



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

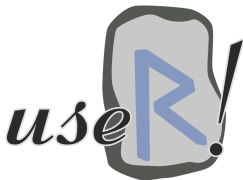
Gergely Daroczi, Ajay Gopal



CARD.COM

CARD.com

07/1/2015



```
> sessionInfo()  
[1] "June 30 - July 3, 2015"  
[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

- 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

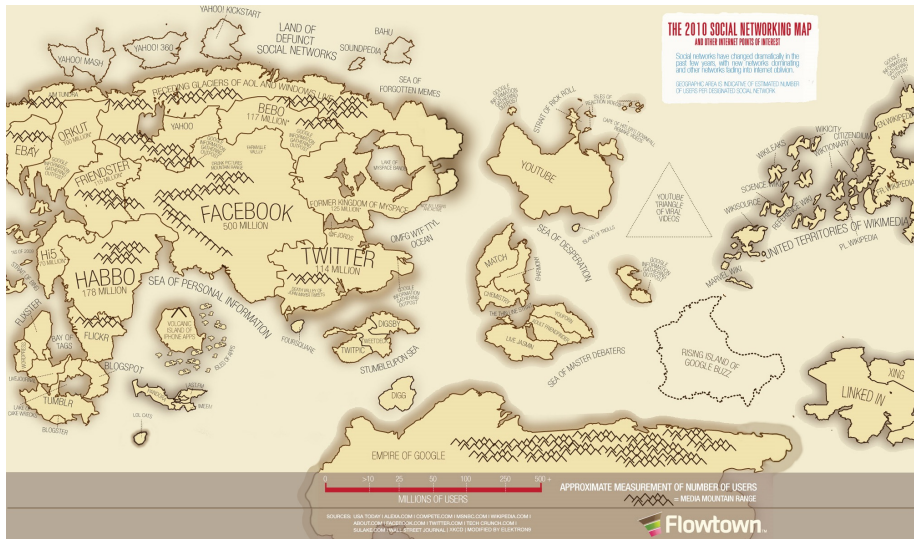
- Yahoo+Bing have joint search ad network & API

NULL

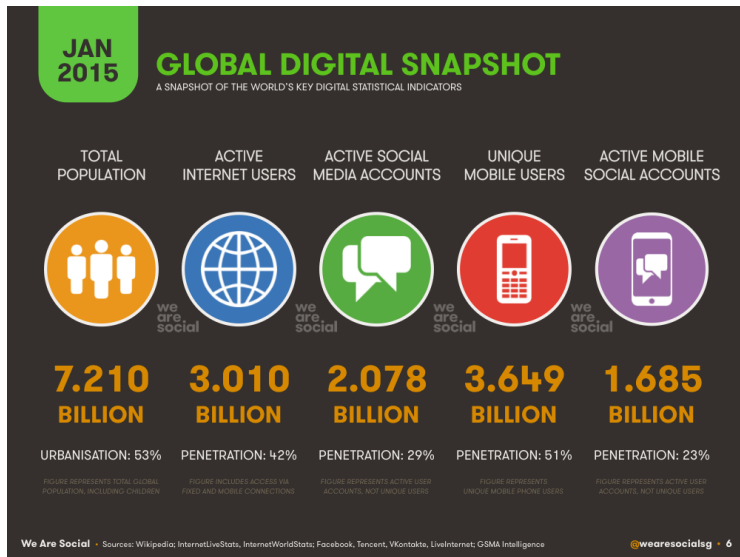
- Utilize Facebook likes & comments data via FB Marketing API

```
devtools::install.github('cardcorp/fbRads')
```

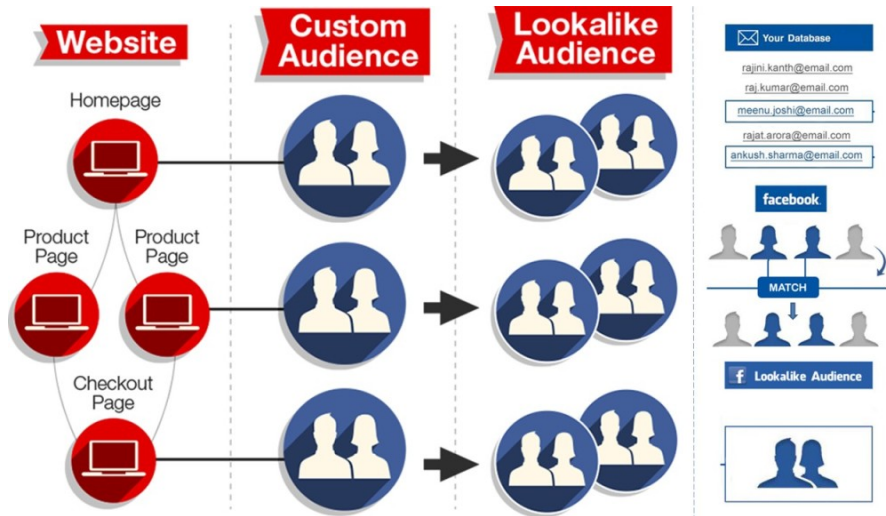
When to advertise on Facebook



When to advertise on Facebook



When to advertise on Facebook



Source: adparlor.com

CARD.com's view of the world



Gergely Daróczy @daroczig · Apr 11

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

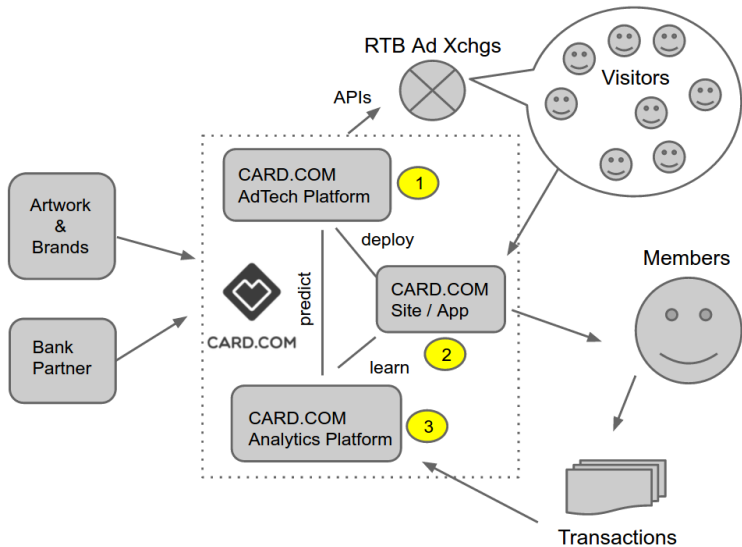


RETWEETS
10

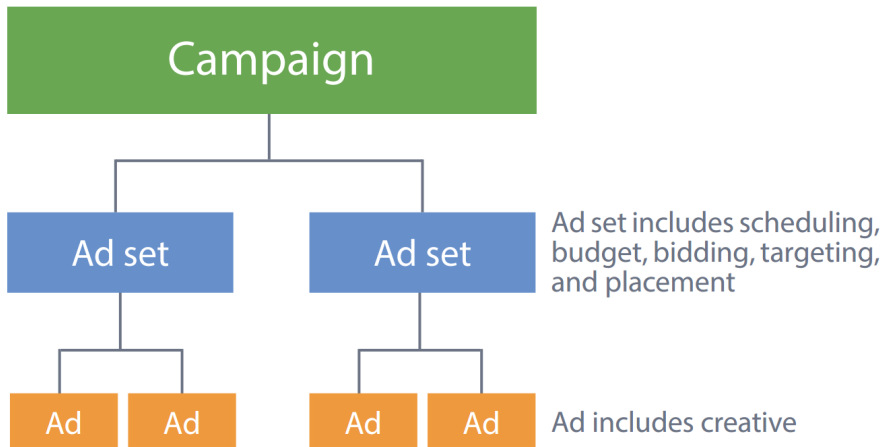
FAVORITES
16



Modern Marketing



Anatomy of a Facebook Ad Campaign



Source: Facebook Marketing API docs

Facebook Ad Campaign Structure Changes

- 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

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

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

```
> 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)
```


Collect R package developer e-mail addresses

```
> 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

The screenshot shows the Facebook Adverts Manager interface. At the top, there is a search bar with 'Find friends' and a user profile for 'Gergely'. Below this, the page is titled 'Adverts Manager > Audiences' with a 'Create Audience' button. The 'Advert Account' is listed as '(GBP)'. The main heading is 'Audiences'. Below the heading are buttons for 'Filters', 'Customise Columns', 'Create Advert', and 'Actions', along with a search box 'Search by name'. A table lists the audience 'R package maintainers' with a size of 900 and a 'Low match rate' status. A tooltip message states: 'We could only match 28% of your entries to Facebook profiles in the last update. Please check your source and try uploading it again.' with a link to 'Upload Data File'.

	Name	Type	Size	Availability
<input type="checkbox"/>	R package maintainers	Custom Audience Customer List	900	Low match rate Get Help

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:

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

We need RCurl for HTTPS:

```
> library(RCurl)
```

Get URL of all archive files:

```
> 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:

```
> 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:

```
> 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:

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

Extract e-mail addresses from these lines:

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

Extracted e-mail addresses from the R-help mailing list

```
> 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

The screenshot shows the Facebook Developers website interface. At the top, there is a navigation bar with links for 'Developers', 'My Apps', 'Products', 'Docs', 'Tools & Support', and 'News'. A search bar is located on the right side of the navigation bar. The main content area features a 'Quick Start for Website' section with a 'Start Over' button on the left and a 'Skip and Create App ID' button on the right. A central 'WWW' icon is visible. A modal dialog box titled 'Create a New App ID' is overlaid on the page. The dialog contains the following elements: a question 'Create **Rads demo app App?', a radio button labeled 'NO' followed by the text 'Is this a test version of another app? [Learn More.](#)', a 'Category' dropdown menu currently set to 'Business', and a footer with the text 'By proceeding, you agree to the Facebook Platform Policies', a 'Cancel' button, and a 'Create App ID' button.

<https://developers.facebook.com/apps/>

Authenticate with the Facebook API

Create a token:

```
> 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:

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

Initialize connection to Facebook Marketing API:

```
> fbacc <- fbad_init(fid, tkn)
```

Create custom audience

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

Reading audience info:

```
> fbacc_read_audience(fbacc, aud_id, 'approximate_count')  
20
```

Adding e-mails to audience:

```
> fbacc_add_audience(fbacc, aud_id, 'EMAIL', mails)
```

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

```
> fbacc_read_audience(fbacc, aud_id, 'approximate_count')  
8700
```


Create lookalike audiences

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

```
> 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:

```
> 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:

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

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

	country	size		country	size
1:	Australia	173000	13:	Ireland	32800
2:	Austria	41500	14:	Italy	336200
3:	Belgium	72400	15:	Latvia	7800
4:	Brazil	1280400	16:	Mexico	758100
5:	Canada	253100	17:	Netherlands	110900
6:	Colombia	308200	18:	New Zealand	34500
7:	Faroe Islands	400	19:	Norway	36500
8:	France	392900	20:	Singapore	257000
9:	Germany	347700	21:	Slovenia	11200
10:	Greece	59900	22:	Spain	284200
11:	Hungary	61500	23:	Switzerland	43100
12:	India	2042000	24:	United Kingdom	478700
			25:	United States	2483200

Create a campaign

```
> 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:

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

The original R-help posters list:

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

The original R package developers list:

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

Prepare JSON list:

```
> target <- list(custom_audiences = target)
```

Create an adset

```
> adset <- fb_ad_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:

```
> img <- 'user_2015_logo.png'  
> download.file('http://user2015.math.aau.dk/gfx/useR2015.png', img)
```

Upload to Facebook:

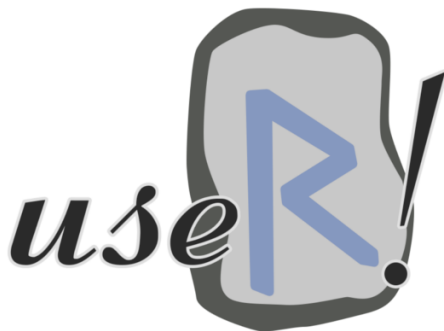
```
> img <- fbacc_create_image(fbacc, img)
```

Take a note on the returned hash:

```
> 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/t45.16  
> img <- img$hash
```

Create a creative

```
> url <- 'http://user2015.math.aau.dk/contributed_talks#210'
> creative <- fb_ad_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)
```



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

```
> ad <- fbacc_create_ad(  
+   fbacc,  
+   name          = 'An ad -- right from the R console',  
+   campaign_id  = adset,  
+   creative     = creative)
```

A/B testing

```
> 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[l], 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[l]), ')'),
+     campaign_id = adset,
+     creative   = creative)
+ }
}
```

Performance metrics

Facebook Ads Manager interface showing performance metrics for a campaign.

Find friends | Gergely Home Find Friends | **Adverts Manager** | Gergely Daróczy

Home | Campaign | Advert Set | **Promote my useR! 2015 talk** | **My budget for promoting my useR! 2015 talk** | How does this page work? [Create Advert in Advert Set](#)

Account
Gergely Daróczy

- Create an advert
- Campaigns**
- Pages
- Reports
- Audience Insights
- Settings
- Billing
- Conversion Tracking
- Power Editor
- Account History
- Audiences

Help Centre
Advertiser Support

Search your adverts

STATUS **DELIVERY** ● Active **SPENT TODAY** **£2.51** of **£2.15** **LIFETIME SPENT** **£41.66** of **£42.00** **END DATE** **Tomorrow** (8 hours left)

RESULTS ? **REACH ?** **75,714** **FREQUENCY ?** **3.40** **TOTAL SPENT ?** **£41.66** **COST PER RESULT ?** **—** **1 June 2015 - 30 June 2015**

All Except Deleted | [Edit Adverts](#) | [View Report](#) | [View History](#) | 3 Results

Status ?	Advert ?	Delivery ?	Results ?	Cost ?	Reach ?	Frequency ?	Clicks ?	Click-Through Rate ?	Relevance Score ?	Spent Today	Total Spent ?	Max Bid ?	Avg. Price ?
<input checked="" type="checkbox"/>	An ad – right from the R console (B)	● Active	Not Available	—	55,421	2.43	169	0.125%	7/10	£0.89	£23.61	£0.42 CPC	£0.14 CPC
<input checked="" type="checkbox"/>	An ad – right from the R console (A)	● Active	Not Available	—	16,332	1.94	24	0.076%	7/10	£0.00	£3.28	£0.42 CPC	£0.14 CPC
<input checked="" type="checkbox"/>	An ad – right from the R console	● Active	Not Available	—	40,560	2.24	107	0.118%	8/10	£1.62	£14.77	£0.42 CPC	£0.14 CPC

3 Results

[About](#) [Create Advert](#) [Create Page](#) [Developers](#) [Careers](#) [Privacy](#) [Cookies](#) [AdChoices](#) [Terms](#) [Help](#)

Facebook © 2015
English (UK)

Performance metrics

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

```
> 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
```

Performance metrics

```
> 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

```
> 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
```



Jennifer Bryan

@JennyBryan



Following

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 ·

Like Page

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

<https://github.com/cardcorp/fbRads>

