Exploratory analysis of crowdfunding data from Kickstarter

David Ika

23/08/2020

1 - Introduction

This Project will apply various methods in to a dataset to clean, transform, visualise and report on observations.

Chosen dataset is titled “Funding Successful Projects on Kickstarter” and can be found on Kaggle here (https://www.kaggle.com/codename007/funding-successful-projects). Uploaded by user Lathwal (https://www.kaggle.com/codename007)

The dataset was released by company, Kickstarter (https://www.kickstarter.com/), who connects community investors with start-up projects in an ‘all-or-nothing’ fashion: The user sets a goal for their project, and if it falls short by even $1, zero funding is attained.

Data was initially released to help early prediction of whether a project will be successfully funded, but also provides other information that potential authors may find useful.

2 - Preparation

2.1 Libraries

```
#install.packages()
```

```
#general
library(dplyr) #data cleaning.
library(tidyverse)
library(anytime) #time formats.
library(forcats) #data sorting.
library(scales) #labelling axes.
library(lubridate) #manipulate date/time.
library(stringr) #splitting columns
library(countrycode) #country codes.
library(tidyquant) #xts convert

#Plotting
library(corrplot)
library(ggplot2)
library(tidyverse)
library(gridExtra)
library(ggthemes)
library(vcd)
library(forecast) #seasonal

#Text analysis
library(tm)
library(wordcloud)
library(wordcloud2)
library(RColorBrewer)
library(extrafont)

#Mapping
library(sf)
library(rvest)
library(stringr)
library(scales)
library(viridis)
```

2.2 Theme

Theme will be automatically applied to future ggplots without further code.

```
theme_set(theme_minimal()+
              theme(text = element_text(size = 9, colour = "grey20"),
                    axis.text = element_text(size = 10, colour = "grey10"),
                    axis.title = element_text(size=11,face="bold"),
                    plot.title = element_text(size=12,face="bold"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.line = element_line(colour = "grey20",
                      size = 1, linetype = "solid"),
    axis.ticks = element_line(size = 0.5)))
```

2.3 Dataset

```
kstrain <- read.csv("C:/Users/David/OneDrive/Employment/DS Projects/CITS4009 - P1/train.csv")
```

3 - Initial Observations

3.1 Structure and summary

```
str(kstrain)
```

- Total of 108,129 projects analysed across 14 variables.
- Shows various data types: character strings, integers, numerical values and Booleans.
- Some chr variables may need to be converted to factors or numeric values.
- Some formats will need to be transformed to be useful.
- Good mix of info: geographic, time-related, author-related, text-based.

```
summary(kstrain)
```

- Huge variance in goal amount: 0 to 10 million.
- To be converted to factor: country, currency, outcome.
- Time-related variables to be transformed: deadline, state_changed_at, created_at, launched_at.

3.2 Variables and context

```
names(kstrain)
```

```
##  [1] "project_id"           "name"                 "desc"
##  [4] "goal"                 "keywords"             "disable_communication"
##  [7] "country"              "currency"             "deadline"
## [10] "state_changed_at"     "created_at"           "launched_at"
## [13] "backers_count"        "final_status"
```

- project_id: unique id of project.
- name: name of project.

- desc: description of project.
- goal: $ amount required for project.
- keywords: words describing project.
- disable_communication: whether project author has opted to disable communication with donors.
- country: country of project's author.
- currency: currency of goal amount.
- deadline: goal must be achieved on or before this date (unix time format).
- state_changed_at: at this time, project status changed to successful or otherwise (1,0). Unix time format.
- created_at: at this time, project was posted to Kickstarter (unix time format).
- launched_at: at this time, project went live on website (unix time format).
- backers_count: number of people who backed the project.
- final_status: whether project was successfully funded (1 = True; 0 = False).

Renaming some variables for clarity:

```
names(kstrain)[6] <- "disable_comms"
names(kstrain)[13] <- "backers"
names(kstrain)[14] <- "outcome"
```

3.3 Subsetting and summarising numerical data

```
kstrain_num <- kstrain[,!sapply(kstrain, is.character)]

summary(kstrain_num)
```

```
##       goal           disable_comms   state_changed_at     launched_at
##  Min.   :        0   Mode :logical   Min.   :1.241e+09   Min.   :1.241e+09
##  1st Qu.:     2000   FALSE:107806    1st Qu.:1.347e+09   1st Qu.:1.344e+09
##  Median :     5000   TRUE :323       Median :1.394e+09   Median :1.391e+09
##  Mean   :    36726                   Mean   :1.380e+09   Mean   :1.377e+09
##  3rd Qu.:    13000                   3rd Qu.:1.416e+09   3rd Qu.:1.413e+09
##  Max.   :100000000                   Max.   :1.433e+09   Max.   :1.433e+09
##  NA's   :2                           NA's   :3           NA's   :7
```

- disable_comms: Only 323 out of 108,129 elected to disable this communication. Exclude from analysis (immaterial).
- Time conversions required, as noted.
- NAs have been observed; to be dealt with.

3.4 Examples of each variable

```
head(kstrain,3)
```

```
##       project_id                                              name
## 1 kkst1000004038                                 Production Elvis Show
## 2  kkst100004721                                                     ?
## 3 kkst1000064918 Designing a Map for the Dallas Pedestrian Network
## 
desc
## 1 A Live stage production based on a chapter of the life of 
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
lvi
s""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
 & his impact on the music world, Aimed towards the REAL Elvis fans!
## 2                     Buffalo based blues and folk/rock artist Andrew Robert Mitchell will
release his new record,
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
un Me Out Of Tow
n.""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

## 3 
Challenged by a lack of wayfinding, the Dallas Pedestrian Network needs a map clearly showing
connections and accessibility
##    goal                                          keywords disable_comms
## 1 10000                             production-elvis-show         FALSE
## 2  3200 andrew-robert-mitchells-new-album-run-me-out-of-to         FALSE
## 3  1200  designing-a-map-for-the-dallas-pedestrian-network         FALSE
##   country currency   deadline state_changed_at created_at launched_at backers
## 1      US      USD 1416537300       1416537300 1411689838  1411895590      42
## 2      US      USD 1382882927       1382882927 1357311101  1380290927      17
## 3      US      USD 1308454787       1308454787 1305843570  1305862787      31
##   outcome
## 1       1
## 2       0
## 3       0
```

- project_id: merely an identifier and does not add value to this project. To be removed.
- desc: some messy text.
- name: “?” to be dealt with. May imply invalid data of row.

4 - Data Cleaning & Transformation

4.1 Checking NAs

Per column:

```
(apply(is.na(kstrain), 2, sum))
```

```
##       project_id             name             desc             goal
##                0                6                0                2
##         keywords    disable_comms          country         currency
##                1                0                0                1
##         deadline state_changed_at       created_at      launched_at
##                2                3                0                7
##          backers          outcome
##                0                0
```

Total NAs:

```
sum(apply(is.na(kstrain), 2, sum))
```

```
## [1] 22
```

Thus far, only 22 NAs from entire dataset out of 108,129 obs. Safe to remove without affecting dataset. Assigning non-NA data to kstrain1.

```
kstrain1 <- na.omit(kstrain)
```

4.1.1 Transforming further invalid data into NAs

Recall that some "?" values were identified. Converting these, along with blanks and "NA" chr strings, to actual NAs.

```
kstrain1[kstrain1 == "NA"] <- NA
kstrain1[kstrain1 == ""] <- NA
kstrain1[kstrain1 == "?"] <- NA
```

4.1.2 Re-running the check

```
sum(apply(is.na(kstrain1), 2, sum))
```

```
## [1] 61
```

And again, removing NAs

```
kstrain2 <- na.omit(kstrain1)
```

4.2 Sense checks

With prior context, checking for nonsensical data:

- goal should not be negative.
- state_changed_at should not be before created at nor launched_at.
- deadline should not be before created at nor launched_at.

Unless these count for a large portion, we will remove those rows.

```
count(kstrain2[kstrain2[4] < 0, ])
```

```
##   n
## 1 0
```

```
count(kstrain2[kstrain2$deadline < kstrain2$launched_at,])
```

```
##   n
## 1 0
```

```
count(kstrain2[kstrain2$deadline < kstrain2$created_at,])
```

```
##   n
## 1 0
```

```
count(kstrain2[kstrain2$state_changed_at < kstrain2$launched_at,])
```

```
##   n
## 1 0
```

```
count(kstrain2[kstrain2$state_changed_at < kstrain2$created_at,])
```

```
##   n
## 1 0
```

No anomalies.

4.3 Where data is categorical, converting strings to factors

```
kstrain2$country <- factor(kstrain2$country)
kstrain2$currency <- factor(kstrain2$currency)
kstrain2$outcome <- factor(kstrain2$outcome)
```

4.4 Where data should be numeric and continuous, converting strings into numerals

```
kstrain2$deadline <- as.numeric(kstrain2$deadline)
kstrain2$created_at <- as.numeric(kstrain2$created_at)
kstrain2$backers <- as.numeric(kstrain2$backers)
```

4.5 Country acronyms to names

Converting the country acronyms to long-handed characters, then back into factors.

```
kstrain2$country <- factor(countrycode(kstrain2$country, "iso2c", "country.name"))
```

4.6 Dropping variable

As mentioned, dropping due to redundancy, but will use a new variable should we wish to revert.

```
kstrain3 <- select(kstrain2,-1)
```

4.7 Unix time formats to date objects

As mentioned, the following variables are in unix time format which will now be converted into a more usable date object. Again, assigning converted columns + dataset to a new variable, should we wish to revert.

- deadline
- state_changed_at
- created_at
- launched_at

```
kstrain4 <- kstrain3
kstrain4[8:11] <- lapply(kstrain4[8:11], anydate)
head(kstrain4[8:11],5)
```

```
##     deadline state_changed_at created_at launched_at
## 1 2014-11-21       2014-11-21 2014-09-26  2014-09-28
## 3 2011-06-19       2011-06-19 2011-05-20  2011-05-20
## 4 2011-05-15       2011-05-15 2011-03-18  2011-04-15
## 7 2011-06-14       2011-06-14 2011-04-22  2011-04-22
## 8 2015-04-14       2015-04-14 2015-03-12  2015-03-18
```

Variables that were in unix time formats now show as yyyy-mm-dd.

4.8 Re-summarise

```
summary(kstrain4)
```

```
##      name               desc                goal             keywords
##  Length:108053      Length:108053      Min.   :        0   Length:108053
##  Class :character   Class :character   1st Qu.:     2000   Class :character
##  Mode  :character   Mode  :character   Median :     5000   Mode  :character
##                                        Mean   :    36739
##                                        3rd Qu.:    13000
##                                        Max.   :100000000
##
##  disable_comms             country         currency        deadline
##  Mode :logical   United States :91974   USD    :91974   Min.   :2009-05-03
##  FALSE:107731    United Kingdom: 8746   GBP    : 8746   1st Qu.:2012-09-04
##  TRUE :322       Canada        : 3734   CAD    : 3734   Median :2014-03-01
##                  Australia     : 1879   AUD    : 1879   Mean   :2013-09-27
##                  Netherlands   :  705   EUR    :  817   3rd Qu.:2014-11-11
##                  New Zealand   :  353   NZD    :  353   Max.   :2015-06-01
##                  (Other)       :  662   (Other):  550
##  state_changed_at       created_at          launched_at
##  Min.   :2009-05-03   Min.   :2009-04-22   Min.   :2009-04-25
##  1st Qu.:2012-09-04   1st Qu.:2012-06-19   1st Qu.:2012-08-02
##  Median :2014-02-28   Median :2013-11-14   Median :2014-01-28
##  Mean   :2013-09-25   Mean   :2013-07-17   Mean   :2013-08-23
##  3rd Qu.:2014-11-10   3rd Qu.:2014-09-02   3rd Qu.:2014-10-09
##  Max.   :2015-06-01   Max.   :2015-05-23   Max.   :2015-05-27
##
##     backers         outcome
##  Min.   :     0.0   0:73514
##  1st Qu.:     2.0   1:34539
##  Median :    17.0
##  Mean   :   123.6
##  3rd Qu.:    65.0
##  Max.   :219382.0
##
```

Overall summary now makes a lot more sense.

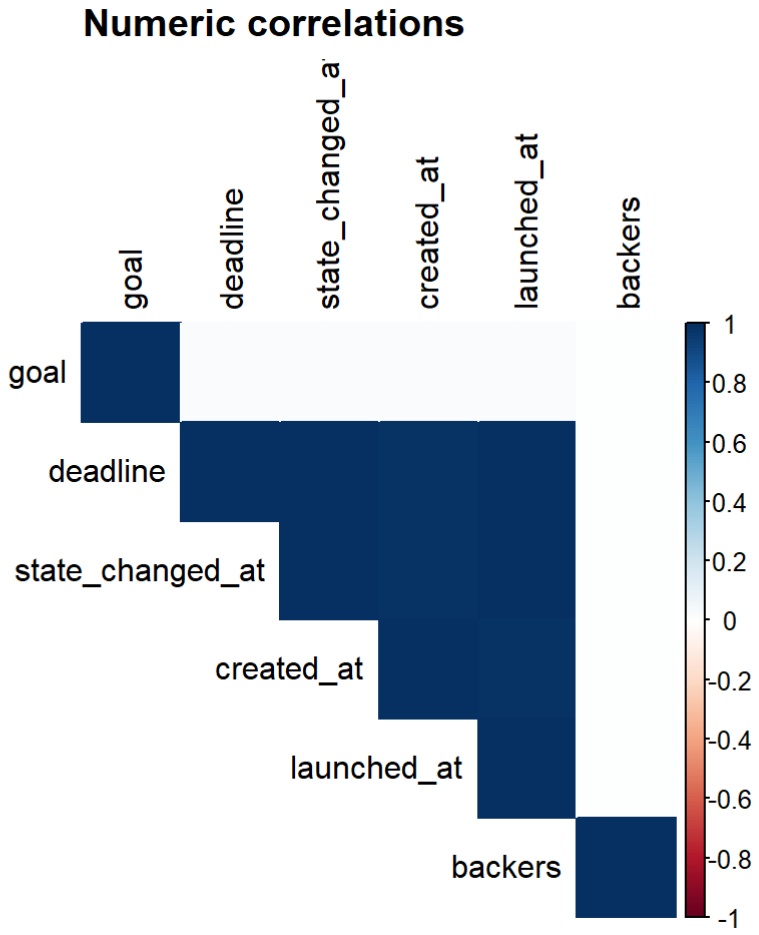
5 - Exploring and visualising variables

5.1 Correlations

Using a correlogram to give a high-level overview of correlations between numerical variables.

5.1.1 Numerical

```
#prep
kstrain4_nums <- as.data.frame(lapply(kstrain4, as.numeric))
kstrain4_nums <- kstrain4_nums[c(3,8:12)]
kstrain4_corr <- cor(kstrain4_nums)
#plot
corrplot(kstrain4_corr,
         method="color",
         type="upper",
         tl.col="black",
  title = "Numeric correlations",
  mar=c(0,0,2,0))
```

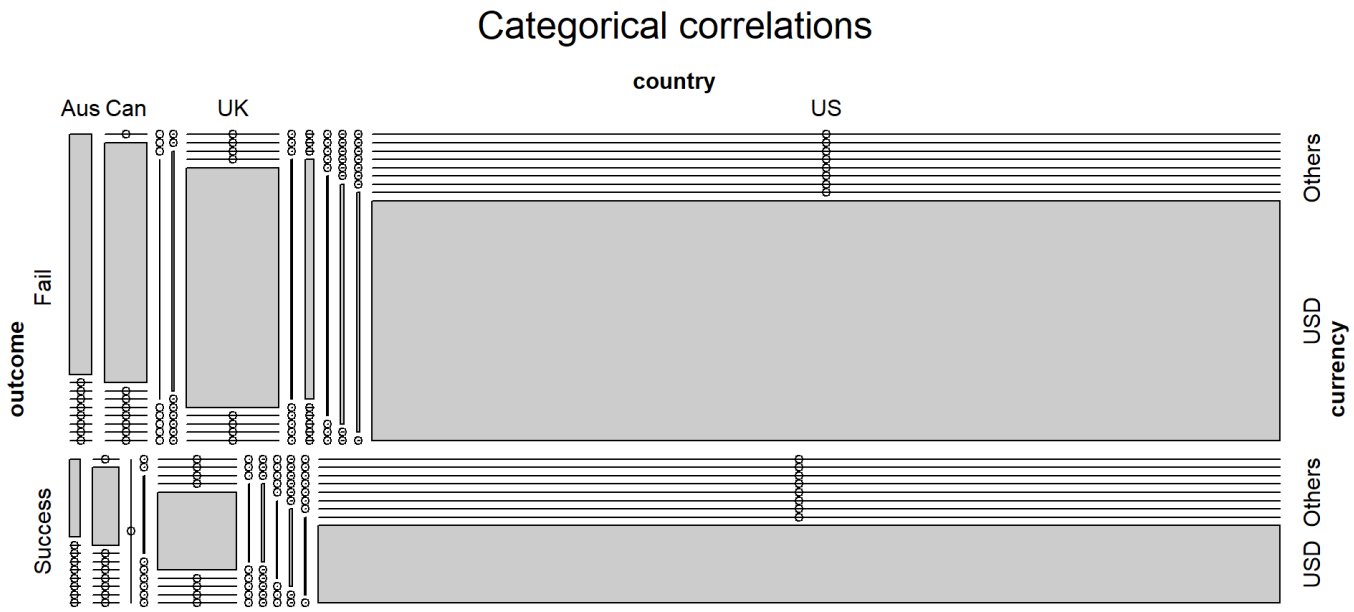


Strong correlations between time data, which makes sense: deadline will often equal state_changed_at unless the user cancels project early. People would also often launch on creation date. Surprisingly no correlation between goal and backers, perhaps implying that the size of the goal does not influence the size of a person's donation.

5.1.2 Categorical

Now using a mosaic chart to observe correlations between categorical data.

```
#prep
kstrain4_cat <- as.data.frame(lapply(kstrain4[c(6,7,13)], as.factor))

kstrain4_cat$country <- kstrain4_cat$country %>%
  as.character() %>%
  countryname(destination = "iso2c")

kst <- xtabs(~outcome + country + currency,
                kstrain4_cat)

#plot
mosaic(kst,
       main = "Categorical correlations", size=2,
       set_labels = list(outcome = c("Fail", "Success"),
                         country = c("Aus","Can","","","UK","","","","","","US"),
                         currency = c("","","","","Others","","","","USD")))
```



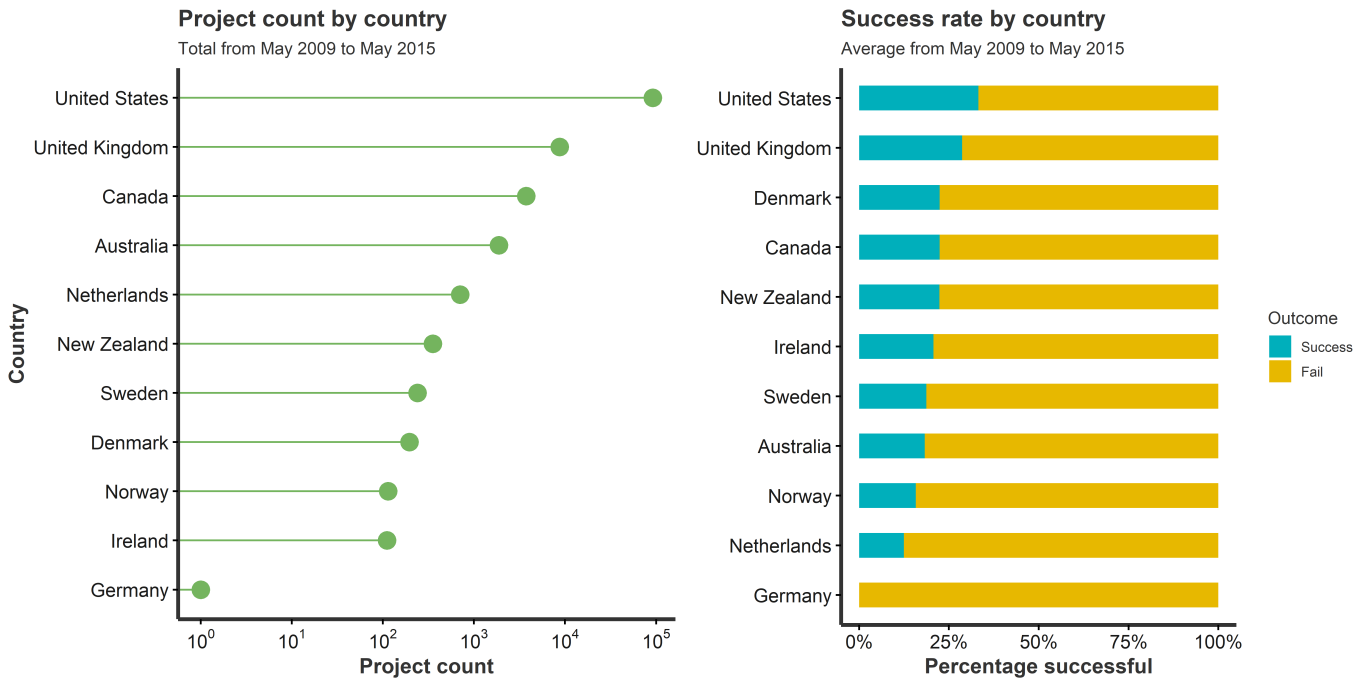
The mosaic plot gives us an idea of correlation between categorical variables based on their relative proportion to each another (this is not observing absolute counts). Some observations:

- More outcomes failed than succeeded, roughly 70:30.
- US is the most prevalent country; USD the most prevalent currency.
- US-based projects are slightly more correlated with success than failure.
- Using USD seems equally correlated with success and failure.

5.2 By country

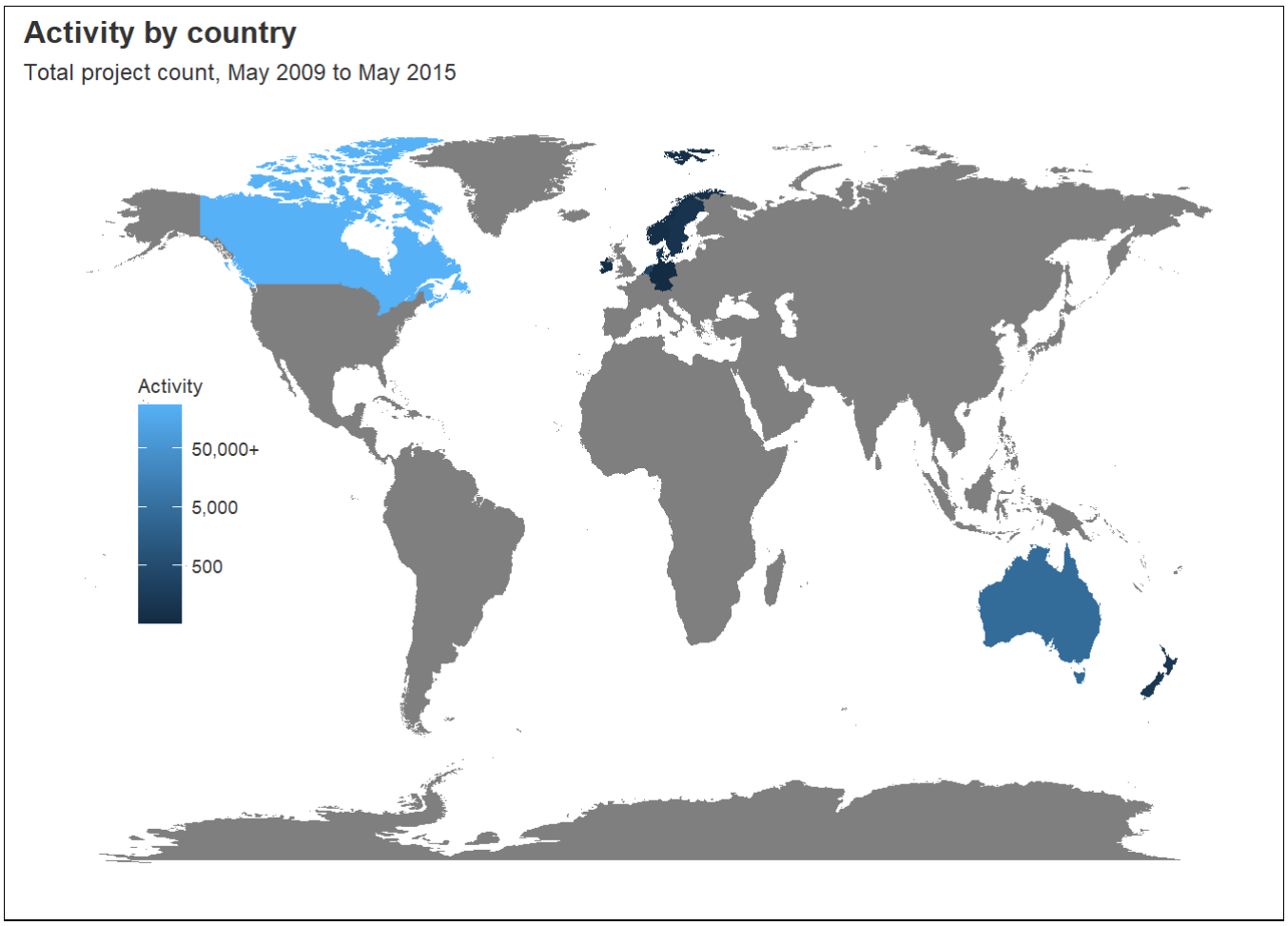
Observing total project count across countries on a log scale (due to US count being far higher than others), and then success rates across countries.

```
#prep co1
kstrain4.seg <- kstrain4 %>%
    group_by(country) %>%
    summarise(Freq=n())

#prep co2
prep.levels <- function(variable){
  names(sort(tapply(kstrain4$outcome == "1", variable, mean)))
}
successlvls <- prep.levels(kstrain4$country)

#plot
co1 <- ggplot(kstrain4.seg,
       aes(x=reorder(country,Freq),
           y = Freq)) +
    geom_segment(aes(xend=country, yend=0), colour="#74b45e") +
    geom_point(size=4, colour="#74b45e") +
    coord_flip() +
    scale_y_log10(breaks = scales::trans_breaks("log10", function(x) 10^x),
                  labels = scales::trans_format("log10", scales::math_format(10^.x))) +
    labs(title = "Project count by country",
         subtitle = "Total from May 2009 to May 2015",
         y = "Project count",
         x = "Country")

co2 <- ggplot(data = kstrain4, aes(factor(country, levels = successlvls), fill = outcome)) +
     geom_bar(position = "fill", width = 0.5) +
     labs(title = "Success rate by country",
          subtitle = "Average from May 2009 to May 2015",
          x = NULL,
          y = "Percentage successful",
          color = "test") +
     scale_y_continuous(labels = percent) +
    scale_fill_manual(labels = c("Fail","Success"),
                      values=c("#E7B800", "#00AFBB"),
                      name = "Outcome",
                      guide = guide_legend(reverse = TRUE)) +
    coord_flip() +
    theme(legend.key.size = unit(0.9,"line"))

grid.arrange(co1, co2, nrow = 1)
```



Further visualising with a global map:

```
#Prep
country_freq <- data.frame(table(kstrain4$country))
map_freq <- map_data('world') #load map data
map_join <- left_join(map_freq, country_freq, by = c('region' = 'Var1')) #join

#Plot
ggplot(map_join, aes( x = long, y = lat, group = group )) +
    geom_polygon(aes(fill = Freq), lwd = 1) +
    labs(fill = 'Activity',
         title = 'Activity by country',
         subtitle = 'Total project count, May 2009 to May 2015',
         x = NULL,
         y = NULL) +
  scale_fill_continuous(breaks = c(1000,2000,3000), labels = c("500","5,000","50,000+")) +
  theme(axis.ticks = element_blank(),
        axis.text = element_blank(),
        panel.grid = element_blank(),
        plot.background = element_rect(fill = 'white'),
        legend.position = c(.14,.50),
        legend.background = element_blank(),
        legend.key = element_blank(),
        legend.title = element_text(size = 8),
        legend.text = element_text(size = 7),
        axis.line.y = element_line(colour = "white"),
        axis.line.x = element_line(colour = "white"))
```

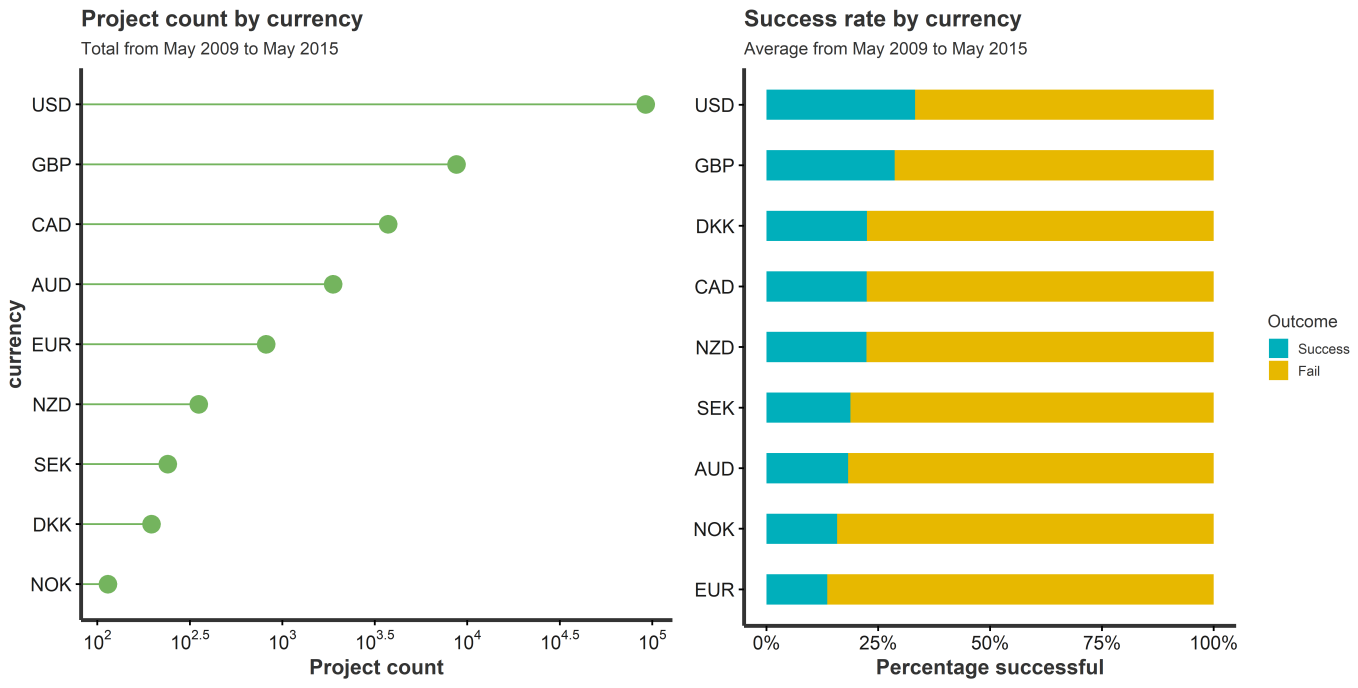


Far more projects with authors in the US compared to other countries. Aligns with notion of pronounced start-up culture in the US. Note, however, Kickstarter was founded in the US and only went global 4 years later, likely affecting regional presence.

US out on top, not only in overall project count, but also in success rate. Impressive, but other countries' success rates were not far off.

5.3 By currency

```
#prep co1
kstrain4.seg <- kstrain4 %>%
    group_by(currency) %>%
    summarise(Freq=n())

#prep co2. prior funtion used.
currency.success <- prep.levels(kstrain4$currency)

#plot
co1 <- ggplot(kstrain4.seg,
       aes(x=reorder(currency,Freq),
           y = Freq)) +
    geom_segment(aes(xend=currency, yend=0), colour="#74b45e") +
    geom_point(size=4, colour="#74b45e") +
    coord_flip() +
    scale_y_log10(breaks = scales::trans_breaks("log10", function(x) 10^x),
                  labels = scales::trans_format("log10", scales::math_format(10^.x))) +
    labs(title = "Project count by currency",
         subtitle = "Total from May 2009 to May 2015",
         y = "Project count",
         x = "currency")
co2 <- ggplot(data = kstrain4, aes(factor(currency, levels = currency.success), fill = outcom
e)) +
     geom_bar(position = "fill", width = 0.5) +
     labs(title = "Success rate by currency",
          subtitle = "Average from May 2009 to May 2015",
          x = NULL,
          y = "Percentage successful",
          color = "test") +
     scale_y_continuous(labels = percent) +
    scale_fill_manual(labels = c("Fail","Success"),
                      values=c("#E7B800", "#00AFBB"),
                      name = "Outcome",
                      guide = guide_legend(reverse = TRUE)) +
    coord_flip() +
    theme(legend.key.size = unit(0.8,"line"))
grid.arrange(co1, co2, nrow = 1)
```

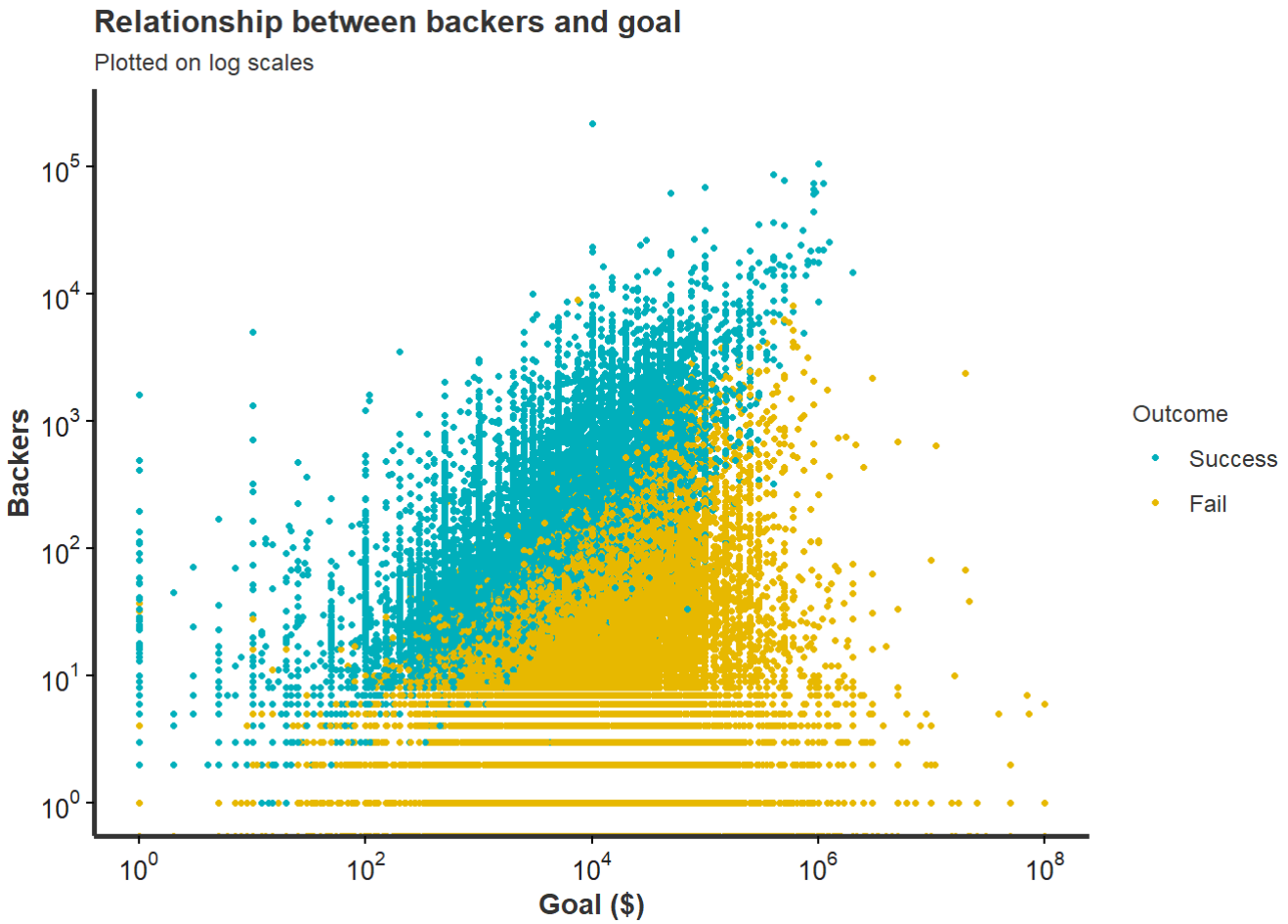


Corresponding to country, the USD had the largest project count and success rate, regardless of where it was used. Plots also show that currency frequencies had a tighter spread (count and rate) than countries, likely because other countries use the USD.

5.4 By backers and goal

Using log scale due to heavy right skew.

```
ggplot(kstrain4,aes(x=goal,y=backers,colour=outcome))+
        geom_point(shape=20, size=1.5)+
    xlim(0,NA)+
        labs(title = "Relationship between backers and goal",
             subtitle = "Plotted on log scales",
             x = "Goal ($)",
             y = "Backers",
             fill = "Outcome")+
        scale_x_log10(limits=c(1,NA),
                      breaks = scales::trans_breaks("log10", function(x) 10^x),
                   labels = scales::trans_format("log10", scales::math_format(10^.x)))+
    scale_y_log10(breaks = scales::trans_breaks("log10", function(x) 10^x),
                   labels = scales::trans_format("log10", scales::math_format(10^.x))) +
    scale_colour_manual(values=c("#E7B800", "#00AFBB"),
                        labels = c("Fail", "Success"),
                        name = "Outcome",
                        guide = guide_legend(reverse = TRUE)) +
  theme(legend.text = element_text(size=9))
```



Clustering shows projects of higher goals being harder to achieve, as expected. Higher goals also generally need more backers.

Turning point for success or fail is around where blue meets orange, and might be seen as the average of the lowest amount of backers required for a certain goal. That is, potential authors may size a market of backers and work back to optimise goal amount, or vice versa.

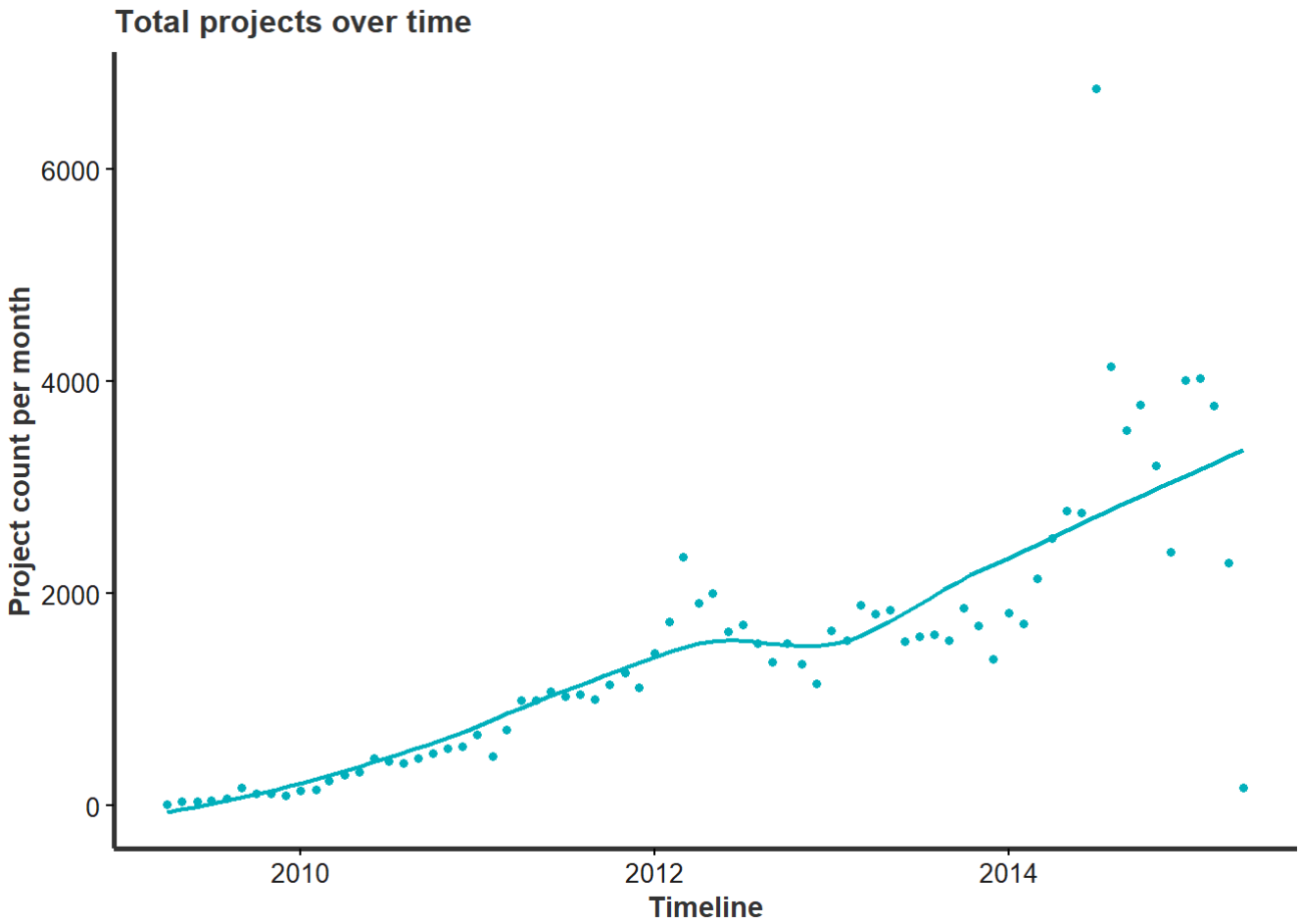
5.5 By time

We have 4 key variables to explore time-related trends. Recall that we previously converted them from unix time formats to date objects.

- 1. deadline: due date for project to reach goal.
- 2. state_changed_at: date project changed status (i.e. success, fail)
- 3. created_at: date that project was posted to Kickstarter.
- 4. launched_at: date that project went live.

5.5.1 Across years

```
#Extract dates
activity.month <- kstrain4[10]
#Show count for yyyy-mm
activity.month$created_at <- format(as.Date(activity.month$created_at), "%Y-%m")
count_activity.m <- count(activity.month, created_at)
#convert chr to date
count_activity.m$created_at <- anydate(count_activity.m$created_at)

# Plot
ggplot(data = count_activity.m,
               aes(x = created_at,
                   y = n)) +
    geom_point(colour = "#00AFBB",
               group=1, size = 1.3) +
    stat_smooth(lwd = 0.8, colour = "#00AFBB", se = FALSE) +
    labs(title = "Total projects over time") +
    xlab('Timeline') +
    ylab('Project count per month')
```

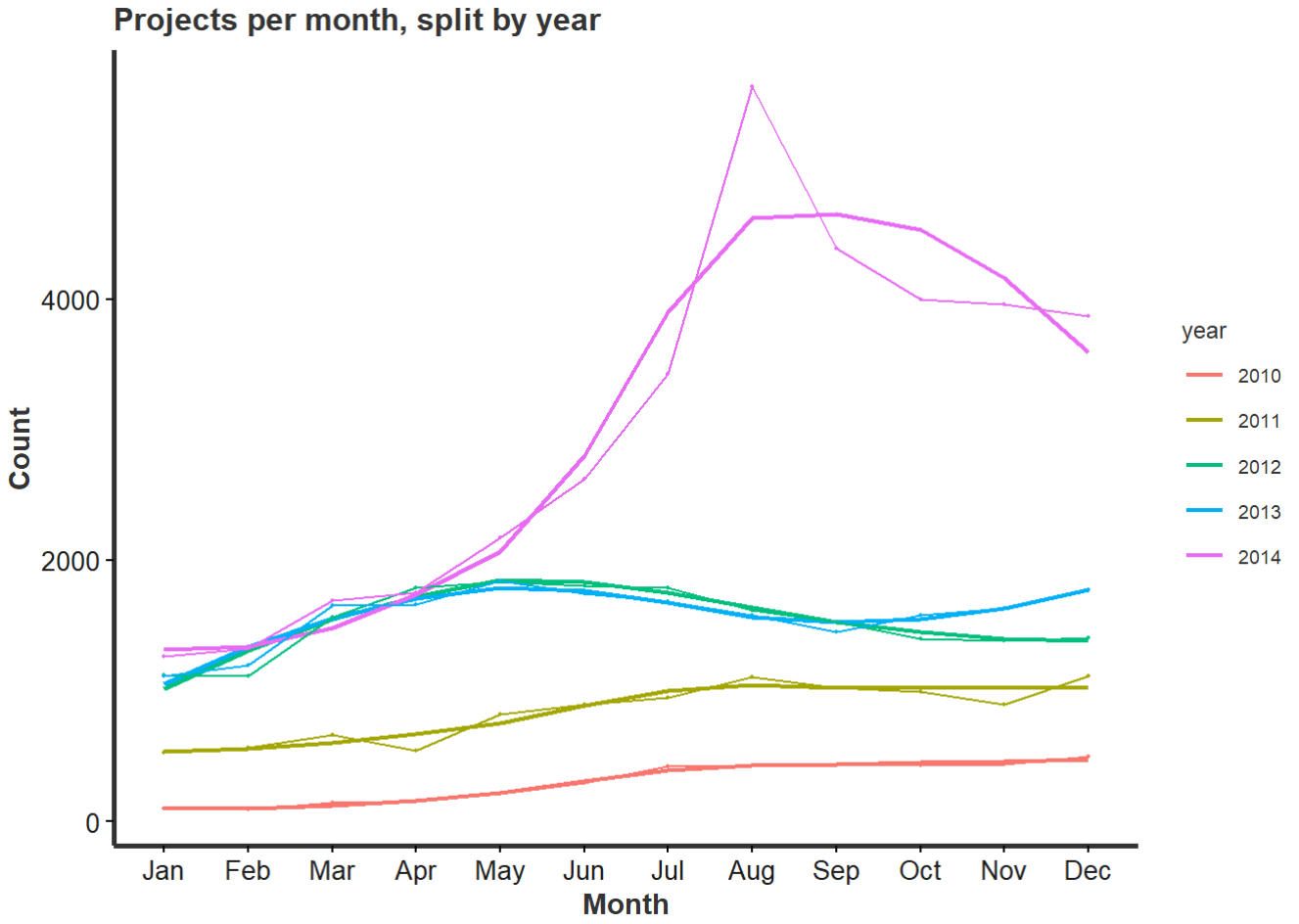


Positive linear increase in project counts from 2010 to 2014 inclusive.

From brief research, I was unable to find reasoning for the strong spike around mid-2014. Data still seems legitimate and aligns with activity on Google Trends. See here (https://trends.google.com/trends/explore?date=2010-01-01%202014-12-31&geo=US&q=%2Fm%2F0bwhy_7), which shows activity with reference to that high point. Unsure on low-point in late 2014.

Below is a similar plot, but instead observing each year's activity relative to one another.

```
monthsrate <- kstrain4[c(9)]#Extract dates

monthsrate$yearmo <- format(as.Date(monthsrate$state_changed_at), "%Y-%m")#Form new column to
show yyyy-mm

monthsratefreq <- data.frame(table(monthsrate$yearmo))#Create freq table

monthsratefreq$year <- format(str_sub(monthsratefreq$Var1, 1, 4))#form year col
monthsratefreq$month <- format(str_sub(monthsratefreq$Var1, -2))#Form month col
monthsratefreq$year <- as.character(monthsratefreq$year)
monthsratefreq1 <- subset(monthsratefreq, year!="2009" & year!="2015")#exclude 2009 & 2015 (n
on-whole years)

#plot
ggplot(data = monthsratefreq1,
        aes(x = month,
            y = Freq,
            group = year,
            colour = year)) +
  geom_smooth(size = 0.75, se = FALSE) +
  geom_line(size = 0.5) +
  geom_point(lwd = 0.3) +
     labs(title = "Projects per month, split by year",
          x = "Month",
          y = "Count") +
     scale_x_discrete(labels = c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct",
"Nov","Dec"))
```



Upward progression of the lines show the growth in project count across years, in line with previous plot. Highly similar progression from January to May in years 2012 to 2014 - possible seasonal trend that may be repeated going forward.

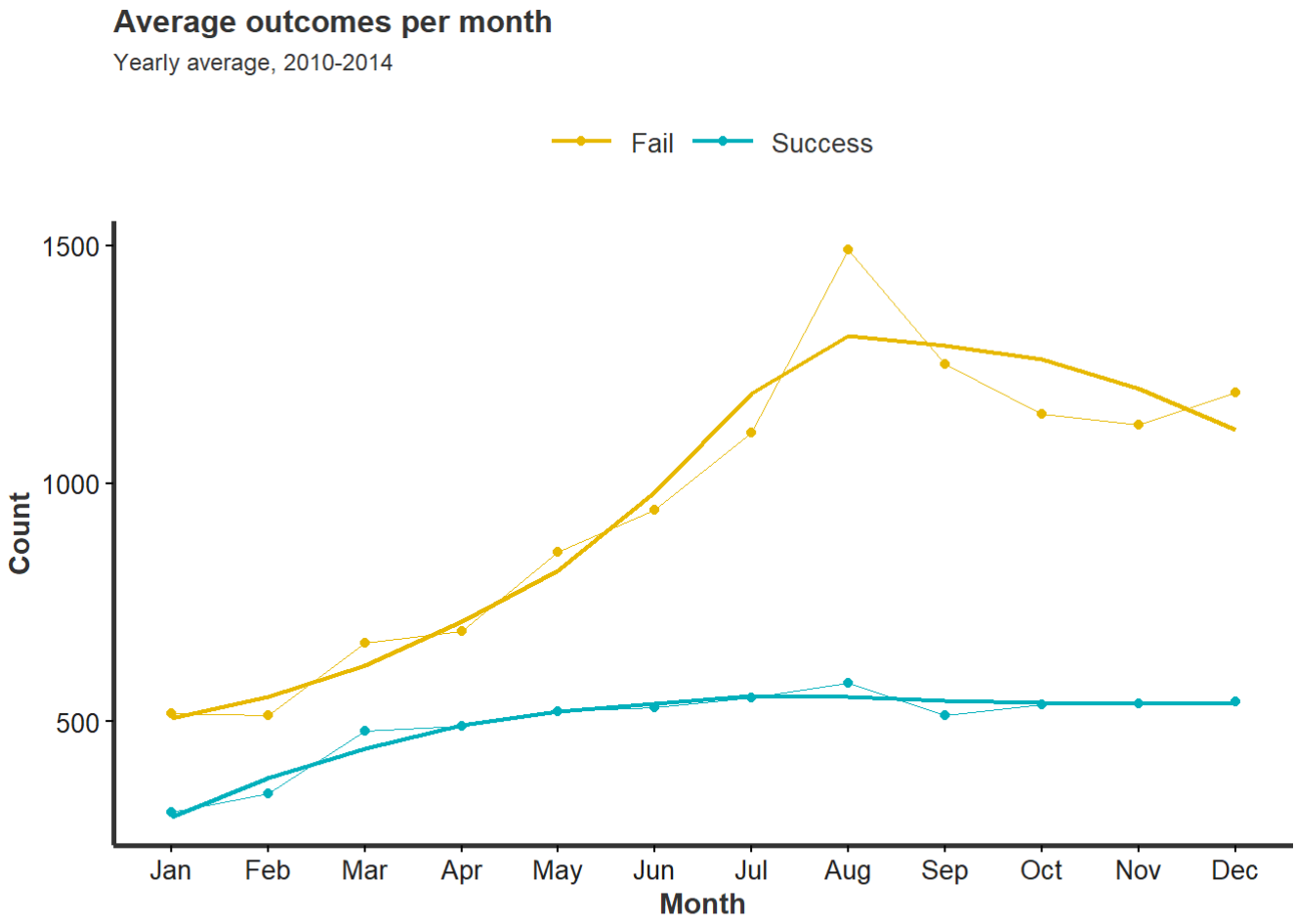
What we see more clearly in this graph is the stagnation from 2012 to 2013, before the impressive surge around May 2014; perhaps due to the early 2014 change in management (https://www.vox.com/2014/3/6/11624254/at-a-billion-dollars-pledged-kickstarter-ceo-yancey-strickler-reflects).

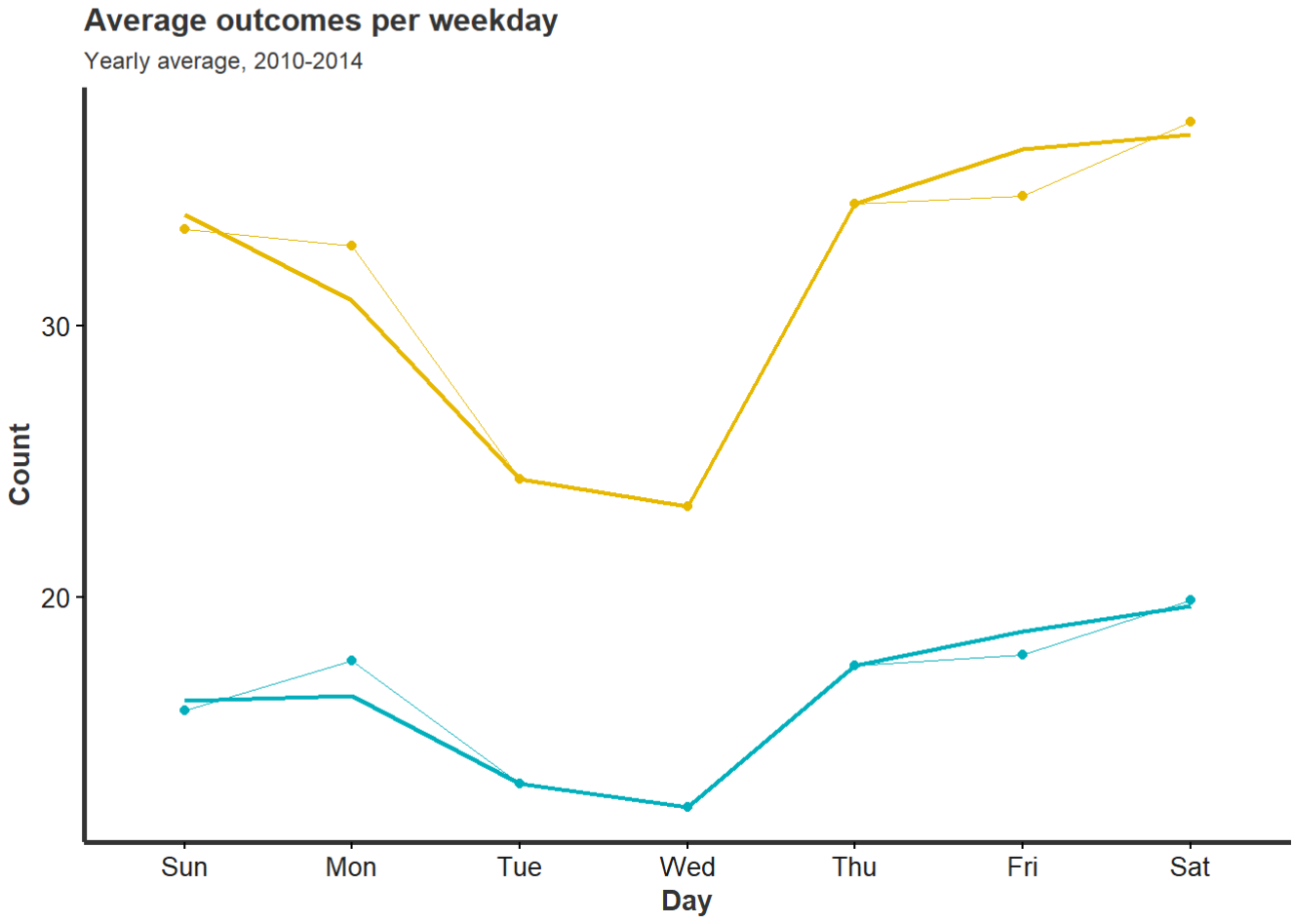
5.5.2 Between months and days

To observe activity levels between months and days, we will observe variables state_changed_at and outcome, again only for full-years 2010 - 2014.

```
#prep o1
months <- kstrain4[c(9,13)]

#create function to limit dates:
lim.dates <- function(df,col){
  subset(df, col >= '2010-01-01' & col <= '2014-12-31')
}
months0 <- lim.dates(months, months$state_changed_at)

months1 <- months0
months1$state_changed_at <- strftime(months0$state_changed_at, "%m")#convert to month only
monthsfreq <- data.frame(table(months1$state_changed_at, months1$outcome)) #new df for freqs
monthsfreq$Freq <- (monthsfreq$Freq)/5 #avg freq count

#prep o2
days <- kstrain4[c(9,13)]

#reapply function lim.dates
days0 <- lim.dates(days,days$state_changed_at)

days0$state_changed_at <- wday(days0$state_changed_at, label = TRUE) #convert to day only
daysfreq <- data.frame(table(days0$state_changed_at, days0$outcome)) #new df for freqs
daysfreq$Freq <- (daysfreq$Freq/(5*52)) #avg freq count

#plot
o1 <- ggplot(data = monthsfreq,
             aes(x = Var1,
                 y = Freq,
                 group = Var2,
                 colour = Var2)) +
    geom_line(lwd=0.3) +
    geom_point() +
    stat_smooth(lwd = 0.8, se = FALSE) +
    labs(title = "Average outcomes per month",
         subtitle = "Yearly average, 2010-2014",
         x = "Month",
         y = "Count") +
    scale_x_discrete(labels = c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct",
"Nov","Dec")) +
    scale_colour_manual(values=c("#E7B800", "#00AFBB"),
                        labels = c("Fail", "Success"),
                        name = "") +
    theme(legend.position = "top",
          legend.text = element_text(size=10),
          legend.key.size = unit(2,"line"))


o2 <- ggplot(data = daysfreq,
             aes(x = Var1,
                 y = Freq,
                 group = Var2,
                 colour = Var2)) +
    geom_line(lwd=0.3) +
    geom_point() +
    stat_smooth(lwd = 0.8, se = FALSE) +
    labs(title = "Average outcomes per weekday",
         subtitle = "Yearly average, 2010-2014",
```

```
          x = "Day",
          y = "Count") +
    scale_colour_manual(values=c("#E7B800", "#00AFBB"),
                        labels = c("Fail", "Success"),
                        name = "Outcome") +
    theme(legend.position = "none")

grid.arrange(o1, o2, nrow=2)
```





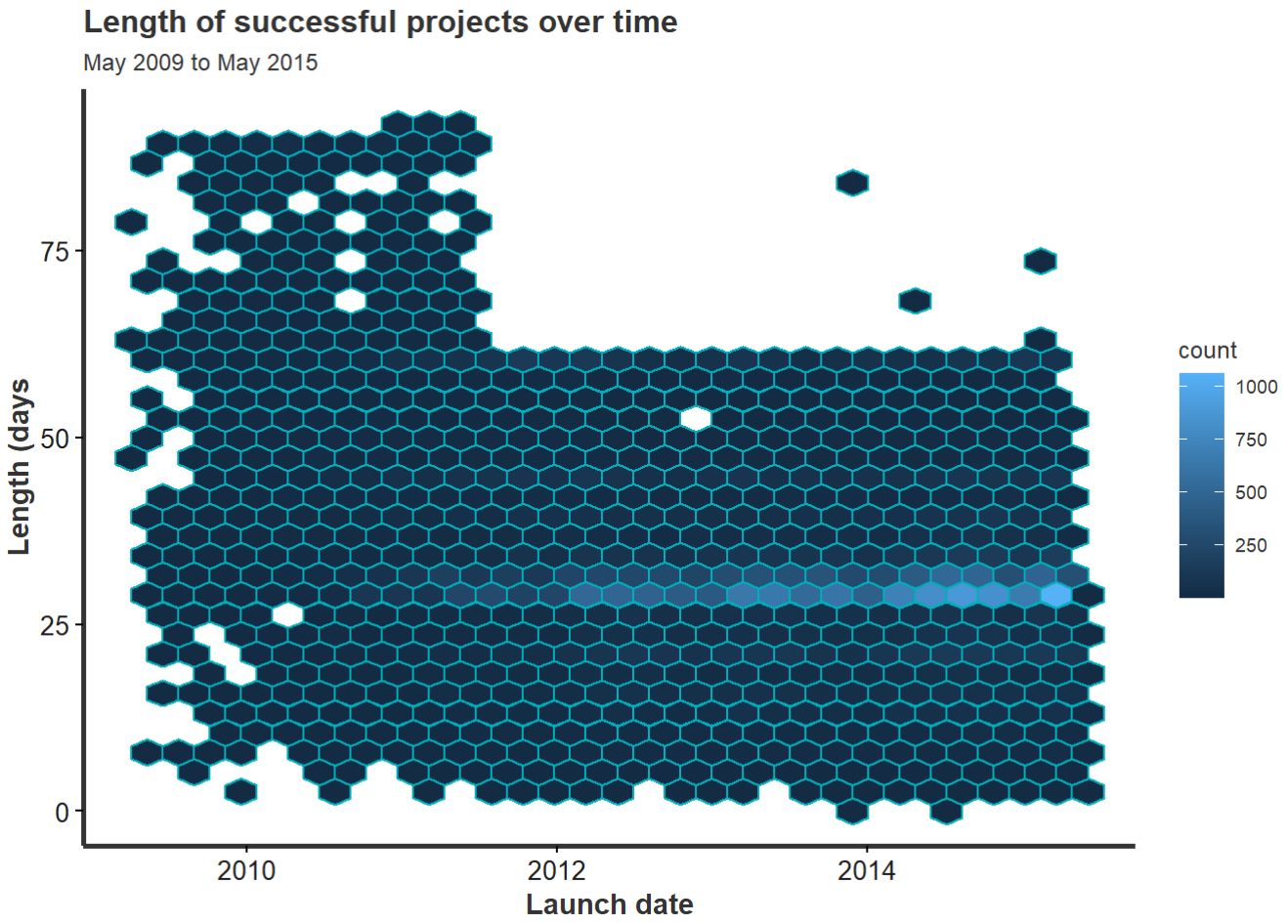
Heightened activity in second half of calendar years. Observing the gap between lines, we see that Feb/Mar had the highest success rates (smaller gap = higher portion of successful projects).

By similar observation, we see most activity occurred Thursday to Saturday, with dips from Monday to Wednesday. Highest success rates were on Tuesdays and Wednesdays.

5.6 By length of project

We can find length of project by subtracting launched_at from state_changed_at. We will observe this and filter for only successful projects.

```
#prep
kstrain5 <- kstrain4[c(9,11,13)]
kstrain5$outcome <-  as.character(kstrain5$outcome)
kstrain5 <- subset(kstrain5, kstrain5$outcome == "1") #successful only
kstrain5$length <- kstrain5$state_changed_at - kstrain5$launched_at #length in days


#plot
ggplot(data = kstrain5,
       aes(x = launched_at, y = length)) +
  stat_binhex(colour = "#00AFBB") +
  labs(title = "Length of successful projects over time",
       subtitle = "May 2009 to May 2015",
       x = "Launch date",
       y = "Length (days") +
  theme(text = element_text(size = 9, colour = "grey20"),
                    axis.text = element_text(size = 10, colour = "grey10"),
                    axis.title = element_text(size=11,face="bold"),
                    plot.title = element_text(size=12,face="bold"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.line = element_line(colour = "grey20",
                      size = 1, linetype = "solid"),
    axis.ticks = element_line(size = 0.5))
```



Around mid 2011 we see a noticeable drop in the variance of successful project lengths, reducing to about 5 to 60 days.

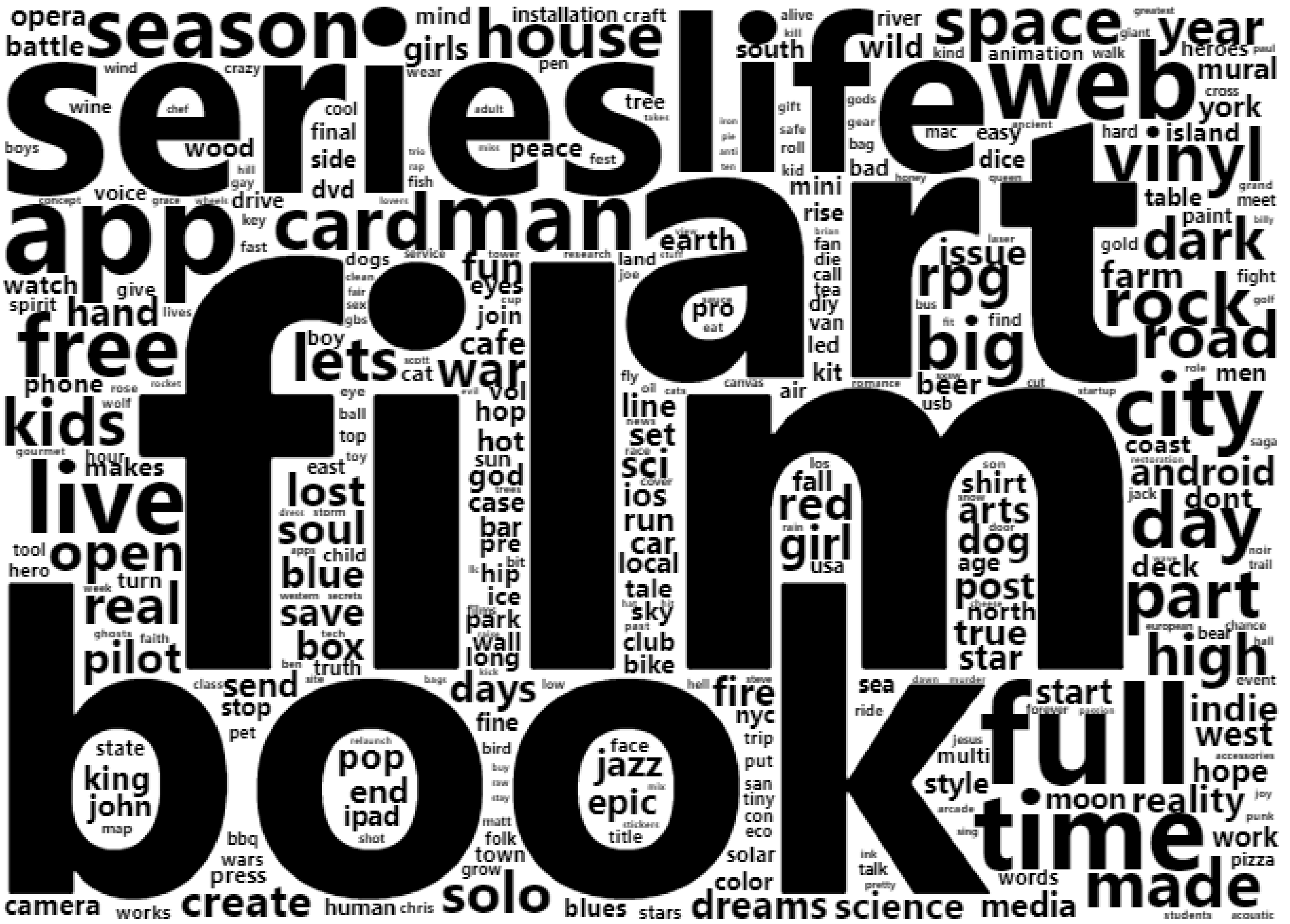
We then see a trend forming around 2012 onward, where the largest portion of successful projects had a length of about 30 days.

5.7 Text analysis

The dataset presents us with 3 columns for text analysis: name, desc and keywords. Only keywords will be analysed as it is the most consistent in format and cleanliness.

The word cloud below is based on all projects, with a larger size indicating greater relative frequency. Hover over words for total count. Note that there may be slight differences in word clouds based on browser window size (i.e. can resize window then refresh).

```
#prep
keywords <- strsplit(kstrain4$keywords, split = "-") #select, xform
keywords_0 <- data.frame(unlist(keywords))
keywords_1 <- paste(keywords_0$unlist.keywords., collapse=" ") #combine txt
#Form & clean corpus
keywords.corpus <- Corpus(VectorSource(keywords_1))
keywords.corpus = keywords.corpus %>%
    tm_map(removeNumbers) %>%
    tm_map(removePunctuation) %>%
    tm_map(stripWhitespace) %>%
    tm_map(content_transformer(tolower)) %>%
    tm_map(removeWords, stopwords("english")) %>%
    tm_map(removeWords, stopwords("SMART"))
#Form matrix from corpus
tdm <- TermDocumentMatrix(keywords.corpus) %>%
    as.matrix()
words = sort(rowSums(tdm), decreasing = TRUE)
keywords.df <- data.frame(word = names(words), freq = words)

#plot fn
plot.wc <- function(x){
  wordcloud2(x,
             color = "black",
             backgroundColor = "white",
             size = 2.5,
             minSize = 5,
             rotateRatio = 0)
}

plot.wc(keywords.df)
```



The cloud gives us an idea of popular themes/categories in projects (successful or otherwise), and how they were described. For a potential new author of a project, they may use a live version of this to indicate saturation of certain themes/categories/topics in the crowdfunding market.

Below, we run the same code but subsetting for only successful projects.

```
#Selection
obstext_0 <- kstrain4[c(4,13)]
obstext_suc <- obstext_0 %>%
group_by(keywords) %>%
filter(any(outcome == "1"))
obstext_suc_1 <- obstext_suc[-2]
#Transformation
keywords_suc <- strsplit(obstext_suc_1$keywords, split = "-")
keywords_suc_0 <- data.frame(unlist(keywords_suc))
keywords_suc_1 <- paste(keywords_suc_0$unlist.keywords_suc., collapse=" ") #combine text
keywords_suc.corpus <- Corpus(VectorSource(keywords_suc_1))
#Cleaning corpus with tm_map
keywords_suc.corpus = keywords_suc.corpus %>%
    tm_map(removeNumbers) %>%
    tm_map(removePunctuation) %>%
    tm_map(stripWhitespace) %>%
    tm_map(content_transformer(tolower)) %>%
    tm_map(removeWords, stopwords("english")) %>%
    tm_map(removeWords, stopwords("SMART"))
#Form matrix from corpus
tdm <- TermDocumentMatrix(keywords_suc.corpus) %>%
    as.matrix()
words = sort(rowSums(tdm), decreasing = TRUE)
keywords_suc.df <- data.frame(word = names(words), freq = words)

#plot w/prior function
plot.wc(keywords_suc.df)
```

Observing differences in word size between the two clouds may indicate that films and games had a higher success rate than, for example, books, art or tours. *debut* making an interestingly large appearance whilst not noticeable in the total projects word cloud (again, dependent on browser size).

6 - Summary

That was fun. After some cleaning and transformations, we were able to explore and visualise correlations and trends by country, currency, the amount of backers, the goal amount, time and text.

Throughout, we saw trends that may assist:

- Kickstarter itself: such as with marketing or analysis of areas and periods of time to focus on.
- Project creators: such as knowing that certain times of the year (down to the day) have clearly had more success than others. Or, knowing what types of projects are trending, or what country has been most successful, or the most successful project lengths.
- Donors: similarly to creators, may consider that certain times of the year show more success, and so may put their effort into browsing and donating at that time, where perhaps there is a higher likelihood that their donation will lead to a successful funding.

6.1 References

- (n.d.). Retrieved September 1, 2020, from https://rkabacoff.github.io/datavis/Models.html (https://rkabacoff.github.io/datavis/Models.html)
- Abhimotgi. (2020, July 15). Abhimotgi/dataslice. Retrieved August 27, 2020, from https://github.com/abhimotgi/dataslice/blob/master/R/Word (https://github.com/abhimotgi/dataslice/blob/master/R/Word) Clouds in R.R
- Kassambara. (2017, November 17). Plot Time Series Data Using GGPlot. Retrieved August 20, 2020, from http://www.sthda.com/english/articles/32-r-graphics-essentials/128-plot-time-series-data-using-ggplot/ (http://www.sthda.com/english/articles/32-r-graphics-essentials/128-plot-time-series-data-using-ggplot/)
- Kickstarter, L. (2017, June 20). Funding Successful Projects on Kickstarter. Retrieved August 17, 2020, from https://www.kaggle.com/codename007/funding-successful-projects (https://www.kaggle.com/codename007/funding-successful-projects)
- PHPBoost. (n.d.). Visualize correlation matrix using correlogram. Retrieved September 1, 2020, from http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram (http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram)
- Prabhakaran, S. (2017). Top 50 ggplot2 Visualizations - The Master List (With Full R Code). Retrieved August 28, 2020, from http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html (http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html)
- Sharp Sight. (2019, November 15). Mapping oil production by country using R. Retrieved August 20, 2020, from https://www.sharpsightlabs.com/blog/map-oil-production-country-r/ (https://www.sharpsightlabs.com/blog/map-oil-production-country-r/)