Investigation

MP

29 4 2021

```r
library(tidyverse)
library(GGally)
library(lubridate)
setwd("D:/Python Projects/codechallenge_001")
```

```r
users <- read_csv("data/user_info.csv",
    col_types = cols(
      user_id = col_double(),
      country_id = col_character(),
      next_exam_type = col_character(),
      marketing_source = col_character(),
      signup_device = col_character(),
      signup_os = col_character(),
      activated = col_double(),
      register_date = col_datetime(format = "")
    )
  )
activations <- read_csv("data/code_activations.csv",
    col_types = cols(
      user_id = col_double(),
      code_activation = col_datetime(format = ""),
      access_start = col_datetime(format = ""),
      access_end = col_datetime(format = ""),
      days = col_double()
    )
  )
chapters <- read_csv("data/chapters_read.csv",
   col_types = cols(
      user_id = col_double(),
      created_at = col_datetime(format = ""),
      referer = col_character(),
      time_spent = col_double(),
      chapter_id = col_double(),
      subjects = col_character()
    )
 )
questions <- read_csv("data/questions_read.csv",
    col_types = cols(
      user_id = col_double(),
      question_id = col_double(),
      answer_id = col_double(),
      collection_id = col_double(),
```

```
      created_at = col_datetime(format = ""),
      was_answer_correct = col_double(),
      is_completed = col_double(),
      has_given_up = col_double(),
      time_spent = col_double(),
      used_highlight = col_double(),
      used_case_highlight = col_double(),
      used_hint_as_help = col_double(),
      used_hint_as_nonsense = col_double(),
      chapter_ids = col_character()
  )
)
```

Hello reader, I will take you through my journey on the data. As this is an exploratory data analysis, I will not revise this afterwards, but show you my way of working through this dataset.

After reading it in, I want to get a feel for the data. I usually do this by looking at summary statistics of individual columns.

Let's start by looking at the users first, as they are the most important stakeholder for us. We start by looking at where they are coming from.

```
users %>%
  group_by(country_id) %>%
  summarise(n=n()) %>%
  arrange(-n)
```

```
## # A tibble: 205 x 2
##    country_id     n
##    <chr>      <int>
##  1 NULL       10086
##  2 US          5192
##  3 IN          2010
##  4 LV          1908
##  5 RO          1478
##  6 AU          1435
##  7 DE          1184
##  8 UA           837
##  9 PK           756
## 10 MX           751
## # ... with 195 more rows
```

The first things we notice, is that there are 205 distinct countries. Random fact about me: I am kinda a geo-nerd. So I know that there are only 195 countries (because I learned all the flags at some point…). Well, thats interesting isn't it? We learn that for 10k cases we don't have a country information. We could check later if these are users who don't end up having code activations. From my experience those cases are either incomplete profiles or cases where tracking is disabled -> GDPR and such. But they could also indicate a bug or a path through the onboarding which is not tested enough by the product team.

Later, when we figure out how to capture the US market, I would focus down the analysis on the US customers. It is rare to see that you really can transfer learnings between continents. "transfer learnings" in an analytical as well as ML sense.

```r
users %>%
  filter(country_id=="VA")
```

```
## # A tibble: 1 x 8
##   user_id country_id next_exam_type marketing_source signup_device signup_os
##     <dbl> <chr>      <chr>          <chr>            <chr>         <chr>
## 1 1084797 VA         NULL           NULL             mobile        Android
## # ... with 2 more variables: activated <dbl>, register_date <dttm>
```

We have someone from Vatican City who signed up via his Android phone. Funny, huh?

```r
users %>%
  group_by(marketing_source) %>%
  summarise(n=n(),
            us_n = sum(country_id=="US", na.rm=TRUE)) %>%
  ungroup() %>%
  mutate(p = round(n/sum(n)*100,1),
         us_p = round(us_n/sum(us_n)*100,1)) %>%
  arrange(-n)
```

```
## # A tibble: 16 x 5
##    marketing_source      n  us_n     p  us_p
##    <chr>             <int> <int> <dbl> <dbl>
##  1 NULL              24538  2774  56    53.4
##  2 facebook           6415   613  14.7  11.8
##  3 friends            4567   567  10.4  10.9
##  4 advertisement      2417   311   5.5   6
##  5 google             1813   396   4.1   7.6
##  6 press_online       1339   242   3.1   4.7
##  7 university          908    68   2.1   1.3
##  8 conference          638    92   1.5   1.8
##  9 other               475    75   1.1   1.4
## 10 youtube             324    53   0.7   1
## 11 students_work       287     0   0.7   0
## 12 student_committee    27     0   0.1   0
## 13 library              20     0   0     0
## 14 flyer                13     0   0     0
## 15 amazon                1     1   0     0
## 16 bookstore             1     0   0     0
```

Looking at marketing sources now, we could interpret NULL values as "organic". We also see a steep decline here. I am surprised that "university" is so far down the list, as the tool is fairly useful for students.

Looking at US again, we see a roughly similar distribution versus the world.

Next I want to have a look into the onboarding process from a tech perspective. Looking at activation rates of different divices could shed some light on the quality of the products accessability across platforms.

```r
users %>%
  group_by(signup_device) %>%
  summarise(n=n(),
            activated_sum = sum(activated),
            activated_p = round(activated_sum/n*100,2))
```

```
## # A tibble: 4 x 4
##   signup_device     n activated_sum activated_p
##   <chr>         <int>         <dbl>       <dbl>
## 1 desktop       14290         12890        90.2
## 2 mobile        22370         17715        79.2
## 3 NULL           4923          3097        62.9
## 4 tablet         2200          1886        85.7
```

Desktops have by far the highest activation rate, followed by tablet and mobile, each with a 5% performance drop to the afore-mentioned. We also detected another case of NULLs, potentially of tracking reasons. Just based on this, I would use the signup-device as a parameter for any marketing strategy, it clearly has an influence on conversion. Noteable is also that most customers are signing up via mobile. My first questions are: Are we advertising on mobile more than on other platforms? If not, and marketing is spend evenly, our users tend to sign up via mobile. And if that's the case, let's get the product team together and rethink the onboarding experience for mobile!

Also for the interested reader, I am now roughly 45 mins in, I love taking my time understanding what's going on and to be fair, the narrative is also taking quite some time :)

```
table(users$next_exam_type)
```

```
## 
##                anatomy-embryology               behavioral-sciences 
##                                14                                 2 
##                      biochemistry           histology-cell-biology 
##                                 2                                 4 
##                      microbiology                      neuroscience 
##                                 7                                 7 
##                              NULL                         pathology 
##                             40204                                13 
##                      pharmacology                        physiology 
##                                 4                                 5 
## shelf-adult-ambulatory-medicine             shelf-anesthesiology 
##                                46                                79 
##             shelf-family-medicine           shelf-internal-medicine 
##                                98                               260 
##                   shelf-neurology     shelf-obstetrics-gynecology 
##                               167                               133 
##                  shelf-pediatrics                  shelf-psychiatry 
##                               160                               110 
##                     shelf-surgery                            step-1 
##                               306                              1173 
##                            step-2                            step-3 
##                               813                               176
```

Looking at chapters now. First thing I feel like knowing is if the subjects on the chapters_ids are consistent or fluctuate. My thought was to split the subjects up to dive deeper into them, but they need to be consistent for it to make any sense on chapter_id level.

```
chapters %>%
  group_by(chapter_id, subjects) %>%
  summarise(visits=n())
```

```
## `summarise()` has grouped output by 'chapter_id'. You can override using the `.groups` argument.
```

```
## # A tibble: 766 x 3
## # Groups:   chapter_id [766]
##    chapter_id subjects                                                     visits
##         <dbl> <chr>                                                         <int>
##  1          0 <NA>                                                              1
##  2          1 ['Pediatrics', 'Otolaryngology', 'Infectiology']                510
##  3          2 ['Pneumology', 'Infectiology']                                  486
##  4          3 ['Neurology', 'Infectiology', 'General surgery']                981
##  5          4 ['General surgery', 'Infectiology']                             213
##  6          6 ['Obstetrics', 'Urology', 'Pneumology', 'Gynecology', 'Oph~   1365
##  7         14 ['Infectiology', 'Hygiene,microbiology,virology', 'Gastr~    344
##  8         16 ['Gastroenterology', 'Hygiene,microbiology,virology', 'I~    645
##  9         17 ['Infectiology', 'Neurology', 'Dermatology']                  1648
## 10         18 ['Infectiology']                                               392
## # ... with 756 more rows
```

```
chapters$chapter_id %>% unique() %>% length()
```

```
## [1] 766
```

Good news! Chapter subjects don't change, otherwise we would have found duplicates in the chapter_ids.

Seeing the subjects listed like this makes me want to show them as graphs, but I havn't done it in ages. What I will do now is cleaning up the subjects, e.g. create a new table for chapters alone with a row being a combination of chapter_id and a single subject. This will be of tremendous value for any analysis to follow.

note: I am dealing here with a typical `['','','']` list structure. It always feels like they are more home in Python than they are in R. So I am treating them as strings, removing their formatting before splitting them apart.

```
chapter_subject_mapper = chapters %>%
  group_by(chapter_id) %>%
  mutate(rank = row_number()) %>% # running index
  ungroup() %>%
  filter(rank==1) %>% # select the first entry
  select(chapter_id, subjects) %>%
  mutate(subjects = str_replace_all(subjects,"\\[|\\]|'","")) %>% # remove brackets and single quotatio
  separate_rows(subjects, sep = ",") %>% # split by comma into new rows
  filter(subjects!="") %>% # remove parsing errors
  rename(subject=subjects) %>% # renaming
  mutate(subject = tolower(subject)) # lowercasing
chapter_subject_mapper
```

```
## # A tibble: 1,673 x 2
##    chapter_id subject
##         <dbl> <chr>
##  1        274 "occupational medicine,social medicine"
##  2        274 " pneumology"
##  3        269 "imaging,radiotherapy,radiation protection"
##  4        269 " pneumology"
##  5        622 "infectiology"
##  6        622 " otolaryngology"
##  7        622 " pediatrics"
```

```
## 8          937 "emergency medicine"
## 9          937 " anesthesiology"
## 10         937 " pneumology"
## # ... with 1,663 more rows
```

In the chunk above I take the first entry of all chapter_ids, drop the rest as well as all cols which are not relevant and then turn every subject into an individual row.

Now we could easily answer a question like which subject has the highest average reading time. Or which subject has the lowest amount of readers. Let's do that quickly!

```
chapters %>%
  left_join(chapter_subject_mapper) %>%
  group_by(subject) %>%
  summarise(number_reads = n(),
            average_reading_time = mean(time_spent, na.rm=TRUE),
            number_of_readers = length(unique(user_id)))
```

```
## Joining, by = "chapter_id"
```

```
## # A tibble: 92 x 4
##    subject                          number_reads average_reading_t~ number_of_reade~
##    <chr>                                   <int>              <dbl>            <int>
##  1 " abdominal surgery"                    33191               53.4             3730
##  2 " anatomy"                                 27                4                 23
##  3 " anesthesiology"                       15067               42.0             3340
##  4 " biochemistry"                             7              NaN                  4
##  5 " cardiology and angiology"             66420               46.9             7479
##  6 " child and adolescent psyc~             2504               58.1              886
##  7 " clinical-pathological con~              708               55.0              248
##  8 " clinical chemistry,labor~            31062               52.6             4408
##  9 " clinical pharmacology / p~            38588               50.9             5620
## 10 " dermatology"                          38113               47.9             5037
## # ... with 82 more rows
```

Time Investment 75 mins so far.

Let's define some performance KPIs and apply them to different origins of customers.

KPI definitions

Interaction Volumne

With the Interaction Volumne KPIs we want to express a customers behavior in the first 5 days, a foundation for predicting the likelyhood use the tool after the free trial.

We start by defining up until which point the free trial worked, ultimately we should exclude those cases where there was an activation within those 5 days, but I skip that for time sake.

```
user_regis = users %>%
  mutate(register_date = date(register_date),
         free_trial_until = register_date+5) %>%
  select(user_id, register_date, free_trial_until)
user_regis
```

```
## # A tibble: 43,783 x 3
##    user_id register_date free_trial_until
##      <dbl> <date>        <date>
##  1 1079934 2017-10-01    2017-10-06
##  2 1080081 2017-10-01    2017-10-06
##  3 1080241 2017-10-01    2017-10-06
##  4 1080395 2017-10-02    2017-10-07
##  5 1080507 2017-10-02    2017-10-07
##  6 1080765 2017-10-02    2017-10-07
##  7 1080839 2017-10-03    2017-10-08
##  8 1080864 2017-10-03    2017-10-08
##  9 1081069 2017-10-03    2017-10-08
## 10 1081363 2017-10-04    2017-10-09
## # ... with 43,773 more rows
```

Next, we combine it with their chapters information. I want to derive 3 metrics here: Number of chapters they "started", number of chapters they have not spend any time on, and the sum of time_spent. All 3 should grant us insights. Note that time_spent is only available for mobile users.

```r
chapter_KPIs = user_regis %>%
  left_join(chapters, by="user_id") %>%
  mutate(chapter_c_date = date(created_at)) %>%
  filter(chapter_c_date <= free_trial_until) %>%
  group_by(user_id) %>%
  summarise(n_chapters = n(),
            n_spent_time = n_chapters - sum(is.na(time_spent)),
            sum_time_spent = sum(time_spent, na.rm = TRUE))
chapter_KPIs
```

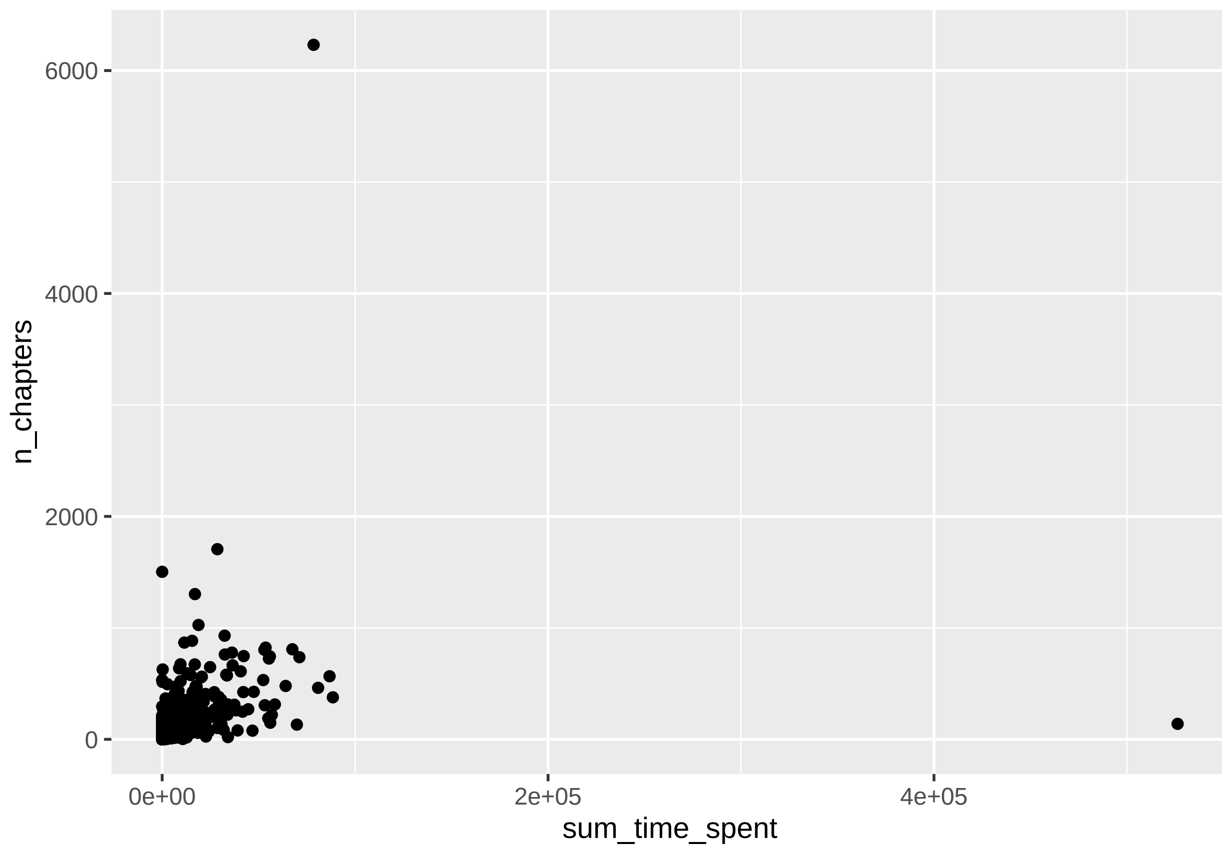
```
## # A tibble: 18,452 x 4
##    user_id n_chapters n_spent_time sum_time_spent
##      <dbl>      <int>        <int>          <dbl>
##  1 1079921          1            0              0
##  2 1079924          2            0              0
##  3 1079926         48            0              0
##  4 1079931          7            7            129
##  5 1079932          1            0              0
##  6 1079933         11            0              0
##  7 1079934         16            0              0
##  8 1079939          1            0              0
##  9 1079940         22           21            819
## 10 1079941         20            4             25
## # ... with 18,442 more rows
```

We immediately see cases where there are a lot of chapters opened, but no time spent at all.

Now we look into their behavior on questions, before combining them back together and aggregate on different levels.

We can see that there are two outlines in the dataset, so I will ignore those two cases going on.

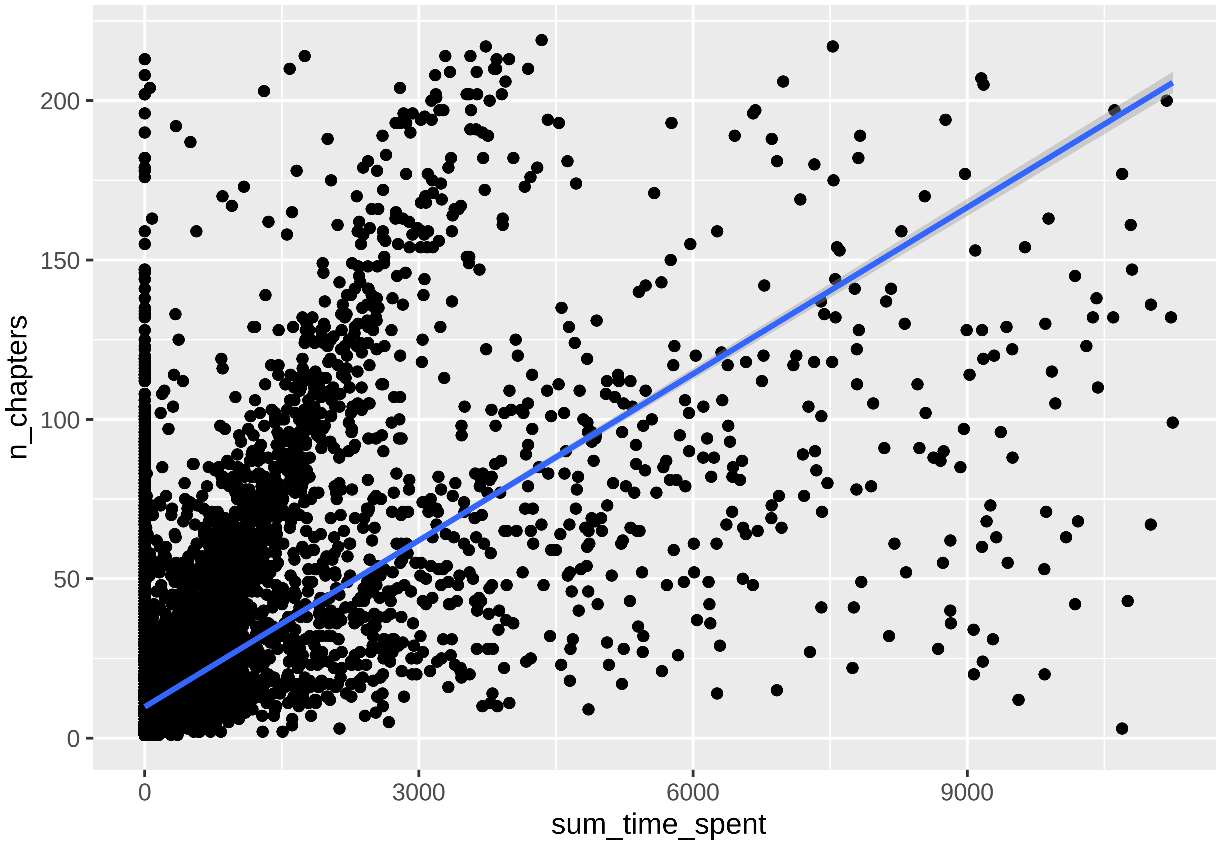
```r
chapter_KPIs %>%
  ggplot(aes(x=sum_time_spent,y=n_chapters)) +
  geom_point()
```




```r
chapter_KPIs = chapter_KPIs %>%
  filter(sum_time_spent <= 11300 & n_chapters <= 220) # 99th percentile
```

```r
chapter_KPIs %>%
  ggplot(aes(x=sum_time_spent,y=n_chapters)) +
  geom_point()+
  geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



I looked at time spend versus the number of chapters to see if I infer missing datapoints for time_spent for the non-mobile users. Truth be told, not sure if I should infer from mobile behavior to non-mobile behavior in the first place. But looking at the scatter plot above there is a really interesting insight: It looks like two overlapping distributions! This cloud of data points above the linear abline looks like this trade-off between chapters and time spent has a third, yet unaccounted variable.

Just as a sidenote, the questions offer a tremendous amount of insights. Hint-usage, answer-rate, difficulty... and we can link it all back to the chapters and their content. That's a holy grail for analytical purposes.

```
question_KPIs = user_regis %>%
  left_join(questions, by="user_id") %>%
  mutate(question_c_date = date(created_at)) %>%
  filter(question_c_date <= free_trial_until) %>%
  group_by(user_id) %>%
  summarise(n_questions = n(),
            given_up_flag = any(has_given_up==1),
            right_answer_rate = round(mean(was_answer_correct, na.rm = TRUE)*100,2),
            sum_time_spent_q = sum(time_spent, na.rm = TRUE))
question_KPIs
```

```
## # A tibble: 12,683 x 5
##    user_id n_questions given_up_flag right_answer_rate sum_time_spent_q
##      <dbl>       <int> <lgl>                     <dbl>            <dbl>
##  1 1079919          27 FALSE                      7.41             9500
##  2 1079921           6 FALSE                     16.7               991
##  3 1079923           1 FALSE                      0               16069
```

```
##   4 1079924           1 FALSE                     0                48
##   5 1079926           4 FALSE                    50               415
##   6 1079934          16 FALSE                    37.5            1126
##   7 1079939           1 FALSE                     0               102
##   8 1079941           1 FALSE                     0                24
##   9 1079942           1 FALSE                     0                90
##  10 1079943          13 FALSE                    46.2             491
## # ... with 12,673 more rows
```

The questions also do have a time spent columns, my instinct is to sum it up with the time spend of chapters to get a fuller picture.

I created a flag if they ever given up, counted questions, calculated the share of right answers and calculated time spent on questions.

Now I bring them together!

```
interaction_volume = chapter_KPIs %>%
  left_join(question_KPIs, by = "user_id") %>%
  mutate(time_spent_trial = sum_time_spent+sum_time_spent_q)
interaction_volume
```

```
## # A tibble: 18,173 x 9
##    user_id n_chapters n_spent_time sum_time_spent n_questions given_up_flag
##      <dbl>      <int>        <int>          <dbl>       <int> <lgl>
##  1 1079921          1            0              0           6 FALSE
##  2 1079924          2            0              0           1 FALSE
##  3 1079926         48            0              0           4 FALSE
##  4 1079931          7            7            129          NA NA
##  5 1079932          1            0              0          NA NA
##  6 1079933         11            0              0          NA NA
##  7 1079934         16            0              0          16 FALSE
##  8 1079939          1            0              0           1 FALSE
##  9 1079940         22           21            819          NA NA
## 10 1079941         20            4             25           1 FALSE
## # ... with 18,163 more rows, and 3 more variables: right_answer_rate <dbl>,
## #   sum_time_spent_q <dbl>, time_spent_trial <dbl>
```

The lack of adaptation of the questions feature makes it hard to judge the meaningfulnis of those KPIs. For me, every NA in these KPIs is also telling a story. The fact, that a customer has not discovered or used the questions feature is a valid observation which could have a big impact. You always have to question if your missing data is actual data!

timespent: 1:45h

Analysis for interaction volumn in trial period

Given the KPIs we just developed, we are able to answer some questions. First off, let's try to find a link between the first 5 days of activity and the number of code activations throughout their lifetime.

```
activation_counts = activations %>% group_by(user_id) %>% count()
df = interaction_volume %>%
  left_join(activation_counts, by="user_id") %>%
  select(n_chapters, n_questions,right_answer_rate, given_up_flag, time_spent_trial, n)
df
```

```
## # A tibble: 18,173 x 6
##    n_chapters n_questions right_answer_rate given_up_flag time_spent_trial     n
##         <int>       <int>             <dbl> <lgl>                    <dbl> <int>
##  1          1           6              16.7 FALSE                      991     1
##  2          2           1               0   FALSE                       48    NA
##  3         48           4              50   FALSE                      415    NA
##  4          7          NA              NA   NA                          NA    NA
##  5          1          NA              NA   NA                          NA    NA
##  6         11          NA              NA   NA                          NA     1
##  7         16          16              37.5 FALSE                     1126     2
##  8          1           1               0   FALSE                      102    NA
##  9         22          NA              NA   NA                          NA    NA
## 10         20           1               0   FALSE                       49    NA
## # ... with 18,163 more rows
```

This dataframe above could be a foundation for modeling. A couple of steps need to be thought of: 1. The scale of the numbers. e.g. time_spent_trial ranges significantly. One could set boundaries or categories beforehand. e.g. "low time spent" "mid time spent" "high time spent"... Or you use a model which tries to capture it. 2. Infer further features. time spent = NA might be a more meaningful feature then the numeric value of time_spent_trial. (remember, this is only valid for mobile users) 3. Take the time and go through features and combinations on the search for significance.

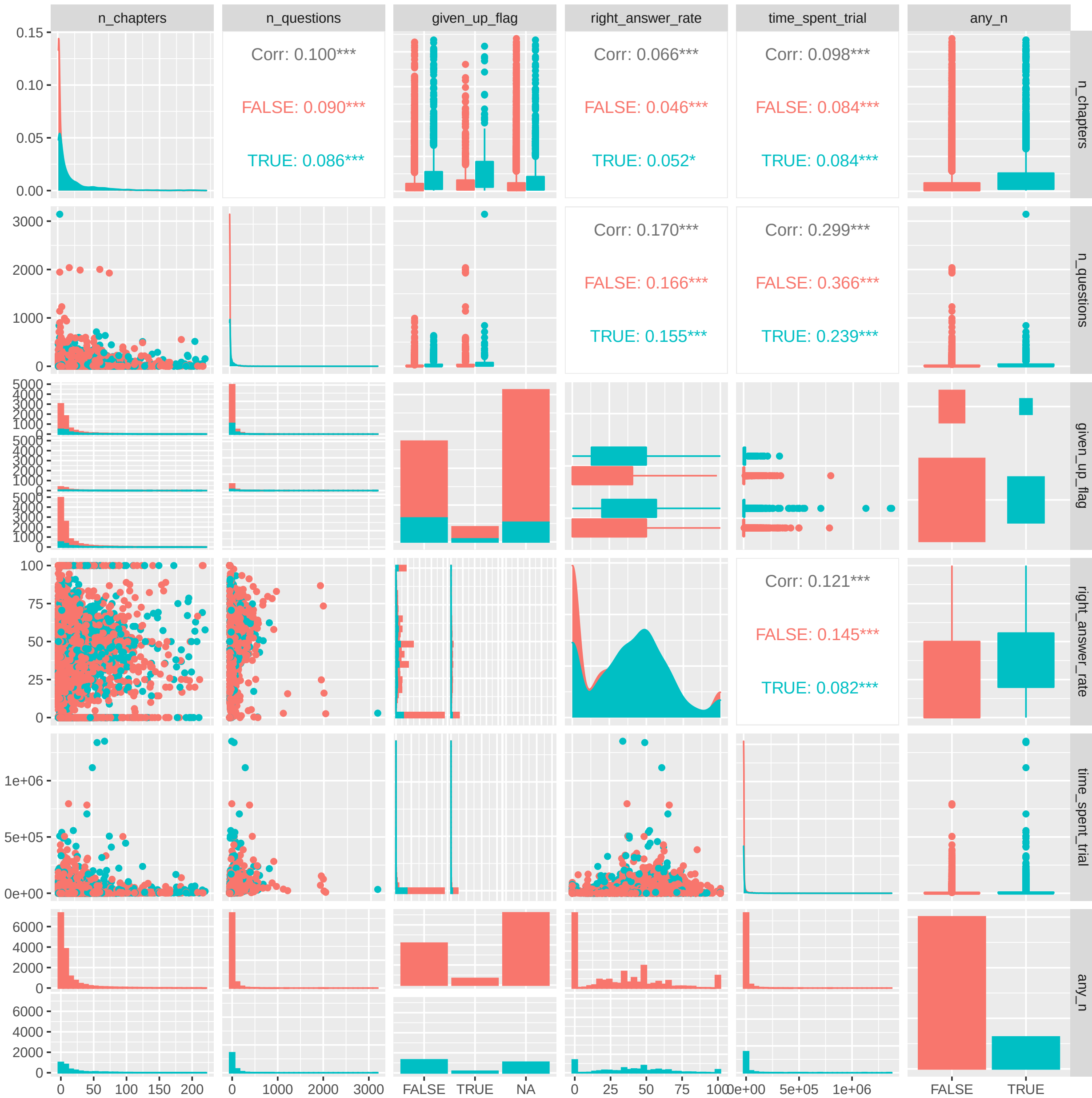
I want to quickly get a feeling for activations, before I wrap things up.

```
interaction_volume %>%
  left_join(activation_counts, by="user_id") %>%
  mutate(any_n = !is.na(n)) %>%
  group_by(any_n) %>%
  summarise(chapters_avg = mean(n_chapters, na.rm = TRUE),
            questions_avg = mean(n_questions, na.rm = TRUE),
            RAR_avg = mean(right_answer_rate, na.rm = TRUE),
            given_up_avg = mean(given_up_flag, na.rm = TRUE))
```

```
## # A tibble: 2 x 5
##   any_n chapters_avg questions_avg RAR_avg given_up_avg
##   <lgl>        <dbl>         <dbl>   <dbl>        <dbl>
## 1 FALSE         13.3          26.3    30.9        0.136
## 2 TRUE          22.0          43.4    38.8        0.111
```

Just by looking at some aggregated numbers, we figure out that interaction in terms of number of chapters and questions will separate those who later activate. Yet the right_answer rate and given up ratio doesn't seem to indicate big differences.

```
interaction_volume %>%
  left_join(activation_counts, by="user_id") %>%
  mutate(any_n = !is.na(n)) %>%
  select(n_chapters, n_questions, given_up_flag, right_answer_rate, time_spent_trial, any_n) %>%
  ggpairs(aes(fill=as.factor(any_n),color=as.factor(any_n)))
```



To wrap things up, here is a really bright graph of some selected features from within the first 5 days in relation to each other and split by if they will ever activate or not. Are there any learnings here? Well, the same learnings as before, just wrapped into an overwhelming plot :) It would be something you print on canvas and frame it, not something you would actually use to come to conclusions with. Anyway, I felt this doc needs some more plotting and color, so here we are.

final words

This is where I will end. I worked a total of ~3 hours on it and I did not spend any time on trying to build machine learning models. Why you may ask? You need to build up understanding before your train any model. This understanding, if done correctly can be of great benefit for the whole company. Your feature engineering, or the creation of new KPIs can have a long lasting impact. It all depends dramatically if you, the data scientist, have the business case in mind, or just focus on data.