Facebook Ad Analysis

Karthik

7/24/2020

Initial Thoughts This data is originally downloaded from ***Kaggle Competition - Sales COnversion Optimization***.

I thank the contributor of the dataset.

What do we need from Facebook ad 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.

It is important to know the company's marketing strategy or campaign objectives so that we do know which key performance indicators (KPIs) are the most important. For example, if a new company focusses on brand awareness then they 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.

Loading the data

```
setwd("C:/Users/karthikkannepalli/Downloads/My Projects/FB Ad Analysis")
ad_data = read.csv("datasets_2678_4448_KAG_conversion_data.csv")
df = ad_data
```

Quick look at the data

```
library(dplyr)
glimpse(ad_data)
```

```
## Rows: 1,143
## Columns: 11
## $ ad_id               <int> 708746, 708749, 708771, 708815, 708818, 708820,...
## $ xyz_campaign_id     <int> 916, 916, 916, 916, 916, 916, 916, 916, 916, 91...
## $ fb_campaign_id      <int> 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            <int> 15, 16, 20, 28, 28, 29, 15, 16, 27, 28, 31, 7, ...
## $ Impressions         <int> 7350, 17861, 693, 4259, 4133, 1915, 15615, 1095...
## $ Clicks              <int> 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,...
## $ Total_Conversion    <int> 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ Approved_Conversion <int> 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,...
```

The documentation 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.

Replace character string age ranges with number

```
library(dplyr)
df$age[df$age == '30-34'] = 32
df$age[df$age == '35-39'] = 37
df$age[df$age == '40-44'] = 42
df$age[df$age == '45-49'] = 47

df$age = as.integer(df$age)
```

convert gender variable to integer

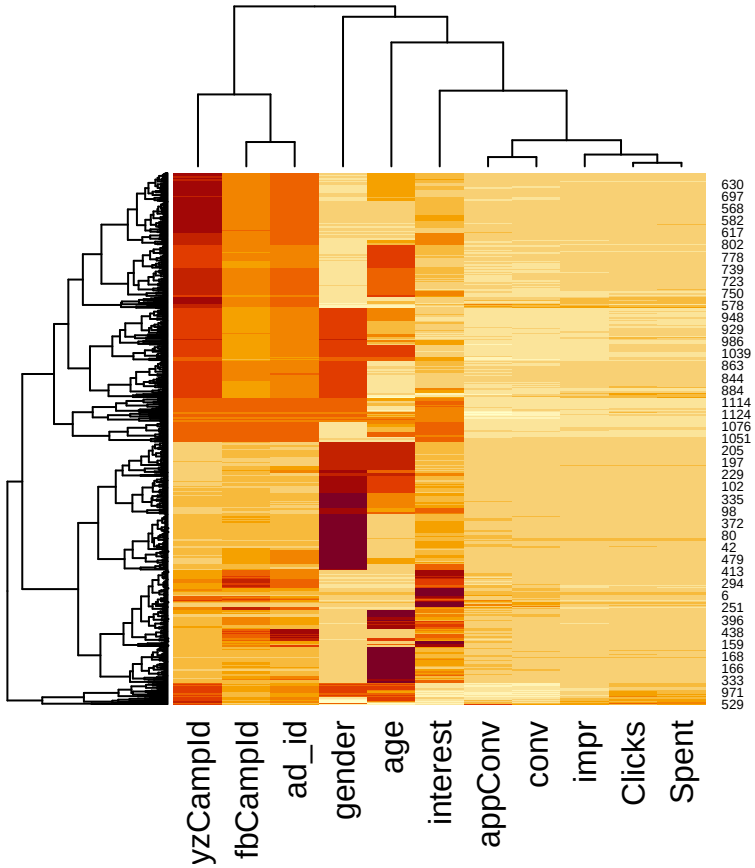
```
df$gender[df$gender == 'M'] = 0
df$gender[df$gender == 'F'] = 1

df$gender = as.integer(df$gender)
```

abbreviate some variable names

```
df = df %>%
  rename(xyzCampId = xyz_campaign_id, fbCampId = fb_campaign_id, impr = Impressions,
         conv = Total_Conversion, appConv = Approved_Conversion)
```

Plotting HeatMap

```
library(heatmaply)
dataMatNorm = as.matrix(normalize(df, method = "standardize"))
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.

Create Additional KPIs

```
df = df %>%
  mutate(CTR = ((Clicks / impr) * 100), CPC = Spent / Clicks)

df$CTR = round(df$CTR, 4)
df$CPC = round(df$CPC, 2)
```

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.

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.

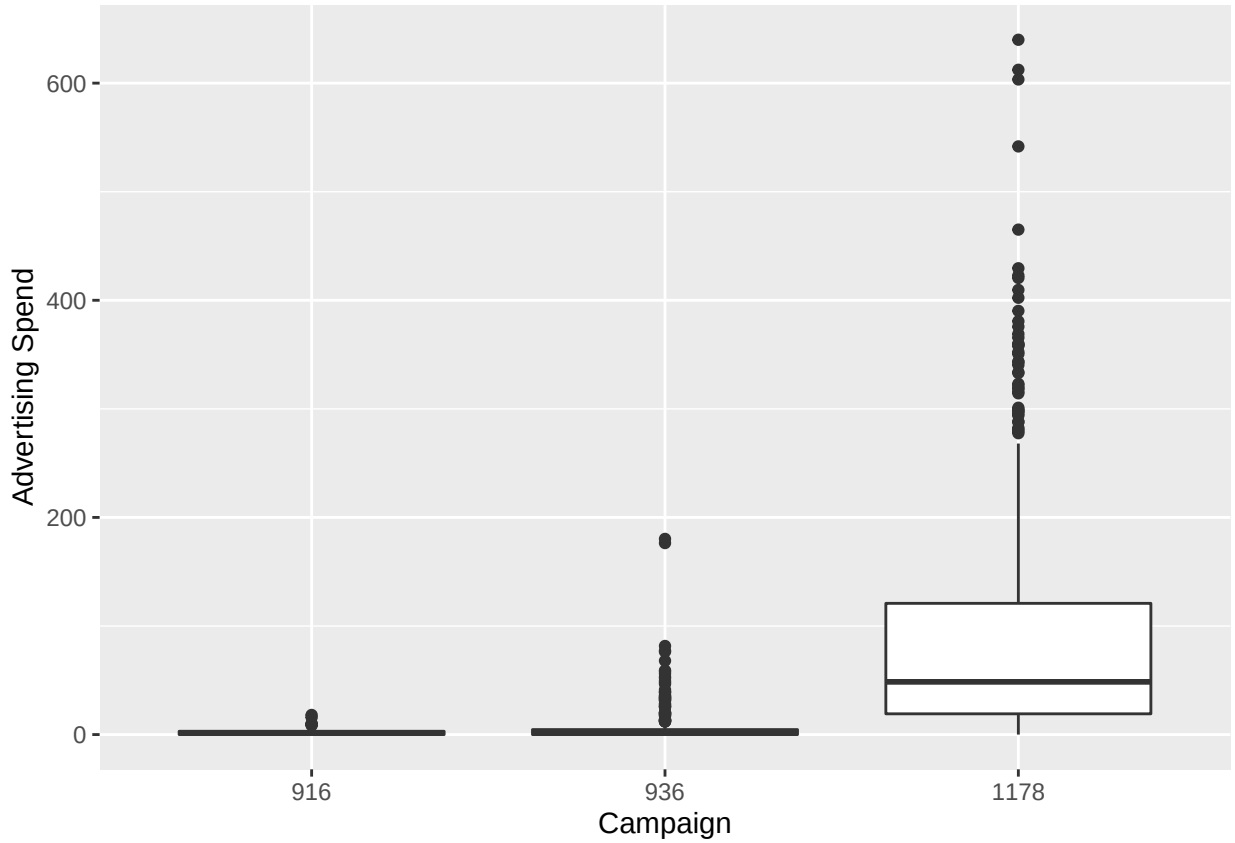
Create Trimmed dataset

```r
dfTrim = df %>%
  select(CTR, CPC, appConv, conv, impr, Spent, Clicks)
```

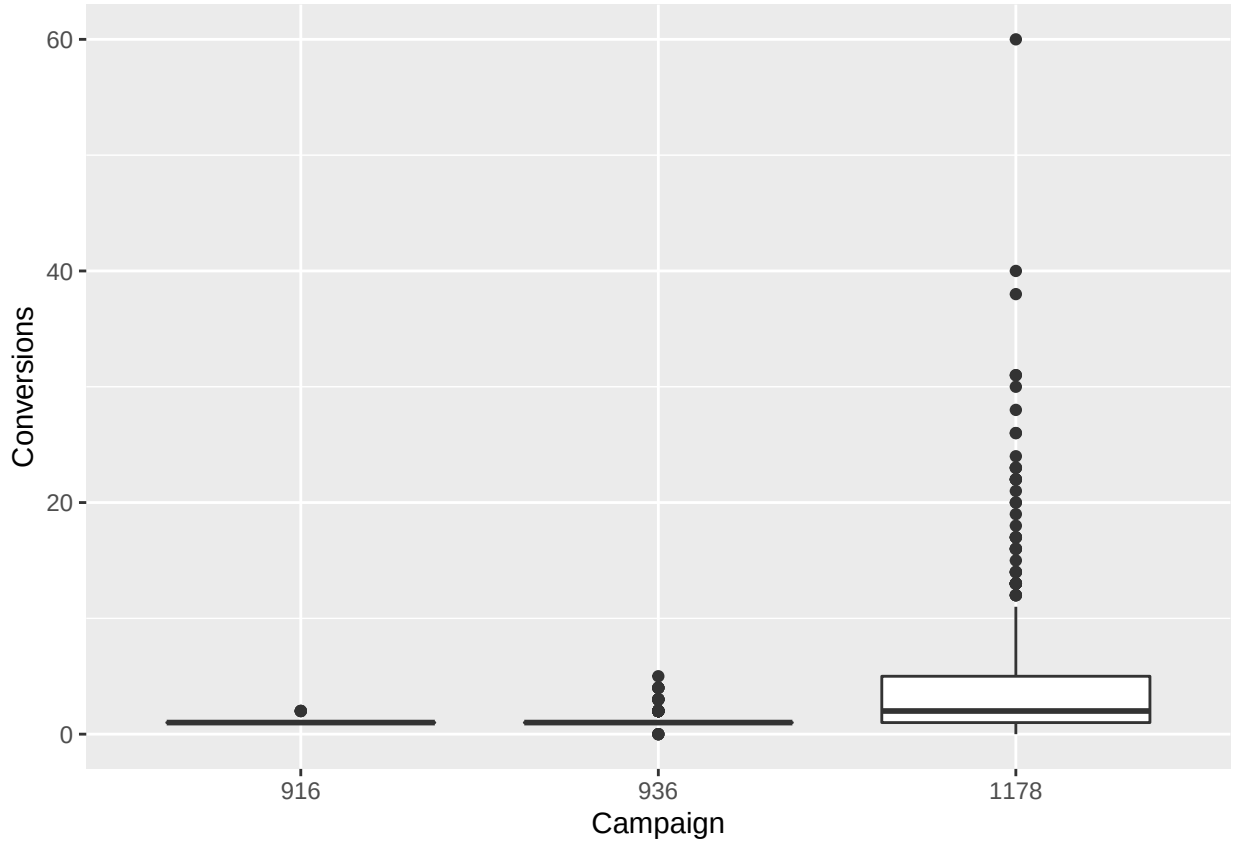
For our next stage in the analysis, we'll look at a specific campaign in more detail and see what we can pull out of it. First of all, let's choose a campaign, the one on which we regularly spend the most money and regularly get the most conversions (and for which we have the most data!).

Create Trimmed dataset

```r
library(ggplot2)
ggplot(df, aes(as.factor(xyzCampId), Spent)) + geom_boxplot() + labs(x = "Campaign", y = "Advertising Sp
```




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



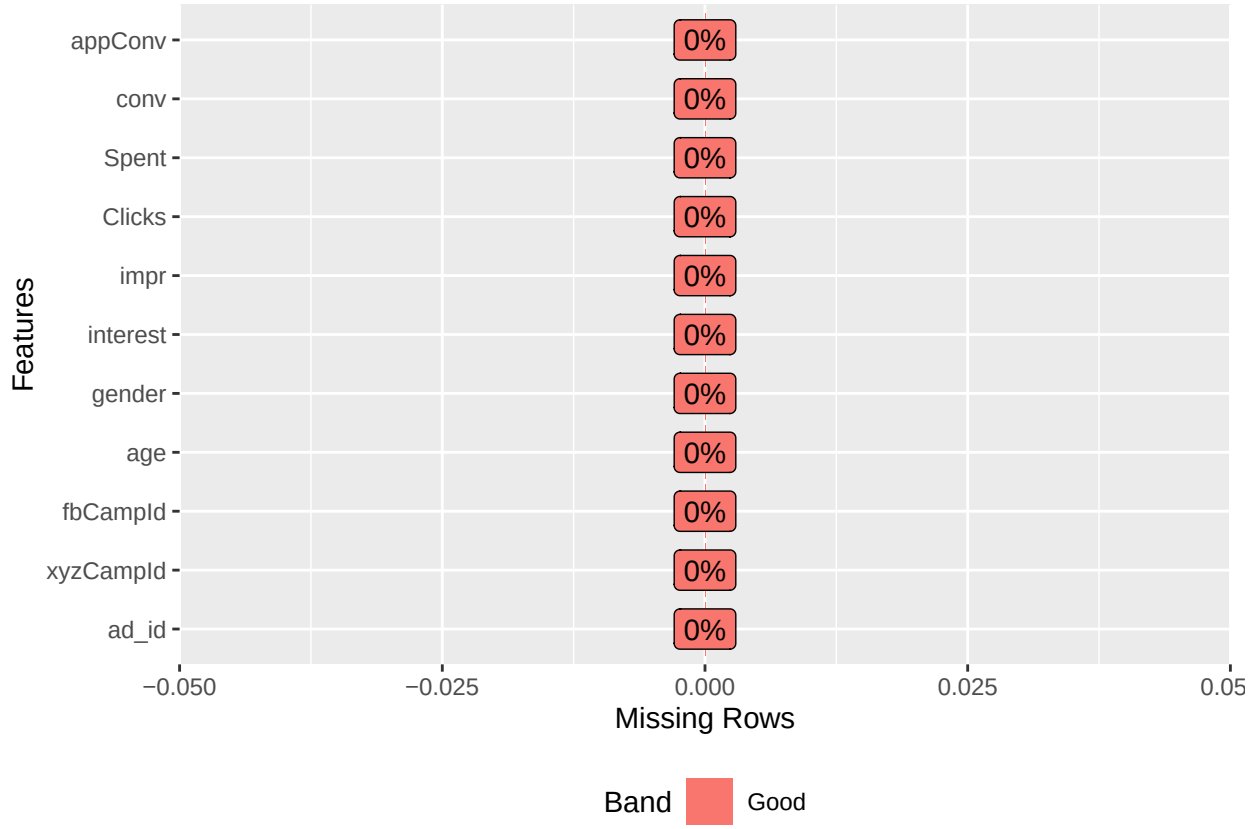
Campaign '1178' is the one to go for, so we'll create a new dataframe that just includes the data from that campaign.

Filtering for Ad_campaign = 1178

```
data1178 = ad_data %>%
  rename(xyzCampId = xyz_campaign_id, fbCampId = fb_campaign_id, impr = Impressions,
         conv = Total_Conversion, appConv = Approved_Conversion) %>%
  filter(xyzCampId == 1178)
```

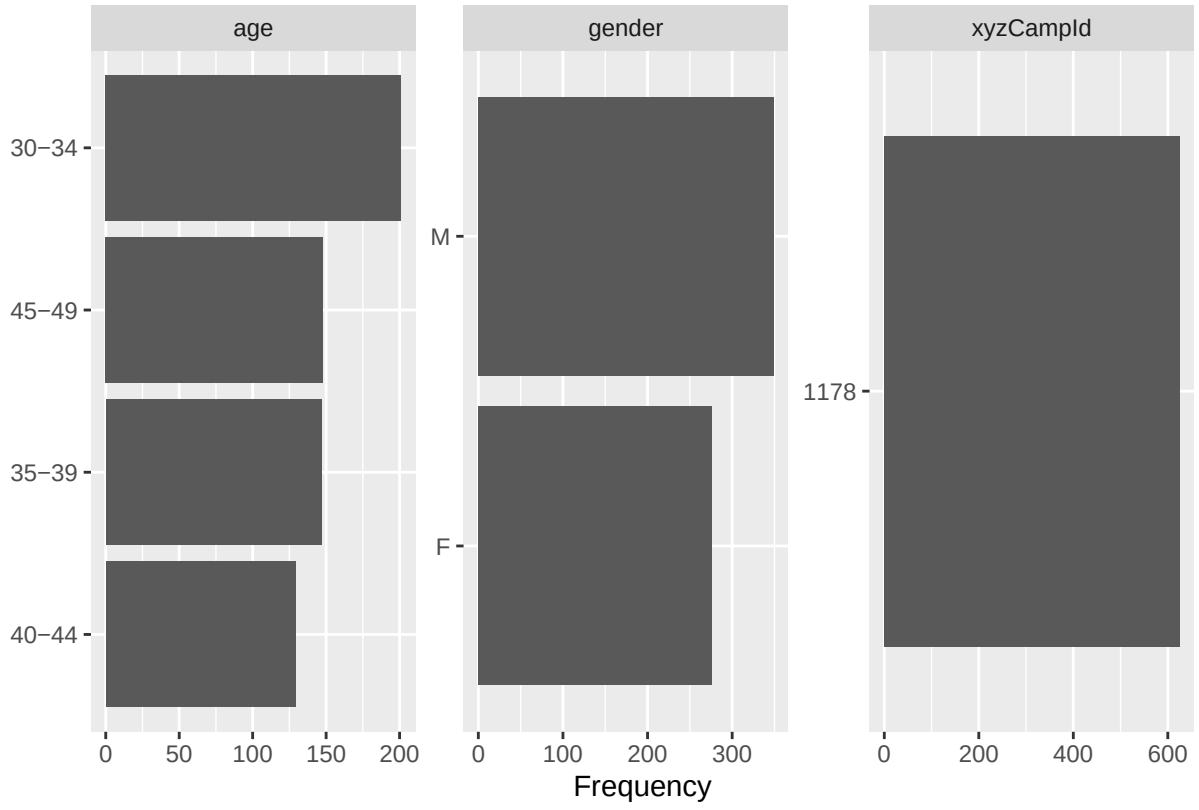
Looking for missing data

```
library(DataExplorer)
plot_missing(data1178)
```

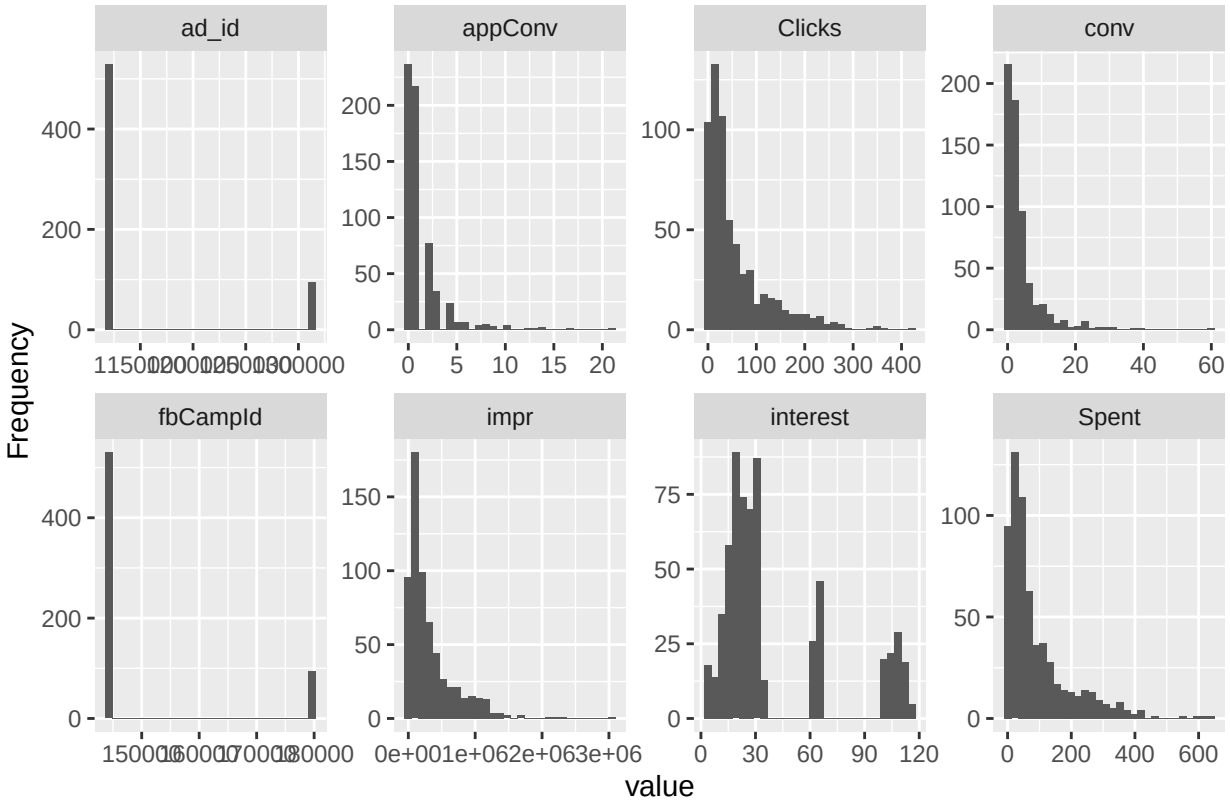


Distribution of data

```
library(DataExplorer)
plot_bar(data1178)
```

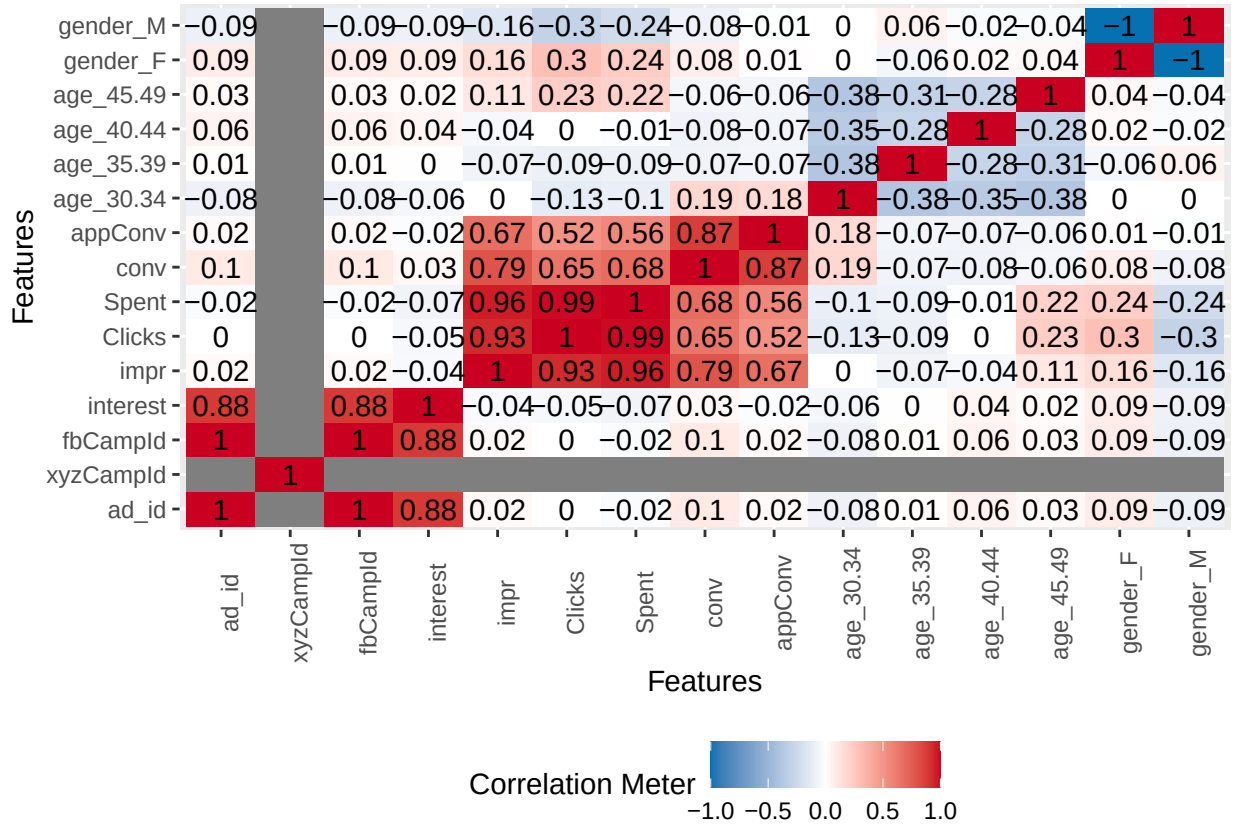



```
plot_histogram(data1178)
```



Correlation matrix for the 1178 campaign

```
library(DataExplorer)
plot_correlation(data1178)
```



Feature Engineering

We don't have the actual numbers at our disposal here, but as an example, we 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

```
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))
```

Cost Per Mile (CPM) is the cost of one thousand impressions. If the objective is ad exposure to increase brand awareness, this might be an important KPI to measure.

Create CPM

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

Glimpse of new variables

```
head(data1178)
```

```
##      ad_id xyzCampId fbCampId   age gender interest     impr Clicks  Spent conv
## 1 1121091      1178   144531 30-34      M       10 1194718    141 254.05   28
## 2 1121092      1178   144531 30-34      M       10  637648     67 122.40   13
```

```
## 3 1121094      1178   144531 30-34      M       10   24362      0   0.00    1
## 4 1121095      1178   144531 30-34      M       10  459690     50  86.33    5
## 5 1121096      1178   144531 30-34      M       10  750060     86 161.91   11
## 6 1121097      1178   144532 30-34      M       15   30068      1   1.82    1
##   appConv totConv conVal appConVal totConVal costPerCon ROAS  CPM
## 1      14      42    140      1400      1540       6.05 6.06 0.21
## 2       5      18     65       500       565       6.80 4.62 0.19
## 3       1       2      5       100       105       0.00  Inf 0.00
## 4       2       7     25       200       225      12.33 2.61 0.19
## 5       2      13     55       200       255      12.45 1.57 0.22
## 6       0       1      5         0         5       1.82 2.75 0.06
```

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.

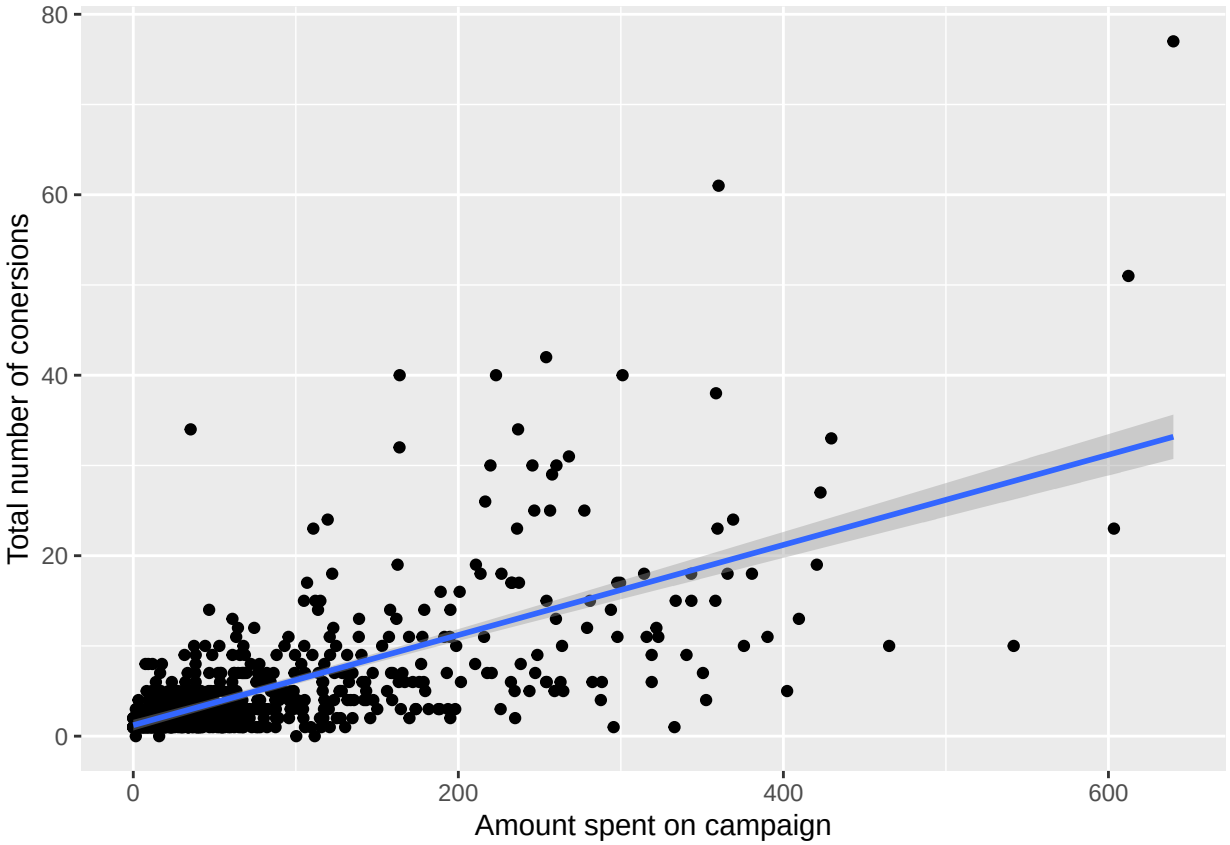
It's still a conversion, so we want it in there for the purposes of our aggregate statistics, but we do need to remember that it's there and consider what that might be doing as we work through our analyses.

Analysis of Campaign 1178

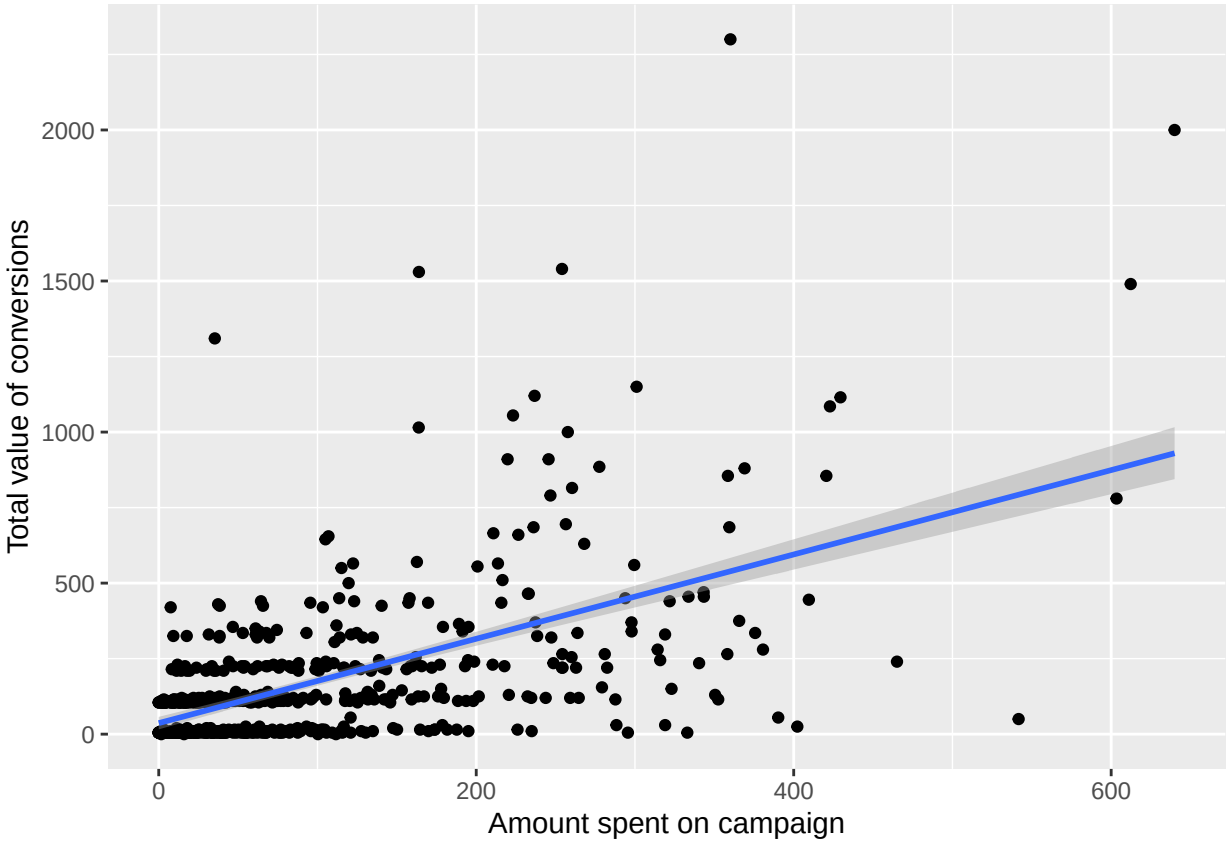
we'll assume that this is an e-commerce business that is purely focused on maximizing revenue.

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?

```
library(ggplot2)
ggplot(data1178, aes(Spent, totConv)) + geom_point() + geom_smooth(method = "lm") +
  labs(x = "Amount spent on campaign", y = "Total number of conersions")
```



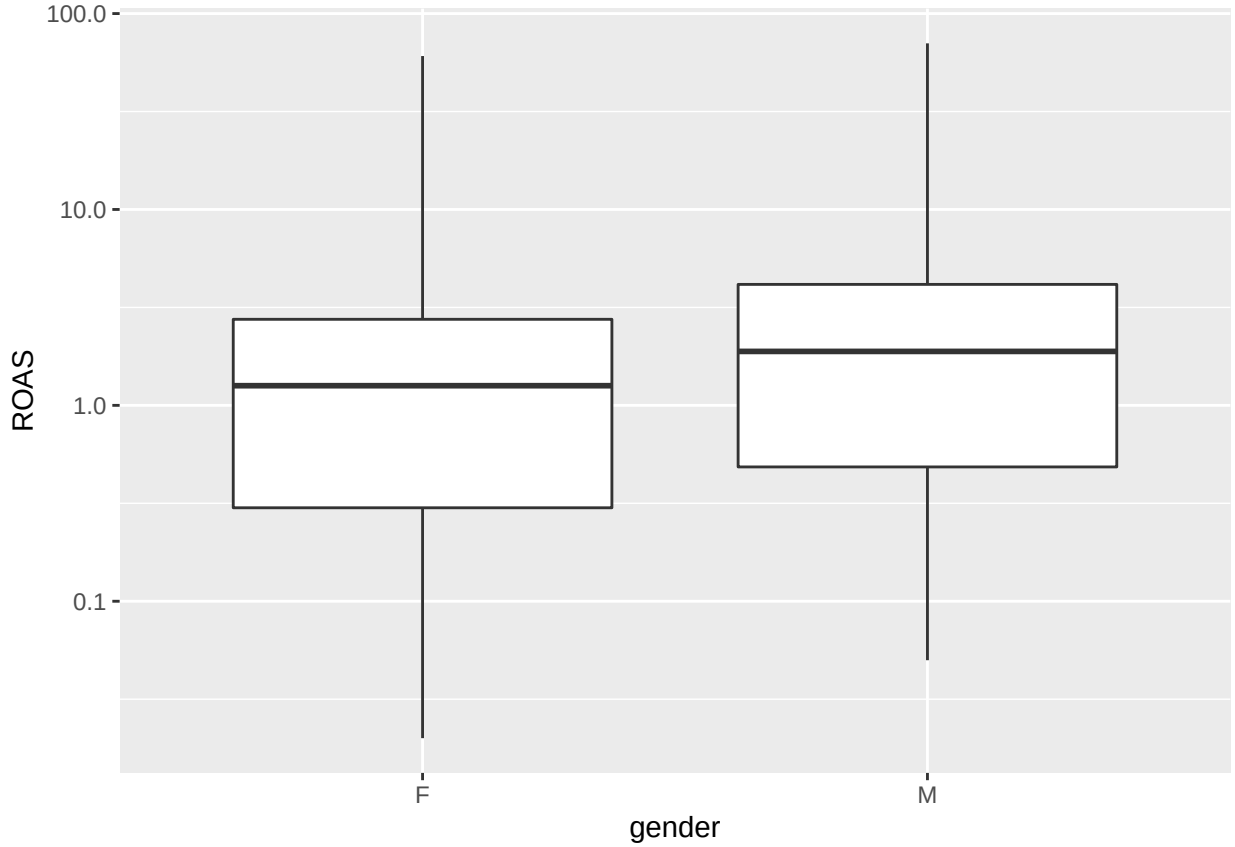

```
ggplot(data1178, aes(Spent, totConVal)) + geom_point() + geom_smooth(method = "lm") +
  labs(x = "Amount spent on campaign", y = "Total value of conversions")
```



At first glance, 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 have to dig deeper before we start recommending that we should just increase our advertising budget.

Splitting the data by gender

```
library(ggplot2)
ggplot(data1178, aes(gender, ROAS)) + geom_boxplot() + scale_y_log10()
```



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:

Wilcox test

```
library(ggplot2)
wilcox.test(ROAS ~ gender, data=data1178)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  ROAS by gender
## W = 38870, p-value = 3.397e-05
## alternative hypothesis: true location shift is not equal to 0
```

Analysis by Gender

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

```
## # A tibble: 2 x 3
##   gender medianROAS meanROAS
```

```
##   <chr>       <dbl>    <dbl>
## 1 F            1.23     2.82
## 2 M            1.88     4.50
```

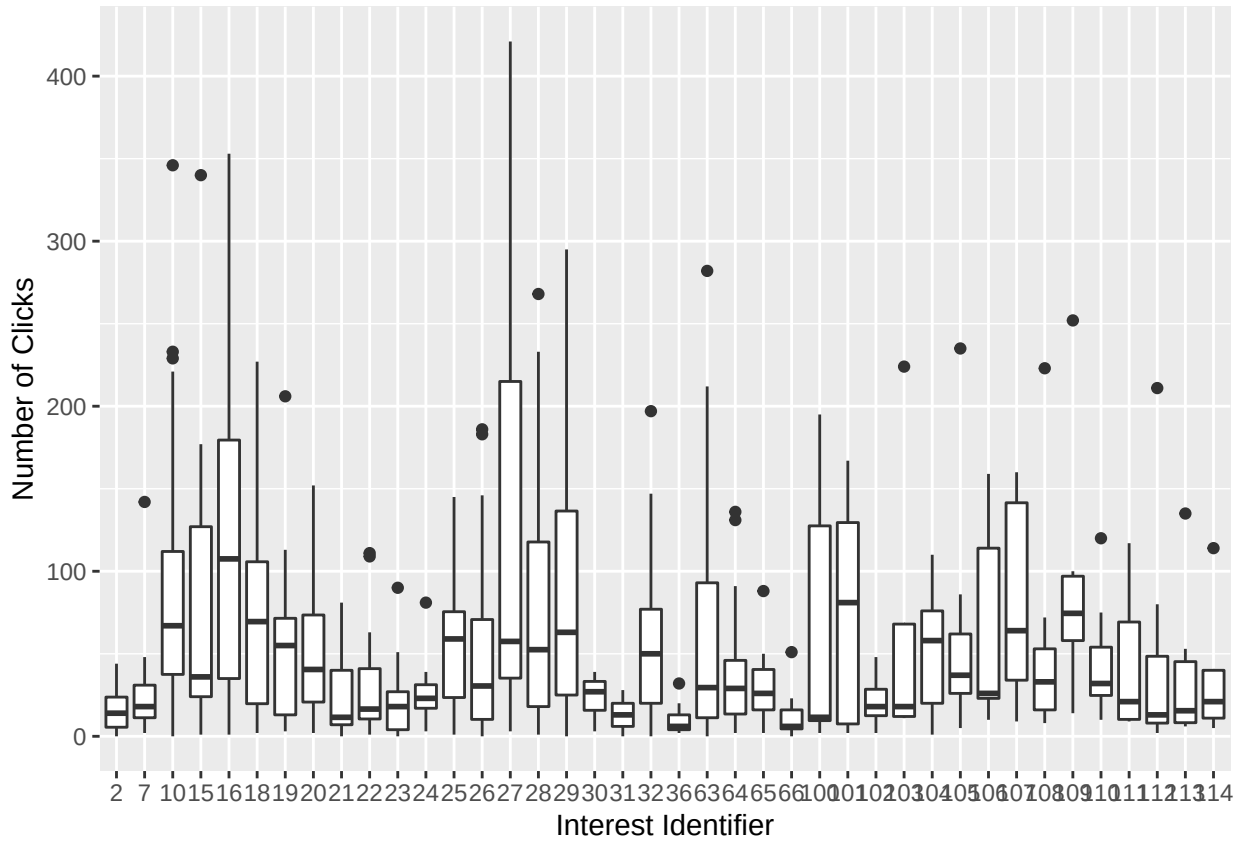
We observe that ROAS is higher for males than females and that this difference is statistically significant ($p < 0.01$).

In this case, while the median does give us a more accurate estimation of what our ROAS would be for a particular ad_ID, 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.

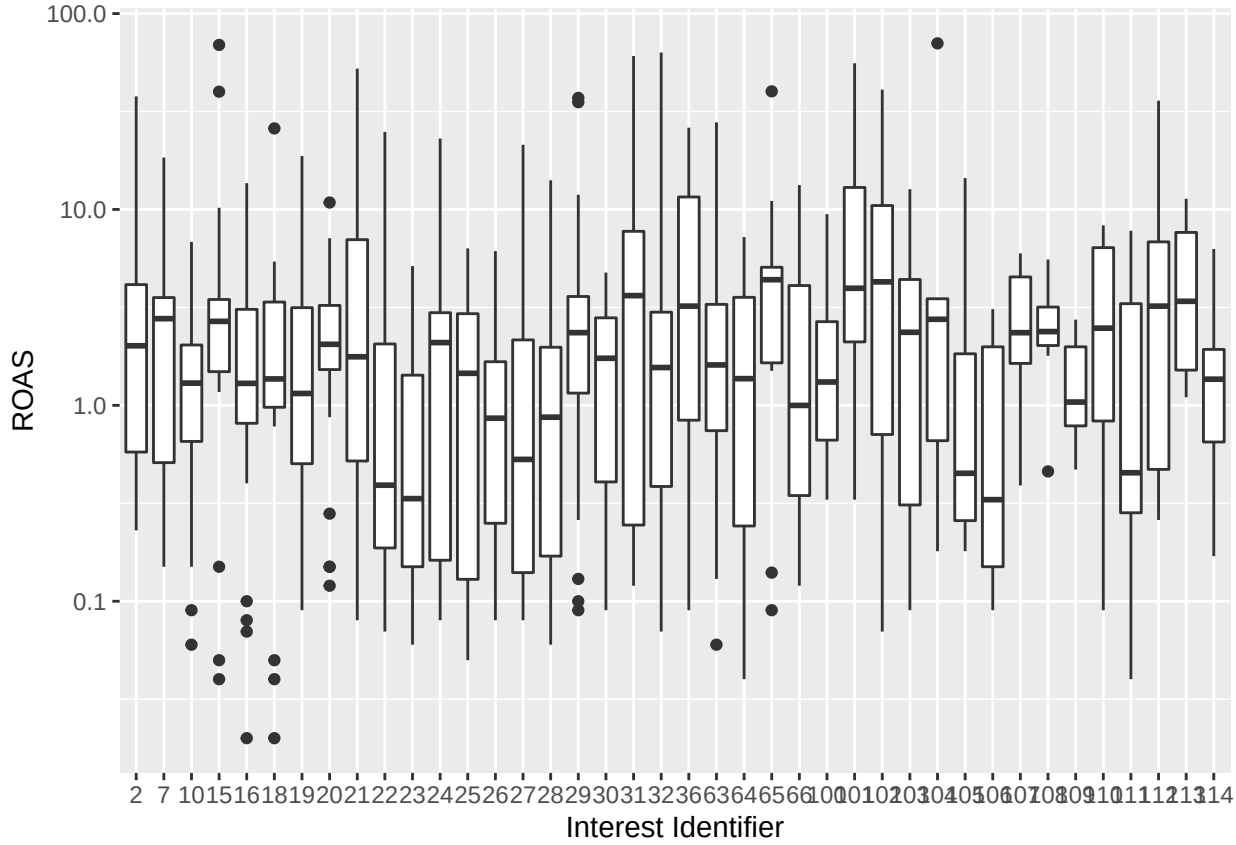
However, we have the data to go a lot more granular than this.

Splitting the data by Interest

```
# Boxplot - Clicks vs interests
ggplot(data1178, aes(as.factor(interest), Clicks)) + geom_boxplot() +
  labs(x = "Interest Identifier", y = "Number of Clicks")
```




```
# Boxplot - ROAS vs interests
data1178 %>%
  ggplot(aes(as.factor(interest), ROAS)) + geom_boxplot() + scale_y_log10() +
  labs(x = "Interest Identifier", y = "ROAS")
```



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

Analysis by Interest

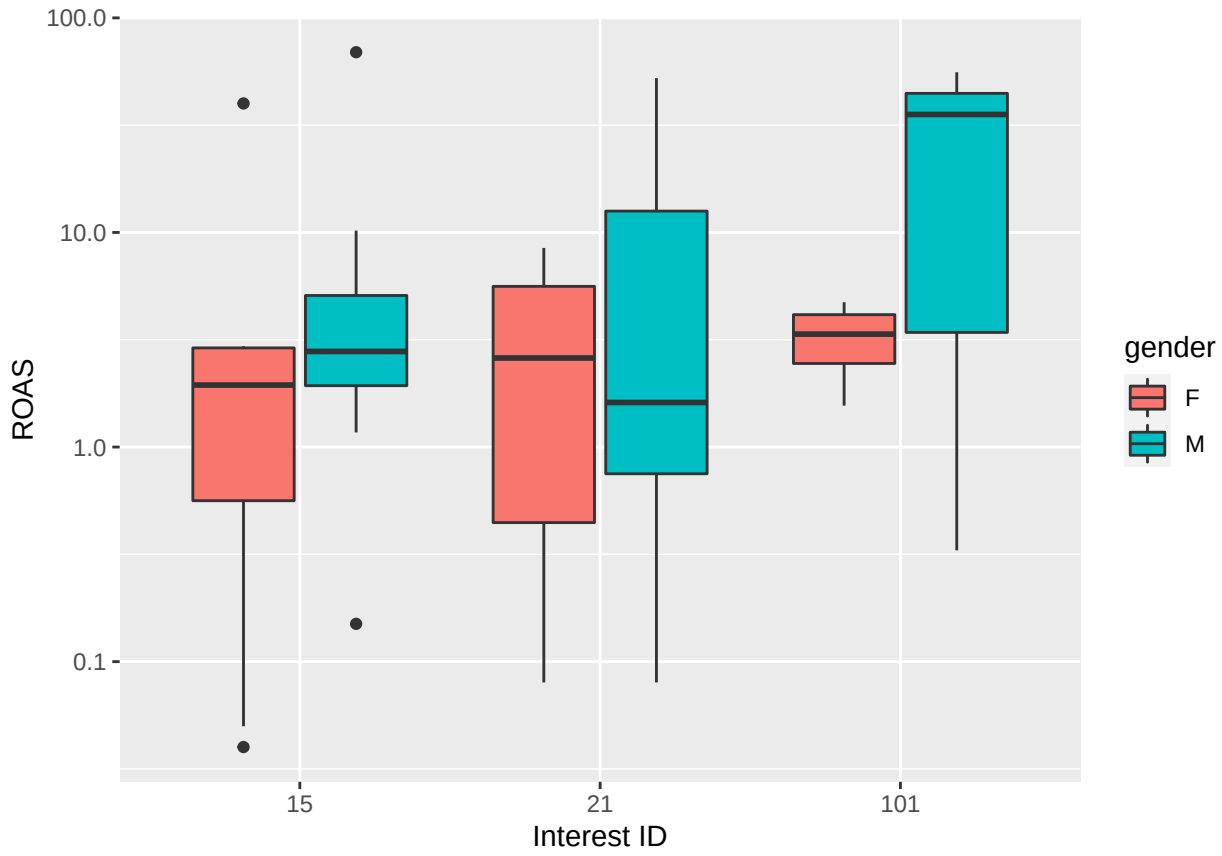
```
data1178 %>%
  select(interest, ROAS, Clicks) %>%
  group_by(interest) %>%
  filter(ROAS != 'Inf') %>%
  summarise(medianROAS = round(median(ROAS) ,2), meanROAS = round(mean(ROAS), 2), clicks = sum(Clicks)) %>%
  arrange(desc(meanROAS)) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 4
##    interest medianROAS meanROAS clicks
##       <int>      <dbl>    <dbl>  <int>
##  1      104       2.75    15.5     265
##  2      101       3.96    15.0     524
##  3      102       4.27    10.4     150
##  4       31       3.64     8.26    189
##  5      112       3.21     8.06    339
##  6       15       2.68     7.89   1554
##  7       36       3.21     7.38    126
##  8       65       4.38     7       343
##  9       21       1.77     6.34    493
## 10        2       2.08     5.41    306
```

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.

Analysis by gender for selected interests

```
data1178 %>%
  filter(interest == 101 | interest == 15 | interest == 21) %>%
  ggplot(aes(x = as.factor(interest), y = ROAS, fill = gender)) + geom_boxplot() + scale_y_log10() +
  labs(x = 'Interest ID', y = 'ROAS')
```




```
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))
```

```
## # A tibble: 6 x 5
## # Groups:   interest [3]
##   interest gender medianROAS meanROAS clicks
##      <int> <chr>       <dbl>    <dbl>  <int>
## 1      101 M           35.5     30.5      17
## 2       21 M            1.62     9.63    200
## 3       15 M            2.79     8.89    827
```

```
## 4       15 F            1.98     6.38    727
## 5       21 F            2.6      3.36    293
## 6      101 F            3.41     3.28    507
```

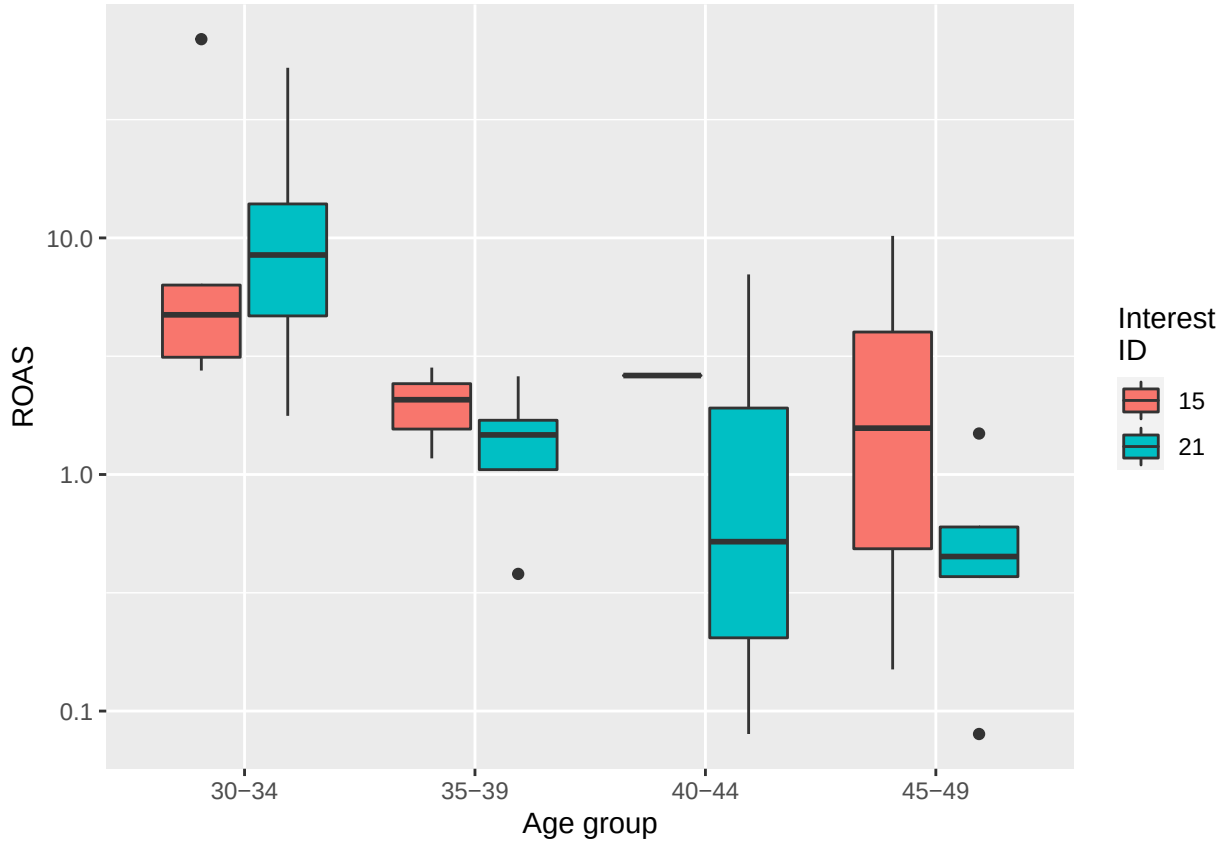
Looking at the results above, increasing our budget to display our ads to males with interest 101 might make a lot of sense as it's currently bringing in over £30 for every £1 spent. However, this is with a small number of clicks (17), so could just be due to chance.

Given this finding, it is tempting to recommend a modest increase in the budget for this demographic, only 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. Let's see if we can be even more targeted:

```
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() + scale_y_log1
  labs(x = 'Age group', y = 'ROAS') + scale_fill_discrete(name="Interest\nID")
```




```
data1178 %>%
  select(age, interest, gender, ROAS, Clicks) %>%
  group_by(age, interest) %>%
```

```r
  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))
```

```
## # A tibble: 7 x 5
## # Groups:   age [4]
##   age   interest medianROAS meanROAS clicks
##   <chr>    <int>      <dbl>    <dbl>  <int>
## 1 30-34       21      13.9     18.4      58
## 2 30-34       15       4.73    17.2     495
## 3 45-49       15       1.57     3.97    138
## 4 40-44       15       2.62     2.62     26
## 5 35-39       15       2.07     2.02    168
## 6 35-39       21       1.47     1.47     44
## 7 45-49       21       0.45     0.38     98
```

Result It looks like we're getting the best ROAS with 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 chance/noise.

Final thoughts This notebook is aimed at analyzing pay-per-click advertising to try and improve ROI from digital campaigns.

But it really is only a starting point. The right type of analysis and measures of success will be driven by business model and underlying marketing objectives.