How to create ads from R?
user2015.math.aau.dk
Learn how to create Facebook ads from R at a contributed talk at the useR! 2015 conference
fbRads
Analyzing and managing Facebook ads from R

Gergely Daroczi, Ajay Gopal
CARD.com
07/1/2015

> sessionInfo()
[2] "Aalborg, Denmark"
Modern advertising

- Google knows what you are searching for
- Amazon knows what you are in the market for
- Facebook knows what you like
Modern advertising

- Google knows what you are searching for
- Amazon knows what you are in the market for
- Facebook knows what you like

- This info can be is used to advertise to you
Ad Platforms

- Utilize Google search data via AdWords API

## by Johannes Burkhardt

devtools::install.github('jburkhardt/RAdwords')

- Utilize Amazon purchase history via Amazon Ads

NULL

- Yahoo+Bing have joint search ad network & API

NULL

- Utilize Facebook likes & comments data via FB Marketing API

devtools::install.github('cardcorp/fbRads')
When to advertise on Facebook
When to advertise on Facebook
When to advertise on Facebook

Source: adparlor.com
CARD.com’s view of the world

Gergely Daróczi @daroczig · Apr 11

Just received my "I ❤ R" prepaid debit card from @CARD. Will be fun to use this #rstats designed card at #user2015 :)

![Image of a prepaid debit card]
Modern Marketing

RTB Ad Xchgs

1. CARD.COM AdTech Platform
   - Artwork & Brands
   - Bank Partner

2. CARD.COM Site/App
   - Deploy
   - Learn

3. CARD.COM Analytics Platform

Visitors

Members

Transactions
Anatomy of a Facebook Ad Campaign

Campaign

Ad set

Ad set includes scheduling, budget, bidding, targeting, and placement

Ad
Ad

Ad
Ad

Ad
Ad

Source: Facebook Marketing API docs
Anyone here from Facebook?
Facebook Ad Campaign Structure Changes

- Anyone here from Facebook?
- 2 changes in the campaign structure in 2014
- 2 -> 3 hierarchical categories
- Before July 2014, “Ad Sets” were called “Campaigns”
- At the API endpoints:
  - campaigns are called `adcampaign_groups`
  - ad sets are called `adcampaigns`
  - ads are called `adgroups`
- When creating an ad via the API, the adset id is called `campaign_id`
- 4 new Facebook Marketing API versions in October 2014
Anyone here from Facebook?

2 changes in the campaign structure in 2014
2 -> 3 hierarchical categories
Before July 2014, “Ad Sets” were called “Campaigns”
At the API endpoints:
- campaigns are called `adcampaign_groups`
- ad sets are called `adcampaigns`
- ads are called `adgroups`

When creating an ad via the API, the adset id is called `campaign_id`
4 new Facebook Marketing API versions in October 2014

But it’s pretty damn good
Facebook Ad Campaign Structure Changes

- Anyone here from Facebook?

- 2 changes in the campaign structure in 2014
  - 2 -> 3 hierarchical categories
  - Before July 2014, “Ad Sets” were called “Campaigns”

- At the API endpoints:
  - campaigns are called `adcampaign_groups`
  - ad sets are called `adcampaigns`
  - ads are called `adgroups`

- When creating an ad via the API, the adset id is called `campaign_id`

- 4 new Facebook Marketing API versions in October 2014

- But it’s pretty damn good

- Really!
Collect R package developer e-mail addresses

```r
> url <- 'http://cran.r-project.org/web/checks/check_summary.html'
> packages <- readHTMLTable(url, which = 2)
> mails <- sub('.*<(.*>)', '\1', packages$' Maintainer')
> mails <- sub(' at ', '@', mails)
```

```r
## Dirk Eddelbuettel (35)
## Kurt Hornik (29)
## Scott Chamberlain (24)
## Martin Maechler (24)
```

```r
## 4023
```

```r
tail(sort(table(sub('.*@', '', mails))))
```

```r
## gmail.com (1778)
## R-project.org (84)
```

Gergely Daroczi, Ajay Gopal (CARD.com)  
fbRads: Facebook ads from R
Collect R package developer e-mail addresses

```r
> url <- 'http://cran.r-project.org/web/checks/check_summary.html'
> packages <- readHTMLTable(url, which = 2)
> mails <- sub('.*<(.*)>', '\1', packages$ 'Maintainer')
> mails <- sub(' at ', '@', mails)

> tail(sort(table(mails)))
## Dirk Eddelbuettel (35)
## Kurt Hornik (29)
## Scott Chamberlain (24)
## Martin Maechler (24)
## Paul Gilbert (22)

> length(unique(mails))
## 4023

> tail(sort(table(sub('.*@', '', mails)))))
## gmail.com (1778)
## R-project.org (84)
## edu
```
Collect R package developer e-mail addresses

28% match: only 900 accounts for 6,000+ R packages
Collect e-mail addresses from the R-help mailing list

Get the location of the archives:

```r
> url <- 'https://stat.ethz.ch/pipermail/r-help/
```

We need RCurl for HTTPS:

```r
> library(RCurl)
```

Get URL of all archive files:

```r
> R.help.toc <- htmlParse(getURL(url))
> R.help.archives <- unlist(
+   xpathApply(R.help.toc, "//table//td[3]/a", xmlAttrs),
+   use.names = FALSE)
```

Download archive files:

```r
> dir.create('r-help')
> for (f in R.help.archives)
+   download.file(url = paste0(url, f),
+                  file.path('help-r', f), method = 'curl')
```
Extract e-mail addresses from the R-help mailing list

Regular expression matching date format in “From” lines:

```r
> dateregex <- paste('^[A-Za-z]{3} [A-Za-z]{3} [0-9]{1,2}',
+ '[0-9]{2}:[0-9]{2}:[0-9]{2} [0-9]{4}','$')
```

grep for lines matching the From field:

```r
> mails <- system(paste0(
+ "zgrep -E '^[\w\s]*From .* at .* ",
+ dateregex,
+ '.*\./help-r/*.txt.gz"),
+ intern = TRUE)
```

Extract e-mail addresses from these lines:

```r
> mails <- sub('.*From ', '', mails)
> mails <- sub(paste0('[ ]*', dateregex, '$'), '', mails)
> mails <- sub(' at ', '@', mails)
```
> length(mails)
266449

> head(sort(table(mails), decreasing = TRUE))
  ripley@stats.ox.ac.uk    dwinsemius@comcast.net
                 8611             7064
  ggrothendieck@gmail.com p.dalgaard@biostat.ku.dk
                 5386             3243
  jholtman@gmail.com    smartpink111@yahoo.com
                 3193             2999

> length(unique(mails))
29266

> 29266 > 4023
TRUE  \o/
Authenticate with the Facebook API

https://developers.facebook.com/apps/
Authenticate with the Facebook API

Create a token:

```r
> library(httr)
> app <- oauth_app('facebook', 'your_app_id', 'your_app_secret')
> tkn <- oauth2.0_token(
+   oauth_endpoints('facebook'), app, scope = 'ads_management',
+   type = 'application/x-www-form-urlencoded')
> tkn <- tkn$credentials$access_token
```

Save this secret token (never commit to git repository) and load it in any later session:

```r
> saveRDS(tkn, 'token.rds')
> tkn <- readRDS('token.rds')
```

Initialize connection to Facebook Marketing API:

```r
> fbacc <- fbad_init(fid, tkn)
```
Create custom audience

```r
> aud_id <- fbad_create_audience(fbacc, 'R-help posters', +  'Unique e-mail addresses in R-help 1997-2015')
```

Reading audience info:

```r
> fbad_read_audience(fbacc, aud_id, 'approximate_count')
20
```

Adding e-mails to audience:

```r
> fbad_add_audience(fbacc, aud_id, 'EMAIL', mails)
```

It usually takes a day or two for custom audiences to fully populate :(

```r
> fbad_read_audience(fbacc, aud_id, 'approximate_count')
8700
```
Create lookalike audiences

Load the number of attendees per country (see my poster tomorrow!):

```r
> url <- 'http://rapporter.net/custom/R-activity/data/RStats_2015.csv'
> library(data.table)
> RpC <- fread(url)
> user2015 <- RpC[user_2015 > 0, ]
```

Create a lookalike audience for each country:

```r
> aud_ids <- sapply(1:nrow(user2015), function(i) {
+   try(fbad_create_lookalike_audience(
+     fbacc,
+     name = paste('R-help posters in', user2015[i, NAME]),
+     origin_audience_id = aud_id,
+     ratio = 0.01,
+     country = toupper(user2015[i, ISO2C])))
+   Sys.sleep(20)
+ })
```
Read lookalike audiences

Get the approximate count of each lookalike audience:

```r
> lookalikes[!is.na(audience),
+   size := fbad_read_audience(fbacc, audience, 'approximate_count')[[1]]
+   by = country]
```

```r
> lookalikes[!is.na(audience), c('country', 'size'), with = FALSE]
```

<table>
<thead>
<tr>
<th>country</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>173000</td>
</tr>
<tr>
<td>Austria</td>
<td>41500</td>
</tr>
<tr>
<td>Belgium</td>
<td>72400</td>
</tr>
<tr>
<td>Brazil</td>
<td>1280400</td>
</tr>
<tr>
<td>Canada</td>
<td>253100</td>
</tr>
<tr>
<td>Colombia</td>
<td>308200</td>
</tr>
<tr>
<td>Faroe Islands</td>
<td>400</td>
</tr>
<tr>
<td>France</td>
<td>392900</td>
</tr>
<tr>
<td>Germany</td>
<td>347700</td>
</tr>
<tr>
<td>Greece</td>
<td>59900</td>
</tr>
<tr>
<td>Hungary</td>
<td>61500</td>
</tr>
<tr>
<td>India</td>
<td>2042000</td>
</tr>
<tr>
<td>Ireland</td>
<td>32800</td>
</tr>
<tr>
<td>Italy</td>
<td>336200</td>
</tr>
<tr>
<td>Latvia</td>
<td>7800</td>
</tr>
<tr>
<td>Mexico</td>
<td>758100</td>
</tr>
<tr>
<td>Netherlands</td>
<td>110900</td>
</tr>
<tr>
<td>New Zealand</td>
<td>34500</td>
</tr>
<tr>
<td>Norway</td>
<td>36500</td>
</tr>
<tr>
<td>Singapore</td>
<td>257000</td>
</tr>
<tr>
<td>Slovenia</td>
<td>11200</td>
</tr>
<tr>
<td>Spain</td>
<td>284200</td>
</tr>
<tr>
<td>Switzerland</td>
<td>43100</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>478700</td>
</tr>
<tr>
<td>United States</td>
<td>2483200</td>
</tr>
</tbody>
</table>
```

Gergely Daroczi, Ajay Gopal (CARD.com)

fbRads: Facebook ads from R

useR! 2015, Aalborg
Create a campaign

```r
> campaign <- fbad_create_campaign(fbacc,
+       name = 'Promote my useR! 2015 talk'
>
> fbad_read_campaign(fbacc, campaign)

$id
1 "*******"

$account_id
1 "*******"

$buying_type
1 "AUCTION"

$campaign_group_status
1 "ACTIVE"

$objective
1 "NONE"

$name
1 "Promote my useR! 2015 talk"
```
Define target for an adset

All valid lookalike audiences:

```r
> target <- lookalikes[!is.na(audience)]
> setnames(target, c('name', 'id'))
```

The original R-help posters list:

```r
> target <- rbind(target, list('R-help poster list', id1))
```

The original R package developers list:

```r
> target <- rbind(target, list('R pkg developers list', id2))
```

Prepare JSON list:

```r
> target <- list(custom_audiences = target)
```
Create an adset

> adset <- fbad_create_adset(
+   fbacc,
+   name = 'My budget for promoting my useR! 2015 talk',
+   campaign_group_id = campaign,
+   bid_type = 'CPC',
+   bid_info = list(CLICKS = 42),
+   campaign_status = 'ACTIVE',
+   lifetime_budget = 4200,
+   end_time = as.numeric(as.POSIXct('2015-07-01')),
+   targeting = target)
Upload image

Get an image for the ad:

```r
> img <- 'user_2015_logo.png'
```

Upload to Facebook:

```r
> img <- fbad_create_image(fbacc, img)
```

Take a note on the returned hash:

```r
> str(img)
List of 3
$ filename: chr  "user_2015_logo.png"
$ hash : chr  "140423d688a2e71a6b0cbfd4d65526aa"
$ url : chr  "https://fbcdn-creative-a.akamaihd.net/hads-ak-xft1/t45.1600-4/11410296_6024889747763_1444747968_n.png"
> img <- img$hash
```
Create a creative

```r
url <- 'http://user2015.math.aau.dk/contributed_talks#210'
creative <- fbad_create_creative(
+   fbacc,
+   name = 'How to create ads from R?',
+   body = paste(
+     'Learn how to create Facebook ads from R',
+     'at a contributed talk',
+     'at the useR! 2015 conference'),
+   title = 'How to create ads from R?',
+   object_url = url,
+   image_hash = img$hash)
```

Gergely Daroczi, Ajay Gopal (CARD.com) fbRads: Facebook ads from R useR! 2015, Aalborg 26 / 34
Create a creative

How to create ads from R?
user2015.math.aau.dk
Learn how to create Facebook ads from R
at a contributed talk at the useR! 2015
conference
Create an ad

```r
> ad <- fbad_create_ad(
+   fbacc,
+   name = 'An ad -- right from the R console',
+   campaign_id = adset,
+   creative = creative)
```
> images <- c('http://www.r-project.org/Rlogo.png',
  +   'http://user2014.stat.ucla.edu/images/useR-middle.png')

> for (l in 1:length(images)) {
+   ## download image
+   img <- tempfile(fileext = '.png'); download.file(images[i], img)
+   ## resize, then upload image
+   system(paste('convert', img, '-resize 350x150^', img))
+   img <- fbad_create_image(fbacc, img)
+   ## create creative
+   url <- 'http://user2015.math.aau.dk/contributed_talks#210'
+   creative <- fbad_create_creative(
+       fbacc,
+       name = 'How to create ads from R',
+       body = paste(
+           'Learn how to create Facebook ads from R',
+           'at a contributed talk at the useR! 2015 conference'),
+       title = 'How to create ads from R',
+       object_url = url,
+       image_hash = img$hash)
+   ## create ad
+   ad <- fbad_create_ad(
+       fbacc,
+       name = paste0('An ad -- right from the R console (', toupper(letters[i]), ')'),
+       campaign_id = adset,
+       creative = creative)
+ }

Gergely Daroczi, Ajay Gopal (CARD.com) fbrads: Facebook ads from R useR! 2015, Aalborg 29 / 34
Performance metrics

No functions yet, DIY! But it’s pretty easy:

```r
> res <- fbRads:::fbad_request(
+   path = paste0(fbacc$acct_path, 'adgroupstats'),
+   params = list(access_token = fbacc$access_token))

> jsonlite::fromJSON(res)$data[, c(
+   'impressions',
+   'unique_impressions',
+   'clicks',
+   'unique_clicks',
+   'spent')]

  impressions unique_impressions clicks unique_clicks spent
1      31655          16332    24        23    328
2     134902          55421   169       153   2361
3      90769          40560   107        98   1477
```

Gergely Daroczi, Ajay Gopal (CARD.com)
Performance metrics

```r
> power.prop.test(p1 = 169 / 135000, p2 = 24 / 31000, power = 0.5, sig.level = 0.05)
```

Two-sample comparison of proportions power calculation

```
n = 34077.72
p1 = 0.001251852
p2 = 0.0007741935
sig.level = 0.05
power = 0.5
alternative = two.sided
```

NOTE: n is number in *each* group

```r
> fisher.test(data.frame(B = c(169, 135000), A = c(24, 31000)), conf.int = FALSE)
```

Fisher's Exact Test for Count Data

```
data:  data.frame(B = c(169, 135000), A = c(24, 31000))
p-value = 0.02617
alternative hypothesis: true odds ratio is not equal to 1
sample estimates:
  odds ratio
  1.61696
```
Performance metrics

Jennifer Bryan
@JennyBryan

The universe has confirmed my \#rstats bona fides. Just got hit with @daroczig’s Facebook ad.

Copenhagen, Capital Region of Denmark

Suggested Post

CARD.com
Sponsored ·

Use R to create and manage Facebook ads like this one. Attend the demo of the fbRads package at useR! 2015.
Advertising success with

https://github.com/cardcorp/fbRads