Project 2 Group 10

Campaign Data Analysis

By: Robert Williard, Grant Hagen,
and Jason Wloszek

Contents

Introduction

In this group project we were given two data sets: crowdfunding and contacts. The crowdfunding dataset contained details about various crowdfunding campaigns. Some of the more prominent details being: the company name, contact ids, goal amount, pledged amount, number of backers, the outcome, country, category & sub-category, the start date, and the end date. The contacts dataset contained the contact information as well as the contact id in the campaign dataset. Our task was to clean/restructure our datasets, load our datasets into Postgres, query the datasets, and then analyze what we queried.

Database Design Considerations

Category

The category table then contained 2 unique columns, the category_ids and the category. The category column was extracted from the campaign dataset (making sure to only bring back unique values.). We then assigned a unique category id to each category. The category_ids column would also become our primary key for this table. Both columns were inserted as varchars. See appendix A for diagram.

Subcategory

The subcategory table then contained 2 unique columns, the subcategory_ids and the subcategory. The subcategory column was extracted from the campaign dataset, also ensuring to only bring back unique values as done with the prior column. We then assigned a unique subcategory id to each subcategory. The subcategory _ids column would also become our primary key for this table. Both columns were inserted as varchars. See appendix A for diagram.

Contacts

The contacts table ended up containing 4 columns: contact_id, first_name, last_name, and email. The contact_id column would become our primary key for the table. The columns first_name, last_name, and email were inserted as varchars, and contact_id was inserted as an integer. See appendix A for diagram.

Campaign

The campaign table ended up containing 14 columns: cf_id contact_id, company_name, description, goal, pledged, outcome, backers_count, country, currency, launched_date, end_date, category_ids, and subcategory_ids. The primary key for this table was cf_id. We also had 3 foreign keys: contact_id, category_ids, and subcategory_ids that correspond to the contacts, category, and subcategory tables respectively. The columns cf_id, contact_id, goal, pledged, and backers_count were entered in as integers. The columns company_name, description, outcome, country, currency, category_ids, and subcategory_ids were entered in as varchars. The columns launched_date and end_date were entered in as datetimes. See appendix A for diagram.

Extract/Transform/Load code overview

After importing our dependencies for our code, we began by taking a look at our campaign dataset. We then began by splitting up the column category & sub-category into two separate columns, category and sub-category. After splitting the columns, we then were able to extract the unique values from those columns and create separate tables with primary key identifiers and the unique values.

During the transform phase of the campaign dataset, we renamed a few columns, changed some data types, merged the category/subcategory tables to bring in the unique keys, and dropped unwanted columns per the instructions. Next, we worked on the contacts dataset. The tricky part about this dataset was that it was a list of dictionaries, so we needed to unravel each row to get the data how we wanted it. After doing so, we were able to create a pandas data frame to include contact_id, first_name, last_name, and email. Lastly, we exported the data frame as a csv.

With the extracting and loading phases completed, we transitioned to the loading phase. Before we used python to load our 4 datasets to our SQL database, we needed to make sure our tables were created in Postgres and had all the appropriate connections. After doing so, we were able to set our connection to the database, create our engine, and load our datasets into Postgres.

Data Analysis

Question 1

The first question we wanted to see about our dataset was which countries had the most successful campaigns. We used raw SQL to query our dataset and found that the United States had the vast majority of successful campaigns (see appendix B).

Question 2

The second question we wanted more clarity on which categories had the most successes. We used raw SQL to query our dataset and found that theater campaigns were the most successful. Followed by film & video and music (see appendix C)

Question 3

The last question we wanted answered was which subcategories were the most popular. We used raw SQL to query our dataset and found that plays were by far the most popular subcategory. Followed by rock and documentary (see appendix D)

Bias/Limitations

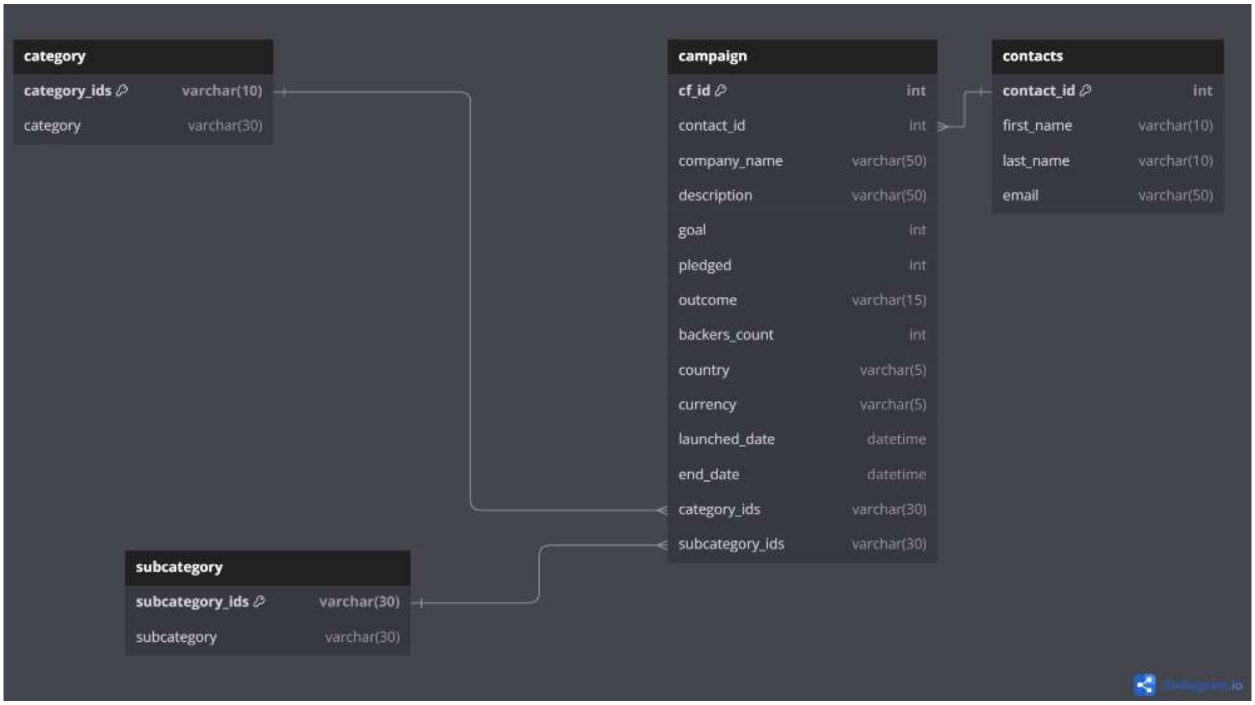
We can see bias/limitations in this dataset, and here a few things we noticed. First, the overwhelming majority of campaigns are US based. Is that actually the case? Does having more US campaigns change the underlying data? If so, how? Also, there is no descriptive data on why certain campaigns failed or succeeded. We can only draw conclusions based on a few aspects of the campaigns. More descriptive data is required to understand better the outcome of the campaigns. Lastly, neither Staff Pick or the Spotlight columns have a detailed explanation, nor can they be directly inferred as to what they are. Are they truly important to gaining a deeper understanding of these campaigns?

Conclusions/Reflection

In summation, we were tasked with creating a database, extracting/transforming/loading datasets to that database, and querying the database to understand the datasets further. This has been a challenging but fun example of what it would be like to handle such tasks in the real-world.

Appendix

Appendix A

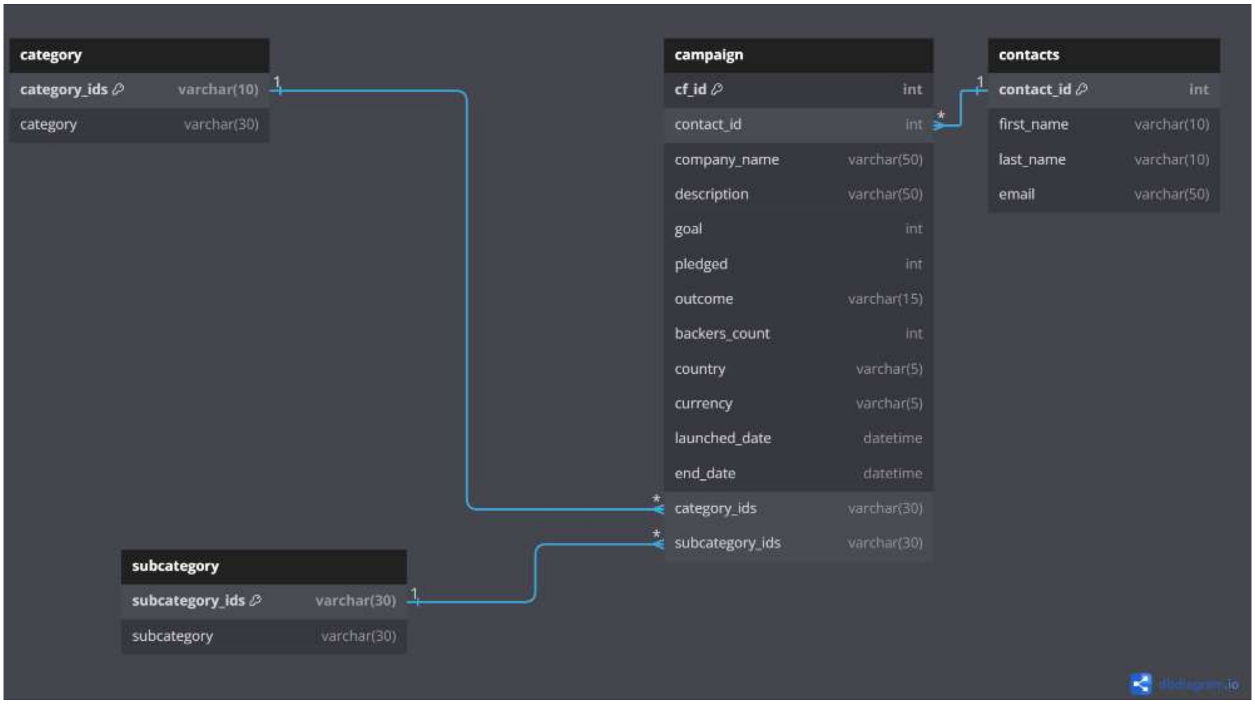


category	
category_ids	varchar(10)
category	varchar(30)

campaign	
cf_id	int
contact_id	int
company_name	varchar(50)
description	varchar(50)
goal	int
pledged	int
outcome	varchar(15)
backers_count	int
country	varchar(5)
currency	varchar(5)
launched_date	datetime
end_date	datetime
category_ids	varchar(30)
subcategory_ids	varchar(30)

contacts	
contact_id	int
first_name	varchar(10)
last_name	varchar(10)
email	varchar(50)

subcategory	
subcategory_ids	varchar(30)
subcategory	varchar(30)



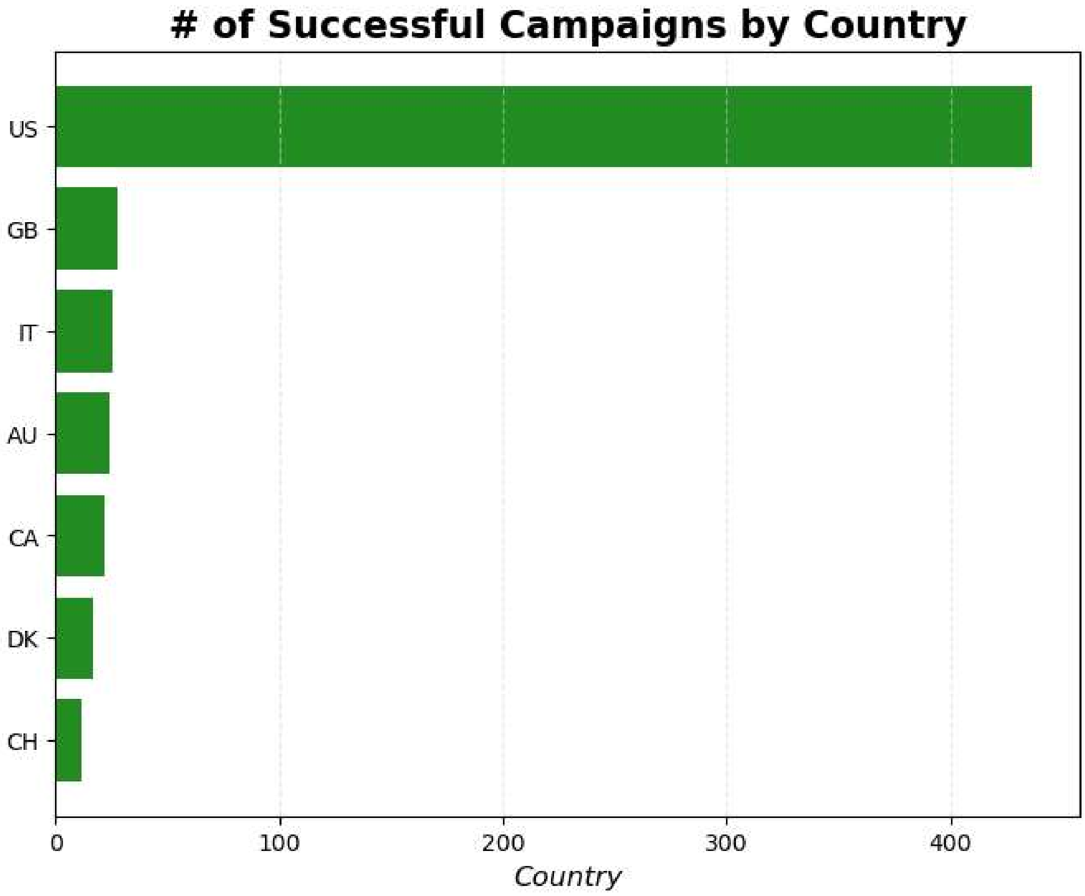
category	
category_ids	varchar(10)
category	varchar(30)

campaign	
cf_id	int
contact_id	int
company_name	varchar(50)
description	varchar(50)
goal	int
pledged	int
outcome	varchar(15)
backers_count	int
country	varchar(5)
currency	varchar(5)
launched_date	datetime
end_date	datetime
category_ids	varchar(30)
subcategory_ids	varchar(30)

contacts	
contact_id	int
first_name	varchar(10)
last_name	varchar(10)
email	varchar(50)

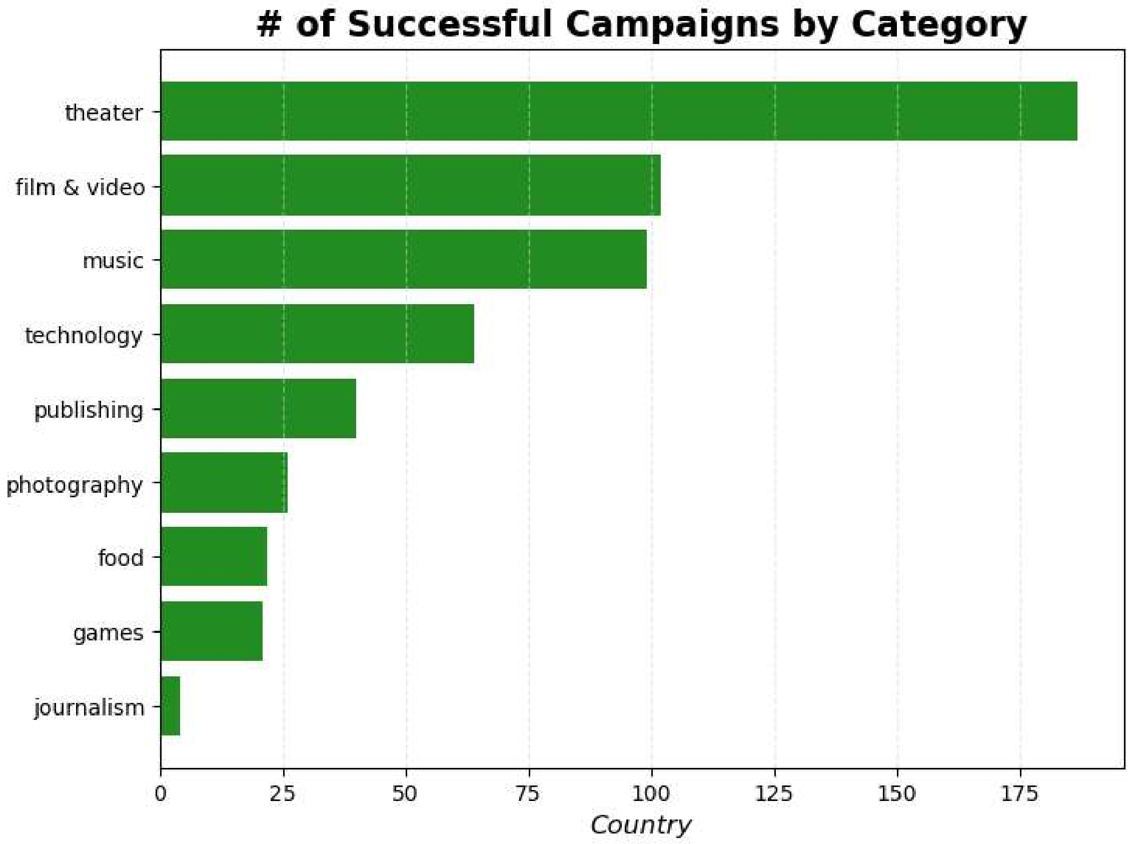
subcategory	
subcategory_ids	varchar(30)
subcategory	varchar(30)

Appendix B



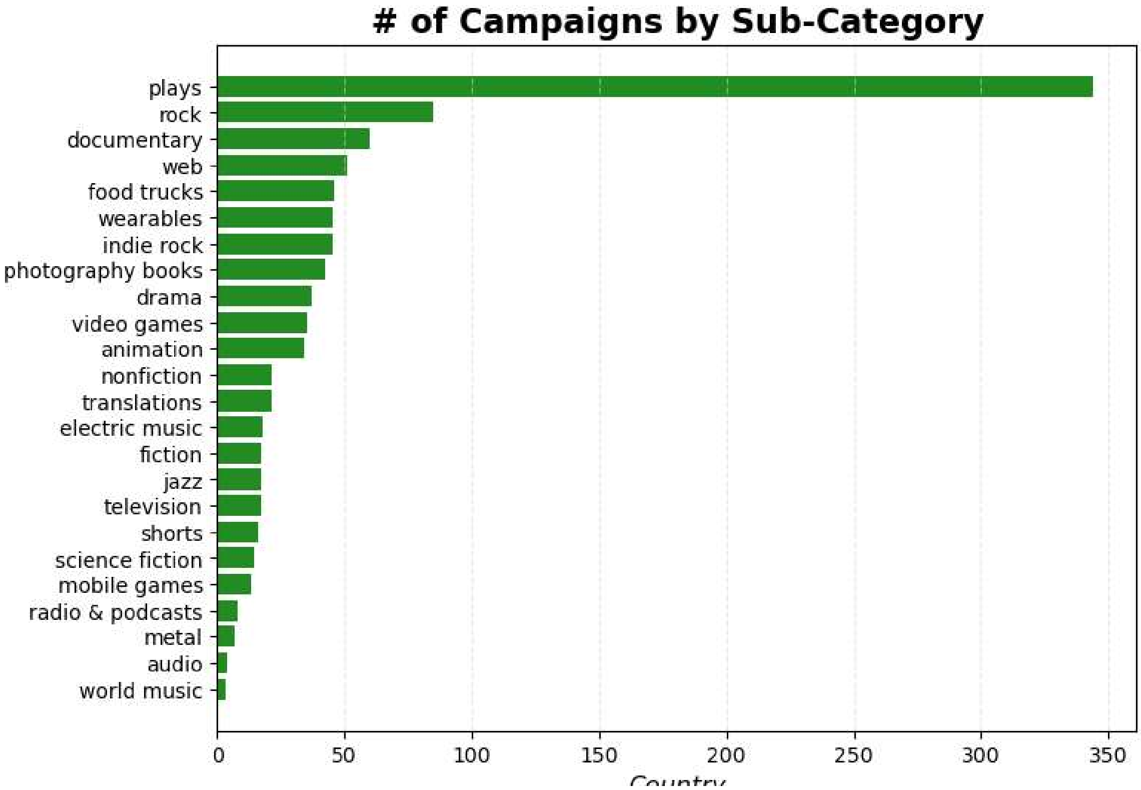
# of Successful Campaigns by Country
US
GB
IT
AU
CA
DK
CH
0
100
200
300
400
Country

Appendix C



# of Successful Campaigns by Category
theater
film & video
music
technology
publishing
photography
food
games
journalism
0
25
50
75
100
125
150
175
Country

Appendix D



# of Campaigns by Sub-Category
plays
rock
documentary
web
food trucks
wearables
indie rock
photography books
drama
video games
animation
nonfiction
translations
electric music
fiction
jazz
television
shorts
science fiction
mobile games
radio & podcasts
metal
audio
world music
0
50
100
150
200
250
300
350
Country