

User Configuration **Manual**

How Cooperative Game Theory can be utilised to enhance marketing analytics attribution.

Master of Science in Data Analytics

James Matthews, X13109260, James.Matthews1@student.ncirl.ie

Contents

1.	User Configuration Overview	3
2.	Prerequisite Google Analytics Configuration	3
2.1	Configuration Goals	3
2.1.1	Destination Goal Count for Completed Conversions	3
2.1.2	Destination Goal for Count Abandoned (null) Conversions	3
2.1.3	Smart Goal to Capture Transaction Value	3
2.2	Configuration Custom Variables	4
2.2.1	Custom Variable to Record Conversion Path.....	4
2.2	Google Analytics OAuth token authentication	4
3.	Mockaroo Dataset Generation	4
3.1	Defining Mockaroo Data Schema	4
3.2	Generating & Downloading Mockaroo Dataset	6
4.	Core Solution Framework Elements.....	6
4.1	Real-time Element for Polling Data.....	7
4.4.1	Scheduling R Script.....	7
4.4.1	Functionality Google_Analytics_Final.r	8
4.2	Attribution Modelling Overview.....	9
4.2.1	Last Click Attribution Model.....	10
4.2.1	Cooperative Attribution Model	11
4.3	Running Attribution Model.....	12
6	References.....	13
7	Appendix	13

1. User Configuration Overview

The solution framework assembled by the author is targeted towards enterprise and small medium businesses that are utilising the Google Technology stack. The solution can be replicated and adapted to work against instances of Google Analytics that have completed the prerequisites customisation steps outlined in section 2.

This user configuration manual details each element of the framework and the necessary steps involved in conducting the attribution benchmark experiment to address How Cooperative Game Theory can be utilised to enhance marketing analytics attribution.

2. Prerequisite Google Analytics Configuration

This section outlines the prerequisite steps that must be completed prior to running the solution framework.

2.1 Configuration Goals

The following goals must be configured in the instance of Google Analytics where the solution framework will be in use. Google Analytics has three different types of goals from which to choose from, please refer to the steps outlined in the [Google Analytics - Create, edit, and share goals](#) guide in setting up the following Goals (Support.google.com, 2016).

2.1.1 Destination Goal Count for Completed Conversions

Destination Goal type, counts the number of successful online transactions and sessions that has been completed on the business website, to be set-up to trigger on displaying of the payment confirmation webpage.

2.1.2 Destination Goal for Count Abandoned (null) Conversions

Destination Goal type, that counts the number of abandoned transactions and sessions that has been completed on the business website, to be set-up to trigger on webpages prior to the payment confirmation.

2.1.3 Smart Goal to Capture Transaction Value

Smart Goal type that captures the total currency value of online transactions that have been completed on the business website, to be set-up to trigger on displaying of the payment confirmation webpage.

2.2 Configuration Custom Variables

The following custom variable must be configured in the instance of Google Analytics where the solution framework will be in use. Use of custom variables gives websites the ability to include & format non-standard data in their instance of Google Analytics. Please refer to the steps outlined in the [Google Analytics - Custom dimensions & metrics](#) guide in setting up the following custom variable (Support.google.com, 2016).

2.2.1 Custom Variable to Record Conversion Path

Session based Custom Variable that captures the conversion path flow of customers; custom variable is to be placed on website landing pages that are being used in Online Marketing activities alongside the website checkout ecommerce pages. Custom Variable is to capture the source/medium values of all active sessions on the website; source/medium will show the marketing channel name and its source of the conversion path a user has taken. For example Display GDN > Organic Google > PPC > Direct > Purchase.

2.2 Google Analytics Oauth token authentication

In order to use the Real-time Element for Polling Data from Google Analytics, a Oauth authentication token must be obtained. To generate the Google Analytics Oauth authentication token please refer to the steps outlined in the [Google Analytics - Using OAuth 2.0 to Access Google APIs](#) guide (Google Developers, 2016). Once the Client ID and Client Secret tokens have been generated, replace the placeholder tokens within the script `Google_Analytics_Final.r` outlined in see section 4.1.

3. Mockaroo Dataset Generation

Mockaroo is an online platform that allows for the generation en masse of synthesized and realistic datasets for use in modelling, testing and demoing purposes. This section outlines the steps involved in generating a synthesized dataset for use in the attribution benchmark experiments (Mockaroo, 2016).

3.1 Defining Mockaroo Data Schema

Mockaroo generates datasets by utilising a data schema, which is based upon an existing underlying data structure predefined by the user. The Mockaroo data schema has two components in the form of a Field Name and Field Type.

The Field Name denotes an attribute name in a given dataset, whilst Field Type being associated with a Field Name contains either a numerical or categorical value. Field Types if numerical are assigned a numeric range that matches the underlying data structure. Moreover, categorical Field Types are assigned all possible characteristic values again based upon the underlying data structure. The data schema is inputted and defined in the Mockaroo platform, in company with the number of required rows needed in the dataset.

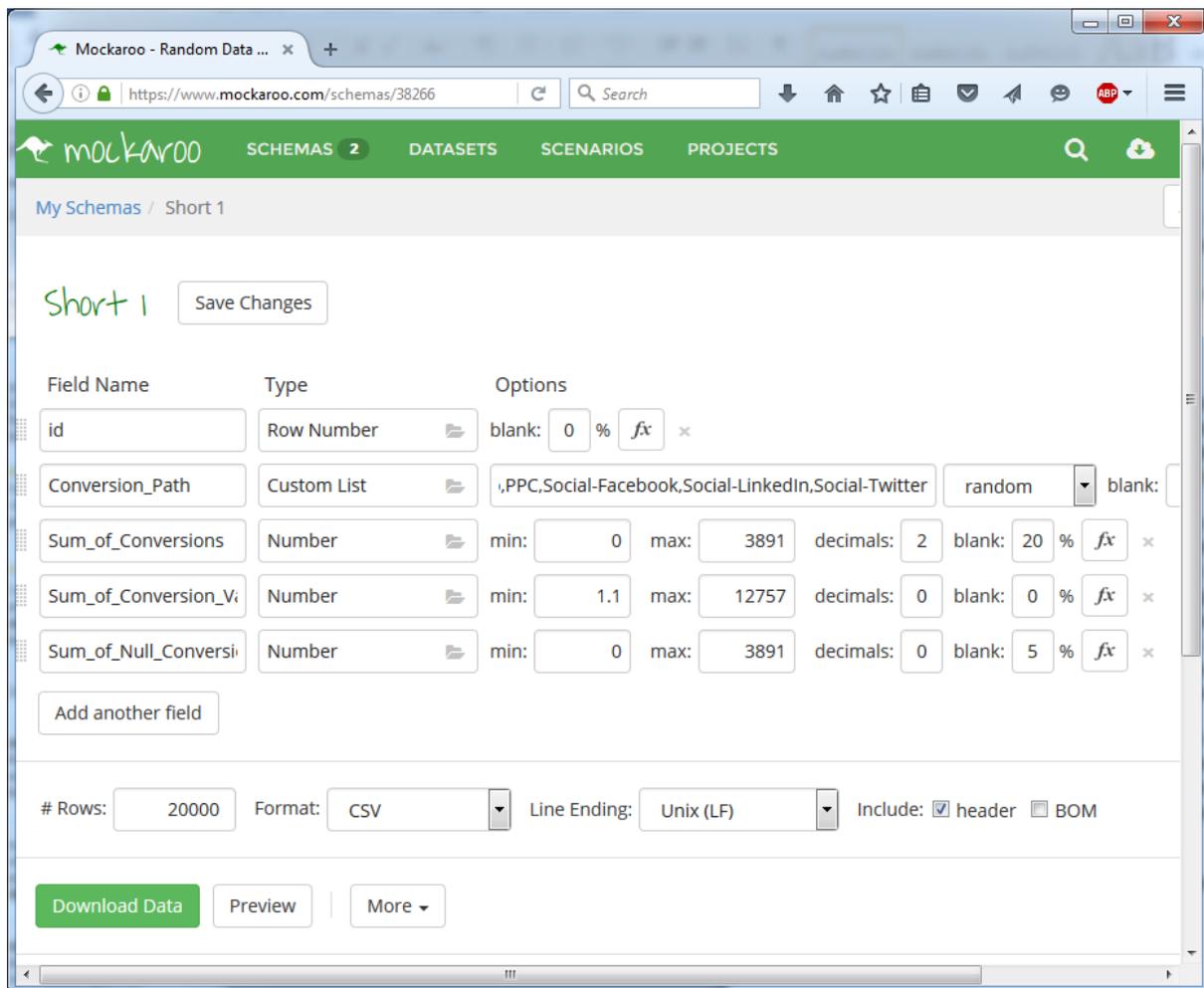


Figure 1 - Example Data Schema

Figure 1 illustrates an example data schema that will generate conversion path data that can be utilised in the attribution model. The schema contains a total of five Field Names alongside their respective Field Types.

- ID generates a unique identification number for each dataset entry.

- Conversion Path field type will randomly assign a conversion path characteristic value from the categorical value list. Which is made up of the Marketing channel names.
- Sum of Conversions and Sum of null conversions assigns a value from a numeric range 0 to 3891.
- Sum of conversions assigns a value from a numeric range of 1.1 through to 12,757.

3.2 Generating & Downloading Mockaroo Dataset

To generate a Mockaroo dataset follow these steps:

1. Go to www.mockaroo.com
2. Register for an account by clicking 'sign in' and follow the on screen prompts.
3. Login to your mockaroo account.
4. Click on data schema.
5. Enter each field name and select a corresponding Field type. Repeat this step until you have the number of desired Field Names & Field Types inputted.
6. Once the schema has been defined in Mockaroo the required number of rows needed is entered into the count rows field and a format for the file type is selected
7. 'Click Download Data'

. Upon clicking the "Download Data" button Mockaroo generates the dataset based upon the data schema creating the Field Names first and then randomly assigning the categorical Field types with their associated numeric range or characteristic values. The generated dataset is then downloaded and saved locally to the user's machine.

4. Core Solution Framework Elements

This section of the user configuration manual outlines the core elements of the solution framework. The framework runs on the statistical programming language called R, which can be downloaded and installed for free from this [link](#). Both files should be copied and save into a working directory on your local PC, all Google Analytics Conversion path data and attribution modelling output will be written to this directory.

4.1 Real-time Element for Polling Data.

The real-time element of the solution framework is an R programming script titled `Google_Analytics_Final.r`. This script utilises the `RGoogleAnalytics` package and the code within is based upon examples provided in the `RGoogleAnalytics` Reference manual (Pearmain, 2016).

4.4.1 Scheduling R Script

The script `Google_Analytics_Final.r` is set to run every 120 minutes via a schedule task set up in windows task scheduler.

To create a schedule task follow these steps:

1. Open the Windows Task Scheduler Application
2. Click Create Task
3. Assign the Task the name of Google Analytics Data Downloader.
4. Click Triggers Tab > New > Daily > Repeat Task Every:
5. Check Repeat Task Every and input 120 minutes, set duration of indefinitely.
6. Click Actions Tab > New > Start a Program
7. From program/script Browse to where you have R installed on your PC.
Typically the installation path will be on your C drive “C:\Program Files\R\R-3.3.1\bin”
8. Enter “`Google_Analytics_Final.r`” as the add arguments. Click OK
9. The create task action window, should be similar to Figure 2 below.
10. Click Ok and Ok again.
11. The data polling element of the framework will now run every 120 minutes.

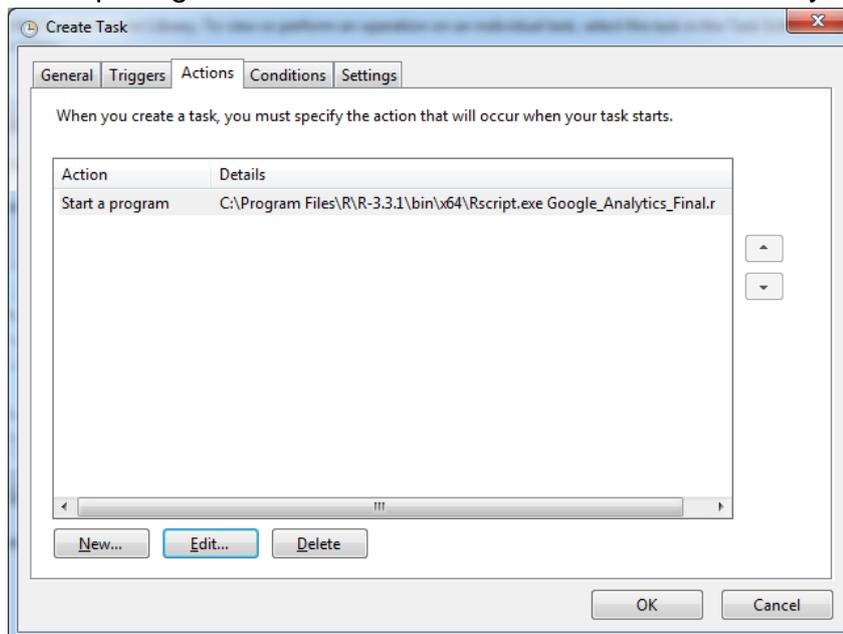


Figure 2 - Scheduled Task Actions Tab for `Google_Analytics_Final.r`

4.4.1 Functionality Google_Analytics_Final.r

The script Google_Analytics_Final.r is set to run every 120 minutes via windows task scheduler. The script will automatically save a CSV file titled “Google_Analytics_Conversion_Path_Data.csv” of conversion path information to your local PC.

Figure 3 below shows the Google API Query Parameters, to fully configure the Google_Analytics_Final.r script, to suit your environment follow these steps to update the query parameters:

1. Install the RGoogleAnalytics package by running the command `install.packages(“RGoogleAnalytics”)` within your R console.
2. Change the Set working file path to match the location of where you have the R scripts saved on your PC.
3. Load RGoogleAnalytics Package – Highlight Line 14 > Click the run command.
This command will load the needed package to connect to the Google Analytics API.
4. Replace the ouath token client.id and client secret id with your newly generated credentials from step 2.2 above.
5. Set Query Start Date to the desired date.
6. Set Query End Date; note this can be set to a future date eliminating the need to set the date each time.
7. Check that the newly created goals and custom variable sequence numbers match that of the script. To note if you have any previous goals created in your Google Analytics account then the number sequences will be different, goal sequence numbers are detailed in your Google Analytics view.
 - a. ga:customVarValue1
 - b. ga:customVarValue2
 - c. ga:goal1Completions
 - d. ga:goal2Completions
 - e. ga:goal3Completions
8. Change the table.id to the Google Analytics View where the query should be run against.
9. Set-up complete.

```

# Generate the oauth_token object
oauth_token <- Auth(client.id = "123456789-XXXXXXXXXXXXXXXXX.apps.googleusercontent.com",
  client.secret = "TUXXXXXXXXXXXXXXXXX_TknUI")
# Save the token object for future sessions
save(oauth_token, file="oauth_token")

# Load the token object
load("oauth_token")

# Create a list of Query Parameters
query.list <- Init(start.date = "2016-07-01",
  end.date = "2016-07-31",
  sort='-ga:date', #Order Returned results by date
  dimensions='ga:date,ga:medium,ga:customVarName1,ga:customVarName2', #Set set dimensions
  metrics = "ga:sessions,ga:pageviews,ga:customVarValue1,ga:customVarValue2,ga:goal1Completions,
  ga:goal2Completions,ga:goal3Completions", #Get Metric Data from set dimensions
  max.results = 1000, #Get 1000 Rows
  table.id = "ga:XXXXXXXX") #Set Profile ID of Google Analytics View where query should be run against.

# Create the query object
ga.query <- queryBuilder(query.list)

# Fire the query to the Google Analytics API
ga.df <- GetReportData(query, oauth_token)

#Write Google Analytics Data to CSV
write.csv(ga.df, file = "Google_Analytics_Conversion_Path_Data.csv")

```

Figure 3 - Google_Analytics_Final.r Google API Query Parameters.

4.2 Attribution Modelling Overview.

The attribution modelling element of the solution framework is a R programming script titled Attribution_model_Final.r. This script utilises the ChannelAttribution package and the code within is based upon examples provided in the ChannelAttribution Reference manual (Altomare, 2016).

To configure the attribution modelling script follow these steps:

1. Install the RGoogleAnalytics package by running the command `install.packages("ChannelAttribution")` within your R console.
2. Change the Set working file path to match the location of where you have the R scripts saved on your PC.
3. Check you have latest conversion path CSV file from Google Analytics, titled `Google_Analytics_Conversion_Path_Data.csv`

4.2.1 Last Click Attribution Model

Figure 4 below shows the Last Click attribution model code in situ.

```
#####  
# LAST CLICK ATTRIBUTION MODEL  
#####  
#Run Last Click Attribution Model  
#Get Last Click volume of Conversions  
heuristic_models(Data,"Conversion_Path","sum_of_Conversions")  
  
#Get Last Click Value of Conversions  
heuristic_models(Data,"Conversion_Path","sum_of_Conversions",var_value="sum_of_Conversion_Value")  
  
#write Results to CSV  
writeResultsCSV <- capture.output(heuristic_models(Data,"Conversion_Path","sum_of_Conversions",  
var_value="sum_of_Conversion_Value"))  
cat("Last Click", writeResultsCSV, file="Last Click Attribution Model.csv", sep="n", append=TRUE)  
#Clear R Console  
cat("\014")
```

Figure 4 – Attribution_model_Final.r - Last Click Attribution Model Functionality

To run the Last Click Attribution Model follow these steps:

1. Load ChannelAttribution Package – Highlight Line 14 > Click the run command.
This command will load the needed package to run the attribution model.
2. Import Dataset – Highlight Lines 20 & 21 > Click the run command
This command will import the latest conversion path information from Google_Analytics_Conversion_Path_Data.csv file.
3. Run Last Click Model > Calculate Last Click Volume of Conversions > Highlight Line 28 > Click the run command
Outputted Last Click Volume of Conversions results can be interpreted onscreen in R.

```
> heuristic_models(Data,"Conversion_Path","sum_of_Conversions")  
  channel_name first_touch last_touch linear_touch  
1          PPC      24331      15062 16667.63929  
2 organic-Google  14415      25272 21309.49286  
3 social-Twitter    215         731  1437.59524  
4 Display-Quantcast 3945         6082 4633.35833  
5          Direct 17027      28230 22117.41548  
6 social-Facebook   533         3581  2462.22143  
7  Display-GDN    22311         5596 14479.62738  
8  Display-AdRoll  7110         4553  6179.56310  
9          Email   914          470   818.10595  
10 social-LinkedIn  517         1723 1200.97619  
11 organic-Yahoo   14           14   17.33333  
12 organic-Bing    6            24   14.67143
```

4. Run Last Click Model > Calculate Last Click Value of Conversions > Highlight Line 31 > Click the run command
Outputted Last Click Value of Conversions results can be interpreted onscreen in R.

```
> heuristic_models(Data, "Conversion_Path", "sum_of_Conversions", var_value="sum_of_Conversion_value")
```

	channel_name	first_touch_conversions	first_touch_value	last_touch_conversions	last_touch_value
1	PPC	24331	55188.1372	15062	36244.366
2	Organic-Google	14415	39855.8513	25272	64613.203
3	Social-Twitter	215	375.9490	731	1453.194
4	Display-Quantcast	3945	9585.5478	6082	14327.896
5	Direct	17027	34531.8616	28230	60936.183
6	Social-Facebook	533	1093.6213	3581	6960.226
7	Display-GDN	22311	49036.2646	5596	10997.882
8	Display-AdRoll	7110	16485.4774	4553	8891.465
9	Email	914	1955.0938	470	1229.025
10	Social-LinkedIn	517	971.9937	1723	3411.371
11	Organic-Yahoo	14	23.6480	14	23.648
12	organic-Bing	6	15.7380	24	30.724

5. Write Last Click Attribution Results to CSV file > Highlight line 34 to 36 > Click the run command.

This command will save to CSV file the outputted results in terms of conversion value volume to a CSV file titled: *Last Click Attribution Model.csv*

6. Process Complete

4.2.1 Cooperative Attribution Model

Figure 5 below shows the Cooperative attribution model code in situ,

```
#####
# COOPERATIVE ATTRIBUTION MODEL
#####
#Run Cooperative Attribution Model
#Calculate cooperative volume of conversions
markov_model(Data, "Conversion_Path", "sum_of_Conversions")

#Calculate cooperative value of conversions
markov_model(Data, "Conversion_Path", "sum_of_Conversions", var_value="sum_of_Conversion_value")

#Calculate Contributions of Individual Marketing Channel
markov_model(Data, "Conversion_Path", "sum_of_Conversions", var_value="sum_of_Conversion_value",
var_null="sum_of_Null_Conversions")
markov_model(Data, "Conversion_Path", "sum_of_Conversions", var_value="sum_of_Conversion_value",
var_null="sum_of_Null_Conversions",
out_more=TRUE)

#Write Attribution Model Results to CSV
writeResultsCSV <- capture.output(markov_model(Data, "Conversion_Path", "sum_of_Conversions",
var_value="sum_of_Conversion_value",
var_null="sum_of_Null_Conversions"))
cat("Cooperative", writeResultsCSV, file="Cooperative Attribution Model.csv", sep="n", append=TRUE)

#Clear R Console
cat("\014")
```

Figure 5 Attribution_model_Final.r - Cooperative Attribution Model Functionality

To run the Last Cooperative follow these steps:

1. Load ChannelAttribution Package – Highlight Line 14 > Click the run command.
This command will load the need package to run the attribution model.
2. Import Dataset – Highlight Lines 20 & 21 > Click the run command
This command will import the latest version of the Google_Analytics_Conversion_Path_Data.csv file.

- Run Cooperative Model > Calculate Cooperative Volume of Conversions > Highlight Line 45 > Click the run command
Outputted Cooperative Volume of Conversions results can be interpreted onscreen in R.

```
> markov_model(Data, Conversion_Path ,
  channel_name total_conversions
1 PPC 17680.272958
2 Organic-Google 16366.026606
3 Social-Twitter 2405.739085
4 Display-Quantcast 6542.061643
5 Direct 18080.168741
6 Social-Facebook 3422.980612
7 Display-GDN 15157.323040
8 Display-AdRoll 8959.468775
9 Email 924.427519
10 Social-LinkedIn 1768.239442
11 Organic-Yahoo 9.016218
12 Organic-Bing 22.275362
```

- Run Cooperative Model > Calculate Cooperative Value of Conversions > Highlight Line 48 > Click the run command
Outputted Cooperative Value of Conversions results can be interpreted onscreen in R.

```
> markov_model(Data, "Conversion_Path", "Sum_of_Conversions",
  channel_name total_conversion total_conversion_value
1 PPC 17719.120612 40990.88216
2 Organic-Google 16483.125717 39786.32682
3 Social-Twitter 2452.294781 5607.87334
4 Display-Quantcast 6436.541890 14981.34068
5 Direct 17891.052718 39883.56656
6 Social-Facebook 3401.381300 7414.58596
7 Display-GDN 15102.878188 33802.88651
8 Display-AdRoll 9032.557007 20350.68649
9 Email 982.088967 2357.91364
10 Social-LinkedIn 1808.214754 3892.02471
11 Organic-Yahoo 9.049058 16.03048
12 Organic-Bing 19.695009 35.06638
```

- Write Cooperative attribution results to CSV file > Highlight line 58 to 61 > Click the run command.
This command will save to CSV file the outputted results in terms of conversion value volume to a CSV file titled: *Cooperative Attribution Model.csv*
- Process Complete.

4.3 Attribution Model Results.

Results from the attribution model can be interpreted onscreen or exported to a CSV file for expanded analysis.

6 References

Altomare, D. (2016). *Package 'ChannelAttribution'*. 1st ed. [ebook] pp.1-6. Available at: <https://cran.r-project.org/web/packages/ChannelAttribution/ChannelAttribution.pdf> [Accessed 1 Jul. 2016].

Google Developers. (2016). *Using OAuth 2.0 to Access Google APIs*. [online] Available at: <https://developers.google.com/identity/protocols/OAuth2> [Accessed 1 Jul. 2016].

"Mockaroo - Random Data Generator | CSV / JSON / SQL / Excel". *Mockaroo.com*. N.p., 2016. Web. 6 May 2016.

Pearmain, M. (2016). *Package 'RGoogleAnalytics'*. 1st ed. [ebook] pp.1-7. Available at: <https://cran.r-project.org/web/packages/RGoogleAnalytics/RGoogleAnalytics.pdf> [Accessed 4 Apr. 2016].

Support.google.com. (2016). *Custom dimensions & metrics - Analytics Help*. [online] Available at: <https://support.google.com/analytics/answer/2709828?hl=en> [Accessed 1 Jul. 2016].

Support.google.com. (2016). *Create, edit, and share goals - Analytics Help*. [online] Available at: <https://support.google.com/analytics/answer/1032415?hl=en> [Accessed 1 Jul. 2016].

7 Appendix

Solution Frame is available for download via [Google Drive](#)

```
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function Addins Project: (None)
Data x Google_Analytics_Final.r x Attribution_model_Final.r* x
Source on Save Run Source
1 #####
2 #REAL-TIME ELEMENT POLLING DATA#
3 #RgoogleAnalytics Package Version: 0.1.1 License Apache License 2.0
4 #Package Authors: Michael Pearmain. Contributions From Nick Michailowski, Vignesh Prajapati, Kushan Shah and Nicolas Remy 2014
5 #https://cran.r-project.org/web/packages/RgoogleAnalytics/index.html
6 #https://cran.r-project.org/web/packages/RgoogleAnalytics/RgoogleAnalytics.pdf
7
8 #R Script below has been adapted & is based upon the RgoogleAnalytics Reference manual.
9 #####
10 |
11 #####
12 # REQUIRED PACKAGES
13 #####
14 library(RgoogleAnalytics)
15
16 #Set the working directory
17 setwd("c:/Users/James/Google Drive/College Lecture Notes - Files/MSCDA/Course Lectures/Semester 4/Model/Final")
18
19 # Generate the oauth_token object
20 oauth_token <- Auth(client.id = "123456789-xxxxxxxxxxxxxxxx.apps.googleusercontent.com",
21                   client.secret = "Txxxxxxxxxxxxx_Tknui")
22 # Save the token object for future sessions
23 save(oauth_token, file="oauth_token")
24
25 # Load the token object
26 load("oauth_token")
27
28 # Create a list of Query Parameters
29 query.list <- Init(start.date = "2016-07-01",
30                  end.date = "2016-07-31",
31                  sort = "ga:date", #Order Returned results by date
32                  dimensions = "ga:date,ga:medium,ga:customVarName1,ga:customVarName2", #Set set dimensions
33                  metrics = "ga:sessions,ga:pageviews,ga:customVarValue1,ga:customVarValue2,ga:goal1Completions,ga:goal2Completions,ga:goal3Completions", #Set metrics
34                  max.results = 1000, #Get 1000 Rows
35                  table.id = "ga:xxxxxxxx") #Set Profile ID of Google Analytics view where query should be run against.
36 # Create the query object
37 ga.query <- queryBuilder(query.list)
38
39 # Fire the query to the Google Analytics API
40 ga.df <- GetReportData(query, oauth_token)
41
42 #Write Google Analytics Data to CSV
43 write.csv(ga.df, file = "Google_Analytics_Conversion_Path_Data.csv")
44
45
10:1 (Top Level) Console R Script
```