Measuring Facebook Advertising ROI

An introduction to the Facebook advertising platform

Along with Google's search and display networks, Facebook is one of the big players when it comes to online advertising. As Facebook users interact with the platform, adding demographic information, liking particular pages and commenting on specific posts, Facebook builds a profile of that user based on who they are and what they're interested in.

This fact makes Facebook very attractive for advertisers. Advertisers can create Facebook adverts, then create an 'Audience' for that advert or group of adverts. Audiences can be built from a range of attributes including gender, age, location and interests. This specific targetting means advertisers can tailor content appropriately for a specific audience, even if the product being marketed is the same.

For example, let's imagine a company wants to advertise its new car. They may wish to promote one set of features, performance and the 2 kW stereo, to women in their early twenties. They might decide that they want to talk about it's fuel efficiency and reduced emmissions to men in their thirties, and they might want to push the spacious interior and safety rating to men and women in their thirties and early forties who are interested in *Families* magazine and who like pages of nappy and baby clothes manufacturers.

In 2016, Facebook's revenue from advertising was $26bn, up from $17bn the year before. This compares to Google's $79bn the $638m that twitter advertising made in Q4 2016 and $173m that linkedIn made from ads in Q3 2016. These figures illustrate just how big an advertising platform is, although it faces challenges for the future with a decline in younger users in 2017 (https://www.recode.net/2018/2/12/16998750/facebooks-teen-users-decline-instagram-snap-emarketer), with generation Z moving to Snapchat and Instagram (http://uk.businessinsider.com/generation-z-facebook-2018-1).

What do we need from our Facebook ads analysis?

When it comes to analysing the Facebook adverts dataset, there are a lot of questions we can ask, and a lot of insight we can generate. However, from a business perspective we want to ask questions that will give us answers we can use to improve business performance.

Without knowing anything of the company's marketing strategy or campaign objectives, we do not know which key performance indicators (KPIs) are the most important. For example, a new company may be focussed on brand awareness and may want to maximise the amount of impressions, being less concerned about how well these adverts perform in terms of generating clicks and revenue. Another company may simply want to maximise the amount of revenue, while minimising the amount it spends on advertising.

As these two objectives are very different, it is important to work with the client to understand exactly what they are hoping to achieve from their

marketing campaigns before beginning any analysis in order to ensure that our conclusions are ***relevant*** and, in particular, ***actionable***. There's not much point in producing a report full of insight, if there's nothing the client can do about it.

Let's start by getting the data imported, taking a look, and we'll work through some analyses that should be relevant for a range of objectives.

In [1]:
```
# load our go-to packages
library(tidyverse)
```

```
── Attaching packages ─────────────────────────────────── tidyverse 1.2.1 ──
✓ ggplot2 2.2.1.9000     ✓ purrr   0.2.4
✓ tibble  1.4.2          ✓ dplyr   0.7.4
✓ tidyr   0.8.0          ✓ stringr 1.2.0
✓ readr   1.2.0          ✓ forcats 0.2.0
── Conflicts ────────────────────────────────────── tidyverse_conflicts() ──
✕ dplyr::filter() masks stats::filter()
✕ dplyr::lag()    masks stats::lag()
```

In [2]:
```
#import data
data <- read_csv("../input/KAG_conversion_data.csv")
```

Output

In [3]:
```
# take a quick look at the data
glimpse(data)
```

```
Observations: 1,143
Variables: 11
$ ad_id               <dbl> 708746, 708749, 708771, 708815, 708818, 708820,...
$ xyz_campaign_id     <dbl> 916, 916, 916, 916, 916, 916, 916, 916, 916, 91...
$ fb_campaign_id      <dbl> 103916, 103917, 103920, 103928, 103928, 103929,...
$ age                 <chr> "30-34", "30-34", "30-34", "30-34", "30-34", "3...
$ gender              <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "M...
$ interest            <dbl> 15, 16, 20, 28, 28, 29, 15, 16, 27, 28, 31, 7, ...
$ Impressions         <dbl> 7350, 17861, 693, 4259, 4133, 1915, 15615, 1095...
$ Clicks              <dbl> 1, 2, 0, 1, 1, 0, 3, 1, 1, 3, 0, 0, 0, 0, 7, 0,...
$ Spent               <dbl> 1.43, 1.82, 0.00, 1.25, 1.29, 0.00, 4.77, 1.27
```

```
$ Spent               <dbl> 1.43, 1.82, 0.00, 1.25, 1.29, 0.00, 4.77, 1.27,...
$ Total_Conversion    <dbl> 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
$ Approved_Conversion <dbl> 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,...
```

The documenation describes the columns in the data as follows:

1.) ad_id: unique ID for each ad.

2.) xyz_campaign_id: an ID associated with each ad campaign of XYZ company.

3.) fb_campaign_id: an ID associated with how Facebook tracks each campaign.

4.) age: age of the person to whom the ad is shown.

5.) gender: gender of the person to whom the add is shown

6.) interest: a code specifying the category to which the person’s interest belongs (interests are as mentioned in the person’s Facebook public profile).

7.) Impressions: the number of times the ad was shown.

8.) Clicks: number of clicks on for that ad.

9.) Spent: Amount paid by company xyz to Facebook, to show that ad.

10.) Total conversion: Total number of people who enquired about the product after seeing the ad.

11.) Approved conversion: Total number of people who bought the product after seeing the ad.

We can see that most of the variables are numerical, but two are character. That may not be a problem, but, in this case, it's quite easy to turn those into numerical data, so let's go ahead and do that, as that will allow us to perform certain functions later.

In [4]:
```
# look for unique values in 'age' column
unique(data$age)
```

'30-34' '35-39' '40-44' '45-49'

In [5]:
```
# create copy of data for editing
```

```
dataTf <- data
```

In [6]:
```
# replace character string age ranges with number
dataTf$age[dataTf$age == '30-34'] <- 32
dataTf$age[dataTf$age == '35-39'] <- 37
dataTf$age[dataTf$age == '40-44'] <- 42
dataTf$age[dataTf$age == '45-49'] <- 47

# convert variable to integer
dataTf$age <- as.integer(dataTf$age)
```

In [7]:
```
# let's just check that age variable now
unique(dataTf$age)
str(dataTf$age)
```

```
  32  37  42  47
```

```
 int [1:1143] 32 32 32 32 32 32 32 32 32 32 ...
```

All looks good, let's go ahead and do the same with our gender variable.

In [8]:
```
# convert gender variable to integer
dataTf$gender[dataTf$gender == 'M'] <- 0
dataTf$gender[dataTf$gender == 'F'] <- 1
dataTf$gender <- as.integer(dataTf$gender)
```

In [9]:
```
# abbreviate some variable names
dataTf <- dataTf %>%
  rename(xyzCampId = xyz_campaign_id, fbCampId = fb_campaign_id, impr = Impressions,
         conv = Total_Conversion, appConv = Approved_Conversion)
```

In [10]:
```
glimpse(dataTf)
```

```
Observations: 1,143
Variables: 11
$ ad_id     <dbl> 708746, 708749, 708771, 708815, 708818, 708820, 708889, 7...
$ xyzCampId <dbl> 916, 916, 916, 916, 916, 916, 916, 916, 916, 916, 916, 91...
$ fbCampId  <dbl> 103916, 103917, 103920, 103928, 103928, 103929, 103940, 1...
$ age       <int> 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 3...
$ gender    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ interest  <dbl> 15, 16, 20, 28, 28, 29, 15, 16, 27, 28, 31, 7, 16, 16, 20...
$ impr      <dbl> 7350, 17861, 693, 4259, 4133, 1915, 15615, 10951, 2355, 9...
$ Clicks    <dbl> 1, 2, 0, 1, 1, 0, 3, 1, 1, 3, 0, 0, 0, 0, 7, 0, 1, 0, 1, ...
$ Spent     <dbl> 1.43, 1.82, 0.00, 1.25, 1.29, 0.00, 4.77, 1.27, 1.50, 3.1...
$ conv      <dbl> 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
$ appConv   <dbl> 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, ...
```

Okay, as you can see from the output above, we've now got all our columns as numerical variables. Let's start our analysis with some unsupervised learning.

I always like to start with a heatmap with hierarchical clustering to see what sort of relationships pop out. I guess it goes back to using heatmaps as a first step in analysing genomic microarray data. As we've converted our data to numeric, we could convert it into a matrix and use the `heatmap` function. However, the *heatmaply library allows us to make our heatmaps a bit more interactive...

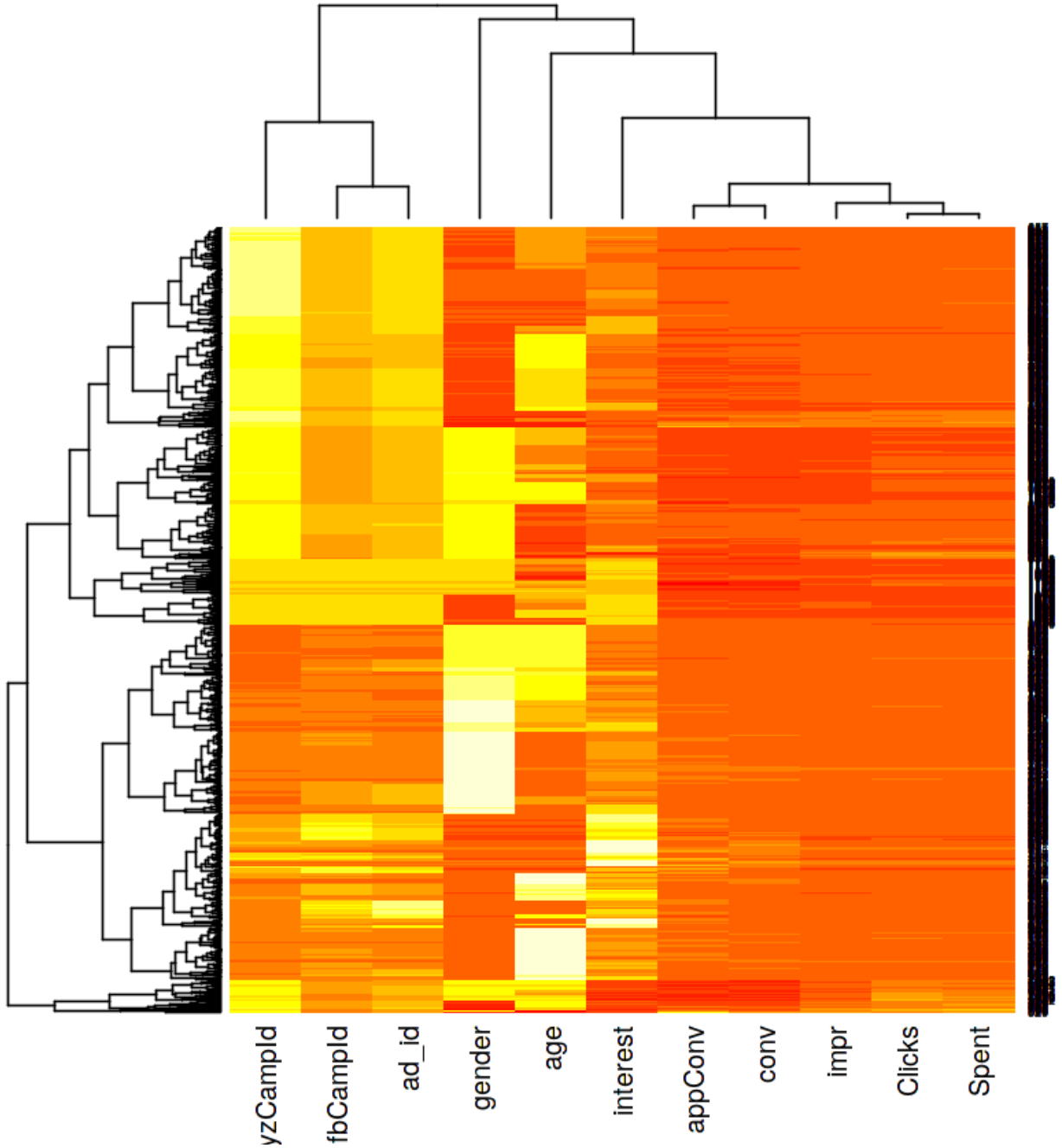
Unfortunately, when I commit and run the notebook, my heatmaply heatmaps don't get rendered, so let's stick with `heatmap` for now, although we will use the heatmaply library for its `normalize` function.

The purpose of this heatmap is just to get a quick overview of the data, so the pre-processing that we have done has been less comprehensive than if we were using it to get down to the serious nuts and bolts of the dataset. As such, some elements are really represented appropriately, but we should be able to get a feel for what's happening.

In [11]:
```
library(heatmaply)
dataMatNorm <- as.matrix(normalize(dataTf, method = "standardize"))
```

Output

In [12]:
```
heatmap(dataMatNorm)
```



Looking at the hierarchically clustered heatmap above, we can see a lot of what we would expect. All our main metrics fall into one major cluster. Our Approved Conversions and Total Conversions cluster together, and what we spent clusters with impressions and clicks, so our first overview of our dataset suggests that it makes sense.

Obviously, we've scaled and normalised features like our Ids, which isn't really appopriate, so we won't try and infer anything here and we'll look at those using other methods later.

Creating additional features

While we have the main 'building blocks' of our KPIs in the original dataset, there are some standard metrics missing, so let's take the opportunity to add them here.

1. **Click-through-rate (CTR)**. This is the percentage of how many of our impressions became clicks. A high CTR is often seen as a sign of good creative being presented to a relevant audience. A low click through rate is suggestive of less-than-engaging adverts (design and / or messaging) and / or presentation of adverts to an inappropriate audience. What is seen as a good CTR will depend on the type of advert (website banner, Google Shopping ad, search network test ad etc.) and can vary across sectors, but 2% would be a reasonable benchmark.
2. **Conversion Rate (CR)**. This is the percentage of clicks that result in a 'conversion'. What a conversion is will be determined by the objectives of the campaign. It could be a sale, someone completing a contact form on a landing page, downloading an e-book, watching a video, or simply spending more than a particular amount of time or viewing over a target number of pages on a website.
3. **Cost Per Click (CPC)**. Self-explanatory this one: how much (on average) did each click cost. While it can often be seen as desirable to reduce the cost per click, the CPC needs to be considered along with other variables. For example, a campaign with an average CPC of £0.5 and a CR of 5% is likely achieving more with its budget than one with a CPC of £0.2 and a CR of 1% (assuming the conversion value is the same.
4. **Cost Per Conversion**. Another simple metric, this figure is often more relevant than the CPC, as it combines the CPC and CR metrics, giving us an easy way to quickly get a feel for campaign effectiveness.

There are other values that are also very useful in assessing the performance of a marketing campaign. One of these is the **conversion value**: how much each conversion is worth. For example, our conversion could be a signup form on a landing page to receive information about a new car. If we know that, on average, 1% of people end up purchasing a car for £10,000, we can use that figure in calculating what our target cost per conversion should be.

For an e-commerce site, we could implement conversion tracking to tie-up the value of specific transactions to particular campaigns, this would allow us to assign the actual amount of revenue generated by each campaign / ad creative.

Knowing the conversion value would allow us to calculate other KPIs such as the **Return on Advertising Spend (ROAS)**. While advertising campaigns have other benefits (such as increased brand awareness and future purchases based on customer lifetime value) that may factor into the over return on investment (ROI), ROAS can quickly tell us how a campaign is paying for itself. It is simply the revenue as a percentage of the advertising spend. If a campaign costs £100 and leads to £400 sales, the ROAS is 400% (or 4).

The importance of understanding the client

Of course, using ROAS requires an understanding of the client's business. For some clients, a ROAS of 400% might be a great number, for others, they might not be covering their costs. This is why it is important to understand the margins of products being sold through these campaigns.

If an advertiser is selling a product for £120 (£100 in the UK after taking off the sales tax) that costs them £40, they are making £60 gross profit and a margin of 60%. If their ROAS is 400% (if calculated using the inc. tax figure), the advertising costs associated with that sale are £30, so there is a net profit of £30.

If, on the other hand, the product cost £80 (20% margin), the gross profit is only £20, therefore there is a net loss of £10 (before other business overheads are considered).

These simple examples show why it is important to understand, not only the strategic objectives of the marketing activities, but also how specific campaigns support these objectives and how their effectiveness is to be measured and, in the case of retail, what type of margins the client is working with across its product mix.

Create the additional KPIs with dplyr

With the variables we have in the dataset, we can easily create the CTR and CPC figures using the `mutate` function from the dplyr package:

In [13]:
```
dataTf <- dataTf %>%
  mutate(CTR = ((Clicks / impr) * 100), CPC = Spent / Clicks)
```

In [14]:
```
dataTf$CTR <- round(dataTf$CTR, 4)
dataTf$CPC <- round(dataTf$CPC, 2)
```

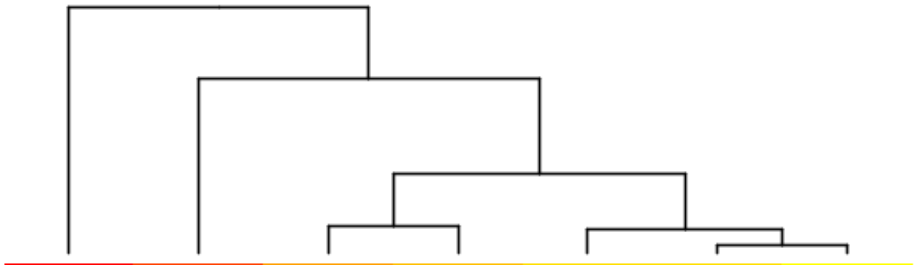
In [15]:
```
glimpse(dataTf)
```

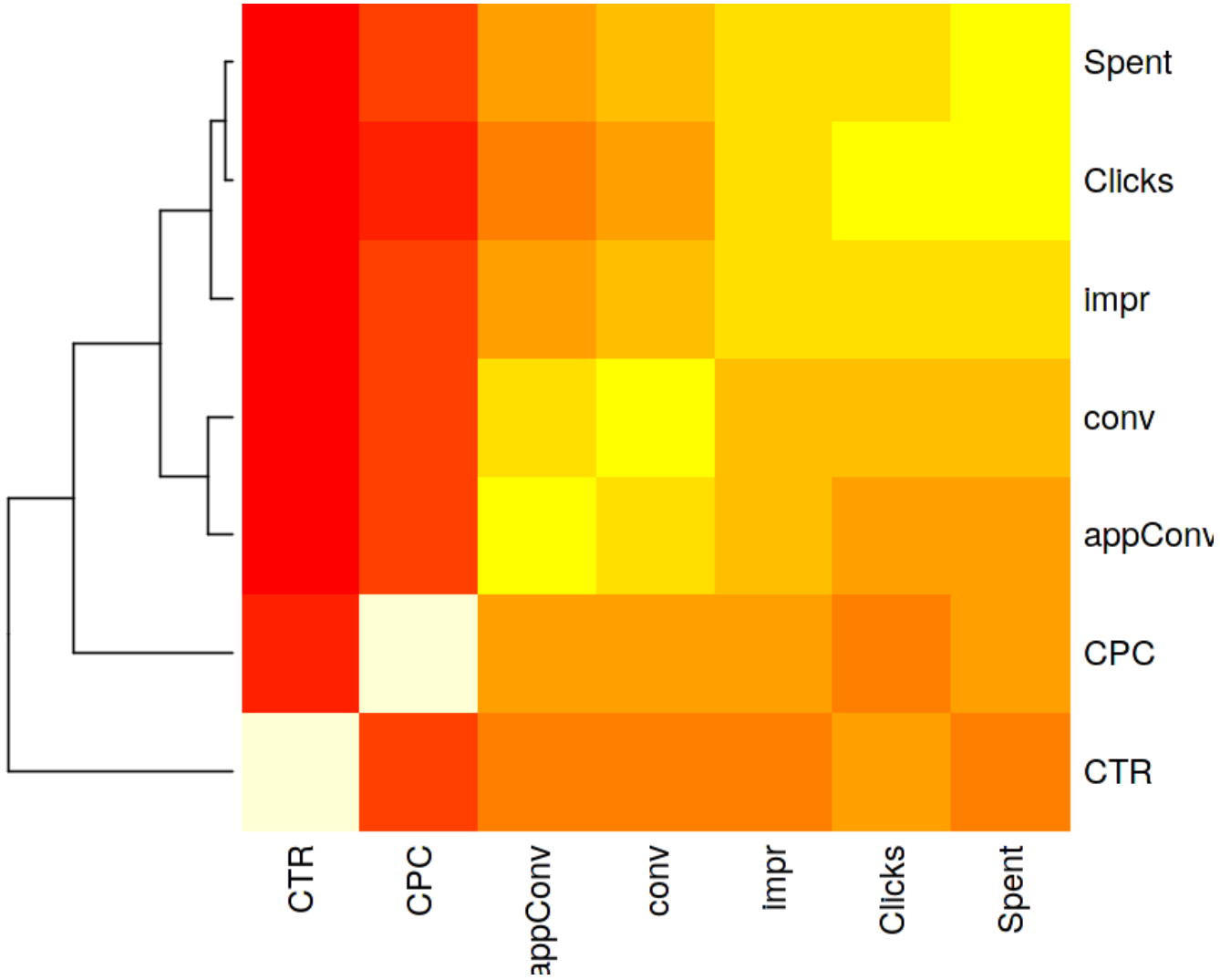
```
Observations: 1,143
Variables: 13
$ ad_id     <dbl> 708746, 708749, 708771, 708815, 708818, 708820, 708889, 7...
$ xyzCampId <dbl> 916, 916, 916, 916, 916, 916, 916, 916, 916, 916, 916, 91...
$ fbCampId  <dbl> 103916, 103917, 103920, 103928, 103928, 103929, 103940, 1...
$ age       <int> 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 3...
$ gender    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ interest  <dbl> 15, 16, 20, 28, 28, 29, 15, 16, 27, 28, 31, 7, 16, 16, 20...
$ impr      <dbl> 7350, 17861, 693, 4259, 4133, 1915, 15615, 10951, 2355, 9...
$ Clicks    <dbl> 1, 2, 0, 1, 1, 0, 3, 1, 1, 3, 0, 0, 0, 0, 7, 0, 1, 0, 1, ...
$ Spent     <dbl> 1.43, 1.82, 0.00, 1.25, 1.29, 0.00, 4.77, 1.27, 1.50, 3.1...
$ conv      <dbl> 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
$ appConv   <dbl> 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, ...
$ CTR       <dbl> 0.0136, 0.0112, 0.0000, 0.0235, 0.0242, 0.0000, 0.0192, 0...
$ CPC       <dbl> 1.43, 0.91, NaN, 1.25, 1.29, NaN, 1.59, 1.27, 1.50, 1.05,...
```

Okay, that's added some more useful variables to our data set. Let's trim out the campaign and demographic variables and look for some correlations:

In [16]:
```
# create trimmed dataset
dataTfTrim <- dataTf %>%
  select(CTR, CPC, appConv, conv, impr, Spent, Clicks)
```

In [17]:
```
# omit missing values, normalise data, calculate correlations and plot heatmap
heatmap(cor(normalize(na.omit(dataTfTrim))))
```





As we might expect, we've got some strong correlations between the amount we spent and how many impressions and clicks we got, with less strong correlations between our spend, clicks and impressions and our conversions. If we wanted to at this point, we could follow this up and calculate the significance of these correlations, but, for now, let's dive into a specific campaign and get a bit more granular.

From our broad overview of the data, we can see that the more we spend, the more clicks and conversions we seem to get. That's quite reassuring to know, but doesn't really give us the 'actionable insight' we were looking for.

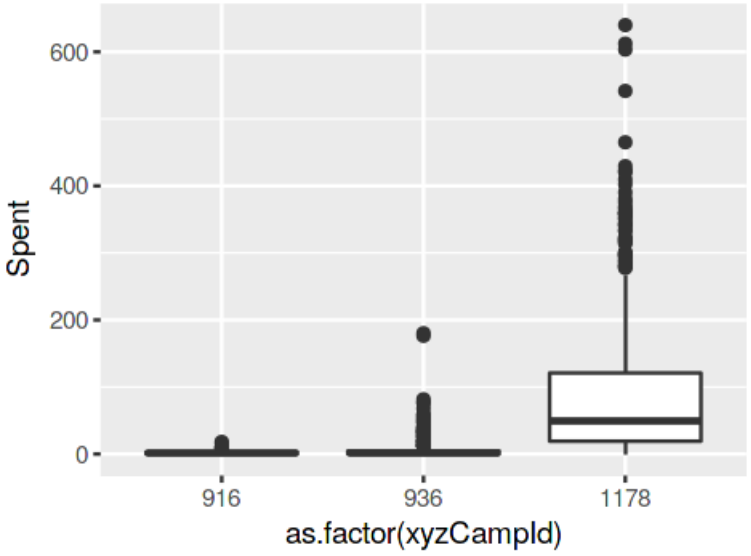
For our next stage in the analysis, we'll look at a specific campaign in more detail and see what we can pull out of it. First of all, let's choose a campaign, the one on which we regularly spend the most money and regularly get the most conversions (and for which we have the most data!) might be a good place to start.

In [18]:
```
# set plot size options
options(repr.plot.width=4, repr.plot.height=3)

ggplot(dataTf, aes(as.factor(xyzCampId), Spent)) + geom_boxplot()
  + labs(x = "Campaign", y = "Advertising Spend")


ggplot(dataTf, aes(as.factor(xyzCampId), conv)) + geom_boxplot()
  + labs(x = "Campaign", y = "Conversions")
```

```
Error in +labs(x = "Campaign", y = "Advertising Spend"): invalid argument to unary operator
Traceback:
```



Looks like campaign '1178' is the one to go for, so we'll create a new dataframe that just includes the data from that campaign using dplyr's `filter` function. As we're done with our heatmaps, we can go back to having our age and gender variables as character strings.

In [19]:
```
data1178 <- data %>%
  rename(xyzCampId = xyz_campaign_id, fbCampId = fb_campaign_id, impr = Impressions,
         conv = Total_Conversion, appConv = Approved_Conversion) %>%
  filter(xyzCampId == 1178)
```

In [20]:
```
glimpse(data1178)
```

```
Observations: 625
Variables: 11
$ ad_id     <dbl> 1121091, 1121092, 1121094, 1121095, 1121096, 1121097, 112...
$ xyzCampId <dbl> 1178, 1178, 1178, 1178, 1178, 1178, 1178, 1178, 1178, 117...
$ fbCampId  <dbl> 144531, 144531, 144531, 144531, 144531, 144532, 144532, 1...
$ age       <chr> "30-34", "30-34", "30-34", "30-34", "30-34", "30-34", "30...
$ gender    <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M...
$ interest  <dbl> 10, 10, 10, 10, 10, 15, 15, 15, 15, 15, 16, 16, 16, 16, 1...
$ impr      <dbl> 1194718, 637648, 24362, 459690, 750060, 30068, 1267550, 3...
$ Clicks    <dbl> 141, 67, 0, 50, 86, 1, 123, 340, 1, 30, 202, 9, 1, 95, 12...
$ Spent     <dbl> 254.05, 122.40, 0.00, 86.33, 161.91, 1.82, 236.77, 639.95...
$ conv      <dbl> 28, 13, 1, 5, 11, 1, 24, 60, 2, 7, 40, 5, 2, 26, 6, 4, 7,...
$ appConv   <dbl> 14, 5, 1, 2, 2, 0, 10, 17, 1, 3, 21, 2, 0, 14, 2, 1, 4, 0...
```

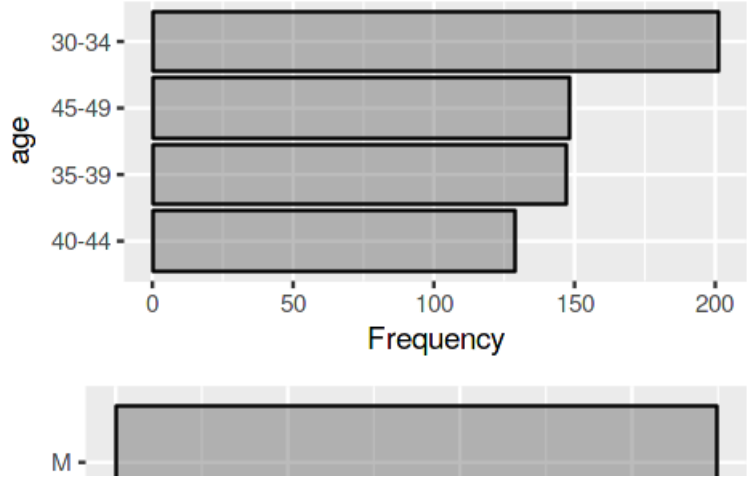
As a first overview of the data, we'll use the DataExplorer package and we'll just have a quick overview

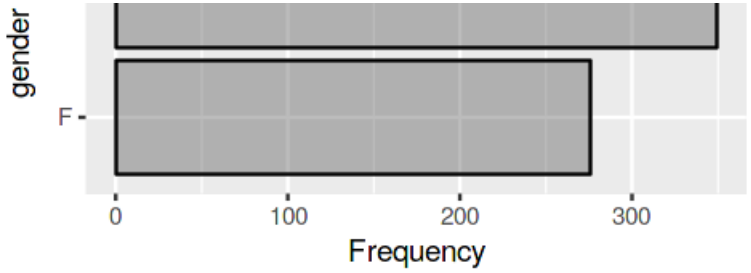
In [21]:
```
library(DataExplorer)
```

In [22]:
```
# look for missing data
plot_missing(data1178)
```

Nothing missing, good to know. Let's look at the distributions of our data, variable by variable:

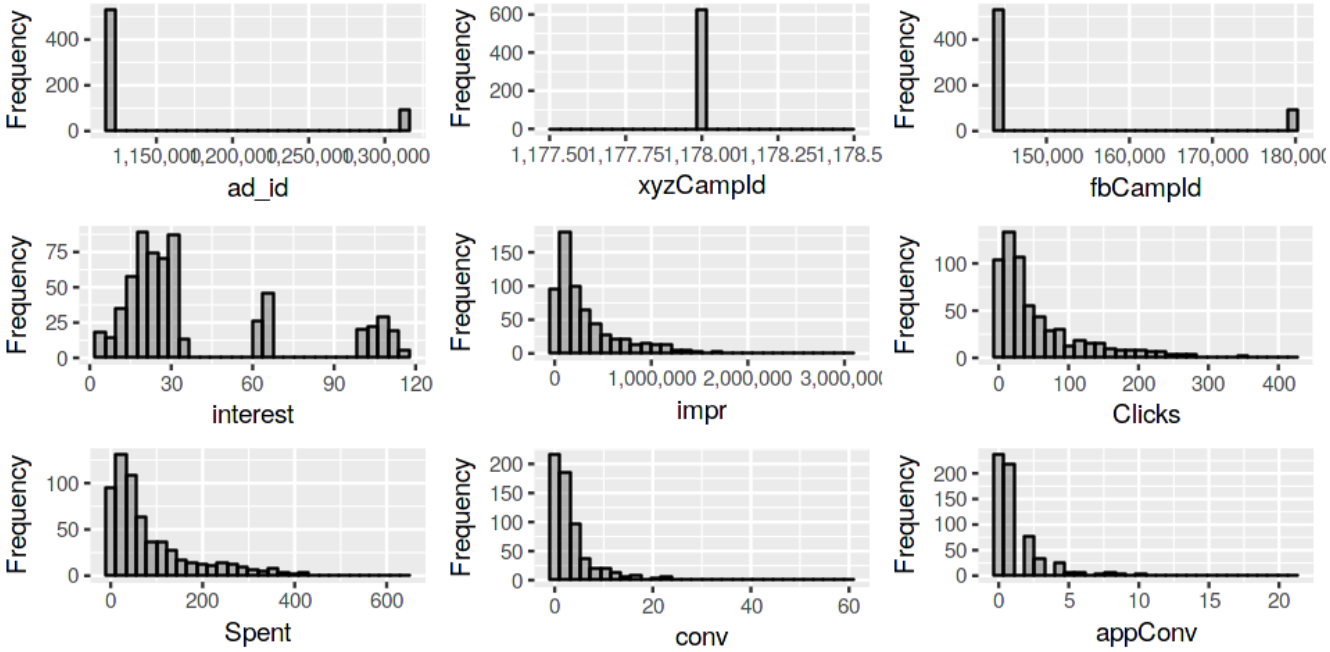
In [23]:
```
options(repr.plot.width=4, repr.plot.height=4)
plot_bar(data1178)
```





In [24]:

```
options(repr.plot.width=8, repr.plot.height=4)
plot_histogram(data1178)
```

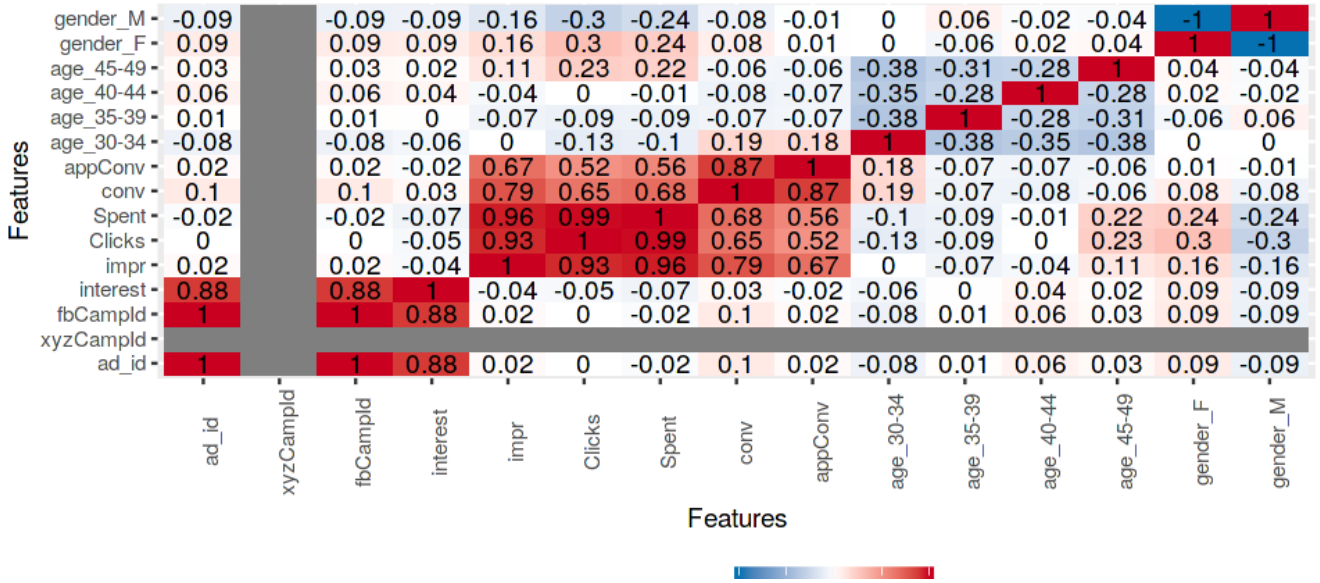


In [25]:

```
# and we'll revisit our correlation matrix for the 1178 campaign
plot_correlation(data1178, use = "pairwise.complete.obs")
```

Warning message in cor(final_data, ...):

```
Warning message in cor(final_data, ...):
"the standard deviation is zero"Warning message:
"Removed 29 rows containing missing values (geom_text)."
```



An introduction to Facebook ad analysis using R

R notebook using data from Sales Conversion Optimization · 10,909 views · 2y ago · data visualization, tutorial, marketing analytics, +1 more

31 Copy and Edit 67

This overview of the dataset allows us to get a feel for what's going on with the variables in this campaign. We have a feel for the distributions and can begin to think about what sort of questions we might want to ask, and what types of analysis might be appropriate.

Version 13

13 commits

More feature engineering

We don't have the actual numbers at our disposal here, but for the purposes of this tutorial, let's assume that an enquiry (Total conversion, conv) is worth £5, and a sale (Approved conversion, appConv), is worth £100. We can now create our conversion value-based variables sing `mutate` :

In [26]:

```
data1178 <- data1178 %>%
```

```
  mutate(totConv = conv + appConv,
         conVal = conv * 5,
         appConVal = appConv * 100) %>%
  mutate(totConVal = conVal + appConVal) %>%
  mutate(costPerCon = round(Spent / totConv, 2),
         ROAS = round(totConVal / Spent, 2))
```

Let's also introduce another KPI that we haven't talked about yet: **Cost Per Mille (CPM)**. This number is the cost of one thousand impressions. If your objective is ad exposure to increase brand awareness, this might be an important KPI for you to measure.

In [27]:

```
data1178 <- data1178 %>%
  mutate(CPM = round((Spent / impr) * 1000, 2))
```

In [28]:

```
# take a look at our new variables
head(data1178)
```


1121091	1178	144531	30-34	M	10	1194718	141	254.05	28	14	42	140	1400
1121092	1178	144531	30-34	M	10	637648	67	122.40	13	5	18	65	500
1121094	1178	144531	30-34	M	10	24362	0	0.00	1	1	2	5	100
1121095	1178	144531	30-34	M	10	459690	50	86.33	5	2	7	25	200
1121096	1178	144531	30-34	M	10	750060	86	161.91	11	2	13	55	200
1121097	1178	144532	30-34	M	15	30068	1	1.82	1	0	1	5	0

The first thing to note is that we can see a row with no clicks, but that has a conversion, giving us a ROAS of infinity. Nice, but probably not what we want in our data. This could perhaps have happened if a conversion was attributed to the campaign, but either the click wasn't tracked, or occurred at a different time and has been attributed elsewhere.

It's still a conversion, so we want it in there for the purposes of our aggregrate statistics, but we do need to remember that it's there and consider what that might be doing as we work through our analyses.

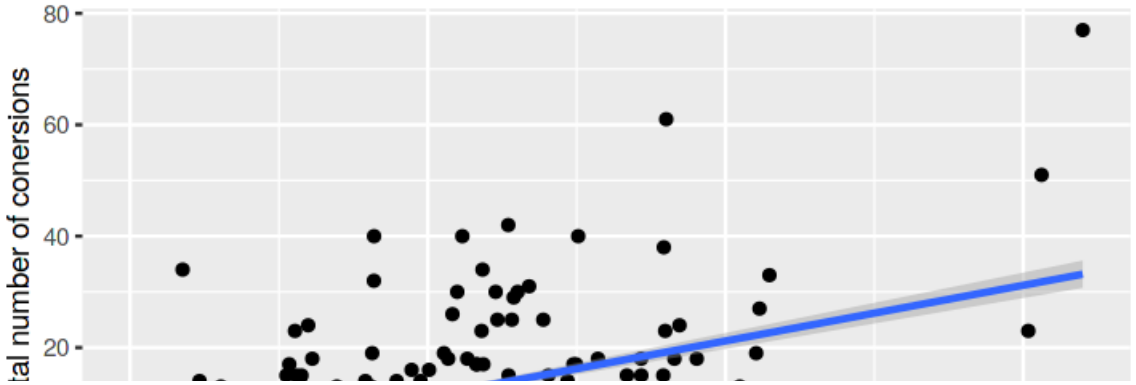
Preliminary analysis of campaign 1178

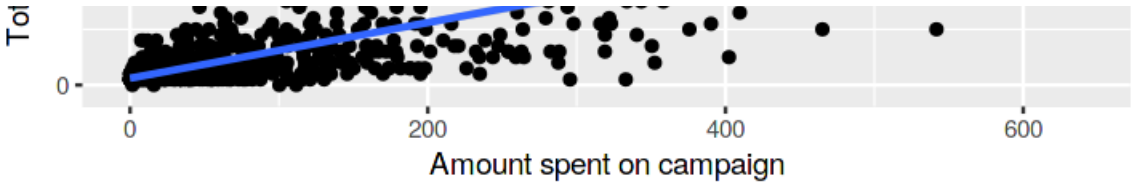
The types of analyses you would want to perform on your own data will depend very much on your campaign objectives, what sort of data you have and what decisions you want to be able to have the insight to support. As we don't know the context for this dataset, we'll take a look through it from the point of view of exploratory data analysis, using the types of tools you could use on your own data.

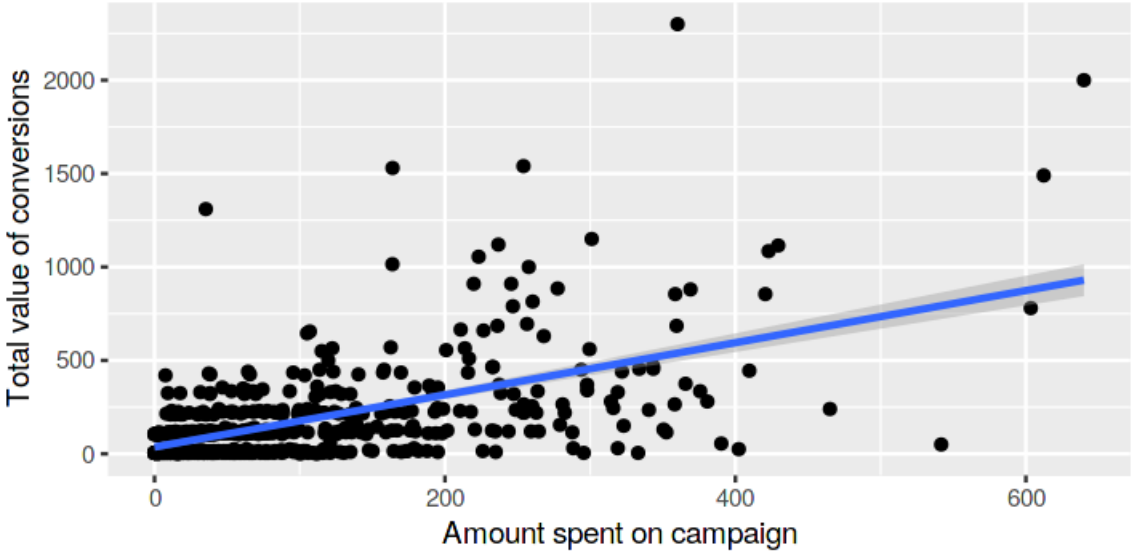
For our purposes, we'll assume that this is an e-commerce business that is purely focussed on maximising revenue.

We'll start by looking at what happens to the number of conversions and the value of our conversions when we spend more money on our campaign. If we spend more, do we get more back?

In [29]:
```
options(repr.plot.width=6, repr.plot.height=3)
ggplot(data1178, aes(Spent, totConv)) + geom_point() + geom_smooth(method = "lm") +
  labs(x = "Amount spent on campaign", y = "Total number of conersions")
ggplot(data1178, aes(Spent, totConVal)) + geom_point() + geom_smooth(method = "lm") +
  labs(x = "Amount spent on campaign", y = "Total value of conversions")
```







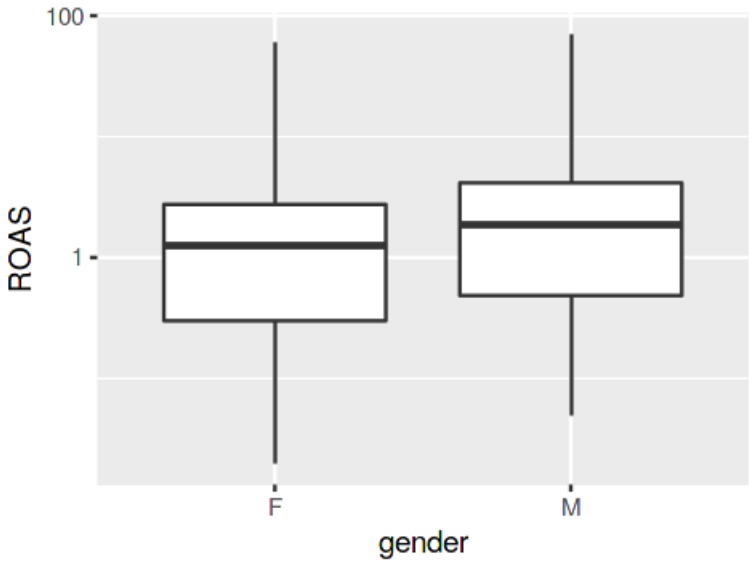
At first glance, then, it looks like the more we spend, the more we get back, but the amount of data is quite sparse at the right-hand side of the budget, so we could just have been lucky there. We'll go into things in a bit more detail before we start recommending that we should just increase our advertising budget...

With Facebook, we have a lot of demographic information we can use, so we'll use that to break apart our dataset, we'll start by splitting the data by gender:

In [30]:
```
options(repr.plot.width=4, repr.plot.height=3)
ggplot(data1178, aes(gender, ROAS)) + geom_boxplot() + scale_y_log10()
```

```
Warning message:
"Transformation introduced infinite values in continuous y-axis"Warning message:
```

```
"Removed 16 rows containing non-finite values (stat_boxplot)."
```



The data look quite symmetrical with a log-transformed axis, but without the log-transformation, it doesn't fit the normal distribution, so we'll look to see if this difference is significant using a non-parametric test:

In [31]:
```
wilcox.test(ROAS ~ gender, data=data1178)
```

```
        Wilcoxon rank sum test with continuity correction

data:  ROAS by gender
W = 37490, p-value = 4.526e-05
alternative hypothesis: true location shift is not equal to 0
```

And let's look for the median and the mean of these data:

In [32]:

```
data1178 %>%
  select(gender, ROAS) %>%
  group_by(gender) %>%
  filter(ROAS != 'Inf') %>%
  summarise(medianROAS = median(ROAS), meanROAS = mean(ROAS))
```

gender	medianROAS	meanROAS
F	1.225	2.819228
M	1.880	4.504633

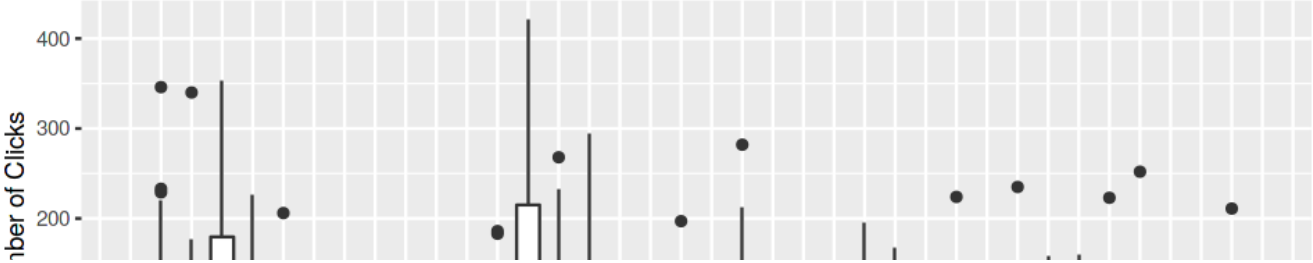
Analysis by interest

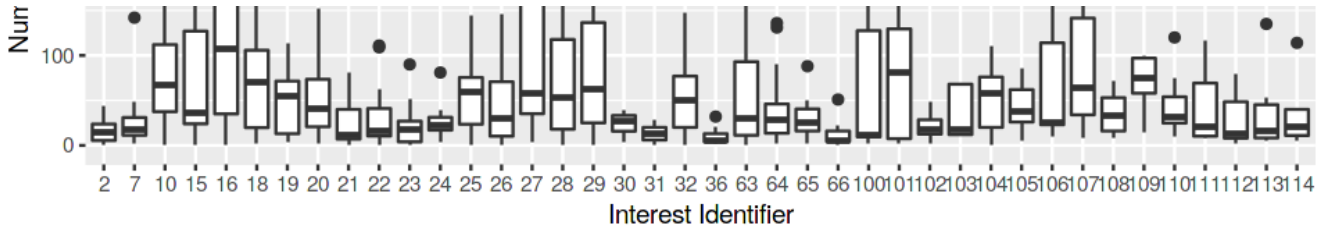
It looks like our ROAS is a higher for males than females and that this difference is statistically significant ($p < 0.01$).

In this case, while the median does give us a more accurate estimation of what our ROAS would be for a particular adID, there are a lot of points that pull the data towards the right. Over time, the ROAS is more likely to tend towards the mean. Using that figure, we can see that the ROAS differences by gender are quite striking and, depending on the profit margins involved, could make the difference between the campaign being profitable or not.

However, we have the data to go a lot more granular than this, so let's see how else we can break the dataset apart. We'll look at interests next:

In [33]:
```
options(repr.plot.width=8, repr.plot.height=3)
ggplot(data1178, aes(as.factor(interest), Clicks)) + geom_boxplot() +
  labs(x = "Interest Identifier", y = "Number of Clicks")
```

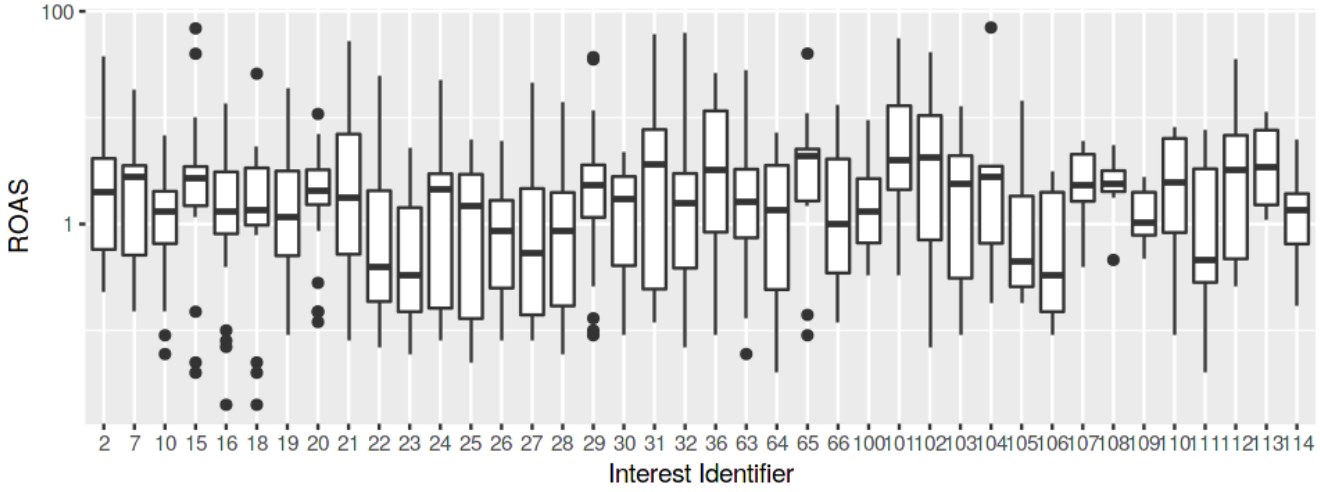




In [34]:

```
options(repr.plot.width=8, repr.plot.height=3)
data1178 %>%
  ggplot(aes(as.factor(interest), ROAS)) + geom_boxplot() + scale_y_log10() +
  labs(x = "Interest Identifier", y = "ROAS")
```

```
Warning message:
“Transformation introduced infinite values in continuous y-axis”Warning message:
“Removed 16 rows containing non-finite values (stat_boxplot).”
```



We can see that our different interest groups are performing differently; we'll quantify that and look at our best performers by ROAS.

In [35]:

```
data1178 %>%
  select(interest, ROAS, Clicks) %>%
  group_by(interest) %>%
  filter(ROAS != 'Inf') %>%
  summarise(medianROAS = round(median(ROAS) ,2),
            meanROAS = round(mean(ROAS), 2), clicks = sum(Clicks)) %>%
  arrange(desc(meanROAS)) %>%
  head(n = 10)
```

interest	medianROAS	meanROAS	clicks
104	2.75	15.51	265
101	3.96	14.95	524
102	4.27	10.35	150
31	3.64	8.26	189
112	3.21	8.06	339
15	2.68	7.89	1554
36	3.21	7.38	126
65	4.38	7.00	343
21	1.77	6.34	493
2	2.08	5.41	306

Analysis by gender

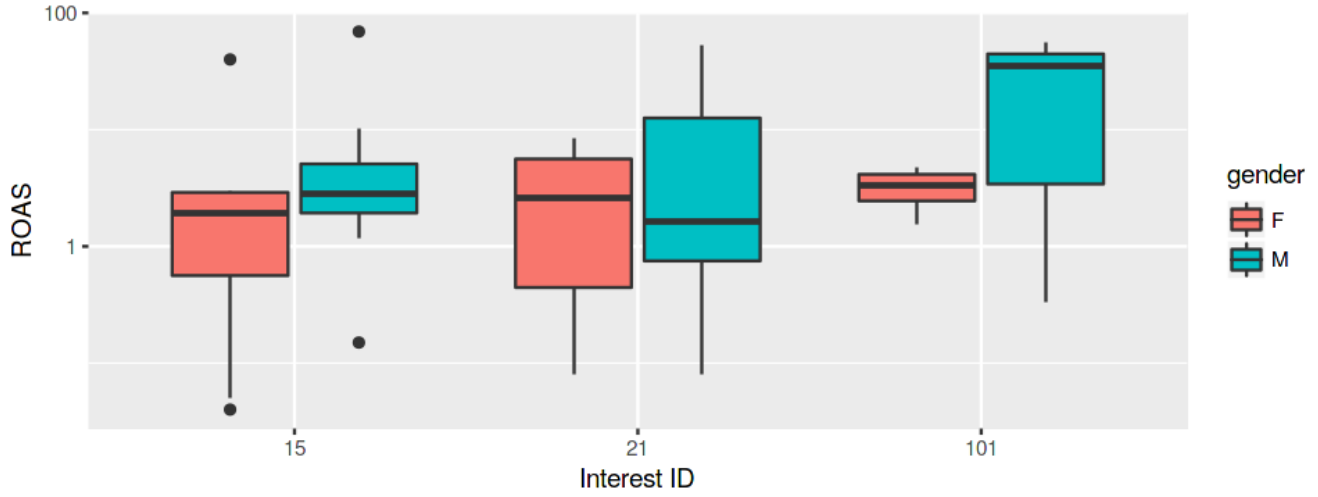
There are a few interests there that are showing a good ROAS and that have a healthy number of clicks associated with them; we'll choose interests 101, 15 and 21 to investigate a little further. Having selected those interests, we'll now break them apart again by gender.

In [36]:

```
options(repr.plot.width=8, repr.plot.height=3)
data1178 %>%
  filter(interest == 101 | interest == 15 | interest == 21) %>%
  ggplot(aes(x = as.factor(interest), y = ROAS, fill = gender)) + geom_boxplot() + scale_y_log1
```

```
0() +
  labs(x = 'Interest ID', y = 'ROAS')
```

```
Warning message:
"Removed 1 rows containing non-finite values (stat_boxplot)."
```



In [37]:

```
data1178 %>%
  select(interest, gender, ROAS, Clicks) %>%
  group_by(interest, gender) %>%
  filter(ROAS != 'Inf', interest == 101 | interest == 15 | interest == 21) %>%
  summarise(medianROAS = round(median(ROAS), 2),
            meanROAS = round(mean(ROAS) ,2), clicks = sum(Clicks)) %>%
  arrange(desc(meanROAS))
```

interest	gender	medianROAS	meanROAS	clicks
101	M	35.47	30.53	17
21	M	1.62	9.63	200
15	M	2.79	8.89	827
15	F	1.98	6.38	727

21	F	2.60	3.36	293
101	F	3.41	3.28	507

Looking at the results above, increasing our budget to display our ads to males with interest 101 might make a lot of sense as it's currently bringing in over £30 for every £1 it's costing. However, this is with a small number of clicks (17), so could just be due to chance.

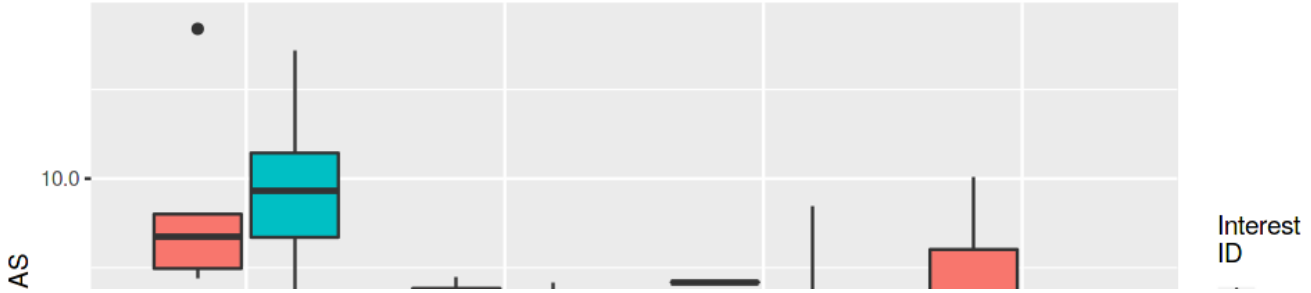
Given this finding, it is tempting to recommend a modest increase in the budget for this demographic, but to follow it closely to see if it maintains this ROAS longer-term. The campaign budgets for males with interests 21 and 15, and females with interest 15 could also be increased, with a reduction in the spend on the demographics with the lowest ROAS.

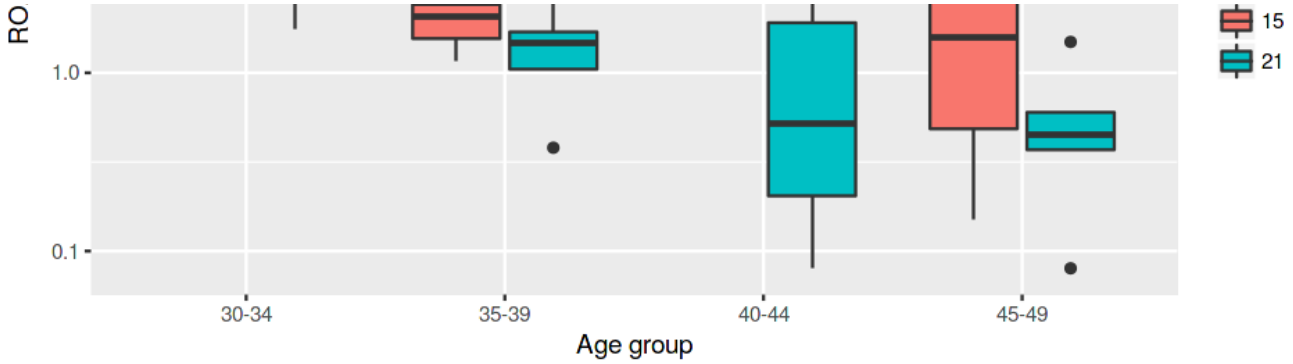
Analysis by age

We've been able to break apart this campaign by interest and gender, with each split allowing us to pull out groups with the highest ROAS. What about the age variable though, we haven't used that yet, so let's see if we can be even more targeted:

In [38]:
```
options(repr.plot.width=8, repr.plot.height=4)
data1178 %>%
  filter(interest == 21 | interest == 15 & gender == 'M') %>%
  group_by(age, interest) %>%
  ggplot(aes(x = as.factor(age), y = ROAS, fill = as.factor(interest))) + geom_boxplot() + scal
e_y_log10() +
  labs(x = 'Age group', y = 'ROAS') + scale_fill_discrete(name="Interest\nID")
```

```
Warning message:
“Removed 1 rows containing non-finite values (stat_boxplot).”
```





In [39]:

```
data1178 %>%
  select(age, interest, gender, ROAS, Clicks) %>%
  group_by(age, interest) %>%
  filter(ROAS != 'Inf', interest == 21 | interest == 15, gender == 'M') %>%
  summarise(medianROAS = round(median(ROAS), 2),
            meanROAS = round(mean(ROAS) ,2), clicks = sum(Clicks)) %>%
  arrange(desc(meanROAS))
```

age	interest	medianROAS	meanROAS	clicks
30-34	21	13.92	18.44	58
30-34	15	4.73	17.22	495
45-49	15	1.57	3.97	138
40-44	15	2.62	2.62	26
35-39	15	2.07	2.02	168
35-39	21	1.47	1.47	44
45-49	21	0.45	0.38	98

It looks like we're getting the best ROAS with aour 30 - 34 year old age group, so we could think about increasing our spend to increase our visibility there. However, the more granular we go with the data, the lower our number of observations and the less sure we can be about these differences being genuine, rather than simply noise.

Of course, once you have performed an EDA such as this and come up with some insight, you can go back through the data, picking your analyses with an endpoint in mind and using the appopriate statistical tests to see how likely it is that these differences are due to chance, giving you more confidence in making appropriate recommendations with the appropriate caveats.

Final thoughts

Hopefully this notebook has been of some use to if you're new to pay-per-click advertising, or if you've been looking for new ways to try and improve ROI from your digital campaigns.

This notebook is just a quick glimpse into the sort of analyses you can do with your digital advertising datasets, but it really is only a starting point: the correct types of analysis and measures of success will be driven by your own business model and your underlying marketing objectives.

For example, if you have a physical business as well as an online presence, how do you factor in people becoming aware of your business, product or promotion online, but converting in store in person? What about products with long buying cycles, where the resulting conversion could be months after the initial

Combining with other data sources such as Google Analytics

As briefly discussed above, while ROAS reports on campaigns tactically, ROI is more strategic. To start to work out ROI, we would likely want to start working with data from other sources, such as our website analytics data. As our Facebook ad campaigns can contain plenty information in the URL that sends visitors to our website, we can look at how much website traffic the campaigns generated and how visitors from that campaign interacted with our website.

With that information, we could look to see if there are other events that we could consider a conversion. Did the visitor subscribe to our email newsletter? Did they spend more than three minutes on the site and browse more than ten pages? Did they bookmark the site and return to make a purchase some time later? If that is the case, their conversion may not be assigned to that campaign in one location, but may be visible as an 'assisted conversion' in the *Multi-Channel Funnels* section of Google Analytics. Then there are other potential values, such as the ability to now 'remarket/retarget (https://support.google.com/adwords/answer/2453998?hl=en-GB)' adverts to that visitor.

Know where your visitors go, how they interact with you and what goals are worth

Additionally, ROI calculations can consider things such as the lifetime value of the customer. With an advertising campaign, you might not get a sale today, but you might get a visit. Will they come back and purchase later? If a customer makes one purchase, do they end up making more over the next few weeks, months, years...? How much do they spend and does that fall off over time? All of these factors can add up to make

over the next few weeks, months, years...? How much do they spend and does that fall off over time? All of these factors can add up to make that initial cost-per-click better value than it might have seemed at the time.

By assigning values to the various goals you have in place on your website, and by knowing where visitors came from and how they interact with you over time, you can make better judgments and decisions on how your marketing campaigns are performing.

Data

Data Sources

- Sales Conversion Optimization
  - KAG_conversion_data.csv — 11 columns

Sales Conversion Optimization

How to Cluster Customer data for campaign marketing

Last Updated: 2 years ago (Version 1)

About this Dataset

Context

Cluster Analysis for Ad Conversions Data

Content

The data used in this project is from an anonymous organisation's social media ad campaign. The data file can be downloaded from here. The file conversion_data.csv contains 1143 observations in 11 variables. Below are the descriptions of the variables.

1.) ad_id: an unique ID for each ad.

2.) xyz_campaign_id: an ID associated with each ad campaign of XYZ company.

3.) fb_campaign_id: an ID associated with how Facebook tracks each campaign.

4.) age: age of the person to whom the ad is shown.

5.) gender: gender of the person to whim the add is shown

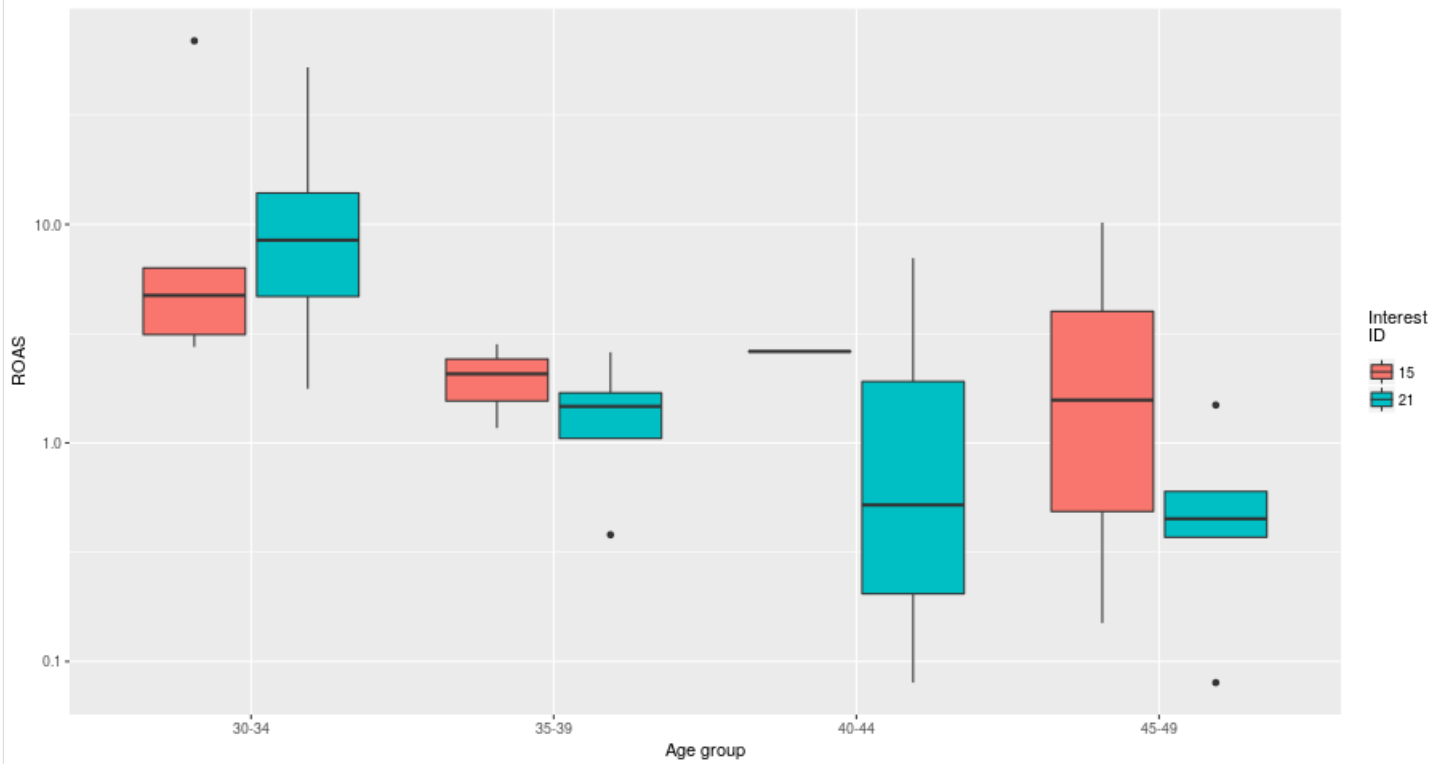
6.) interest: a code specifying the category to which the person's interest belongs (interests are as mentioned in the person's Facebook public profile).

7.) Impressions: the number of times the ad was shown.

8.) Clicks: number of clicks on for that ad.

9.) Spent: Amount paid by company xyz to Facebook, to show that ad.

Output Visualizations



Comments (5)

Sort by

All Comments | Hotness

Click here to comment...

Andrew Lerner • Posted on Latest Version • a year ago • Options • Reply 2

This is really awesome - I am wondering how you got the attributes for each person the ad was shown to e.g. interests, age, etc.?

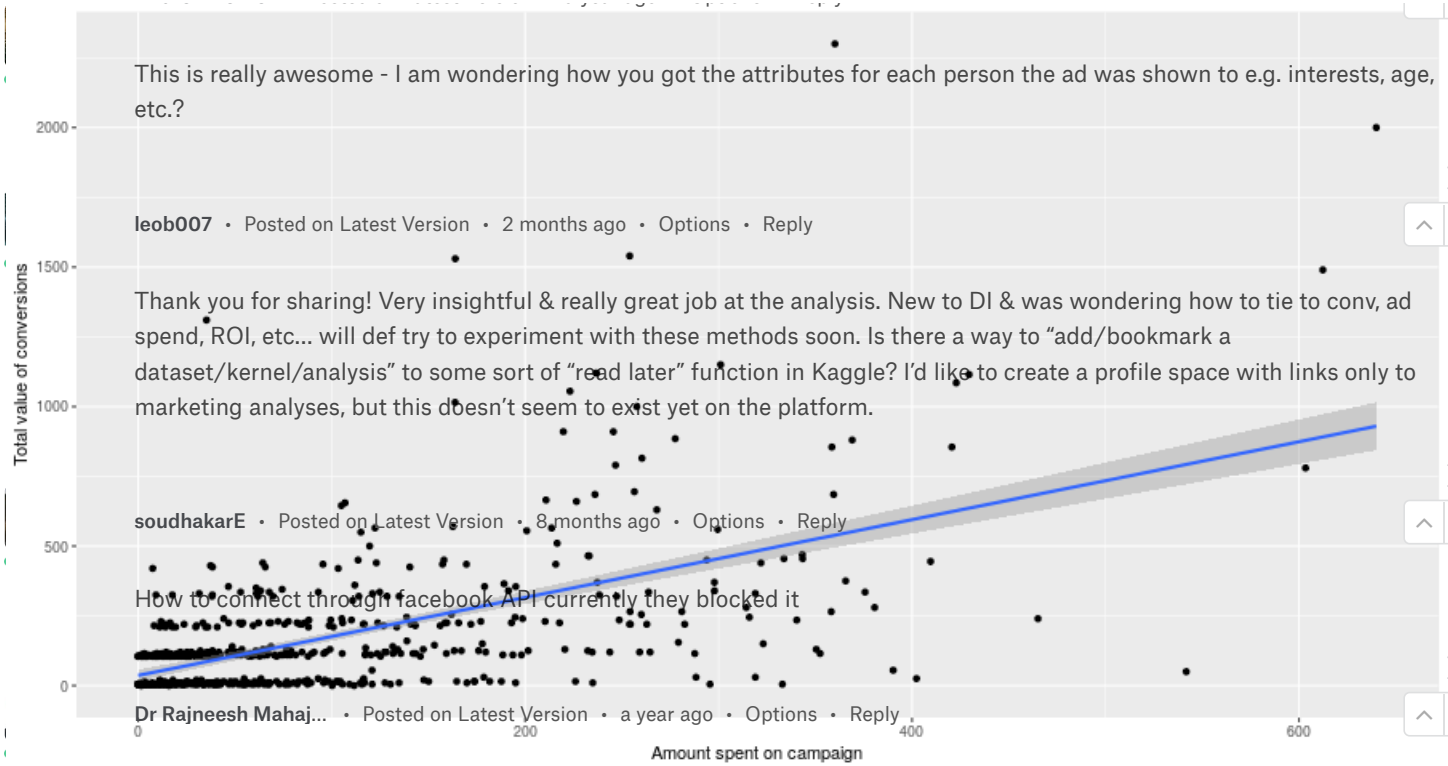
leob007 • Posted on Latest Version • 2 months ago • Options • Reply 0

Thank you for sharing! Very insightful & really great job at the analysis. New to DI & was wondering how to tie to conv, ad spend, ROI, etc... will def try to experiment with these methods soon. Is there a way to "add/bookmark a dataset/kernel/analysis" to some sort of "read later" function in Kaggle? I'd like to create a profile space with links only to marketing analyses, but this doesn't seem to exist yet on the platform.

soudhakarE • Posted on Latest Version • 8 months ago • Options • Reply 0

How to connect through facebook API currently they blocked it

Dr Rajneesh Mahaj... • Posted on Latest Version • a year ago • Options • Reply 0



Hi work is excellent. I tried but unable to run many codes, can you please explain the steps in conducting analysis. My mahajanrajneesh1@gmail.com

Al Mahdi Marhou • Posted on Latest Version • 2 years ago • Options • Reply 0

Great work!

Similar Kernels