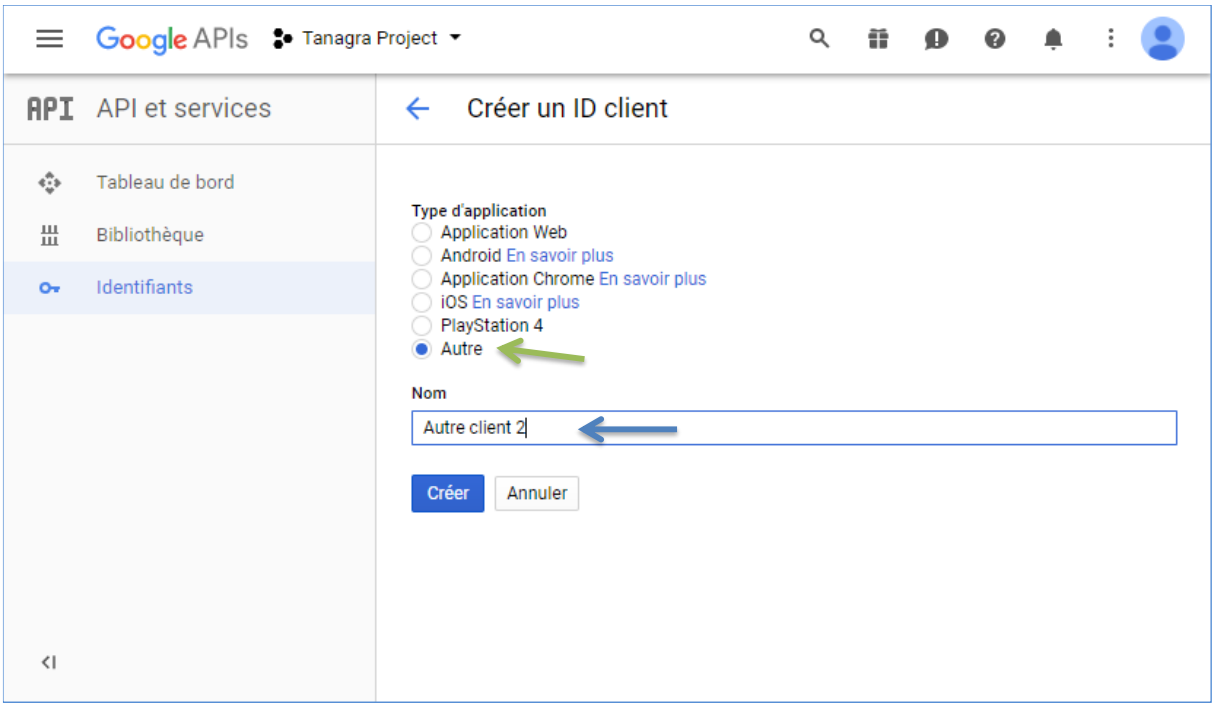


Dans la vaste liste des API disponibles, nous sélectionnons Analytics API.

Une nouvelle page apparaît. Elle nous permet de définir notre identifiant. Nous souhaitons créer un **ID Client Oauth**.

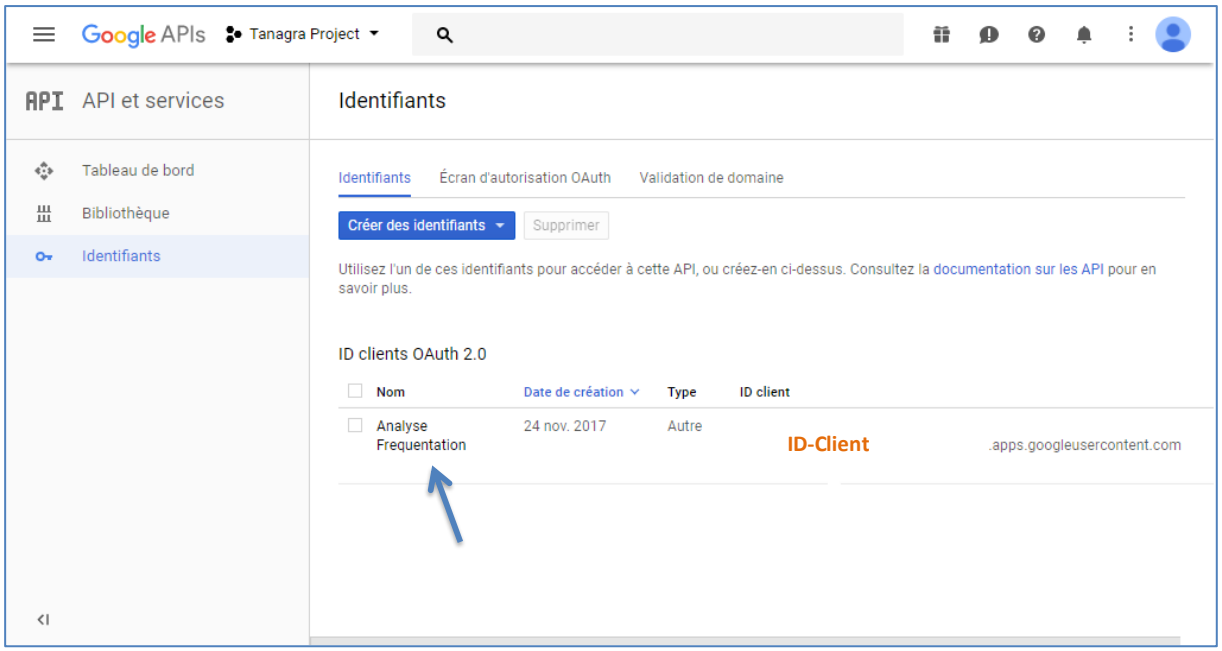


Le type d'application à sélectionner est « Autre ». Nous lui attribuons le nom qui nous convient.



Le nouvel identifiant apparaît dans la liste. Nous disposons d'un ID Client (masqué dans la copie d'écran ci-dessous) et d'un code secret (accessible en cliquant sur l'identifiant) que nous exploiterons dans notre code R.

Pour éviter une recopie fastidieuse, nous pouvons générer un fichier JSON contenant l'ensemble de ces éléments (ID + code secret).



4 Utilisation de Google Analytics sous R

Nous pouvons travailler sous R maintenant. Il faut installer puis charger le package RGoogleAnalytics en préambule de notre programme.

```
#installation de la librairie
#une seule fois
install.packages("RGoogleAnalytics")
#chargement de la librairie
#à chaque utilisation
library(RGoogleAnalytics)
```

4.1 Première connexion - Génération d'un jeton (token) d'authentification

La première étape consiste à obtenir un jeton de connexion (token) à partir de l'identifiant et de la clé secrète obtenus sur le site « Google developers ».

```
#identifiant et clé secrète
client_id <- "xxxxxx.apps.googleusercontent.com"
client_secret <- "xxxxxxxx"
#génération d'un token
oauth_token <- Auth(client.id = client_id, client.secret = client_secret)
```

Lors de l'exécution de cette dernière instruction, votre navigateur web est automatiquement ouvert sur une page d'authentification. Nous devons saisir notre mot de passe Google pour valider l'accès.

La création d'un jeton n'est à réaliser qu'une seule fois. Il convient de le sauvegarder pour une réutilisation lors des prochains traitements. Nous utilisons la commande save().

```
#sauvegarde du token – à faire une seule fois
save(oauth_token,file = "oauth_token")
```

Un fichier binaire « oauth_token » est créé dans le répertoire courant.

Remarque : Notons-le bien, les opérations décrites dans cette section ne sont à réaliser qu'une seule fois. Par la suite, nous nous contenterons de charger et valider le token qui aura été sauvegardé dans un fichier localement.

4.2 Chargement et validation du token

La commande load() permet de charger un token sauvegardé dans un fichier. Nous effectuons cette opération à chaque exécution du programme.

```
#chargement du token
load("oauth_token")
```

Token qu'il faut ensuite valider pour entamer une session de traitements.

```
#validation pour une session d'analyse
ValidateToken(oauth_token)
```

Remarque : Attention, le jeton expire automatiquement après 60 minutes, même au cours de la même session de traitements. Il sera nécessaire de le régénérer à l'aide de la commande ValidateToken().

4.3 Première requête – Nombre de sessions et d'utilisateurs journaliers

Remarque : Attention, les requêtes sont soumises à quota. Même s'il y a de marge, il est prudent de les lancer avec discernement.

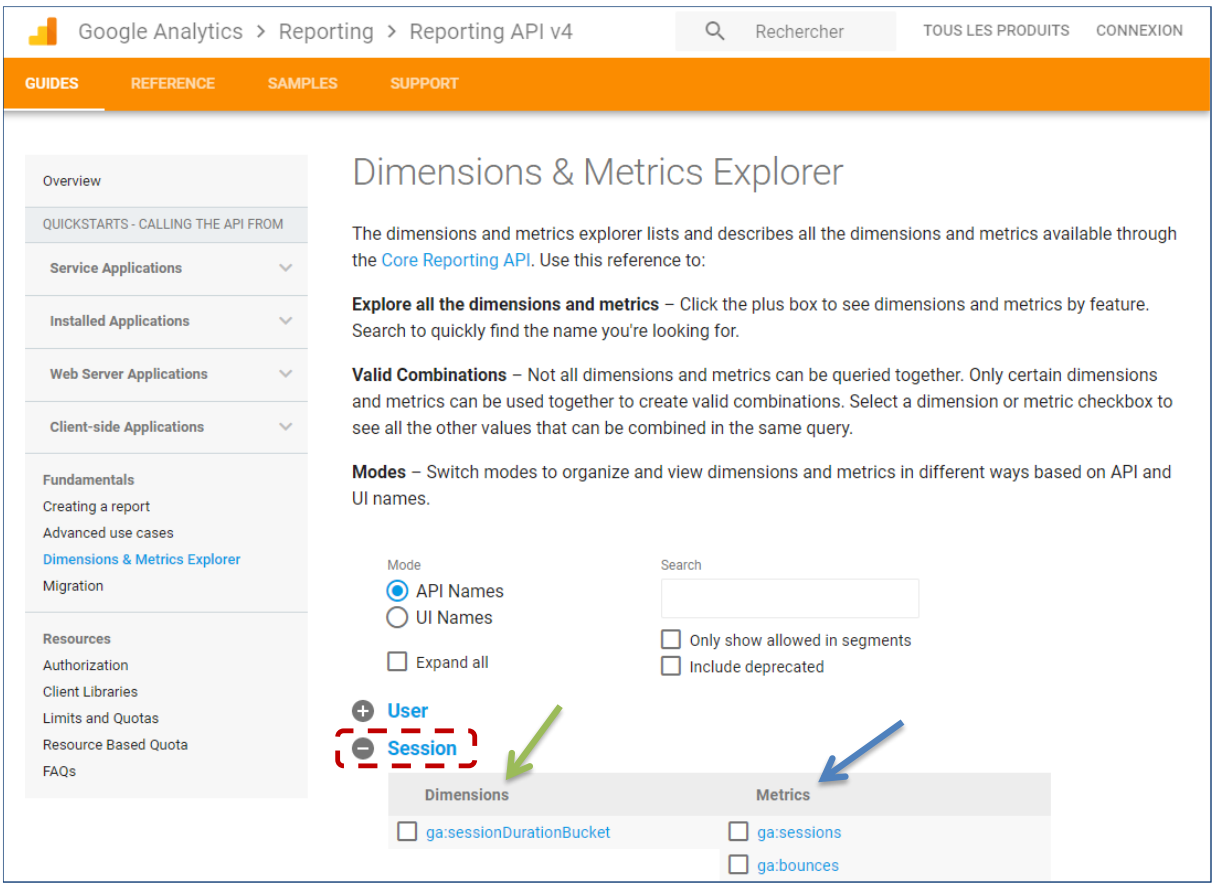
Plusieurs étapes sont nécessaires pour récupérer les données en ligne.

4.3.1 Paramétrage de la requête

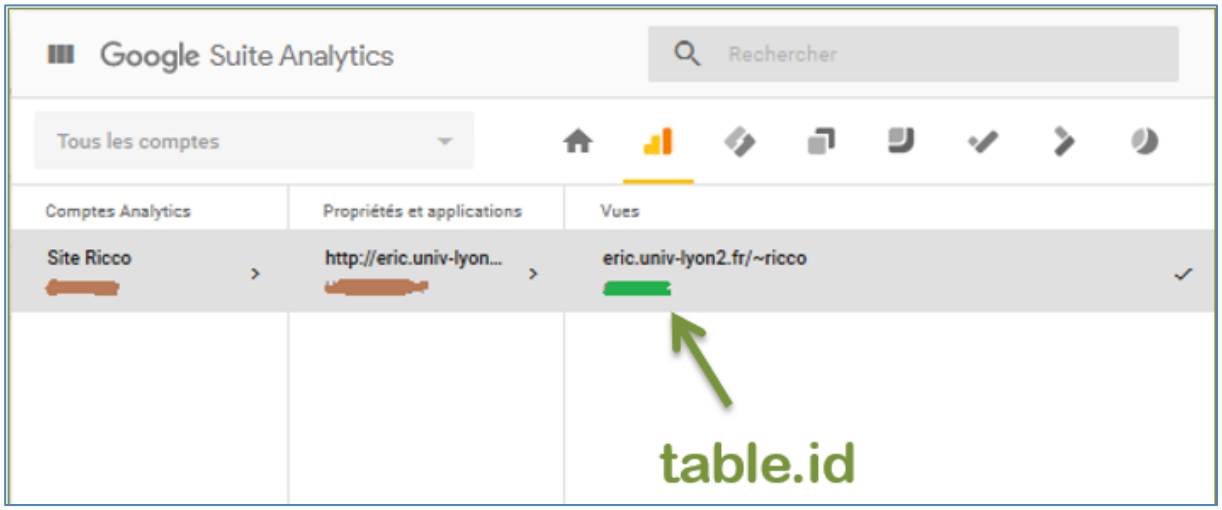
La première consiste à définir la requête que nous souhaitons exécuter. Nous utilisons la commande Init() en spécifiant :

- La période (start.date, end.date) ;
- Les dimensions (axes d'analyse, dimensions) et les mesures (indicateurs, metrics). La liste des items est accessible sur le site de Google. Nous pouvons vérifier en ligne (https://developers.google.com/analytics/devguides/reporting/core/dimsmets) les

combinaisons licites. Dans la copie d'écran ci-dessous, nous distinguons par exemple ce qu'il est possible de faire (dimensions et mesures) sur le thème « Session ».



- Le nombre maximum de lignes de résultats à extraire (max.results).
- L'éventuelle critère de tri à utiliser (sort).
- Le code du site à solliciter (table.id). Nous l'obtenons sur notre page Google Analytics.



Voici donc le paramétrage demandé :

```
#paramétrer la requête
query.list <- Init(start.date = "2017-11-01",
                   end.date = "2017-11-30",
                   dimensions = "ga:date",
                   metrics = "ga:sessions,ga:users",
                   max.results = 10000,
                   sort = "-ga:date",
                   table.id = "ga:TABLE_ID_DE_VOTRE_SITE")
```

Nous essayons de reconstituer la requête réalisée sur le site Google Analytics (Figure 1).

4.3.2 Initialisation et validation

L'étape suivante consiste à valider la requête pour obtenir un objet dédié. C'est le rôle de la commande QueryBuilder().

```
#valider la requête
ga.query <- QueryBuilder(query.list)
```

4.3.3 Réalisation de la requête – Extraction

Nous pouvons utiliser cet objet pour l'extraction des données avec GetReportData().

```
#récupérer les données
ga.data <- GetReportData(ga.query,oauth_token,split_daywise = TRUE)
```

L'option split_daywise permet de produire les données journalières.

Dans la console, R retrace l'accès à chaque valeur. Nous avons bien 30 observations au mois de novembre (n°0 à 29).

```
Access Token is valid
[ Run 0 of 29 ] Getting data for 2017-11-01
[ Run 1 of 29 ] Getting data for 2017-11-02
[ Run 2 of 29 ] Getting data for 2017-11-03
[ Run 3 of 29 ] Getting data for 2017-11-04
[ Run 4 of 29 ] Getting data for 2017-11-05
[ Run 5 of 29 ] Getting data for 2017-11-06
[ Run 6 of 29 ] Getting data for 2017-11-07
[ Run 7 of 29 ] Getting data for 2017-11-08
[ Run 8 of 29 ] Getting data for 2017-11-09
[ Run 9 of 29 ] Getting data for 2017-11-10
[ Run 10 of 29 ] Getting data for 2017-11-11
[ Run 11 of 29 ] Getting data for 2017-11-12
```

```
[ Run 12 of 29 ] Getting data for 2017-11-13
[ Run 13 of 29 ] Getting data for 2017-11-14
[ Run 14 of 29 ] Getting data for 2017-11-15
[ Run 15 of 29 ] Getting data for 2017-11-16
[ Run 16 of 29 ] Getting data for 2017-11-17
[ Run 17 of 29 ] Getting data for 2017-11-18
[ Run 18 of 29 ] Getting data for 2017-11-19
[ Run 19 of 29 ] Getting data for 2017-11-20
[ Run 20 of 29 ] Getting data for 2017-11-21
[ Run 21 of 29 ] Getting data for 2017-11-22
[ Run 22 of 29 ] Getting data for 2017-11-23
[ Run 23 of 29 ] Getting data for 2017-11-24
[ Run 24 of 29 ] Getting data for 2017-11-25
[ Run 25 of 29 ] Getting data for 2017-11-26
[ Run 26 of 29 ] Getting data for 2017-11-27
[ Run 27 of 29 ] Getting data for 2017-11-28
[ Run 28 of 29 ] Getting data for 2017-11-29
[ Run 29 of 29 ] Getting data for 2017-11-30
The API returned 30 results
```

L'objet obtenu est de type data.frame. Nous pouvons afficher les premières lignes.

```
#type d'objet obtenu
print(class(ga.data)) #classe data.frame

#afficher les premières lignes
print(head(ga.data))
```

Nous disposons de la date, du nombre de sessions et d'utilisateurs journaliers.

```
> print(head(ga.data))
      date sessions users
1 20171130      727   634
2 20171129      758   656
3 20171128      873   752
4 20171127      846   744
5 20171126      513   442
6 20171125      453   404
```

Total mensuel. A partir d'ici, nous nous retrouvons dans les conditions d'utilisation usuelles de R. Nous pouvons réaliser différents types de traitements sur l'ensemble de données. Pour calculer le nombre de sessions et utilisateurs mensuels par exemple, nous effectuons une simple somme :

```
#somme mensuelle
print(sapply(ga.data[c("sessions","users")],sum))
```

Pour le nombre de sessions (18708), nous retrouvons bien la valeur fournie par Google Analytics en ligne (Figure 1).

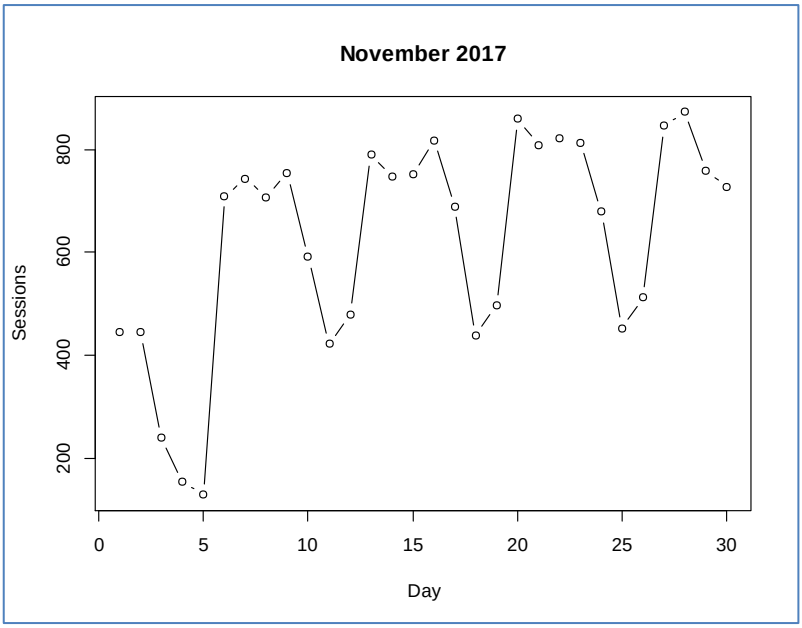
```
> print(sapply(ga.data[c("sessions","users")],sum))
sessions    users
   18708    16417
```

Très étrangement, elles divergent en revanche pour ce qui est du nombre d'utilisateurs (16417 vs. 13006). On pourrait s'étonner de cette différence, d'autant plus qu'en vérifiant les données journalières, elles concordent parfaitement. Il faut revenir à la définition du concept « utilisateurs » sur le site de support de Google pour la comprendre. Un utilisateur est identifié à l'aide d'un cookie. Il peut revenir plusieurs jours de suite. Sommer les valeurs journalières pour produire le total mensuel n'est pas une bonne approche (*d'où l'intérêt de lire attentivement la documentation*).

Graphique. De même, il est possible de reproduire le graphique des sessions journalières.

```
#graphique des sessions
plot(1:30,rev(ga.data$sessions),type="b",xlab="Day",ylab="Sessions",main="November 2017")
```

Attention, avec notre option de tri, les données les plus récentes sont en première position. La commande rev() permet de réorganiser les valeurs dans le bon sens.



4.4 Quelques requêtes

Le principe étant acquis, nous pouvons multiplier les requêtes pour analyser les accès à notre site web.

Nombre de sessions par pays. Pour ventiler les sessions par pays sur le mois de novembre 2017, nous spécifions « ga:country » en dimension, et « ga:sessions » en mesure. Nous trions le tableau de manière décroissante selon le nombre de sessions pour qu'apparaissent dans les premières positions les pays les plus attentifs à mes pages web.

```
#paramétrage : sessions par pays
query.list.pays <- Init(start.date = "2017-11-01",
                    end.date = "2017-11-30",
                    dimensions = "ga:country",
                    metrics = "ga:sessions",
                    max.results = 10000,
                    sort = "-ga:sessions",
                    table.id = "ga:TABLE_ID_DE_VOTRE_SITE ")

#validation
ga.query.pays <- QueryBuilder(query.list.pays)

#résultat
ga.data.pays <- GetReportData(ga.query.pays,oauth_token)
```

La commande nous annonce 136 résultats (*des personnes provenant de 136 pays différents sont venues sur mes pages durant le mois de novembre 2017 !*), nous le vérifions facilement en comptabilisant le nombre de lignes et de colonnes de notre data.frame avec dim(). Nous affichons les 10 premiers pays ensuite.

```
#vérification dimension
print(dim(ga.data.pays)) # (136, 2)
#10 premiers pays
print(head(ga.data.pays,10))
```

Nous retrouvons le rapport précédemment élaboré sur Google Analytics (Figure 2).

```
> print(head(ga.data.pays,10))
         country sessions
1         France    10166
2        Morocco     1235
3        Tunisia     1085
4        Algeria      988
```

```
5          Canada      393
6   United States      326
7   Côte d'Ivoire      324
8         Senegal      302
9         Belgium      300
10          India      265
```

Pages les plus populaires. Si l'on se penche sur les pages les plus populaires maintenant, nous précisons le titre de page « ga:pageTitle » en dimension, et « ga:sessions » toujours en mesure.

```
#paramétrage : sessions par page
query.list.page <- Init(start.date = "2017-11-01",
                        end.date = "2017-11-30",
                        dimensions = "ga:pageTitle",
                        metrics = "ga:sessions",
                        max.results = 10000,
                        sort = "-ga:sessions",
                        table.id = "ga:TABLE_ID_DE_VOTRE_SITE")
#validation
ga.query.page <- QueryBuilder(query.list.page)
#résultat
ga.data.page <- GetReportData(ga.query.page,oauth_token)
```

Nous obtenons 722 résultats. Ce nombre important de pages est dû en grande partie à mon site de tutoriels (1 post = 1 tutoriel = 1 page).

Lorsque nous affichons les dix premières…

```
#affichage des 10 pages les plus populaires
print(head(ga.data.page,10))
```

Nous obtenons :

	pageTitle	sessions
1	Cours Programmation R	2570
2	Supports de cours -- Data Mining, Data Science et Big Data Analytics	2552
3	Programmation Python pour les Statistiques et le Data Science	1334
4	Cours econometrie	1028
5	Tutoriels Tanagra pour le Data Mining et la Data Science	997
6	Cours Excel Avancé et Programmation VBA	787
7	Cours Régression Logistique	783
8	Tanagra EN	754
9	Tanagra FR	698
10	DATA - INFORMATIQUE - STATISTIQUE - Spécialité M2 SISE (Statistique	501

Comme nous le disions plus haut (section 2), les 3 premières pages…

```
#les 3 premières concentrent...
print(sum(ga.data.page$sessions[1:3])/18708) #34.5093
```

…ont concentré 34.5% des sessions en ce mois de novembre 2017. Il s'agit des pages de supports de cours pour : la programmation R, la data science, la programmation python. Ce n'est pas étonnant, elles sont au cœur de mon activité d'enseignant-chercheur à l'Université Lyon 2.

4.5 Analyse approfondie des résultats – Relation « Pays - Pages »

Nous avons utilisé R essentiellement comme outil de reporting jusqu'ici. Son intérêt n'est pas vraiment déterminant dans ce contexte. Pour aller plus loin, nous montrons dans cette section comment rebondir sur le résultat d'une requête pour initier une analyse qui exploite au mieux les capacités de traitements statistiques de R.

Nous souhaitons étudier les éventuelles associations entre les pays et les pages. Est-ce que certains pays sont plus attirés par certains thèmes que d'autres ? Nous verrons cela.

Requête. Nous reparamétrons la requête et nous l'exécutons.

```
#croisement pays - page
query.list.pp <- Init(start.date = "2017-11-01",
                      end.date = "2017-11-30",
                      dimensions = "ga:country,ga:pageTitle",
                      metrics = "ga:sessions",
                      max.results = 10000,
                       sort = "-ga.sessions",
                      table.id = "ga:TABLE_ID_DE_VOTRE_SITE")

#validation
ga.query.pp <- QueryBuilder(query.list.pp)
#résultat
ga.data.pp <- GetReportData(ga.query.pp,oauth_token)
```

R nous annonce 3385 résultats. Lorsque nous affichons les 5 premières…

```
#les 5 premières
print(head(ga.data.pp,5))
```

…, nous observons qu'elles sont monopolisées par la France qui s'intéresse avant tout à mes cours de programmation R, data science, programmation python, mon site de tutoriels, et ma page de cours Excel (au-delà d'Excel, il s'agit surtout de la page de garde du site regroupant mes cours).

	country	pageTitle	sessions
1	France	Cours Programmation R	1609
2	France	Supports de cours -- Data Mining, Data Science et Big Data Analytics	1561
3	France	Programmation Python pour les Statistiques et le Data Science	950
4	France	Tutoriels Tanagra pour le Data Mining et la Data Science	614
5	France	Cours Excel Avancé et Programmation VBA	519

Correspondance « Pays – Page ». Pour savoir si certains pays sont plus attirés par certaines pages que d'autres, nous formons un tableau croisant les pays et les pages en comptabilisant le nombre de sessions. Nous nous restreignons aux 10 pays les plus représentés vus ci-dessus, ainsi que les 10 pages les plus populaires.

```
#tableau croisé des pays et pages
TC <- xtabs(sessions ~ country + pageTitle, data = ga.data.pp)

#restriction aux 10 pays et 10 pages les plus représentés
TC10x10 <- TC[ga.data.pays$country[1:10],ga.data.page$pageTitle[1:10]]
```

Nous renommons les pages pour une meilleure lisibilité puis nous affichons le tableau.

```
#renommage des colonnes
colnames(TC10x10) <- c("R","DataScience","Python","Econometrie","Tutos","Excel","RegLog","TanagraEN","TanagraFR","MasterSISE")
#affichage
print(TC10x10)
```

Nous obtenons :

```
> print(TC10x10)
               pageTitle
country            R DataScience Python Econometrie Tutos Excel RegLog TanagraEN TanagraFR MasterSISE
  France        1609        1561    950         440   614   519    508        28       336        316
  Morocco        140         244     67          92    78    54     74         7        61         37
  Tunisia        160         183     32         102    52    51     33        11        69         15
  Algeria        170         144     57          51    69    30     24         5        48         30
  Canada          43          45     25          10    23    13     27        16        29          3
  United States   13          13      9           3     6     1      5        45         2          4
  Côte d'Ivoire   48          62     26          16    16    14      8         1         4         20
  Senegal         55          33     10          26     9     9     14         1        13         16
  Belgium         58          37     32          15    10     9     20         4        19          5
  India            1           2      3           0     2     0      0        83         0          0
```

Lire tel quel ce tableau n'amènera pas grand-chose. Il faudrait passer par les profils lignes ou colonnes pour identifier les similitudes et les dissemblances, puis calculer le khi-2 du test d'indépendance avec les indices d'attraction et de répulsion pour mettre en évidence les associations fortes. C'est le propos de l'analyse factorielle des correspondances (AFC).

Analyse des correspondances. Nous réalisons une analyse factorielle des correspondances (AFC) à l'aide de l'excellent package « ca » de Greenacre et al. (https://cran.r-project.org/web/packages/ca/index.html).

```
#librairie "ca"
library(ca)

#AFC
afc <- ca(TC10x10,nd=2)
print(afc)
```

Nous obtenons des résultats pour le moins déconcertants.

Principal inertias (eigenvalues):

	1	2	3	4	5	6	7	8	9
Value	0.473857	0.020396	0.006621	0.004718	0.002364	0.001815	0.00045	0.000194	0
Percentage	92.84%	4%	1.30%	0.92%	0.46%	0.36%	0.09%	0.04%	0%

Rows:

	France	Morocco	Tunisia	Algeria	Canada	Jnited States	Côte d'Ivoire	Senegal	Belgium	India
Mass	0.6808	0.0845	0.0701	0.0621	0.0232	0.0100	0.0213	0.0184	0.0207	0.0090
ChiDist	0.1388	0.2734	0.4067	0.2390	0.5287	3.0535	0.3534	0.4500	0.3128	6.3913
Inertia	0.0131	0.0063	0.0116	0.0035	0.0065	0.0932	0.0027	0.0037	0.0020	0.3678
Dim. 1	-0.1642	-0.1221	-0.0496	-0.1235	0.5094	4.4314	-0.1609	-0.1563	-0.0055	9.2838
Dim. 2	0.5567	-1.3644	-2.7580	-0.7412	-0.5623	0.5451	0.5115	-1.6259	0.0391	0.1489

Columns:

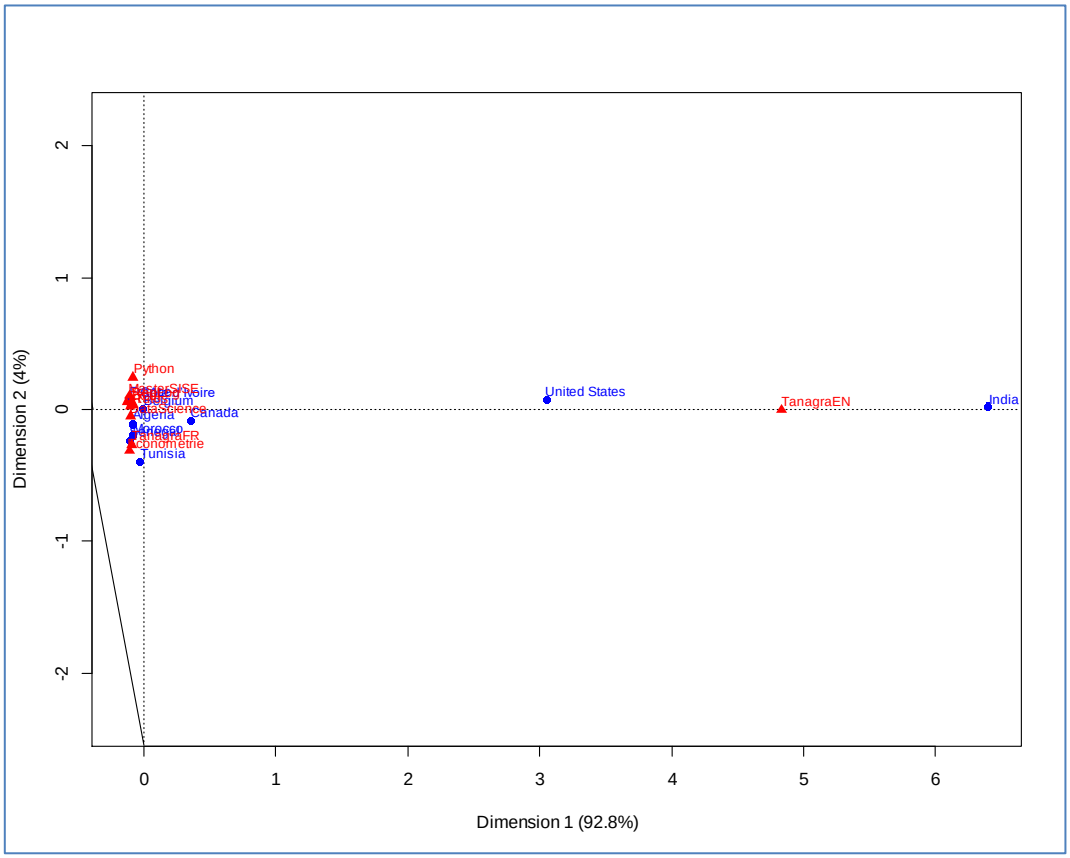
	R	DataScience	Python	Econometrie	Tutos	Excel	RegLog	TanagraEN	TanagraFR	MasterSISE
Mass	0.2273	0.2299	0.1198	0.0747	0.0870	0.0693	0.0705	0.0199	0.0575	0.0441
ChiDist	0.1516	0.1397	0.2637	0.3498	0.1461	0.1846	0.2317	4.8317	0.3505	0.2972
Inertia	0.0052	0.0045	0.0083	0.0091	0.0019	0.0024	0.0038	0.4643	0.0071	0.0039
Dim. 1	-0.1506	-0.1433	-0.1179	-0.1600	-0.1134	-0.1882	-0.1319	7.0190	-0.1278	-0.1542
Dim. 2	0.1764	-0.3428	1.7145	-2.1998	0.2024	0.3848	0.4192	-0.0374	-1.8756	0.7336

Le premier facteur restitue, à lui-seul, 92.84% de l'inertie disponible.

Une telle concentration n'est jamais bon signe. Elle exprime souvent l'existence de situations particulières qui accaparent l'information dans le tableau. On les remarque tout de suite dans le tableau des modalités lignes (Rows) avec une très forte inertie (Inertia) pour l'Inde (surtout) et les Etats-Unis. Ces pays ont une position très excentrée par rapport aux autres sur le premier facteur (Dim.1). Il en est de même dans le tableau des modalités colonnes où TanagraEN tient une place à part (inertie et coordonnées).

```
#représentation simultanée
plot(afc)
```

La représentation simultanée des modalités lignes et colonnes dans le premier plan factoriel précise la nature du résultat.



Ces positions excentrées résultent d'une attraction particulière entre ces deux pays et la page en anglais de Tanagra. Elle masque complètement tout le reste de l'information que l'on pourrait extraire du tableau.

Analyse des correspondances (bis). Nous retirons ces modalités singulières du tableau.

```
#tableau restreint
TC2 <- TC10x10[-c(6,10),-8]
print(TC2)
```

```
              pageTitle
country            R DataScience Python Econometrie Tutos Excel RegLog TanagraFR MasterSISE
  France        1609        1561    950         440   614   519    508       336        316
  Morocco        140         244     67          92    78    54     74        61         37
  Tunisia        160         183     32         102    52    51     33        69         15
  Algeria        170         144     57          51    69    30     24        48         30
  Canada          43          45     25          10    23    13     27        29          3
  Côte d'Ivoire   48          62     26          16    16    14      8         4         20
  Senegal         55          33     10          26     9     9     14        13         16
  Belgium         58          37     32          15    10     9     20        19          5
```

Nous réitérons donc l'AFC sur les pays francophones et les pages exclusivement en français.

```
#afc de nouveau sur le tableau restreint
afc2 <- ca(TC2,nd=2)
print(afc2)
```

Il n'y a plus d'excentricités manifestes dans les résultats. Aucune des modalités n'accapare de manière indue l'information contenue dans les données.

```
Principal inertias (eigenvalues):
           1        2        3       4        5        6       7
Value      0.020656 0.006957 0.00479 0.002379 0.001791 0.00045 0.000171
Percentage 55.54%   18.7%    12.88%  6.4%     4.82%    1.21%   0.46%


 Rows:
           France   Morocco  Tunisia  Algeria    Canada Côte d'Ivoire  Senegal   Belgium
Mass     0.696302  0.086060 0.070819 0.063300  0.022150      0.021744 0.018797  0.020829
ChiDist  0.080968  0.259042 0.406110 0.222627  0.423027      0.335072 0.435099  0.315633
Inertia  0.004565  0.005775 0.011680 0.003137  0.003964      0.002441 0.003558  0.002075
Dim. 1  -0.556124  1.348344 2.744133 0.730398  0.570502     -0.508846 1.602053 -0.051134
Dim. 2   0.001731 -0.116933 0.109316 0.595543 -4.747231      3.593843 2.082782 -2.339198


 Columns:
                R DataScience    Python Econometrie     Tutos     Excel    RegLog TanagraFR MasterSISE
Mass     0.231965    0.234607  0.121825    0.076407  0.088498  0.071022  0.071937  0.058830   0.044910
ChiDist  0.110481    0.098922  0.250582    0.330955  0.123877  0.127278  0.213975  0.340139   0.276813
Inertia  0.002831    0.002296  0.007650    0.008369  0.001358  0.001151  0.003294  0.006806   0.003441
Dim. 1  -0.175402    0.341532 -1.700967    2.173901 -0.194196 -0.395215 -0.411857  1.852534  -0.721927
Dim. 2   0.275177    0.370252 -0.480100    0.966464 -0.242438  0.074050 -1.810606 -2.300198   2.576575
```

Le premier plan restitue 74.24% de l'inertie totale. C'est suffisamment élevé pour que l'analyse du graphique factoriel ne soit pas faussée[2].

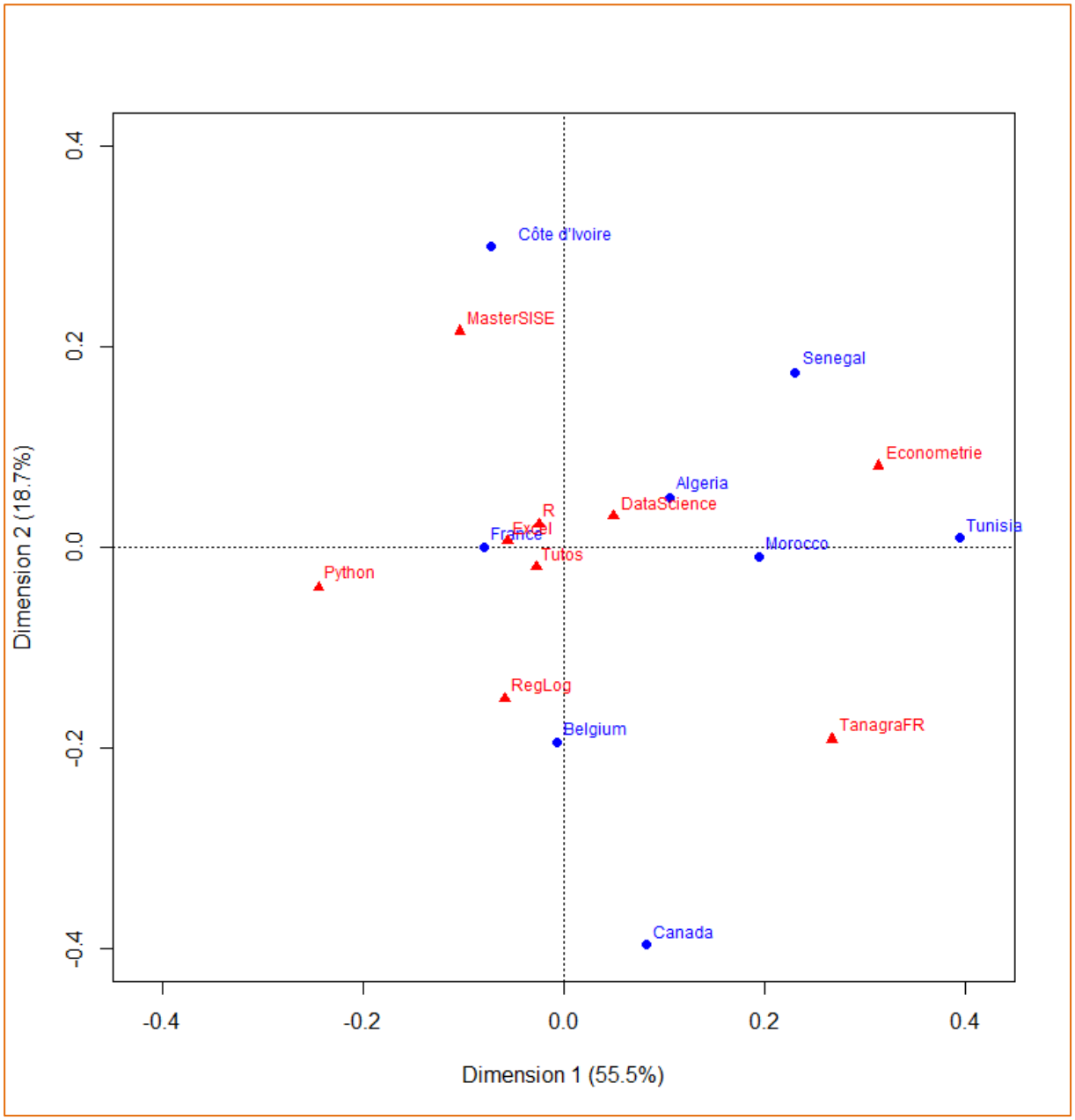
```
#plotting
plot(afc2,xlim=c(-0.4,0.4),ylim=c(-0.4,0.4))
```

Plusieurs éléments attirent notre attention :

- Il y a une attraction entre la {Tunisie (surtout), le Maroc et le Sénégal} avec la page dédiée à l'économétrie (18.05% de l'information totale si l'on va dans le détail).
- La page Tanagra en français est surtout visité par le Canada et la Tunisie (10.79%).

[2] Il faut toujours veiller à ce que le graphique soit bien carré dans une représentation factorielle. En effet, les pourcentages d'inertie en ligne et en colonne ne sont pas identiques (dans notre cas, 55.5% vs. 18.7%). Une différence d'échelle entre l'abscisse et l'ordonnée fausserait la perception des proximités.

- La France détermine en grande partie le profil moyen, elle est en plein centre du repère factoriel, c'est normal au regard de son poids dans le tableau. Ceci étant posé, on remarque qu'elle est surtout attirée par ma page Python. Mes étudiants de la L3 IDS (Informatique et Data Science) de l'Université Lyon 2 qui suivent mon cours de Python comptent pour beaucoup là-dedans je pense.



On pourrait ainsi approfondir l'analyse en rajoutant d'autres pays francophones, ou encore en se retreignant aux pages en anglais, etc. L'accès aux données de Google Analytics sous R nous offre des perspectives élargies d'analyse des données de fréquentation de nos sites web. C'est le principal message de ce tutoriel.

5 Conclusion

Google Analytics est un outil de mesure des audiences web. Avec l'API Google Analytics, nous avons accès à ces données sous R via le package RGoogleAnalytics. Dans ce tutoriel, nous avons montré comment extraire en ligne les informations, puis comment les exploiter par la suite sous R. La vaste liste des dimensions et mesures disponibles (https://developers.google.com/analytics/devguides/reporting/core/dimsmets) nous offre de nombreuses opportunités d'études.

6 Références

Emerson H., « Using Google Analytics with R », R-bloggers, August 2015 ; https://www.r-bloggers.com/using-google-analytics-with-r/

Granowitz A., « Using Google Analytics with R », September 2014 ; https://developers.google.com/analytics/solutions/r-google-analytics

Rakotomalala R., « Analyse factorielle des correspondances – Diapos », Juillet 2013 ; http://tutoriels-data-mining.blogspot.fr/2013/07/analyse-factorielle-des-correspondances.html

Shah K., « How to extract Google Analytics data in R using RGoogleAnalytics », November 2014 ; https://www.tatvic.com/blog/google-analytics-data-extraction-in-r/

Wikipédia, « Audience d'un site web », consulté le 03 décembre 2017.