# **Kickstarter - what separates successful campaign from failed one?**

Crowdfunding is very crowded space in present day where Kickstarter and Indiegogo hold the most market share. My goal is to identify factors for a successful campaign. Successful campaign is one where 'pledged' money (money raised) is greater than or equal to 'goal' (money asked/requested) within a given deadline.

I am using Kaggle data for this analysis which contains below features,

1. ID - unique ID
2. name - name of the campaign
3. category - level 2 category
4. main\_category - level 1 category
5. currency
6. deadline - date stamp
7. goal - amount in local currency
8. launched - date stamp
9. pledged - amount in local currency
10. state - failed, successful, undefined, canceled, live
11. backers - total number of people invested
12. country - country where campaign is launched
13. usd\_pledged - conversion to USD done by Kickstarter
14. usd\_pledged\_real - conversion to USD done by Fixer.io API
15. usd\_goal\_real - conversion to USD done by Fixer.io API

**Data Wrangling**

Load the dataset using pandas and check for column info. Total rows 375,000+ along with 15 features. Two columns 'name' and 'usd pledged' have less entries than other columns. Also the latter has a space character in column name.

There are 3,797 null values in column ‘usd\_pledged’. We could easily drop these rows as it is about 1% of total rows. However, we don't need this feature. Instead we will use ‘usd\_pledged\_real’ where currency conversion is more accurate and in sync with ‘usd\_goal\_real’. Next, analyze the name column to explore missing values.

New information has come to light! There are multiple campaigns under similar name(s) and there are multiple observations where *name* includes a string cancelled or canceled*.* The *state* for these observations is undefined or cancelled. These are bad data and we can drop them to make sure this analysis only includes valid campaigns and not testers. This takes away about 20,000 observations, 230 rows with null values, and 4,000 rows with duplicate data. There is also one incorrect observation for which the launched date is 01-01-1970, which is dropped too. Clean dataset has 346,955 rows.

One last step to make sure our data is ready for analysis is to check the feature *state*. We are only going to look at successful and failed campaigns for the scope of this project. There could be another interesting study with remaining *state* of undefined, cancelled and live. For example, what is the possibility that a campaign is cancelled after registering on kickstarter? But, the problem statement is, what differentiates a successful campaign from failed ones? So we will stick to these two values only and have our ready dataset with 327,355 observations.

**Features**

* Data type for all features is correct except where it contains date entries. Therefore, converting *launched* and *deadline* columns into date-time object
* *Launched* is when campaign was first active. I will consider this date for seasonality. It is better to create new features, *year*, *month* and *day of week* (Monday through Sunday). These will come in handy when looking at time series data
* New feature created, *duration* is the difference between *deadline* and *launched* date. This is the time available for a campaign to pledge money equal to or more than the goal. Remember, for a campaign to be successful, it ought to pledge money at least as much as the goal before or by the deadline date
* *usd\_pledged\_real* and *usd\_goal\_real* are values for each campaign in USD, irrespective of country/currency
* *Backers* is another feature which means total number of pledges made or total number of people who pledged money for a given campaign
* New features created, *p\_timesgoal* - pledged to goal ratio. Note: this should equal to or greater than 1 for successful projects

### **Drop columns**

* *usd\_pledged* is currency conversion done by using exchange rate from different sources. I will keep *usd\_pledged\_real* instead, which uses one exchange rate table for all campaigns
* *ID* is not required as every name is unique
* *goal* is value in local currency, we already have data for all campaigns in USD under column *usd\_goal\_real*
* *pledged* is value in local currency, we already have data for all campaigns in USD under column *usd\_pledged\_real*
* *currency* is the symbol per campaign country. We already have a column country

Dropping these features will use less memory and give better view of dataframe when displayed for analysis.

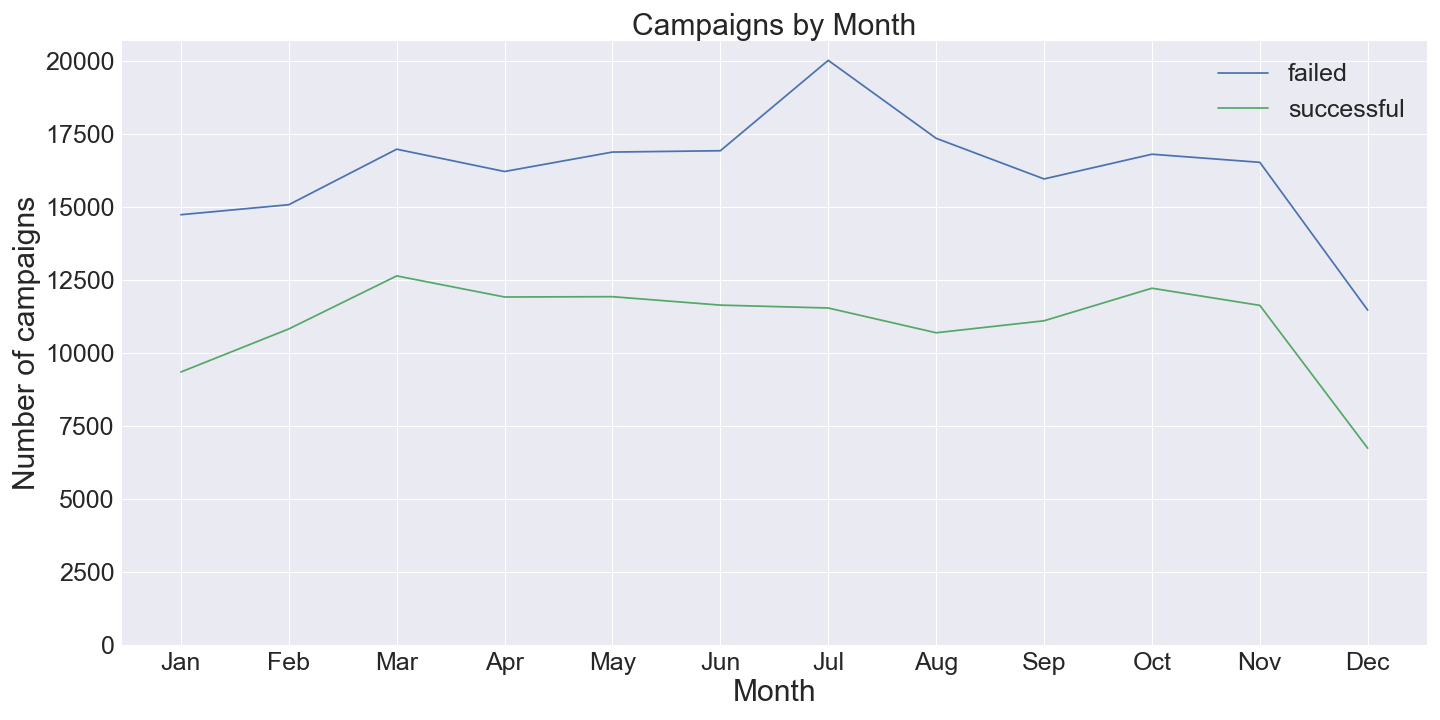
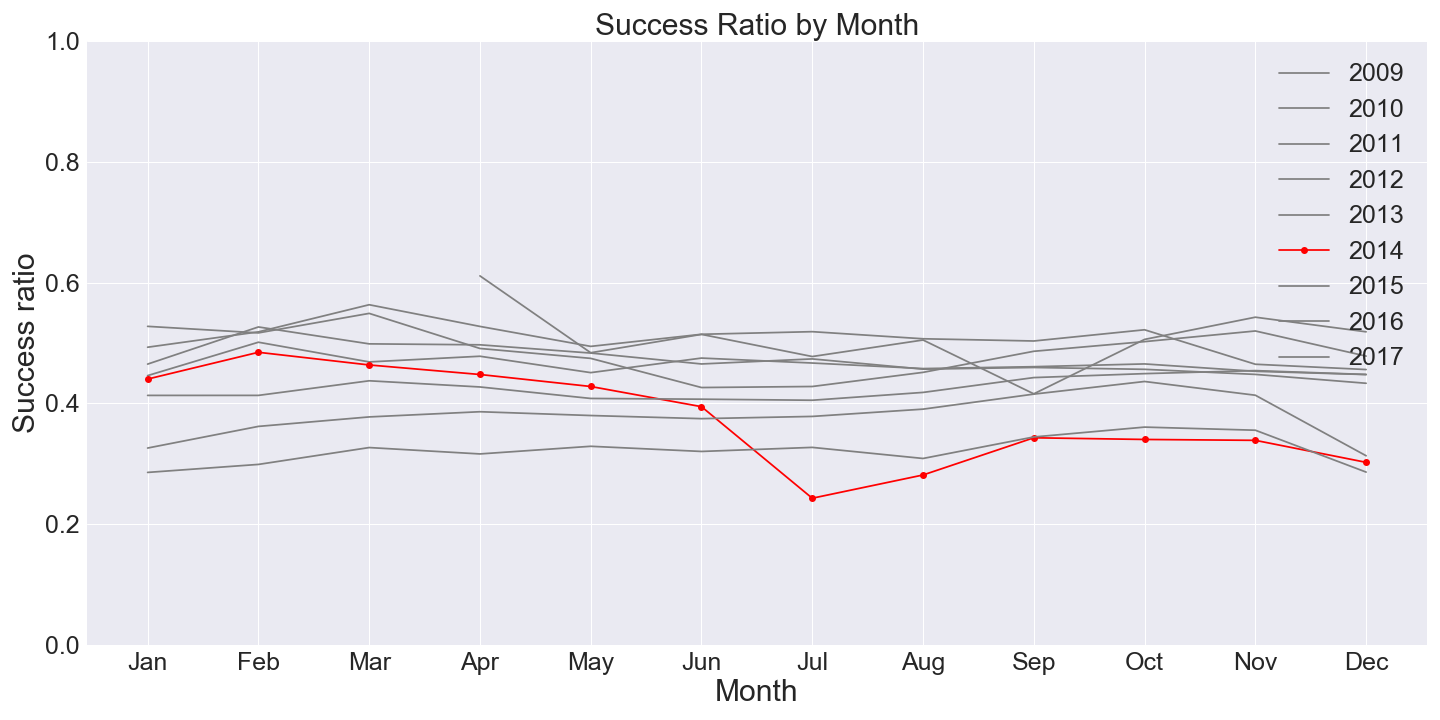
## **df.Info()**

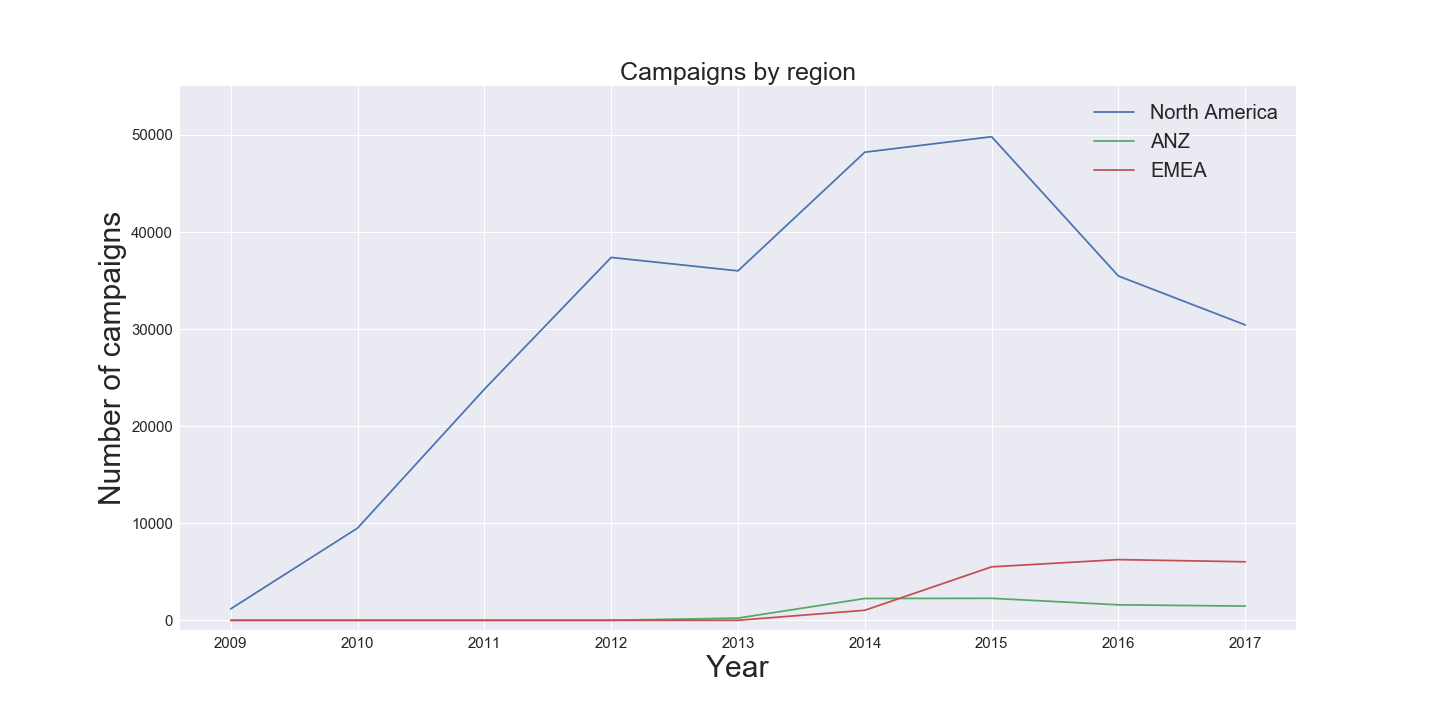
name 346955 non-null object  
category 346955 non-null object  
main\_category 346955 non-null object  
deadline 346955 non-null datetime64[ns]  
launched 346955 non-null datetime64[ns]  
state 346955 non-null object  
backers 346955 non-null int64  
country 346955 non-null object  
usd\_pledged\_real 346955 non-null float64  
usd\_goal\_real 346955 non-null float64  
year 346955 non-null int64  
day 346955 non-null int64  
month 346955 non-null int64  
duration 346955 non-null float64  
p\_timesgoal 346955 non-null float64

Dataset has more than 300,000 observations which is good for finding correlations and trends. This dataset has many useful features.

### **Time series analysis**

Group observations by *month* across all *years* and plot total number of campaigns by *state*. To identify if success is directly correlated to seasonality.

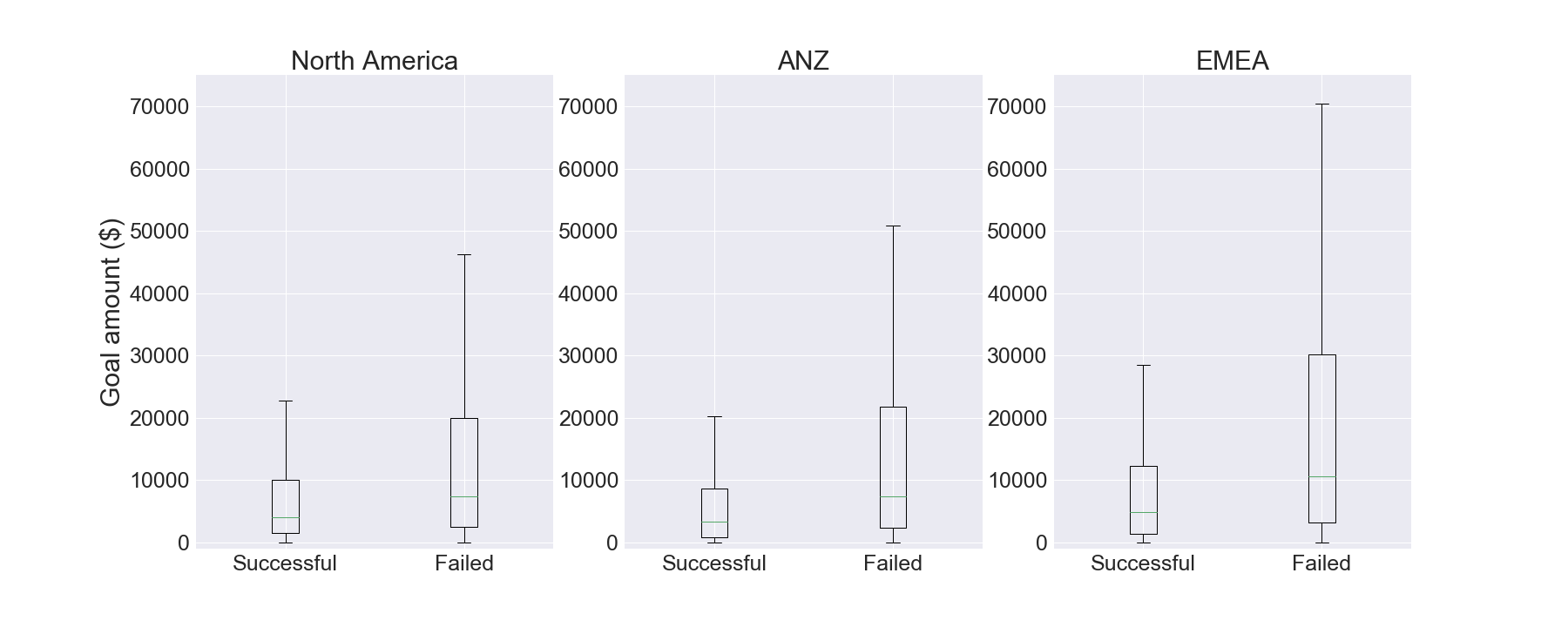
Success ratio, i.e. number of successful campaigns over total campaigns for each year shows a clear dip of 38% in July 2014. Why more campaigns failed during this time? Let’s explore total campaigns per year for three different demographics.



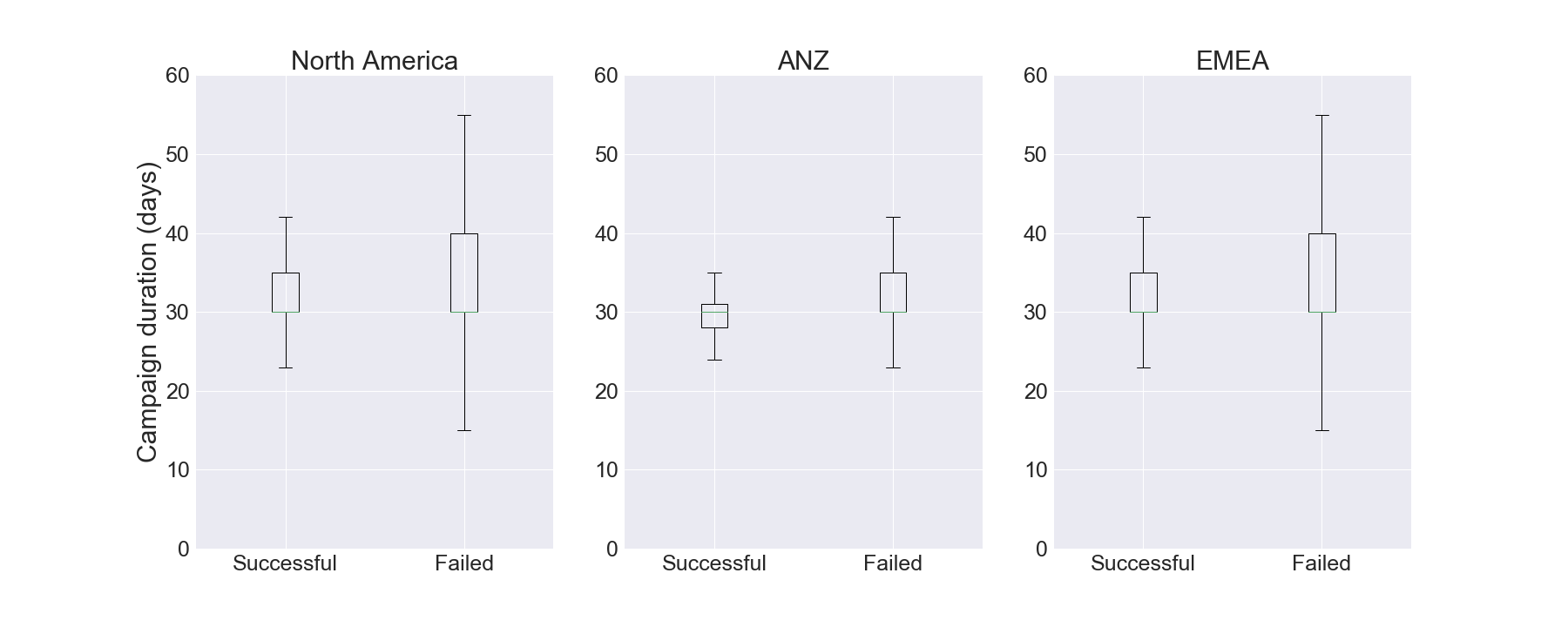
Success ratio declined by 38% from 0.39 to 0.24 and total number of campaigns increased by whopping 107% in July 2014. A lot more countries started running kickstarter campaigns in 2014 (confirmed from online sources like wikipedia).

**Goal amount**

There is strong correlation between *goal* and *state* across regions. This is also true across *main\_category* (analysis not shown). However, this correlation does not signal causation. It will require further analysis and a hypothesis test to prove the statistical significance.

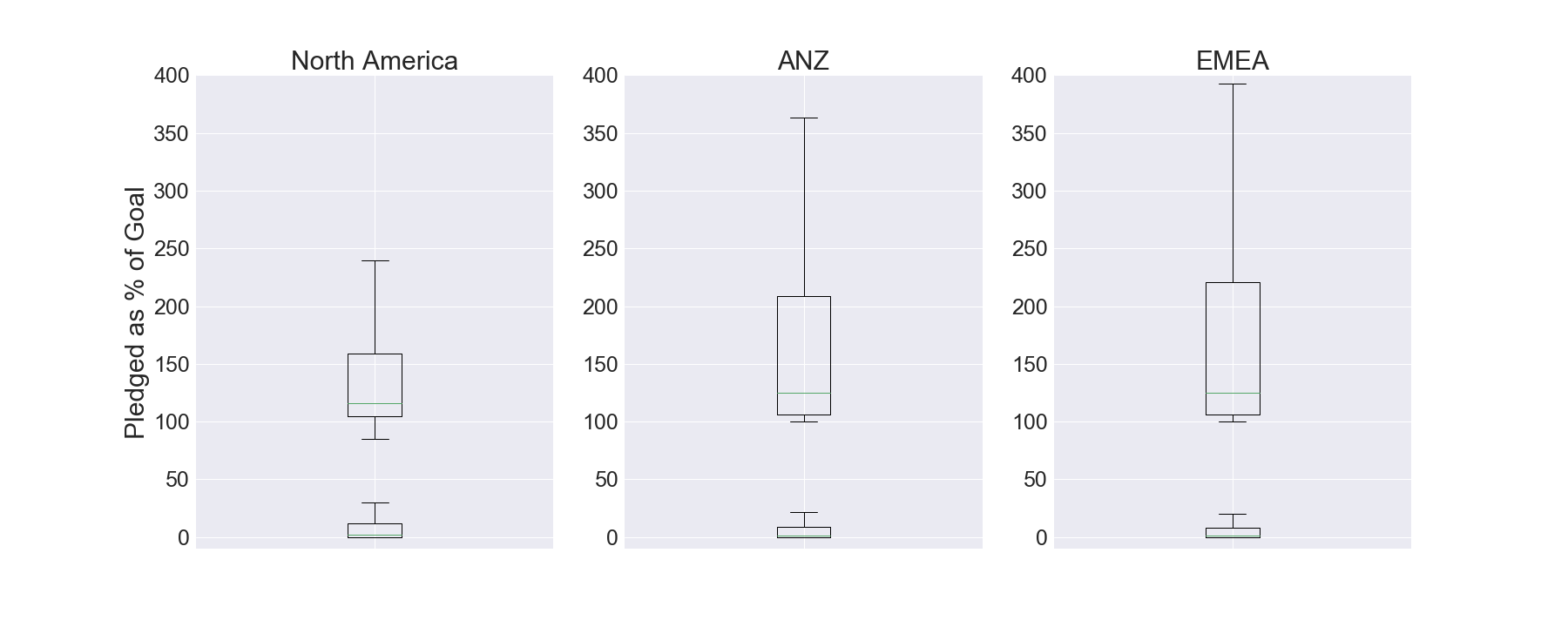
**Duration**

There is a strong correlation between campaign *duration* and *state* across all regions and categories (analysis shown only for regions)



**p\_timesgoal**

One of the derived features in dataset is *pledged* amount as a ratio of *goal* amount. For successful campaign it should be at least 100%, more than 100% is a stretch goal which is happy scenario. **97% of failed campaigns *pledged* half or less of their *goal* amount**. People overvalue their own campaigns or could be out of desperation to raise capital? There is a very interesting problem to compute the probability of success if campaign has raised x% of goal in d days. Daily pledge amount data is required for this, our data set is restricted to one observation per campaign.



**Initial analysis**

Success ratio dropped by 38% in 2014, which is correlated with the increase in total campaigns in the same year as more countries from all regions started to participate in crowdfunding. More projects failed in initial years for countries in these regions. There are few features correlated to campaign success. More significant features are ***duration*** of the campaign and ***goal***. Success is when goal is achieved within predefined duration of the campaign. The *duration* spread is smaller for successful campaigns when compared with failed campaigns. This is true for mean and median values as well, across all demographic regions and categories. Similar results for *goal*when compared between successful and failed campaigns.

**Statistical analysis**

Median *goal*amount is 48% lower for successful campaigns when compared to failed campaigns. There are some major outliers where campaigns have goal for as high as $100 MM, hence chose median instead of mean. *Duration* of campaigns is not skewed similar to *goal*, because campaigns normally have to select deadline date of 1 to 3 months from the launched date. Mean *duration* is lower by 8.5% for successful campaigns.

There was no significant distribution type for both of these features. Visual inspection via histogram shows close to exponential distribution for successful campaigns. However, it is more appropriate to assume there is no identifiable distribution and use non-parametric tests. Spearman test proves **statistical significant correlation** of downward sloping 0.095 between campaign success and *duration* of campaign. And, downward sloping 0.22 for *goal* amount. I split the data into successful and failed campaigns, to compare if there is a difference in duration and goal distribution. Since data is skewed as mentioned earlier, I used Mann Whitney as a non-parametric test to compare median values.

**Hypothesis testing**

**Hₒ**: The median goal amount is same for successful and failed campaigns.

**Hₐ**: The difference in median goal amount is statistically significant between successful and failed campaigns.

P-value is zero and we reject the null hypothesis. Thereby concluding there is a **statistically significant difference between median goal amount for successful and failed campaigns.** P-value is zero when the same hypothesis test is conducted for *duration* of the campaign.

**Machine Learning**

I split the features in two section, one for applying NLP on campaign title and the other set holds all the relevant features derived in earlier analysis.

1. **NLP**

Name is the only feature with text in this data set. However, the contents are not purely numeric or alphanumeric. There is a lot of special characters and symbols, along with numbers. Some examples are, 3X3, 000, #007, etc.

Prepare data

Name text could be vectorize in few different ways. I used word\_tokenize initially to split the text into list of words. This was not the best method as it considered any string between two white spaces as a word. This was corrected by first removing any special character, numbers and punctuation. However, this could be very easily done by using regular expressions as token pattern parameter in any vectorizer.

Vectorizer

I used CountVector (CV) and TfIdfVector (TV) as vectorizers with token\_pattern = '[A-Za-z0-9]+'. TV is not the best option and this was confirmed by reduced cross validation score. TV does not consider words which are commonly occur in all observations. While this may be a better choice in analysis where all the observations belong to similar context, for example fiction books. In this analysis, campaigns are registered by different countries and dialects, humor, emotion, expression is not presented in the same manner as US. This when combined with bi-gram analysis produced better results. CV worked better as it accounted for all words.

Classification model

Naive bayes produced better results than logistic regression or random forest. Cross validation score of 0.645 with bi-gram and parameter tuning with alpha = 1. This is a very much out of the box model for simple analysis to provide structure and basic understanding in capstone 1. Accuracy on training data is 0.92 and on test data is 0.65. This screams overfitting.

Dimensionality reduction

PCA did not work on this since the input matrix is very sparse. I used TruncatedSVD to produce 2 features on a scatter plot. The plot does not show any clear different between two classes, successful and successful. This is pretty obvious as confirmed by poor model accuracy.

1. **Classification model**

I converted few date features into month and day. Here is the list of independent variables.

finish\_month

finish\_day

start\_month

category

main\_category

country

year

duration

goal

Prepare data

First three features are used to assess if there is any seasonality factor which could influence the campaign success. Country, category and main\_category are non number variables. I used them along with year as categorical variables. Year value ranges from 2009 to 2017. This is numeric but it would not make sense to have 2016.5 as a year. Thus, lets treat each year as a category. Duration and goal are numeric variable and the most correlated with result, as seen from earlier analysis.

Pipeline

I created two pipelines, each for numerical and categorical variables. For numeric, steps include simple imputer and standard scaler. For categorical, steps include simple imputer and one hot encoder. A third pipeline is created as column transformer.

I. numeric\_transformer = Pipeline(steps=[('imputer', SimpleImputer(strategy='median')),('scaler', StandardScaler())])  
  
II. categorical\_transformer = Pipeline(steps=[('imputer', SimpleImputer(strategy='constant', fill\_value='missing')),  
 ('onehot', OneHotEncoder(handle\_unknown='ignore'))])  
  
III. preprocessor = ColumnTransformer(transformers=[('num', numeric\_transformer, numeric\_features),  
 ('cat', categorical\_transformer, categorical\_features)])

This preprocessor is then fed in the final pipeline with classifier model. Gradient Boosting produced better results than logistic regression and random forest. Test score of 0.682.