# HarvardX: Project KickStarter

Arnaud Lemaire

Feb 19th, 2022

## Introduction

The goal is to predict the funding a Kickstarter project will receive based on its setup determinants.
For instance we will investigate project categories, funding time of the year, duration, goal, and country. We will use RMSE (Root-Mean-Square-Error) to build and validate the predictions of our model.

## Setup

First we download the dataset (originally found on Kaggle) and segregate for training and validation.

```r
#########################################################
# SETUP
#########################################################

if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
library(caret)
library(data.table)
library(tidyverse)

# Kickstarter Projects dataset: https://www.kaggle.com/kemical/kickstarter-projects
data <- read.csv("https://raw.githubusercontent.com/ArnaudCode/KickStarter/main/ks-projects-201801.csv")

# keeping only target records of project success or failure
# data <- subset(data, state %in% c("successful", "failed") & usd_goal_real >= 100 & usd_goal_real <= 1

# keeping output steady
set.seed(881202)

# leaving 10% of dataset for validation
test_index <- createDataPartition(y = data$ID, times = 1, p = 0.10, list = FALSE)
training <- data[-test_index,]
validation <- data[test_index,]
rm(data, test_index)
```

## Dataset

Then let's explore basic inquiries:
- 379k records: fundings ~ projects
- 15 dimensions: ID, name, category (sub), main category, currency, deadline, goal, launched date, pledged amount, state (status), backers, country, pledged value in USD, real pledged value in USD, goal in USD

```
############################################################
# DATASET
############################################################

# Q1 How many rows and columns?
dim(training)
```

```
## [1] 340793     15
```

```
# Q2 How many projects succeeded, failed, or were canceled?
count(training, state)
```

```
##        state      n
## 1   canceled  34902
## 2     failed 177853
## 3       live   2499
## 4 successful 120700
## 5  suspended   1640
## 6  undefined   3199
```

```
# Q3 How many different main and sub categories ?
n_distinct(training$main_category)
```

```
## [1] 15
```

```
n_distinct(training$category)
```

```
## [1] 159
```

```
# Q4 How many different countries?
n_distinct(training$country)
```

```
## [1] 23
```

```
# Q5 Which project received the highest pledge ?
training[which.max(training$pledged), c(2, 7, 9)]
```

```
##                                                      name  goal  pledged
## 157271 Pebble Time - Awesome Smartwatch, No Compromises 5e+05 20338986
```

```
# Q6 Which project had the most backers ?
training[which.max(training$backers), c(2, 11)]
```

```
##                      name backers
## 187653 Exploding Kittens  219382
```

```
# Q7 Which project had the highest pledged to backers ratio ?
training[which.max(training$pledged / training$backers), c(2, 7, 9, 11)]
```

```
##                               name goal pledged backers
## 170 STREETFIGHTERZ WHEELIE MURICA 6500     555       0
```

# Preparation

A few changes to make data proper for further analysis:
- duration: calculate funding duration (launch to deadline)
- month: determine month deadline
- funded_usd: simplify field referencing instead of "USD_pledged_real"
- goal_usd: simplify field referencing instead of "USD_goal_real"
- then remove un-necessary dimensions
Note: we are going to work with values in USD to ensure comparability

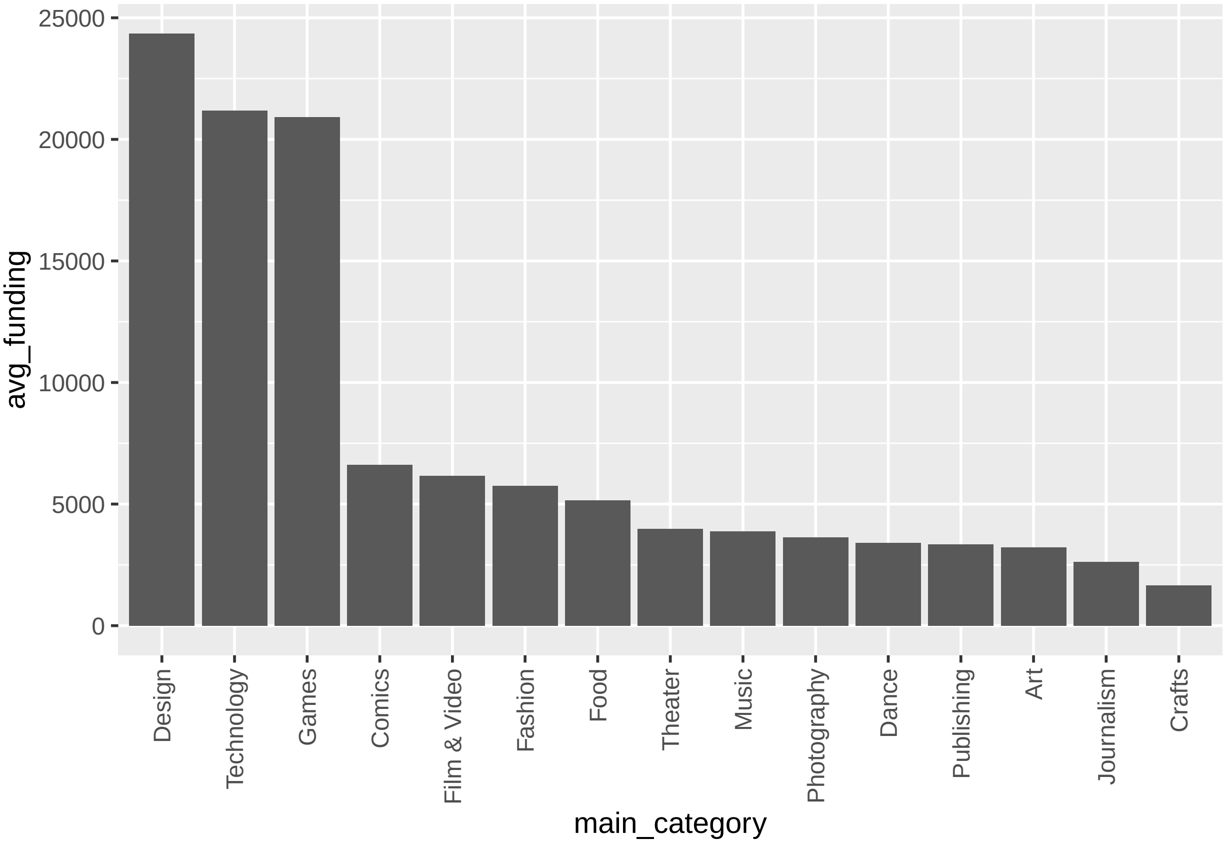
```
##########################################################
# PREPARATION
##########################################################

# training set
training <- training %>% mutate(duration = round(as.Date(deadline) - as.Date(launched), digits = -1))
training <- training %>% mutate(duration = ifelse(duration > 100, 100, duration))
training <- training %>% mutate(month = format(as.Date(deadline), "%m"))
training <- training %>% mutate(funded_usd = usd_pledged_real)
training <- training %>% mutate(goal_usd = round(usd_goal_real, digits = -4))
training <- subset(training, select = -c(currency, deadline, goal, launched, pledged, backers, usd.pledg

# validation set
validation <- validation %>% mutate(duration = round(as.Date(deadline) - as.Date(launched), digits = -1
validation <- validation %>% mutate(duration = ifelse(duration > 100, 100, duration))
validation <- validation %>% mutate(month = format(as.Date(deadline), "%m"))
validation <- validation %>% mutate(funded_usd = usd_pledged_real)
validation <- validation %>% mutate(goal_usd = round(usd_goal_real, digits = -4))
validation <- subset(validation, select = -c(currency, deadline, goal, launched, pledged, backers, usd.p
```

# Exploratory analysis

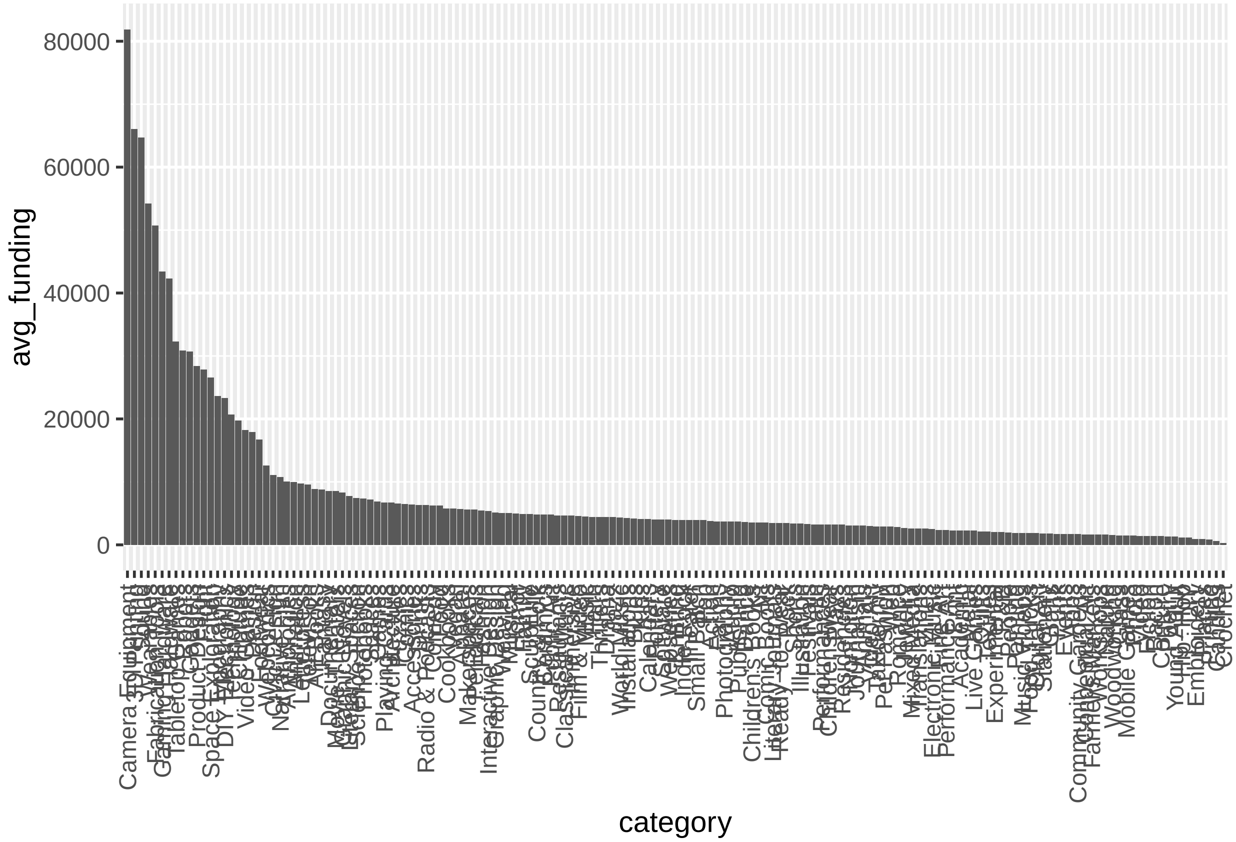
Now that the data is ready, let's explore evolution of ratings through our dimensions.

```
##########################################################
# EXPLORATORY
##########################################################
```

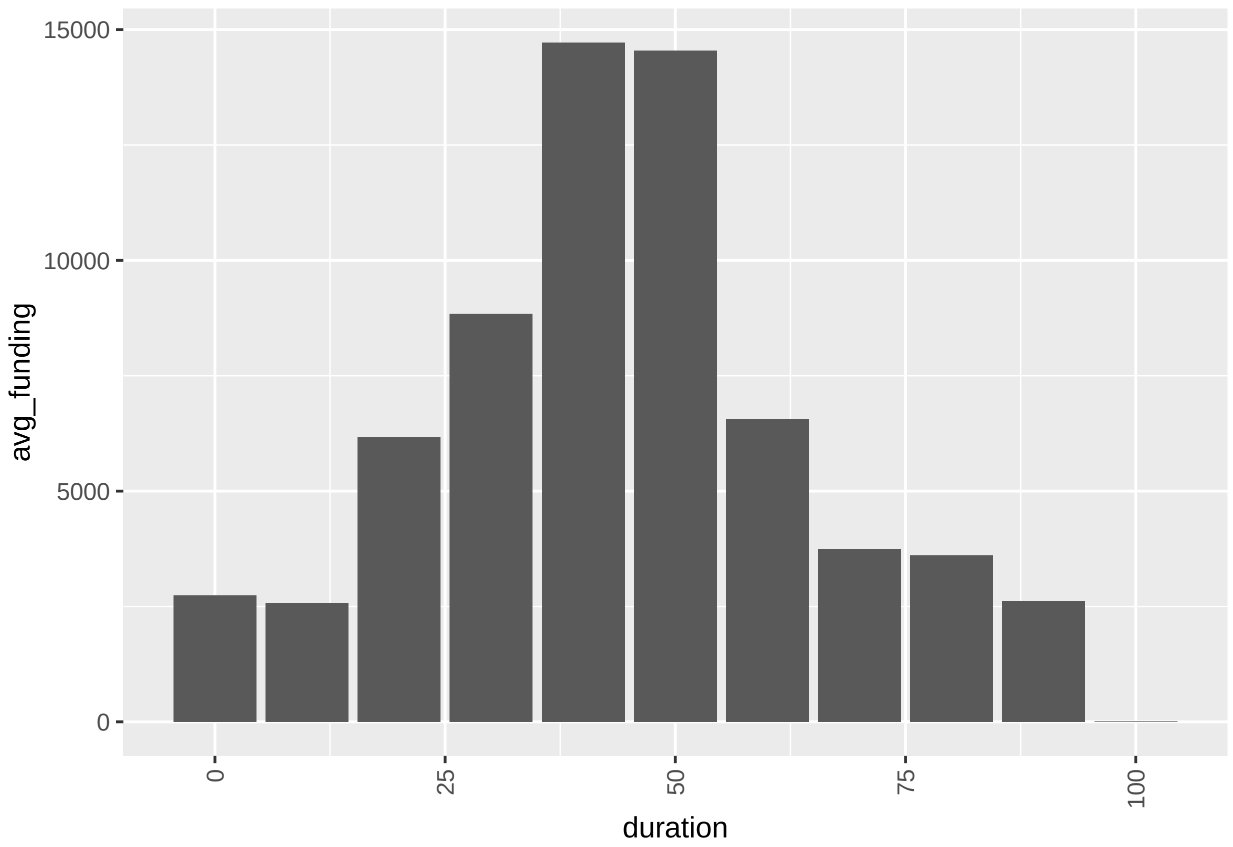
```
training %>%
  group_by(main_category) %>%
  summarise(avg_funding = mean(funded_usd)) %>%
  mutate(main_category = reorder(main_category, -avg_funding)) %>%
  ggplot(aes(x = main_category, y = avg_funding)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



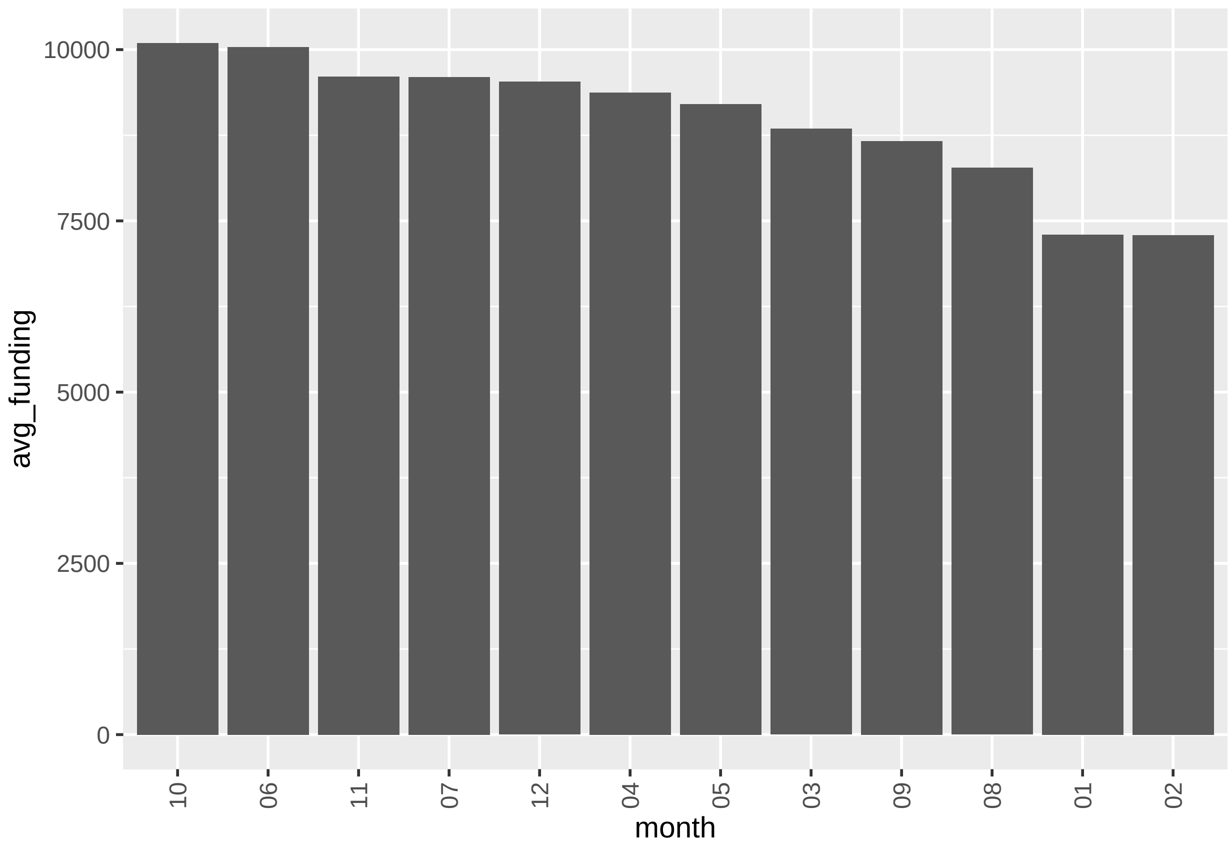
```
training %>%
  group_by(category) %>%
  summarise(avg_funding = mean(funded_usd)) %>%
  mutate(category = reorder(category, -avg_funding)) %>%
  ggplot(aes(x = category, y = avg_funding)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



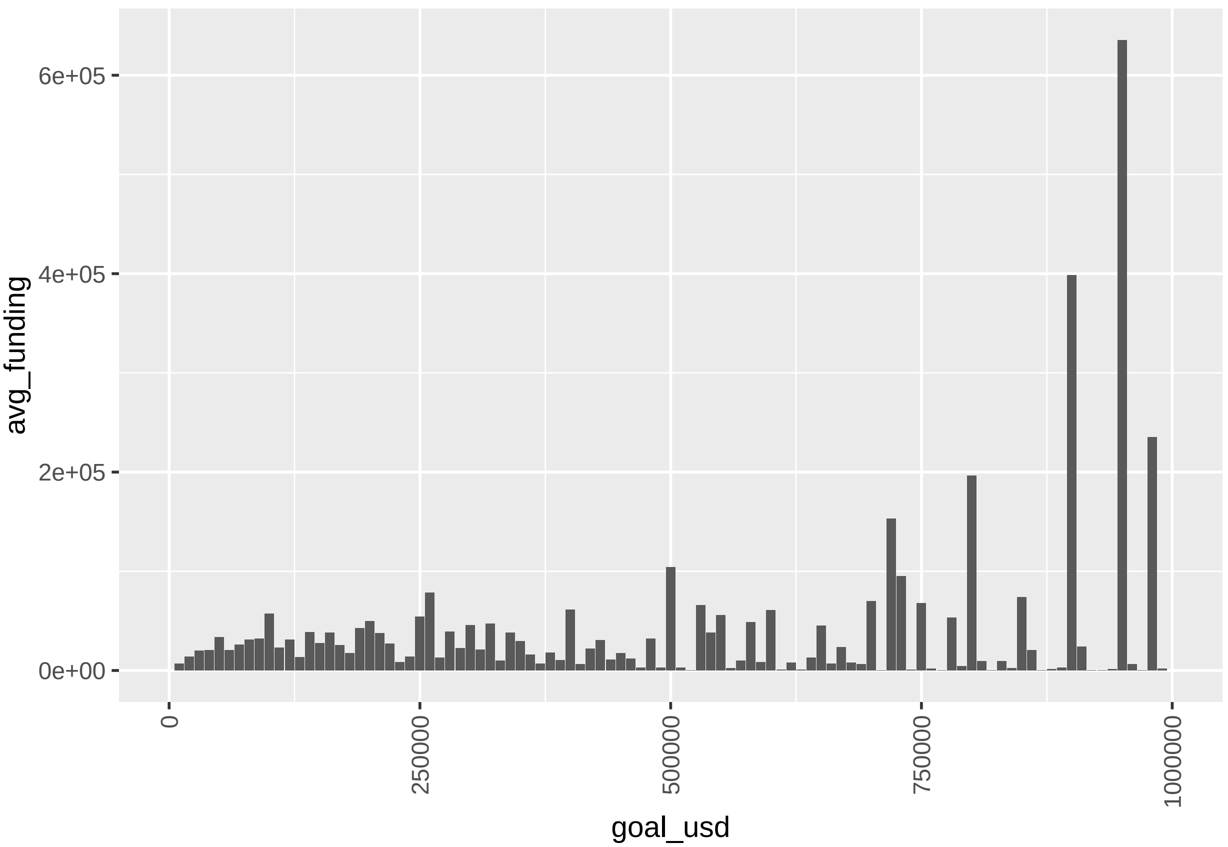
```
# main category and category (sub):
# 1/ design, technology, and games are clearly leading backers' interest with average at $25k+
# 2/ fourth category (comics) is stepped down with average below $10k
# 3/ the hi-tech dominance of the three main categories is showing even higher inner gaps

training %>%
  group_by(duration) %>%
  summarise(avg_funding = mean(funded_usd)) %>%
  ggplot(aes(x = duration, y = avg_funding)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



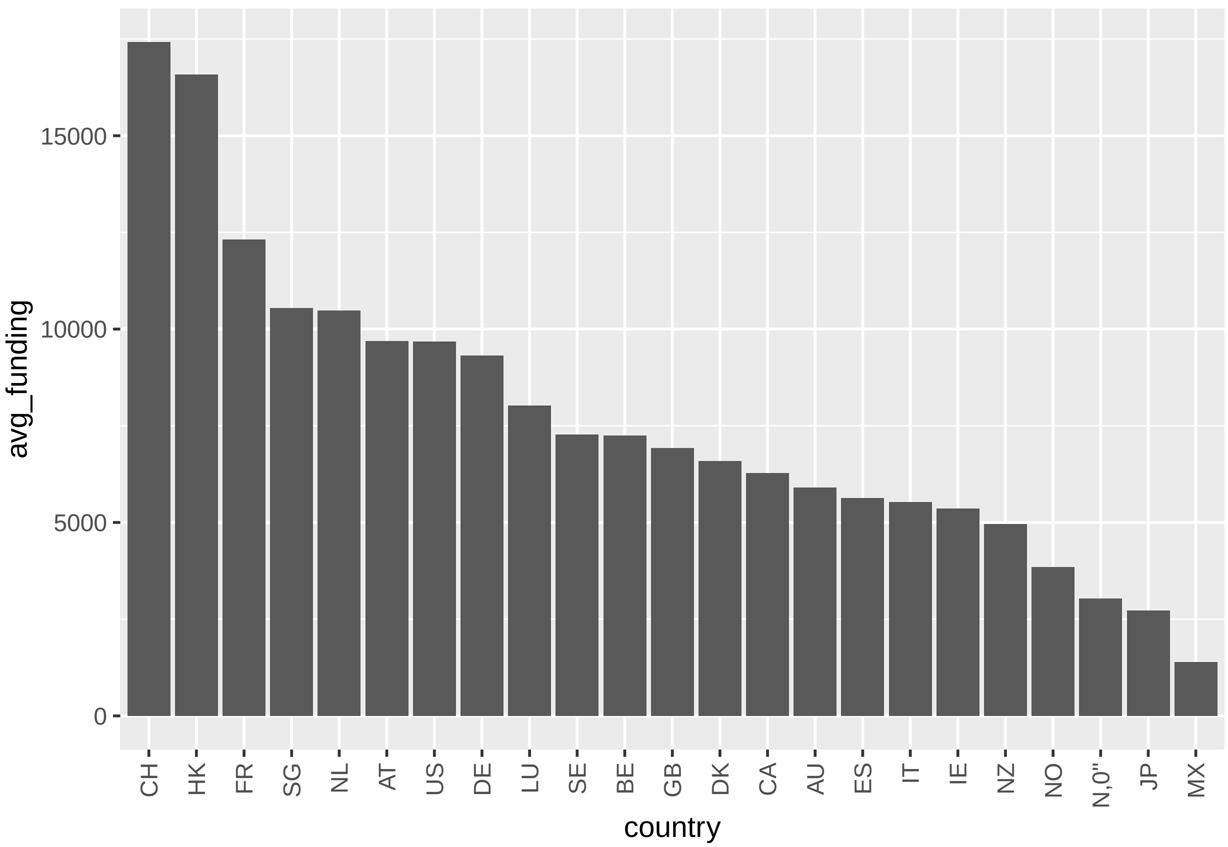
```
training %>%
  group_by(month) %>%
  summarise(avg_funding = mean(funded_usd)) %>%
  mutate(month = reorder(month, -avg_funding)) %>%
  ggplot(aes(x = month, y = avg_funding)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



```
# duration and month:
# 1/ projects with duration from 35 to 55 are showing better funding results than the rest
# 2/ no particular preferential month is identified
# 3/ ... though jan and feb are constantly the lowest rewarding ones (since after Christmas)

training %>%
  group_by(goal_usd) %>%
  summarise(avg_funding = mean(funded_usd)) %>%
  ggplot(aes(x = goal_usd, y = avg_funding)) +
  geom_bar(stat = "identity") +
  xlim(0, 1000000) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



```
training %>%
  group_by(country) %>%
  summarise(avg_funding = mean(funded_usd)) %>%
  mutate(country = reorder(country, -avg_funding)) %>%
  ggplot(aes(x = country, y = avg_funding)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



```
# goal and country:
# 1/ no meaningful info could be extracted from the goal analysis
# 2/ some countries are leading the top: CH Switzerland and HK Hong Kong
# 3/ ... could be explained by population volume and wealth
```

## Methods

From the insights obtained in the previous section, we identify 3 effects of interest for our methods: category, duration, and country. We start modelling with the simple average and integrate progressively respective bias terms.

```
##########################################################
# BASELINE: SIMPLE AVERAGE
##########################################################

mu <- mean(training$funded_usd)
baseline_rmse <- RMSE(validation$funded_usd, mu)
rmse_results <- tibble(method = "baseline: simple average", RMSE = baseline_rmse)
rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---:|
| baseline: simple average | 92423.54 |

```
##########################################################
# MODEL 1: CATEGORY EFFECT (b_c)
##########################################################

main_category_avgs <- training %>%
  group_by(main_category) %>%
  summarize(b_mc = mean(funded_usd - mu))

predicted_funding <- mu + validation %>%
  left_join(main_category_avgs, by='main_category') %>%
  .$b_mc

model_1_rmse <- RMSE(predicted_funding, validation$funded_usd)
rmse_results <- bind_rows(rmse_results, tibble(method="model 1: category effect", RMSE = model_1_rmse )
rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---:|
| baseline: simple average | 92423.54 |
| model 1: category effect | 92067.40 |

```
##########################################################
# MODEL 2: DURATION EFFECT (b_d)
##########################################################

duration_avgs <- training %>%
  left_join(main_category_avgs, by='main_category') %>%
  group_by(duration) %>%
  summarize(b_d = mean(funded_usd - mu - b_mc))

predicted_funding <- validation %>%
  left_join(main_category_avgs, by='main_category') %>%
  left_join(duration_avgs, by='duration') %>%
  mutate(pred = mu + b_mc + b_d) %>%
  .$pred

model_2_rmse <- RMSE(predicted_funding, validation$funded_usd)
rmse_results <- bind_rows(rmse_results, tibble(method="model 2: category + duration effect", RMSE = mode
rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---:|
| baseline: simple average | 92423.54 |
| model 1: category effect | 92067.40 |
| model 2: category + duration effect | 92017.05 |

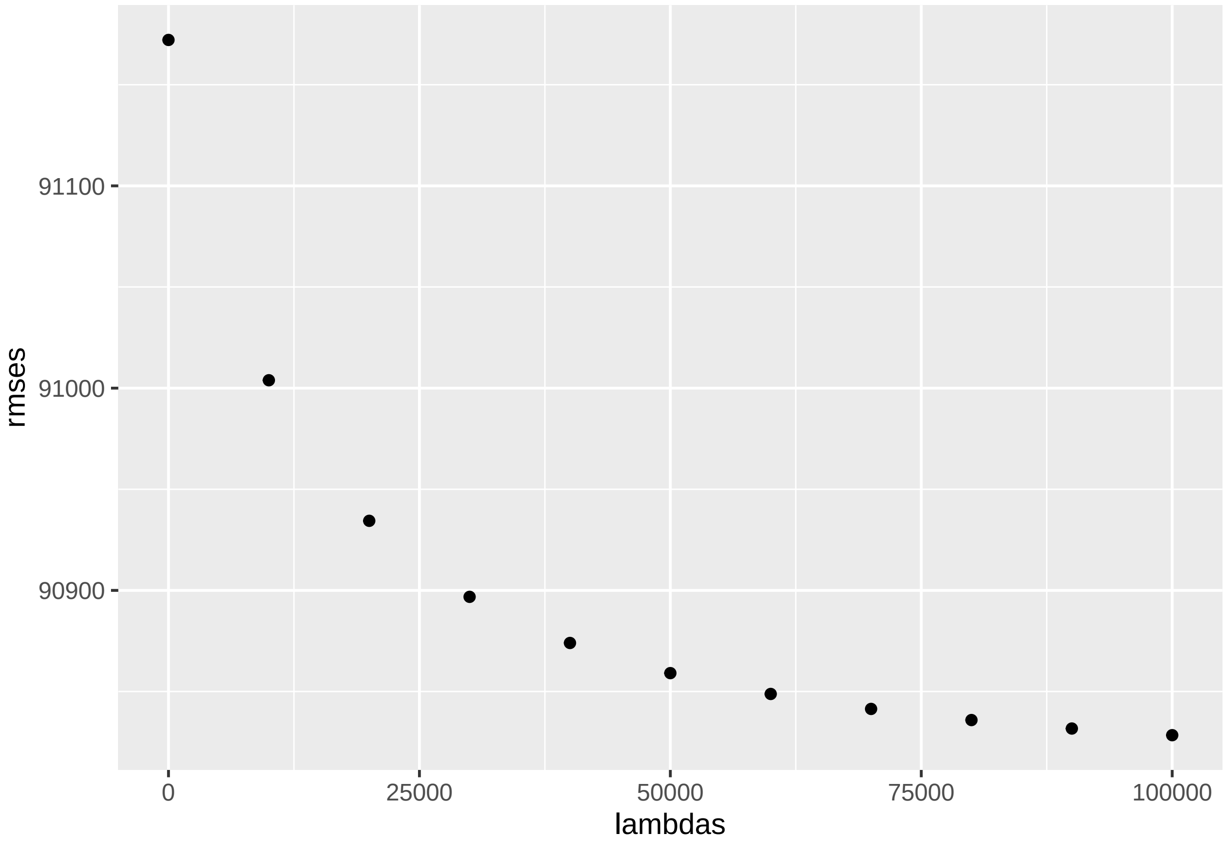
```
###########################################################
# MODEL 3: COUNTRY EFFECT (b_ct)
###########################################################

country_avgs <- training %>%
  left_join(main_category_avgs, by='main_category') %>%
  left_join(duration_avgs, by='duration') %>%
  group_by(country) %>%
  summarize(b_ct = mean(funded_usd - mu - b_mc - b_d))

predicted_funding <- validation %>%
  left_join(main_category_avgs, by='main_category') %>%
  left_join(duration_avgs, by='duration') %>%
  left_join(country_avgs, by='country') %>%
  mutate(pred = mu + b_mc + b_d + b_ct) %>%
  .$pred

model_3_rmse <- RMSE(predicted_funding, validation$funded_usd)
rmse_results <- bind_rows(rmse_results, tibble(method="model 3: category + duration + country effect",
rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---:|
| baseline: simple average | 92423.54 |
| model 1: category effect | 92067.40 |
| model 2: category + duration effect | 92017.05 |
| model 3: category + duration + country effect | 92001.74 |

## Regularization

Finally we have to consider outliers due to the low quantity of records and the specificity of some projects (ex. some could have been heavily promoted = besides standard means) so we need to reduce the effect of such anomalies by regularizing our model.

```
###########################################################
# REGULARISATION
###########################################################

lambdas <- seq(0, 100000, 10000)
rmses <- sapply(lambdas, function(l){
  b_mc <- training %>%
    group_by(main_category) %>%
    summarize(b_mc = sum(funded_usd - mu)/(n()+l))
  b_d <- training %>%
    left_join(b_mc, by="main_category") %>%
    group_by(duration) %>%
    summarize(b_d = sum(funded_usd - mu - b_mc)/(n()+l))
  b_ct <- training %>%
    left_join(b_mc, by="main_category") %>%
    left_join(b_d, by="duration") %>%
    group_by(country) %>%
    summarize(b_ct = sum(funded_usd - mu - b_mc - b_d)/(n()+l))
  predicted_funding <- validation %>%
    left_join(b_mc, by="main_category") %>%
    left_join(b_d, by="duration") %>%
    left_join(b_ct, by="country") %>%
    mutate(pred = mu + b_mc + b_d + b_ct) %>%
    .$pred
  return(RMSE(predicted_funding, training$funded_usd))
})

qplot(lambdas, rmses)
```



```
lambda <- lambdas[which.min(rmses)]
lambda
```

```
## [1] 1e+05
```

```
model_4_rmse <- min(rmses)
rmse_results <- bind_rows(rmse_results, tibble(method="model 4: model 3 regularized", RMSE = model_4_rms
rmse_results %>% knitr::kable()
```

| method | RMSE |
|---|---|
| baseline: simple average | 92423.54 |
| model 1: category effect | 92067.40 |
| model 2: category + duration effect | 92017.05 |
| model 3: category + duration + country effect | 92001.74 |
| model 4: model 3 regularized | 90828.42 |

# Final

To summarize it all:
1/ we reached an RMSE of $90.8k
2/ we have applied the methodology and tools learned in the class
3/ we can't say this is a great result and I wouldn't use it for anything useful at this stage

```
############################################################
# FINAL
############################################################

final_rmse <- model_4_rmse
final_rmse
```

```
## [1] 90828.42
```

Limitations:
1/ the quantity of records is quite low (379k)
2/ the outliers were numerous: feel free to adjust the seed and see how volatile results are (though regularization usually sets back final RMSEaround the $90k)
3/ this could also point that there are more important factors determining the success of your crowdfunding project than those we have worked on

Going beyond:
1/ first, I believe that crowdfunding success is highly tied to the owners dynamism on social networks: it would be interesting to link the funding success to social posts, celebrity (references on web), etc.
2/ second, some (most) campaigns promise valuable content for backers: what if we could value objectively such content and integrate in the model ?
3/ and finally we have seen multiple crowdfunding platforms emerge in the recent years, each trying to fit a market (whether regional/cultural or content-related), therefore I do not think that any recommendation/prevalidation system of budget at your project creation will be satisfying taking in isolation of other platforms => but for this we would need to aggregate data from other companies …