Big Data to the rescue of Google Analytics (I) - BigQuery

I'm a big fun of Google Analytics. I met this tool for the first time 6 years ago, when the mighty *Omniture*, the versatile *Coremetrics* and the reliable and data-protection friendly *Webtrekk* dominated the web analytics market. Ever since, I've been closely following the development of the web analytics baby acquired by Google. At present, Google Analytics aims at going even beyond the web realm, with their universal analytics approach and provides a -to me- unbeatable freemium business analytics suit crafted after many years of innovation and research at Google's Mountain View Headquarters.

The result is at the same time, one of the best and more matured Big Data tools on the market, where you can see all challenges proper of a Big Data product addressed. I particularly like they handled the precision-performance trade-off, just by putting the decision on the user's hands (see image below):

Google Analytics Limitations

In spite of all advanced analytics you get out of the box with Google Analytics, there's a big limitation which prevents you from making the most of the data you collect: you can **just access to aggregated data and sampled reports**, not to individual sessions data. In addition to that, when you handle way too much data, reports get automatically sampled, also for Premium accounts.

Performing analysis **crossing different data sources**, is not an easy task in GA either. Even with the improvements in the Import Data functionality introduced by universal analytics, the still existing constraints make this task really difficult.

The next key limitation is the concept of segments: **managing different segments** and creating combinations of them is everything but simple. You usually are limited to the pre-defined set of filters but your requirements might go beyond that.

GA is a great tool to report on what's happening now vs. what happened in the past in your site, but not appropriate to run typical **Business Intelligence tasks**, like pattern discovery, time series analysis, clustering, regression models,

etc. When you have a reduced amount of data and you don't have to go down to session level, you are probably well off with RGoogleAnalytics, an awesome **#RSTAT** package relying on the Google Analytics API... But for finer granularity analysis you need more.

BigQuery to the rescue!

BigQuery is basically the Google move on the IaaS space for large amount of data. It works on Google Storage and allows for interactive analysis and requires the data to be imported (in CSV or JSON) using the HTTP API.

For Google Analytics Premium accounts, a daily export into BigQuery is free, which makes the combination of both even more attractive. And the data you get exported is raw! (not sampled, not aggregated)... to use the way you want! The specific exporting format is given here. Usually, you get the export with a day of delay, but for your analytics.
To get a feeling of how the query language looks like, have a look at following SQL statements... Does it look like standard SQL? Yes, you are right!

To find out which products have been bought by customers who purchased product A:

```
SELECT hits.item.productName AS other_purchased_products, COUNT(hits.item.productName)
AS quantity   FROM myproject.ga_sessions_20141126 WHERE fullVisitorId IN   ( SELECT ful
lVisitorId FROM myproject.ga_sessions_20141126 WHERE hits.item.productName   CONTAINS '
A' AND totals.transactions>=1 GROUP BY fullVisitorId )   AND hits.item.productName IS N
OT NULL AND hits.item.productName !='A'   GROUP BY other_purchased_products ORDER BY qu
antity DESC;
```

To understand the average revenue spent per visit:

```
SELECT ( SUM(total_transactionrevenue_per_user) / SUM(total_visits_per_user) ) AS avg_r
evenue_by_user_per_visit   FROM   ( SELECT SUM(totals.visits) AS total_visits_per_user,
 SUM( totals.transactionRevenue )   AS total_transactionrevenue_per_user, visitorId   F
ROM myproject.ga_sessions_20141126   WHERE totals.visits>0 AND totals.transactions>=1 A
ND totals.transactionRevenue IS NOT NULL GROUP BY visitorId )
```

Buyers vs. Dwellers for the 26th of November:

```
select type, avg_pageviews_per_user from (  SELECT 'buyers' as type, (SUM (total_pagevi
ews_per_user) / COUNT (users) ) AS avg_pageviews_per_user   FROM (  SELECT fullVisitorI
d AS users, SUM (totals.pageviews) AS total_pageviews_per_user  FROM my project.ga_sess
ions_20141126  where if (totals.transactions is null, 0, totals.transactions) >= 1  GRO
UP BY users ) ),  (SELECT 'dwellers' as type, (SUM (total_pageviews_per_user) / COUNT(u
sers) ) AS avg_pageviews_per_user  FROM (  SELECT fullVisitorId AS users, SUM (total.pa
geviews) AS total_pageviews_per_user  FROM myproject.ga_sessions_20141126  WHERE if(tot
als.transactions is null, 0 , totals.transactions) = 0  GROUP BY users ));
```

Following Videos provide more information on how to analyze GA data with Big Query:

You might be interested in reading more about it:

- Big Query Pricing
- Big Query Browser Toolkit
- Big Query Command Line
- Big Query Command Line

In an upcoming article, I'm going to share with you a few tricks on how to run BigQuery out of your R script.