Kickstarter Statistics

Data 602 Final Project

Jackson Meier, Max Paterson, Noah Seminoff, Weiwei Wang

2024-10-01

Contents

Introduction

The objective of this project will be to explore and analyze a dataset of over 300 thousand projects submitted to the website Kickstarter ("Kickstarter Projects" 2017). This will be done through the use of a variety statistical techniques, such as exploratory data analysis, hypothesis testing, and regression analysis.

The basic concept of Kickstarter is to allow users to raise money for creative projects in a method known as "crowd funding". Specifically, users can submit an idea for a potential project along with a money goal and a date by which they need to reach the goal. Other users can browse through a list of projects, and "back" a project they find interesting by pledging money to it, typically in return for a reward when the project is complete (such as a discounted or free copy of the final project, an exclusive version, etc.). If a project reaches its goal by the deadline, the creator gets the pledged money (minus the cut taken by the website itself) and the project is listed as a success. If they do not reach the goal, the money is returned to each backer. This format allows people to invest in interesting ideas with no downside or risk of losing money if the idea falls through.

The dataset contains information on the name and category of each project, country (and currency) of origin, the start and deadline dates, whether they succeeded, their money goal, how many people contributed, and how much money they raised. There are also additional columns that convert each of the money amounts to US dollars.

Our research question is looking into the relationship between the number of kickstarter projects launched in a given month and their success rate, to see how they correlate to each other. After doing that, we will have insights into how the density of project launches affects the chance of success for the project. A linear regression model will be built to examine and describe the relationship between the number of projects launched in a given month and their success rate, and we will test if it can effectively predict future outcomes based on data.

Exploratory Data Analysis

Setup

Firstly, we need to read the dataset file into R,

```
df18 = read.csv("ks-projects-201801.csv", header = TRUE, sep = ",")
```

EDA

We can look at the first few rows in order to see what we have to work with.

```
head(df18, 4)
```

```
## # A tibble: 4 x 15
##         ID name  category main_category currency deadline  goal launched pledged
##      <int> <chr> <chr>    <chr>         <chr>    <chr>    <dbl> <chr>      <dbl>
## 1   1.00e9 The ~ Poetry   Publishing    GBP      2015-10~  1000 2015-08~       0
## 2   1.00e9 Gree~ Narrati~ Film & Video  USD      2017-11~ 30000 2017-09~    2421
## 3   1.00e9 Wher~ Narrati~ Film & Video  USD      2013-02~ 45000 2013-01~     220
## 4   1.00e9 Tosh~ Music    Music         USD      2012-04~  5000 2012-03~       1
## # i 6 more variables: state <chr>, backers <int>, country <chr>,
## #   usd.pledged <dbl>, usd_pledged_real <dbl>, usd_goal_real <dbl>
```

In addition to columns describing the monetary goal (`goal`), and how much was received (`pledged`) in local currency, there are three columns that convert this to US dollars (USD). The first of these (`usd.pledged`) is conversion done by the website itself, while the other two (`usd_pledged_real` and `usd_goal_real`) are conversions done by the dataset's uploader. We can do a quick check to see how accurate these conversions are by comparing the amount pledged in local currency against the amount pledged converted to USD, specifically for projects already in USD. These two values should be identical if the conversion is being done correctly.

For the websites conversion, the number of errors is:

```
# count of conversion errors for usd.pledged
# i.e., count of the number of rows where converting from USD to USD changes
# the value by more than 1 dollar
```

```
df18 %>% filter(currency == "USD") %>%
  filter(abs(pledged - usd.pledged) > 1) %>% count() %>% pull
```

```
## [1] 41838
```

And for the other conversion, the number of errors is:

```
# count of conversion errors for usd_pledged_real
df18 %>% filter(currency == "USD") %>%
  filter(abs(pledged - usd_pledged_real) > 1) %>% count() %>% pull
```

```
## [1] 0
```

That is, there are over 40 thousand rows in `usd.pledged` for which the conversion to USD changes the dollar amount, while in `usd_pledged_real` this number is zero. Thus, we can drop `usd.pledged` and just use `usd_pledged_real` instead.

```
# drop row with wrongful conversion
df18$usd.pledged = NULL
```

Numerical Summaries

We can also explore the data with some summary statistics.

```
## define useful functions

# mode function
mode = function(x){
  ux = unique(x)
  ux[which.max(tabulate(match(x,ux)))]
}
# function to get number of occurrences of the mode
modeoccurances = function(x){
  modeval = mode(x)
  count(x == modeval)
}
# function to get percent of rows which equal the mode
modeprop = function(x){
  modeoccurances(x)/length(x)
}

## summary statistics: categorical
df18 %>% summarize( # take dataset and summarize it by the following criteria;
  across(
    # for these columns;
```

```
    c("main_category", "category", "country", "currency", "state"),
    # apply these summary functions;
    c("_mode" = mode,
      "_mode_count" = modeoccurances,
      "_mode_prop" = \(x) round(modeprop(x),2))
    )) %>%
  # format to look nice
  pivot_longer(everything(), names_sep = "__",
               names_to = c("variable",".value"))
```

```
## # A tibble: 5 x 4
##   variable      mode           mode_count mode_prop
##   <chr>         <chr>               <int>     <dbl>
## 1 main_category Film & Video        63585      0.17
## 2 category      Product Design      22314      0.06
## 3 country       US                 292627      0.77
## 4 currency      USD                295365      0.78
## 5 state         failed             197719      0.52
```

From the categorical data, we see that the most common general category of project is Film & Video, and sub-category overall is Product Design (a sub-category of Design). Just over 3/4 of projects are from the United States, and about the same proportion are asking for pledges in US dollars. Additionally, we can see the most common outcome for a project is to fail to meet the goal, with 52 percent.

```
## summary statistics: numeric
df18 %>% summarize( # take dataset and summarize it by the following criteria;
  across(
    # for these columns;
    c("backers", "usd_pledged_real", "usd_goal_real"),
    # apply these functions;
    c("_min" = min,
      "_Q1" = \(x) quantile(x, 0.25),
      "_median" = median,
      "_Q3" = \(x) quantile(x, 0.75),
      "_max" = max,
      "_mean" = mean,
      "_mode" = mode,
      "_mode_prop" = modeprop,
      "_IQR" = \(x) IQR(x))
    )) %>%
  # format to look nice
  pivot_longer(everything(), names_sep = "__",
               names_to = c("variable",".value"))
```

```
## # A tibble: 3 x 10
```

```
##   variable           min    Q1 median    Q3     max    mean  mode mode_prop   IQR
##   <chr>            <dbl> <dbl>  <dbl> <dbl>   <dbl>  <dbl> <dbl>     <dbl> <dbl>
## 1 backers            0        2    12     56  2.19e5   106.      0    0.147     54
## 2 usd_pledged_real   0       31   624.  4050  2.03e7  9059.      0    0.139   4019
## 3 usd_goal_real      0.01  2000  5500  15500  1.66e8 45454.   5000    0.0638 13500
```

From the numerical data, we see that a projects exist with anywhere from 0 to 200 thousand backers, but the median number of backers is 12 (and the most common result is zero backers, at 14% of all projects). Because the mean is 106, it is clear there are some large outliers dragging the mean much higher. We can see from the IQR that the middle 50% of data have a spread of around 50 backers. In terms of goal dollars (converted to USD), people set goals anywhere from 1 cent to $166 million, although the median is $5500 (and the most common is $5000). The middle half of data are spread across a range of $13,500. Similarly to backers, we see that the mean ($45,000) is significantly higher than the median, hinting that there are a few large outliers. For money received, projects have gotten anywhere from $0 to $20 million, and the median project makes about $600 (and the most common is zero). In addition, the middle half of data are spread across around $4000.

```
# summary statistics: dates
df18 %>% summarise(
  across(
    # across these columns;
    c("launched", "deadline"),
    # apply these functions;
    c("_min" = min,
      "_max" = max,
      "_mode" = mode,
      "_mode_count" = modeoccurances)
    )) %>%
  # format to look nice
  pivot_longer(everything(), names_sep = "__",
               names_to = c("variable",".value"))
```

```
## # A tibble: 2 x 5
##   variable min                 max                 mode                mode_count
##   <chr>    <chr>               <chr>               <chr>                    <int>
## 1 launched 1970-01-01 01:00:00 2018-01-02 15:02:31 1970-01-01 01:00:~          7
## 2 deadline 2009-05-03          2018-03-03          2014-08-08                 705
```

In the time data summary, we can see that the latest launch date is early January 2018 (which is likely when the dataset was downloaded, because the file name ends with `201801`). The earliest (and most common) launch date is 1970, which is likely a conversion of the NA value of 0 into a date (on computers, time is often represented as seconds after January 1st 1970, so zero will be converted to this date). For deadlines, the earliest is early May 2009 (roughly when the website launched), and the latest is March 2018. The most common deadline is August 8th 2014, which is likely just a coincidence, or when the website was most popular.

In general, this summary tells us that while a few large successful outliers exist, it appears that most projects are relatively small, and don't make very much money, if any.

We can generate a few new columns that could be useful, including the total length of the project (days between `launched` and `deadline`), and the money per backer for each project.

```
## Generate new columns.

# avg money per backer
df18 = df18 %>% mutate(pledged_per_backer_usd_real = usd_pledged_real/backers)

# when backers is zero, pledged per is infinite. replace with zero
df18["pledged_per_backer_usd_real"][sapply(df18[
  "pledged_per_backer_usd_real"],is.infinite)] = 0

# length of time to deadline in days
df18 = df18 %>% mutate(project_days =
                         as.integer(difftime(deadline,
                                             launched, units = "days")))
```

We can also break down the data by category.

```
# unique main categories
df18 %>% pull(main_category) %>% unique()
```

```
##  [1] "Publishing"   "Film & Video" "Music"        "Food"         "Design"
##  [6] "Crafts"       "Games"        "Comics"       "Fashion"      "Theater"
## [11] "Art"          "Photography"  "Technology"   "Dance"        "Journalism"
```

```
# unique sub categories for tech
df18 %>% filter(main_category == "Technology") %>% pull(category) %>% unique()
```

```
##  [1] "Hardware"          "Software"          "Gadgets"
##  [4] "Web"               "Apps"              "Technology"
##  [7] "Flight"            "Makerspaces"       "Fabrication Tools"
## [10] "Sound"             "Wearables"         "DIY Electronics"
## [13] "Camera Equipment"  "3D Printing"       "Space Exploration"
## [16] "Robots"
```

We see there are 15 main categories, and a number of sub-categories for each.

We can also generate some summary statistics, grouped by category:

```
# print out some summary statistics
# summarize number of projects, success, length in days, number of backers
df18 %>%
  group_by(main_category) %>%
```

```
  summarise("Count" = n(),
            "Success_Rate" = round(100 *  sum(state == "successful") / n(),2),
            "Avg_Days" = round(mean(project_days),2),
            "Avg_Backers" = round(mean(backers),0)
            ) %>%
  arrange(desc(Count)) # sort result
```

```
## # A tibble: 15 x 5
##    main_category Count Success_Rate Avg_Days Avg_Backers
##    <chr>         <int>        <dbl>    <dbl>       <dbl>
##  1 Film & Video  63585         37.2     35.6          66
##  2 Music         51918         46.6     35.6          52
##  3 Publishing    39874         30.8     34.3          56
##  4 Games         35231         35.5     32.5         322
##  5 Technology    32569         19.8     35.4         164
##  6 Design        30070         35.1     34.9         241
##  7 Art           28153         40.9     33.0          42
##  8 Food          24602         24.7     34.0          54
##  9 Fashion       22816         24.5     32.8          61
## 10 Theater       10913         59.9     34.9          47
## 11 Comics        10819         54       34.1         135
## 12 Photography   10779         30.7     33.8          40
## 13 Crafts         8809         24.0     31.5          27
## 14 Journalism     4755         21.3     34.5          38
## 15 Dance          3768         62.0     33            43
```

```
# summarize subcategories
df18 %>%
  group_by(main_category) %>%
  summarise("Mode_subcat" = mode(category),
            "Mode_Percent" = round(modeoccurances(category)/n(),2)
            ) %>%
  arrange(desc(Mode_Percent))
```

```
## # A tibble: 15 x 3
##    main_category Mode_subcat    Mode_Percent
##    <chr>         <chr>                 <dbl>
##  1 Design        Product Design         0.74
##  2 Theater       Theater                0.65
##  3 Dance         Dance                  0.62
##  4 Crafts        Crafts                 0.53
##  5 Photography   Photography            0.53
##  6 Food          Food                   0.47
##  7 Comics        Comics                 0.46
##  8 Games         Tabletop Games         0.4
##  9 Fashion       Fashion                0.37
```

```
## 10 Journalism    Journalism             0.37
## 11 Music         Music                  0.3
## 12 Art           Art                    0.29
## 13 Film & Video  Documentary            0.25
## 14 Publishing    Fiction                0.23
## 15 Technology    Technology             0.21
```

```
# summarize some dollar amounts (in USD)
df18 %>%
  group_by(main_category) %>%
  summarise("Median_Goal" = round(median(usd_goal_real),2),
            "Median_Pledge" = round(median(usd_pledged_real),2),
            "Median_Pledge_per_Backer" = round(
              median(pledged_per_backer_usd_real, na.rm = TRUE),2)) %>%
  arrange(desc(Median_Pledge))
```

```
## # A tibble: 15 x 4
##    main_category Median_Goal Median_Pledge Median_Pledge_per_Backer
##    <chr>              <dbl>         <dbl>                    <dbl>
##  1 Design             10000          1923                     56.6
##  2 Dance               3300          1840.                    60.5
##  3 Theater             3300          1550                     57.8
##  4 Comics              3500          1489                     35.4
##  5 Games               8000          1289.                    36.6
##  6 Music               4000          1020                     49.1
##  7 Film & Video        6598.          746                     60.0
##  8 Art                 3000           422                     45
##  9 Technology         20000           321                     52.6
## 10 Publishing          5000           275                     40.7
## 11 Food               10000           255                     48.9
## 12 Fashion             5939.          243.                     50
## 13 Photography         4000           241.                     47.5
## 14 Crafts              2345            94                      31.2
## 15 Journalism          5000            51                      33.7
```

We can see from these tables that the most popular type of project to create is Film & Video with just over 60 thousand projects on the website. The least popular type of project is Dance, with only just shy of 4000 projects. The average length of a project is 30-35 days across the board. The type of project with the highest average number of backers appears to be Games with 322, while the smallest average number is Crafts with 27 backers. In addition, the type of project that has the highest average rate of success is Dance at 62%, while the least successful is Technology, where only 20% are successful.

We can see from the subcategories that the general category with the least variation is Design, where 3/4 of projects are under the Product Design subcategory. On the other hand, Technology seems to be the most varied, since it's most common subcategory (general technology) only holds 20 percent.

In terms of USD, the loftiest goals seem to be in the Technology category with a median goal of $20,000. The smallest is in Crafts, where the median goal is just over $2000. The median Design project nets more money than any other category at just under $2000, while the median Journalism project receives the least at only $50. Projects in Dance net the most per person, with the median amount being just over $60. In contrast, Crafts projects get a median pledge of only $30.

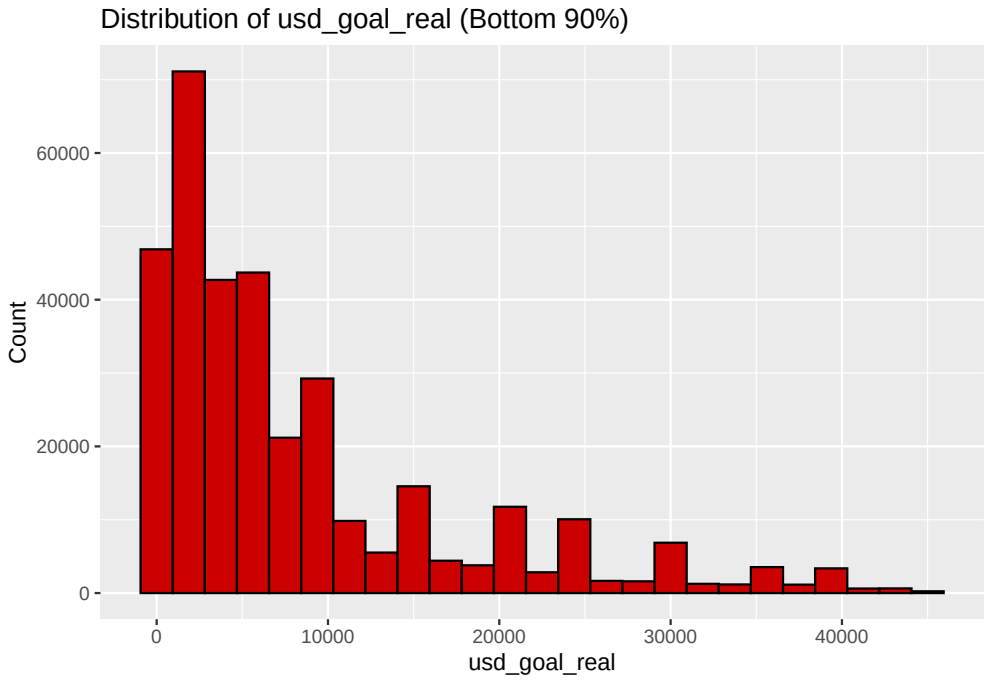
Visual Summaries

For further analysis, we should write a function to return a subset of the dataframe that only contains rows of data below a certain percentile, to filter outliers easier.

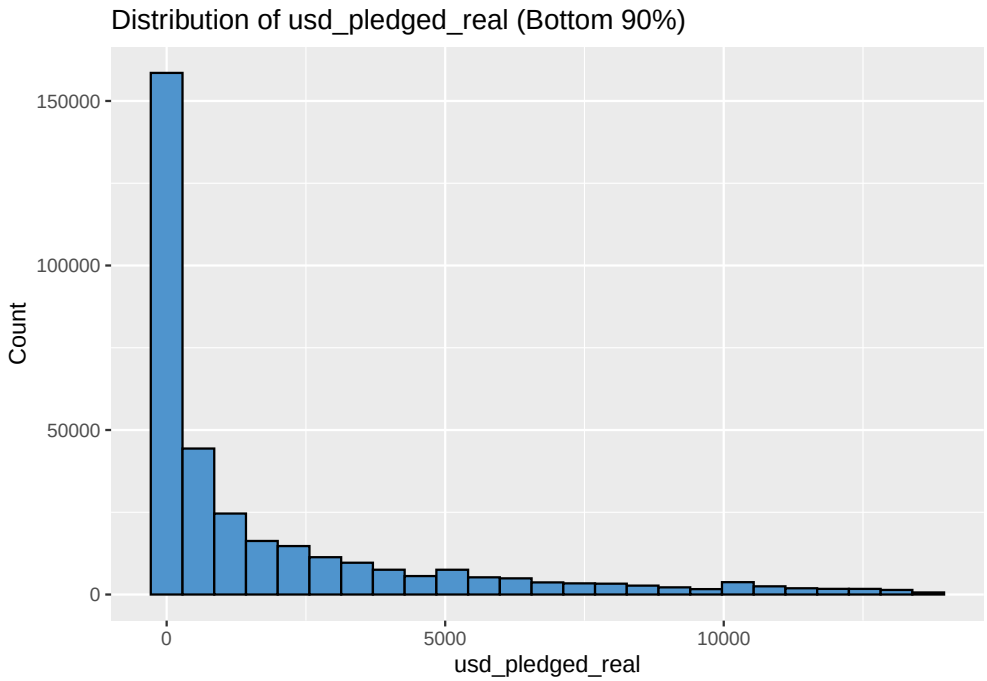
```
# function to get subset of data (bottom X quantile)
filterquantile = function(df, colname, p){
  return(
    df %>% filter(quantile(df[[colname]], p) > df[[colname]])
  )
}
```

We can visualize the numeric distributions with histograms:

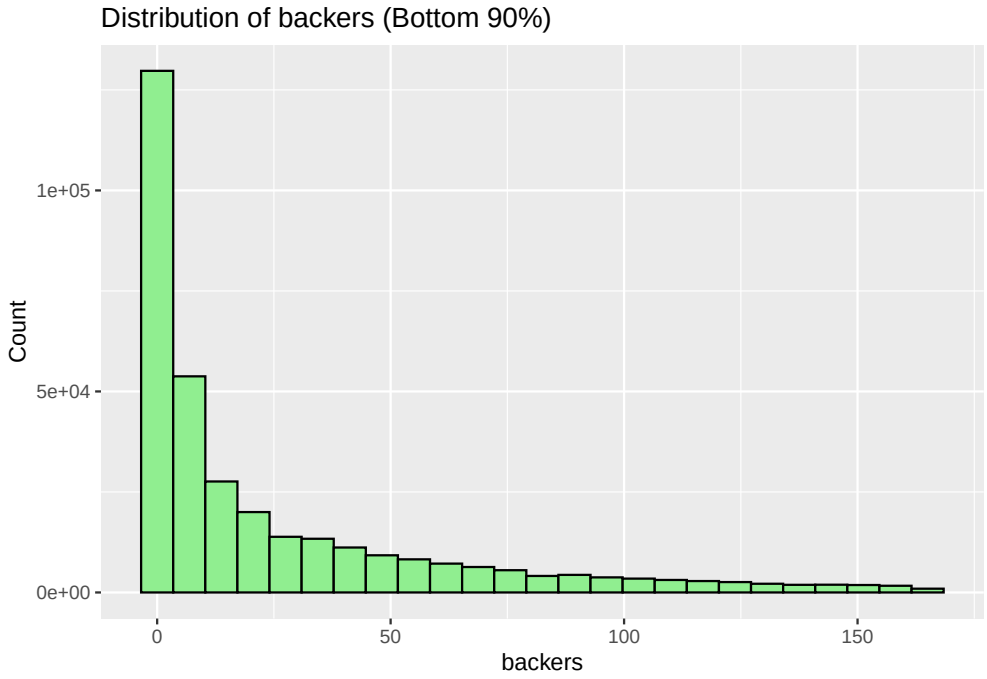
```
# function to plot histogram of column `col`, filtered for the
# bottom `p` quantile, with `bins` number of bins and color `fill`
histfunc = function(col, p, bins, fill){
  # take dataset, filter for p-th quantile
  df18 %>% filterquantile(., col, p) %>%
    ggplot()+
    geom_histogram(aes(x = .data[[col]]),
                   bins = bins, fill = fill,color="black")+
    labs(title = sprintf("Distribution of %s (Bottom %s%%)", col, 100*p),
         x = col,
         y = "Count")
}

# do plots for each numeric column, 90th percentile, 25 bins.
histfunc("usd_goal_real", p = 0.9, bins = 25, fill = "red3")
```




```
histfunc("usd_pledged_real", p = 0.9, bins = 25, fill = "steelblue3")
```



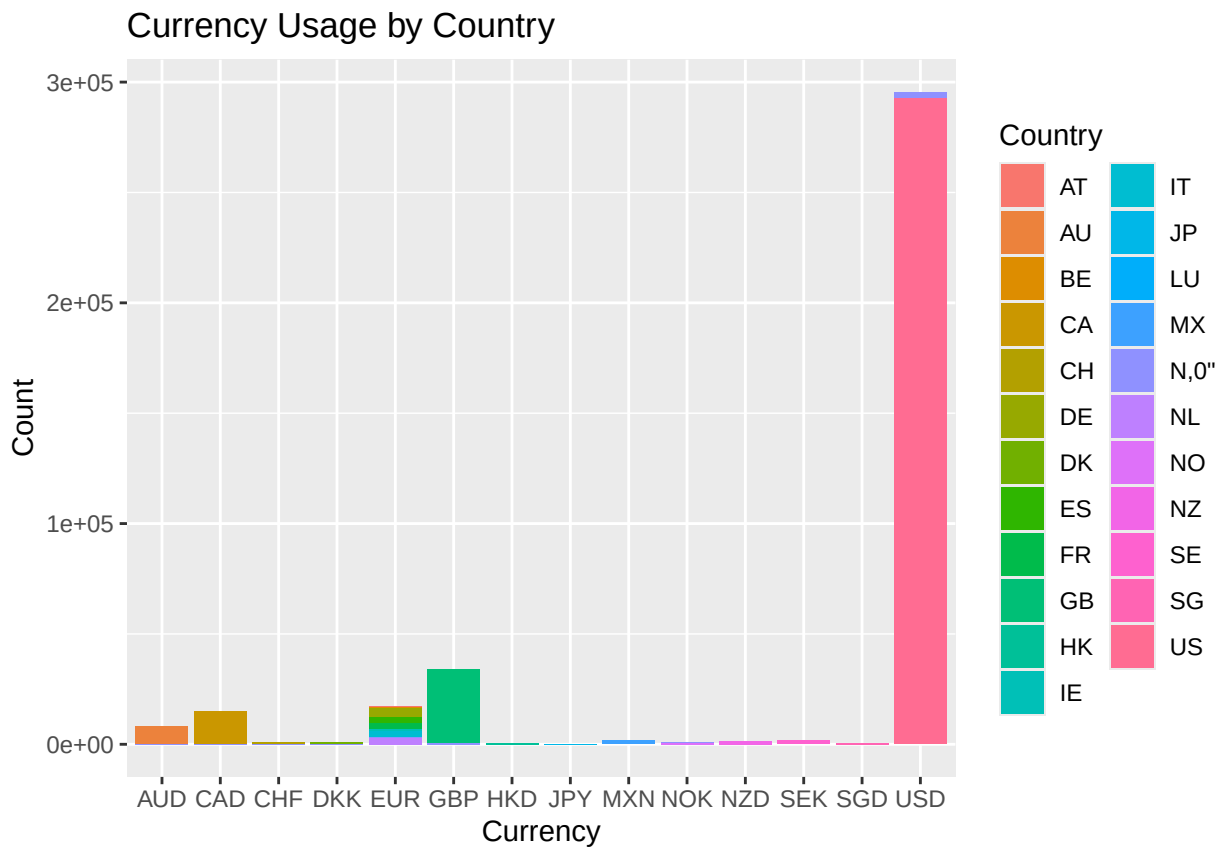
```
histfunc("backers", p = 0.9, bins = 25, fill = "lightgreen")
```



We can see from the histograms, that the majority of projects receive very few pledges and backers. The goals are similarly distributed, with the majority occurring at low values.

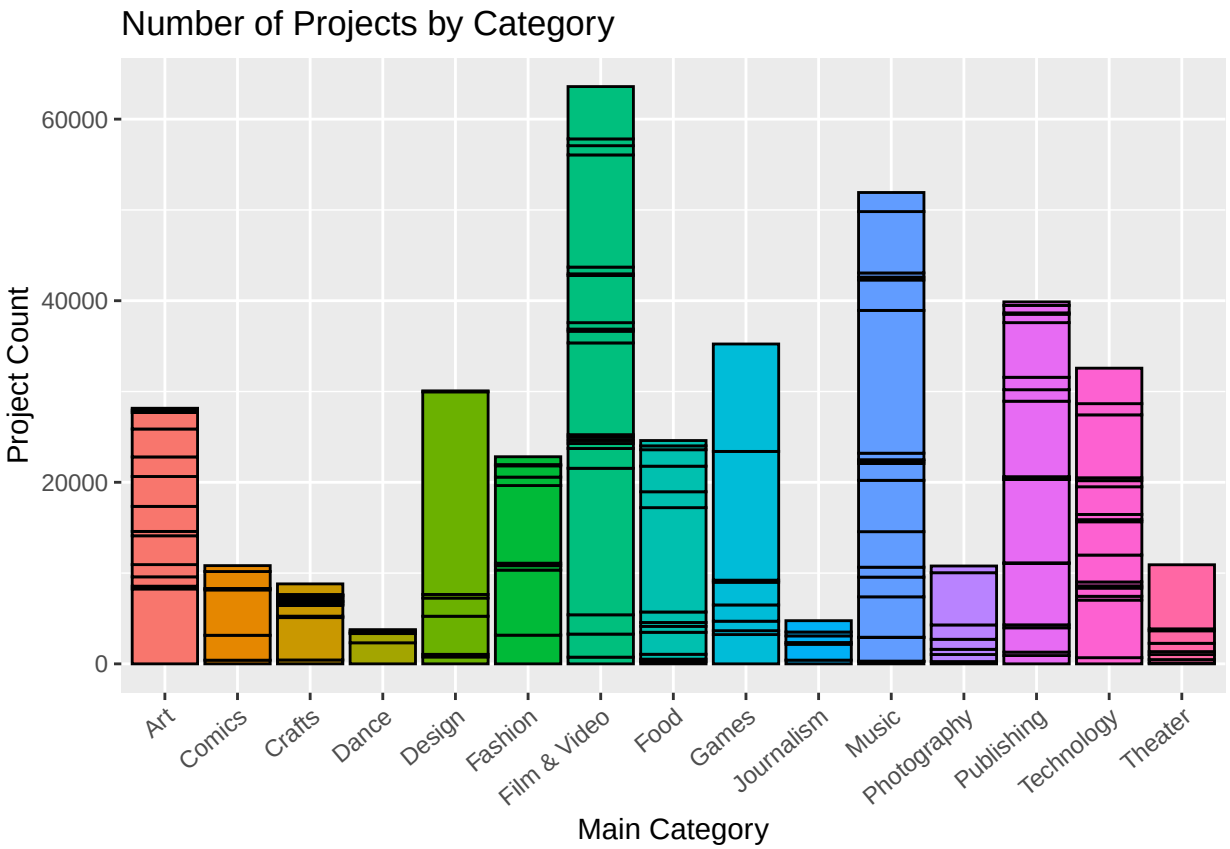
We can visualize some categorical data:

```
## currency and country plot
df18 %>% group_by(currency,country) %>%
  # get count for each group
  summarise(count = n(), .groups = "rowwise") %>%
  ggplot()+
  geom_bar(aes(x = currency, y = count, fill = country), stat = "identity")+
  labs(title = "Currency Usage by Country",
       x = "Currency",
       y = "Count",
       fill = "Country")
```



As we can see, USD is the most popular currency for projects on the website, with the majority originating from the United States, followed by GBP in second. The Euro comes in third, because it is used by a variety of countries.

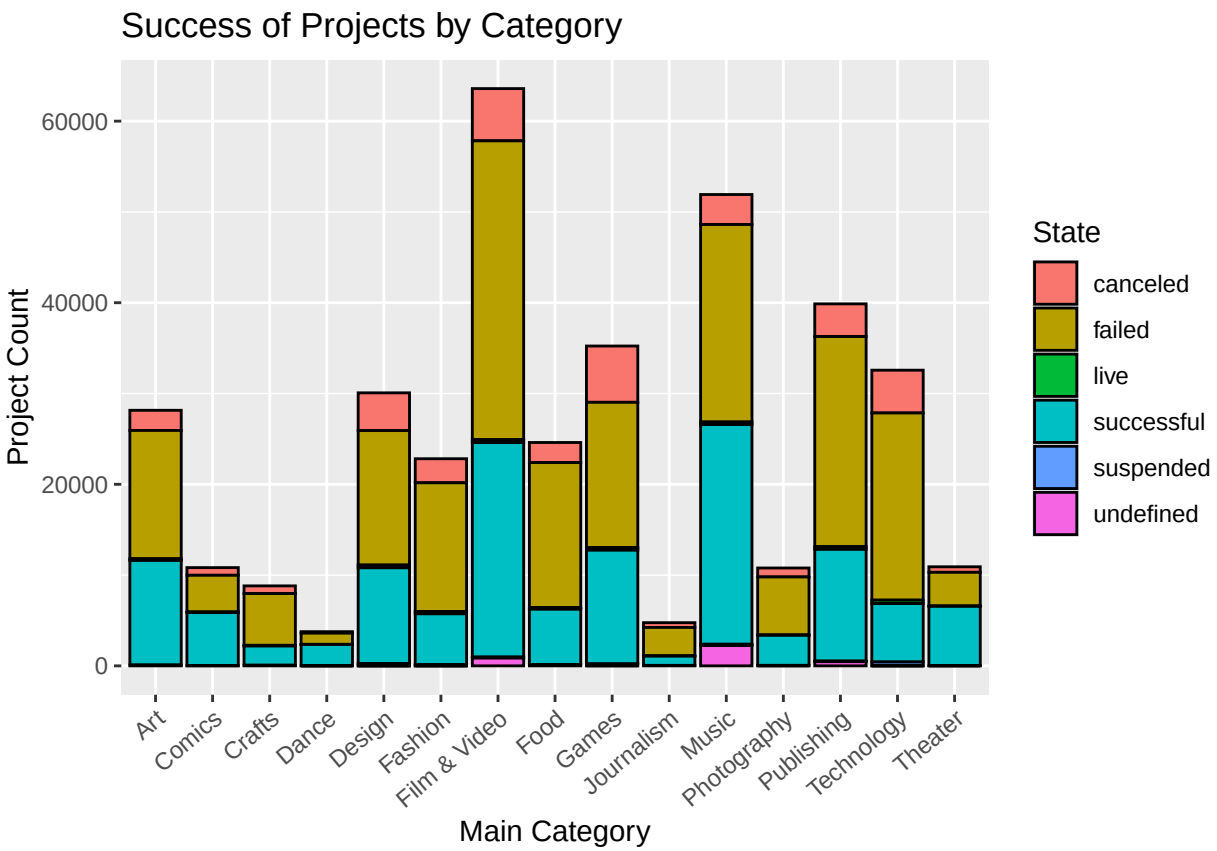
```
## main category and sub category
df18 %>% group_by(main_category, category) %>%
  summarise(count = n(), .groups = "rowwise") %>%
  ggplot()+
  geom_bar(aes(x = main_category, y = count, fill = main_category),
           stat = "identity", color = "black")+
  guides(x = guide_axis(angle = 40),fill = "none")+ # tilt x labels
  labs(title = "Number of Projects by Category",
       x = "Main Category",
       y = "Project Count")
```



Here we can see the distribution of the main categories, divided into sections for the subcategories (unlabeled, because there is 159 of them). Some categories like Art seem to be made up of many equally small subcategories, while some categories like Design are dominated by a single larger subcategory.

We can also similarly plot success or failure by category,

```
## success or failure by category
df18 %>% group_by(main_category, state) %>%
  summarise(count = n(), .groups = "rowwise") %>%
  ggplot()+
  geom_bar(aes(x = main_category, y = count, fill = state),
           stat = "identity", color = "black")+
  guides(x = guide_axis(angle = 40))+
  labs(title = "Success of Projects by Category",
       x = "Main Category",
       y = "Project Count",
       fill = "State")
```



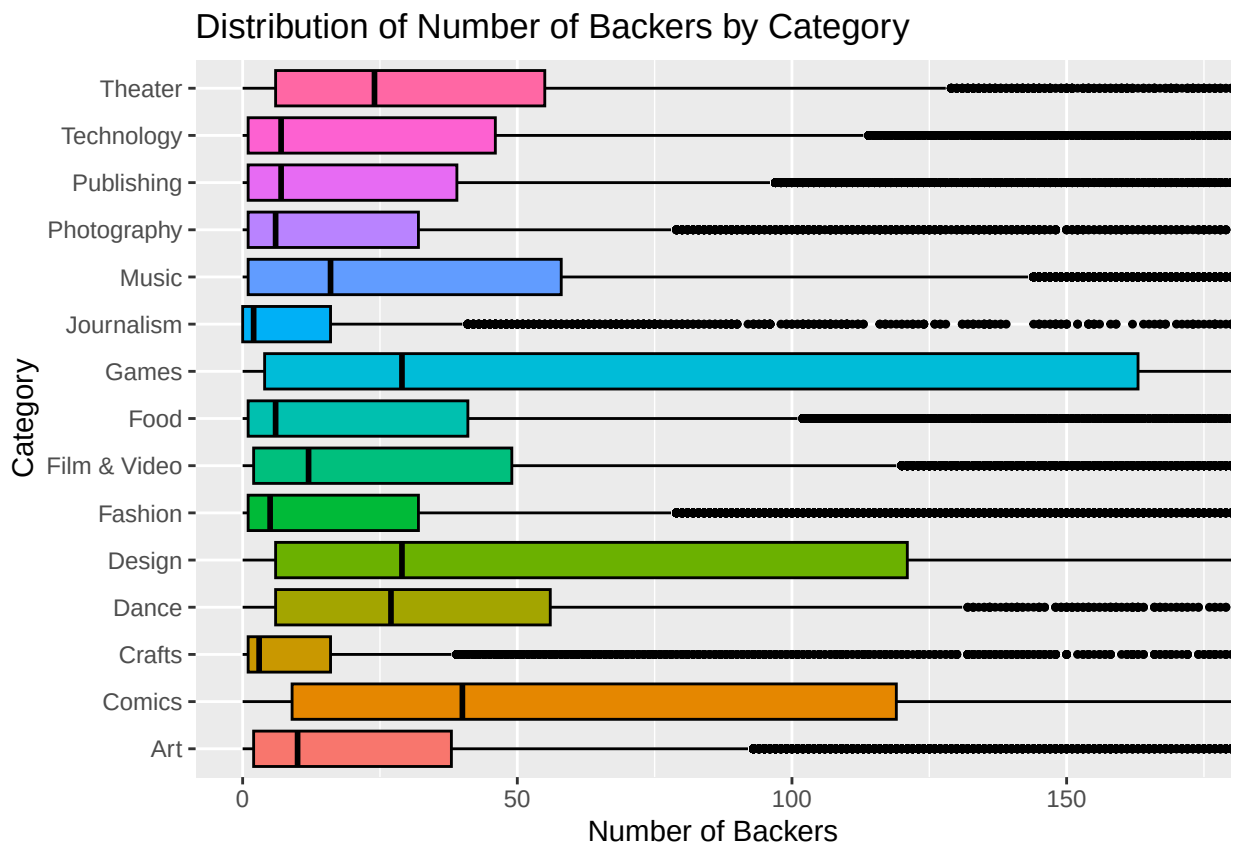
We can see that most of the categories seem to have roughly half failures. Food and Technology seem to have a large proportion of failure than most others.

We can also visualize the numeric data per category with box plots,

```
## function to create boxplot by category for a given column
boxbycat = function(columnname, titlestring){
  xlim_box = boxplot.stats(df18[,columnname])$stats[c(1,5)] # get x limits

  ggplot(df18, aes(x = df18[,columnname]))+
    # do a box plot
    geom_boxplot(aes(y = main_category, fill = factor(main_category)),
                 color = "black",
                 outlier.shape = 20)+

    # set the x limit so as to ignore outliers
    coord_cartesian(xlim = xlim_box*1.25)+

    # remove the legend
    theme(legend.position = "none")+

    # set the labels
    labs(title = sprintf("Distribution of %s by Category", titlestring),
         x = titlestring,
```

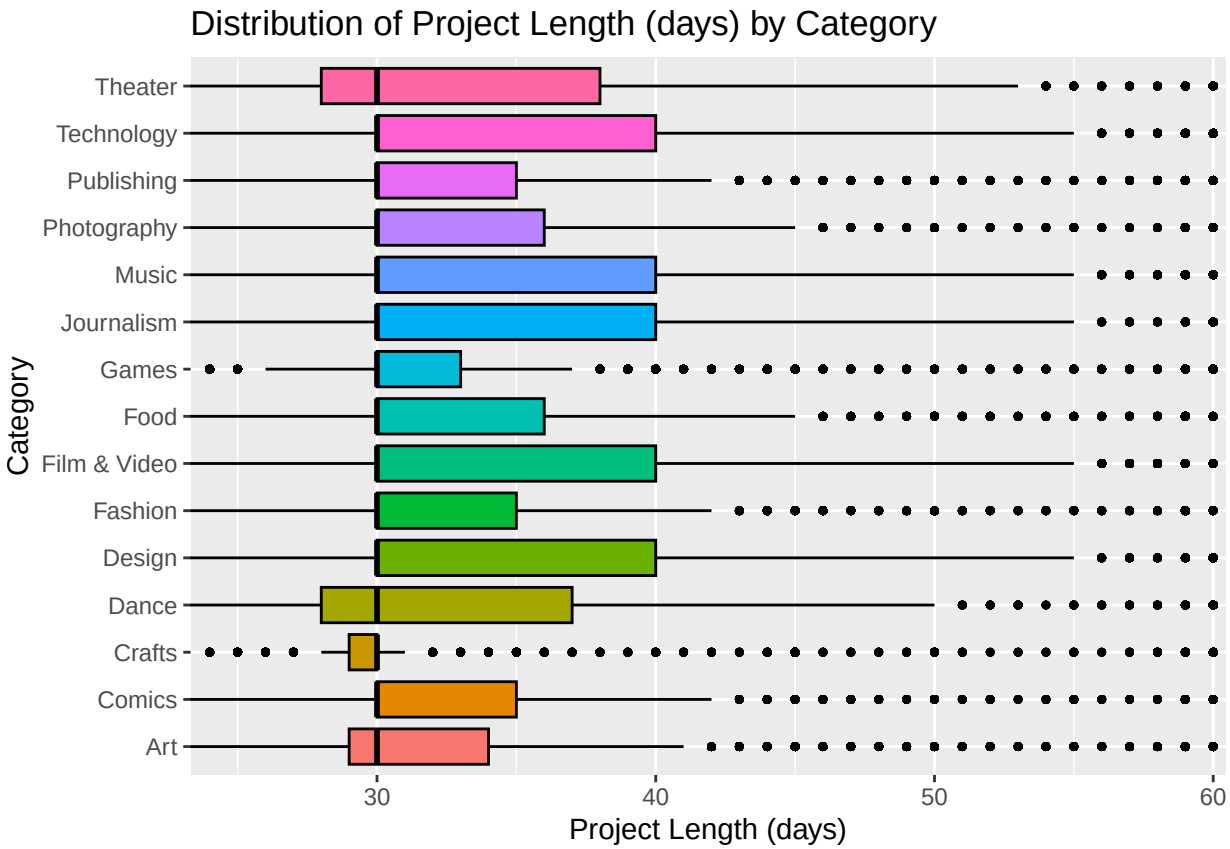
```
          y = "Category")
}
```

```
boxbycat(columnname = "backers", titlestring = "Number of Backers")
```



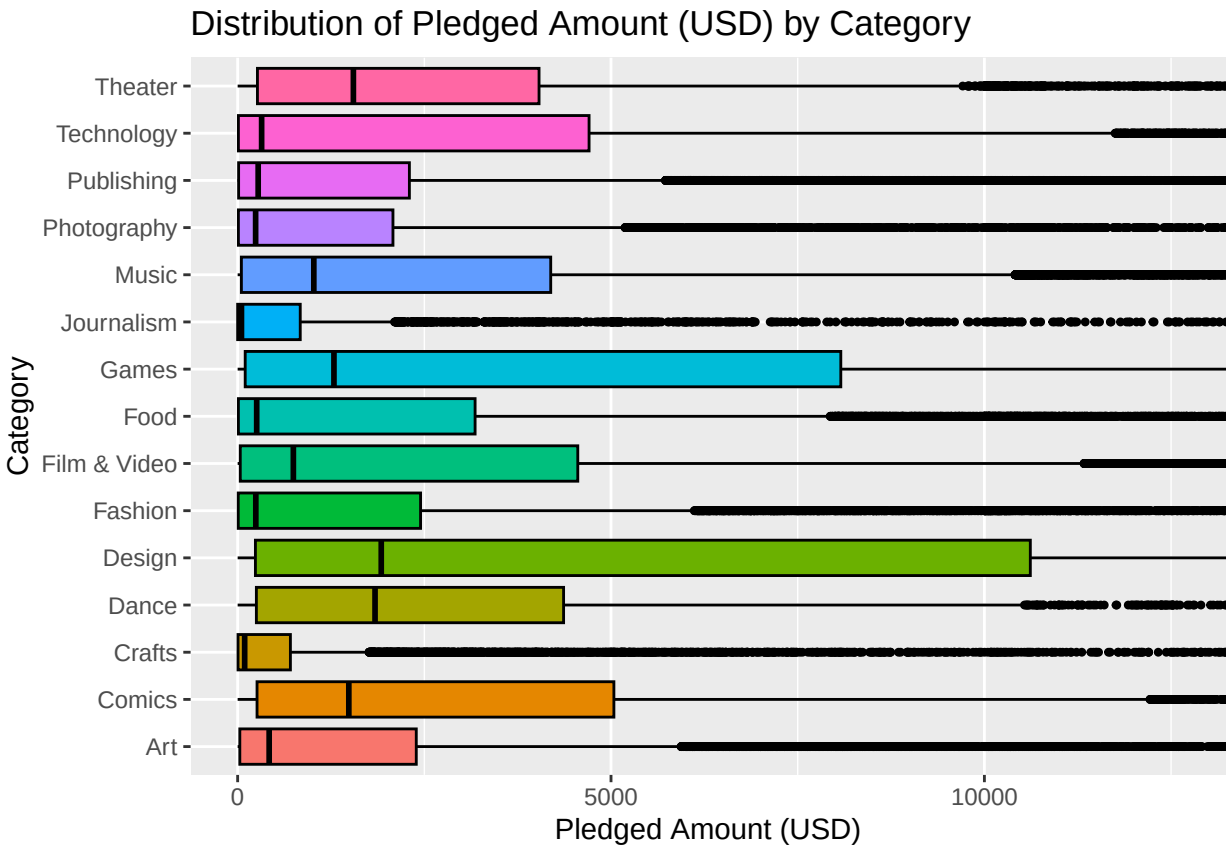
The box plot gives us a sense of the distribution of the number of backers by each category. It appears that most of the categories have right-skewed distributions (likely, a few large outliers in those categories skews the mean above the median). We can also see that Games has the widest spread, and that the median number of backers in almost all categories (except Games, Design, and Dance) is below 25.

```
boxbycat("project_days", "Project Length (days)")
```



We can see from the project length box plot that all the categories have a median of about 30 days. It appears that, with the exception of some outliers, the majority of projects don't last any longer than 6 weeks. Most of the data skew right, except for Crafts which is skewed left.

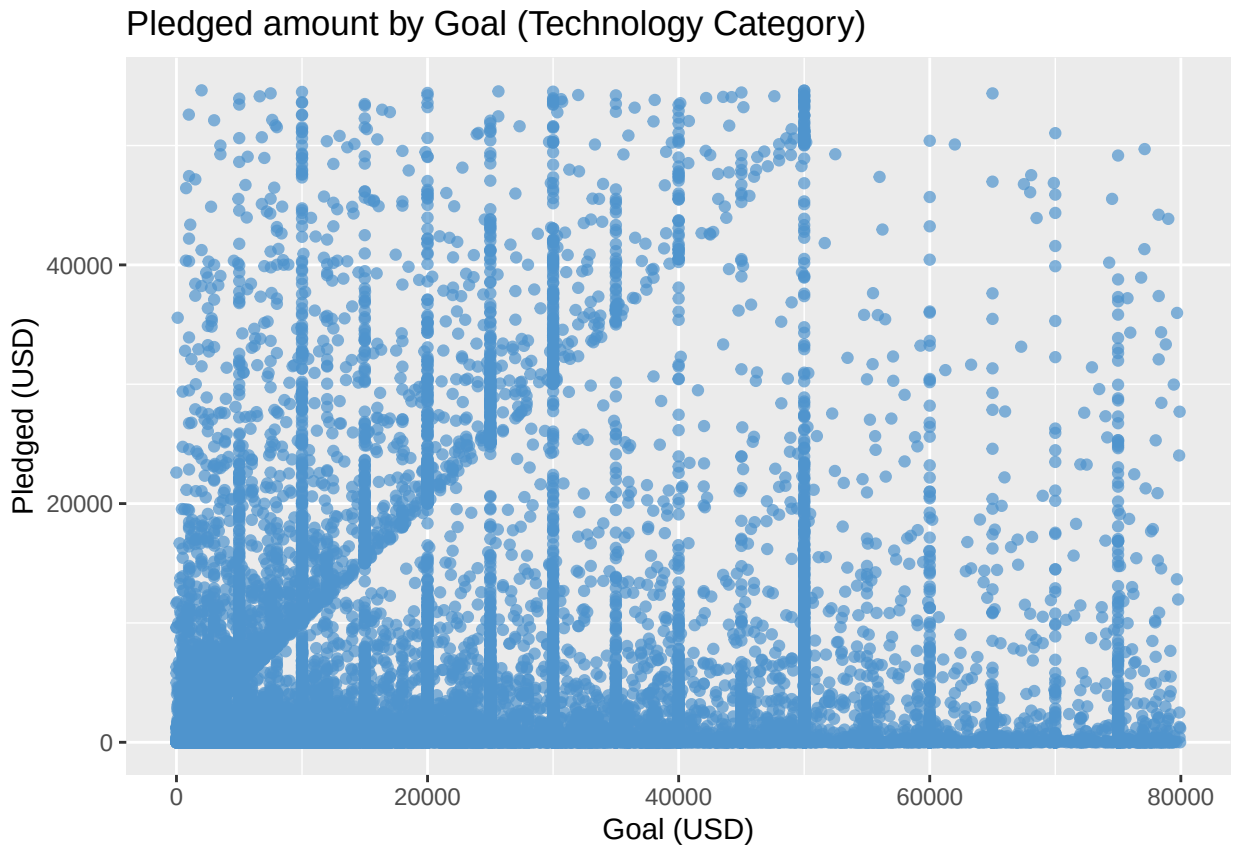
```
boxbycat("usd_pledged_real", "Pledged Amount (USD)")
```



From the boxplot for amount pledged, we see again that most of the data are skewed right, signifying there are a few large outliers dragging the mean higher than the median. We also see that when it comes to money received, Design projects have both the highest median and the largest spread.

We can also plot the pledged amount against the goal amount

```
# filter data for largest outliers in goal and pledged data,
# then filter for tech category (smaller subset is easier to see), and plot
df18 %>%
  filterquantile(., "usd_goal_real", 0.95) %>%
  filterquantile(., "usd_pledged_real", 0.98) %>%
  filter(., .$main_category == "Technology") %>%

  ggplot()+
  geom_point(aes(x = usd_goal_real,y = usd_pledged_real),
             color = "steelblue3", alpha = 0.7)+
  labs(title = "Pledged amount by Goal (Technology Category)",
       x = "Goal (USD)",
       y = "Pledged (USD)")
```



We can see that people tend to set goals at "round" amounts, by the spikes at 5k, 10k and 15k, etc. There also appears to be a straight line on the left which appears to be the line y = x, which may be situations where a project stops when it has reached its goal, and does not continue to accept funds. Points below and to the right of this y = x line are failed projects, and those above and to the left of it are successes; there appear to be many more failures than successes. There also appears to be a gap just below this y = x line, seemingly not many projects get close to their goal but then fail meet it at the end.

Regression Analysis

We are analyzing the correlation between the number of projects launched in a month and their success rates. To start we are going create the variables needed as they are not a part of the data set yet and then group by the year and month so that we will have the number of projects launched per month as well as the success rate for projects in that month. We will then graph the scatterplot between these two variables to confirm or deny the existence of a linear relationship.

But, due to problems encountered as this process went on we are going to filter out any live projects because of data collection problems during the last two months of data that we have

When looking into December of 2017, we see that the problem with this time frame is how the state of the projects is listed. During this month we see that there are many projects deemed as "live", due to the fact that these are not successful yet we have marked them as a zero, thus counting as unsuccessful. However, the interesting thing is that this month contains 76.4% of all the 'live' projects in the entire dataset. This is of course highly unusual and is most likely due to the time that they collected data. Possibly collecting it around the 2018 new year and thus capturing many projects that were still live and failed to get updated because our data ends in the first month of 2018.

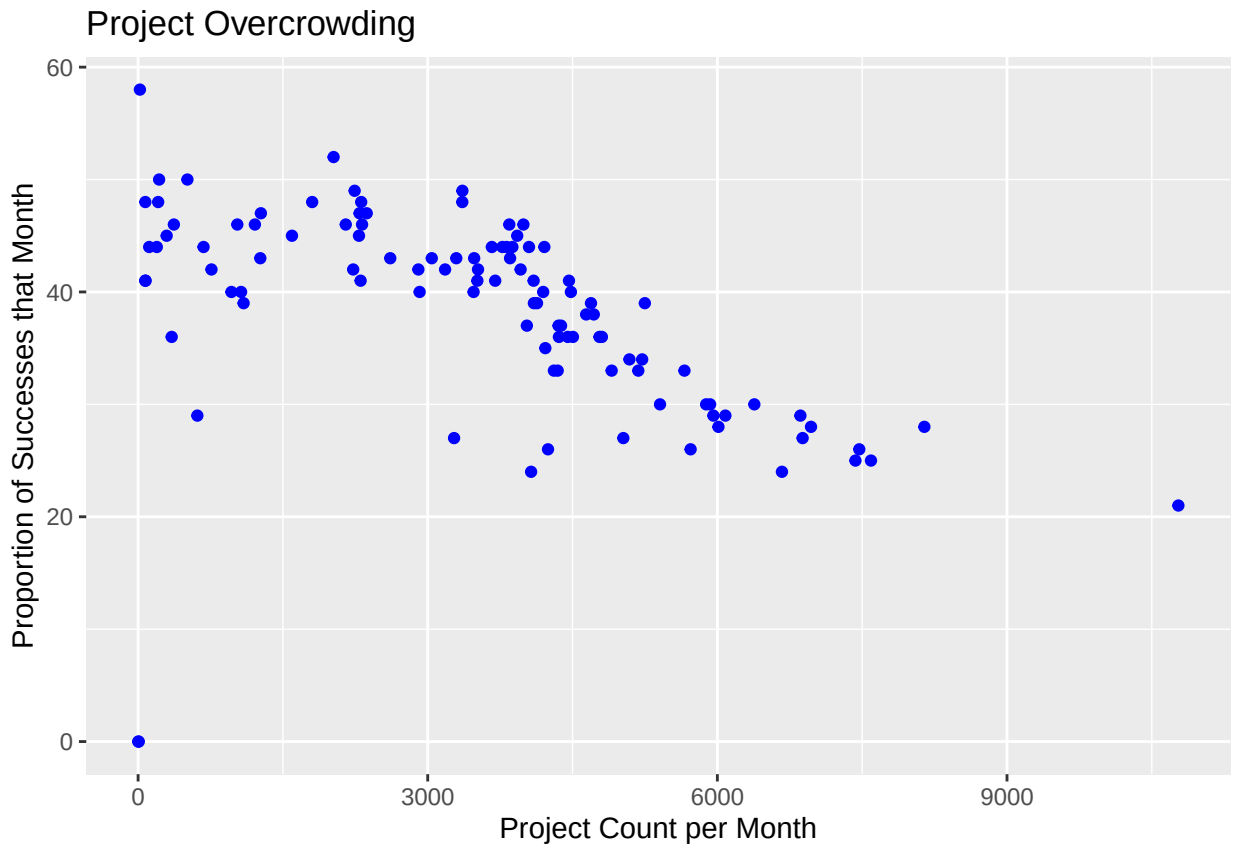
And as suspected we have the same problem with January of 2018, where 98.4% of the entries for that month are still live, the other percentage are projects that were canceled. Due to this we are going to remove all of these entries as we have found data collection problems with them.

Because of this we shall actually filter out all of the live projects.

```
# new column monthstart, with YYYY-MM launched (for grouping)
df18 = df18 %>% mutate(monthstart = (substring(launched,1,7)))

monthgroup = filter(df18, state != "live")

# group data by month, aggregate the total project that month and % success
monthgroup = monthgroup %>%
  group_by(monthstart) %>%
  summarise("Number_Projects" = n(),
            "Success_Rate" = (round((sum(state == "successful")/n())*100)))
```

Now that we have created the needed columns as well as created a new dataframe that is grouped appropriately we can continue on to graphing.

```
# do plot
ggplot(monthgroup, aes(x = Number_Projects, y = Success_Rate))+
  geom_point(col = "blue")+ # scatter
  labs(title = "Project Overcrowding",
       x = "Project Count per Month",
       y = "Proportion of Successes that Month")
```



We can see with this graph that there looks to be a negative correlation between the number of projects per month and the success rate of those projects. We also see that there is possibly a couple outliers in the bottom left corner of the graph. We will investigate these and remove them if we deem them to be outliers.

Before we do so, we will also check the correlation coefficient between the two variables.

```
cor(monthgroup$Success_Rate, monthgroup$Number_Projects)
```

```
## [1] -0.4641619
```

We see that we currently have a correlation coefficient of -0.464 which is a weak negative relationship. However, we will revisit this number after removing any points that we deem to be outliers ("Unname Function - Rdocumentation" n.d.).

```
Q1 = unname(quantile(monthgroup$Success_Rate, 0.25))
Q3 = unname(quantile(monthgroup$Success_Rate, 0.75))
IQR = Q3 - Q1

uf = Q3 + (IQR*1.5)
lf = Q1 - (IQR*1.5)

outliers = filter(monthgroup, monthgroup$Success_Rate < lf)
outliers
```

```
## # A tibble: 2 x 3
##   monthstart Number_Projects Success_Rate
##   <chr>                <int>        <dbl>
## 1 1970-01                  7            0
## 2 2018-01                  2            0
```

We see that there are two points that are considered outliers. Looking further into these specific points we see that there is one with the date of 1970-01, this is actually due to the fact that there are 7 entries in our data that do not have a launched date.

Because we removed the live entries, the only two states of projects that are left in January of 2018 are cancelled projects and that is why we see this small amount of projects. We shall remove this outlier as it doesn't accurately represent where the success rate will end up at.

Because of this we shall actually filter out all of the live projects.

```
monthgroup = filter(monthgroup, between(monthgroup$Success_Rate, lf, uf))
```

```
cor(monthgroup$Success_Rate, monthgroup$Number_Projects)
```

```
## [1] -0.7457337
```

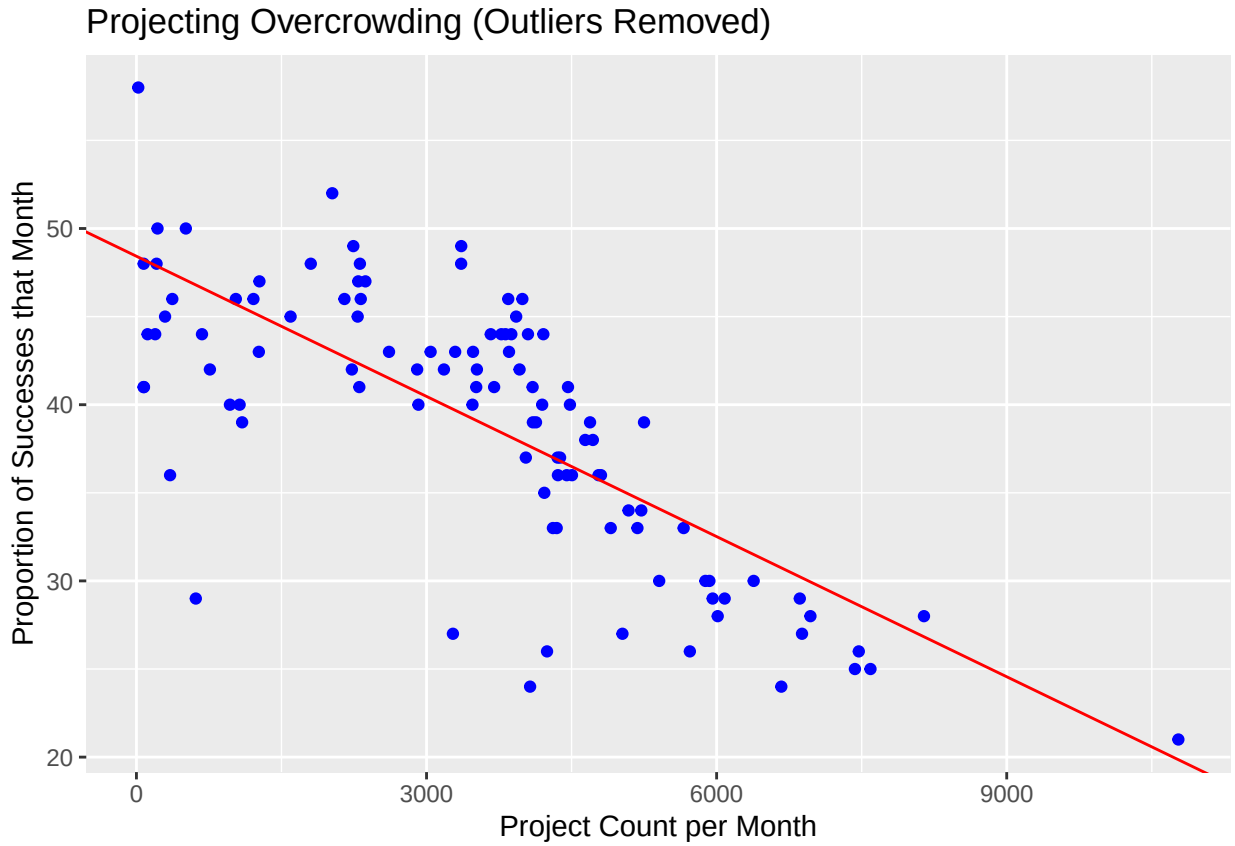
After removing the outliers we see that we now have a correlation coefficient of -0.7457 which is a much stronger relationship and better represents the correlation between the two variables.

We are now going to fit the model in order to find our linear regression line.

```
y = favstats(monthgroup$Success_Rate)
x = favstats(monthgroup$Number_Projects)

beta1 = sum((monthgroup$Number_Projects - x$mean)*
              (monthgroup$Success_Rate - y$mean))/sum((
                monthgroup$Number_Projects-x$mean)^2)

beta0 = y$mean - beta1*x$mean

cat("Our beta_0 (intercept) value is ", beta0,
    ",\n and our beta_1 (slope) value is ", beta1)
```

```
## Our beta_0 (intercept) value is  48.42493 ,
##  and our beta_1 (slope) value is  -0.002651616
```

```
ggplot(monthgroup, aes(x = (Number_Projects), y = (Success_Rate)))+
  geom_point(col = "blue")+
  geom_abline(intercept = beta0, slope = beta1, col = "red")+
  labs(title = "Projecting Overcrowding (Outliers Removed)",
       x = "Project Count per Month",
       y = "Proportion of Successes that Month")
```



Our beta0 value is the y intercept number, and this represents that when the number of projects during a month is zero the success rate will be equal to 48.42%. This is of course a weird interpretation for our model as if there are zero projects there wont be a success rate, but say there was only one project in a month for example, our model predicts the success rate to be very close to 48.42%. Our beta1 value is -0.0026, which states that for every additional project that is launched in a month, the success rate of that month will go down by 0.0026%. Alternatively, for every 1000 projects launched the likelihood of success goes down by 2.6%. Although this may seem insignificant, when there are up to 12,000 projects in a single month, this will decrease the overall success rate by about 31%!

Hypothesis Testing

We need to check if these values are significant and we can do this by hypothesis testing and checking the associated p values.

Our first hypothesis are going to check that β_1 and β_0 are both not equal to zero. This would mean that they are significant and the linear relationship exists, if the slope is not significant then we would not be able to say there is a linear relationship.

$$H_0 : \beta_1 = 0 \quad \text{vs} \quad H_a : \beta_1 \neq 0$$

$$H_0 : \beta_0 = 0 \quad \text{vs} \quad H_a : \beta_0 \neq 0$$

```
stats = summary(lm(formula = Success_Rate ~ Number_Projects, data = monthgroup))
stats
```

```
##
## Call:
## lm(formula = Success_Rate ~ Number_Projects, data = monthgroup)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.7968  -2.7141   0.2471   3.3148   9.6254
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     48.4249332  0.9711780   49.86   <2e-16 ***
## Number_Projects -0.0026516  0.0002334  -11.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.072 on 103 degrees of freedom
## Multiple R-squared:  0.5561, Adjusted R-squared:  0.5518
## F-statistic:   129 on 1 and 103 DF,  p-value: < 2.2e-16
```

We see that both of the p values are less than 0.05 and we can conclude, based on the above tests, that both of the model coefficients are significant. Furthermore, we can also check to confirm that the slope is less than zero by using a one tailed test, instead of a two tailed test as the above uses.

$$H_0 : \beta_1 = 0 \quad \text{vs} \quad H_a : \beta_1 < 0$$

```
pt(-11.36, 103, lower.tail = TRUE)
```

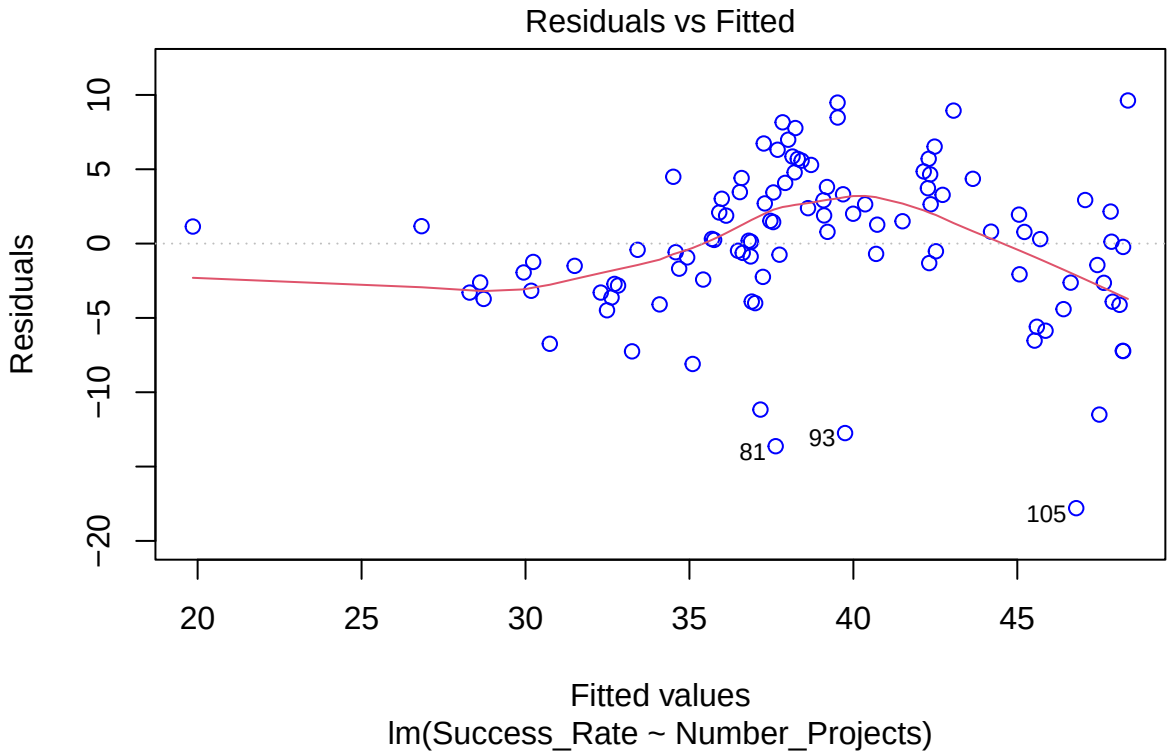
```
## [1] 3.55887e-20
```

We get a p value far less than 0.05, and can further confirm that our slope is less than 0.

Now we can check and confirm if all the assumptions that we have made are correct. The assumptions that we have made, and that we have to check are that the variances equal and independent, as well that they are normally distributed.

The equality and independence of variances can be confirmed by creating a graph of the residuals versus the predicted y values. What we want to see is a straight line that runs through a group of scattered points. This would confirm the assumption that the errors are independent ("Plot.lm: Plot Diagnostics for an Lm Object" n.d.).

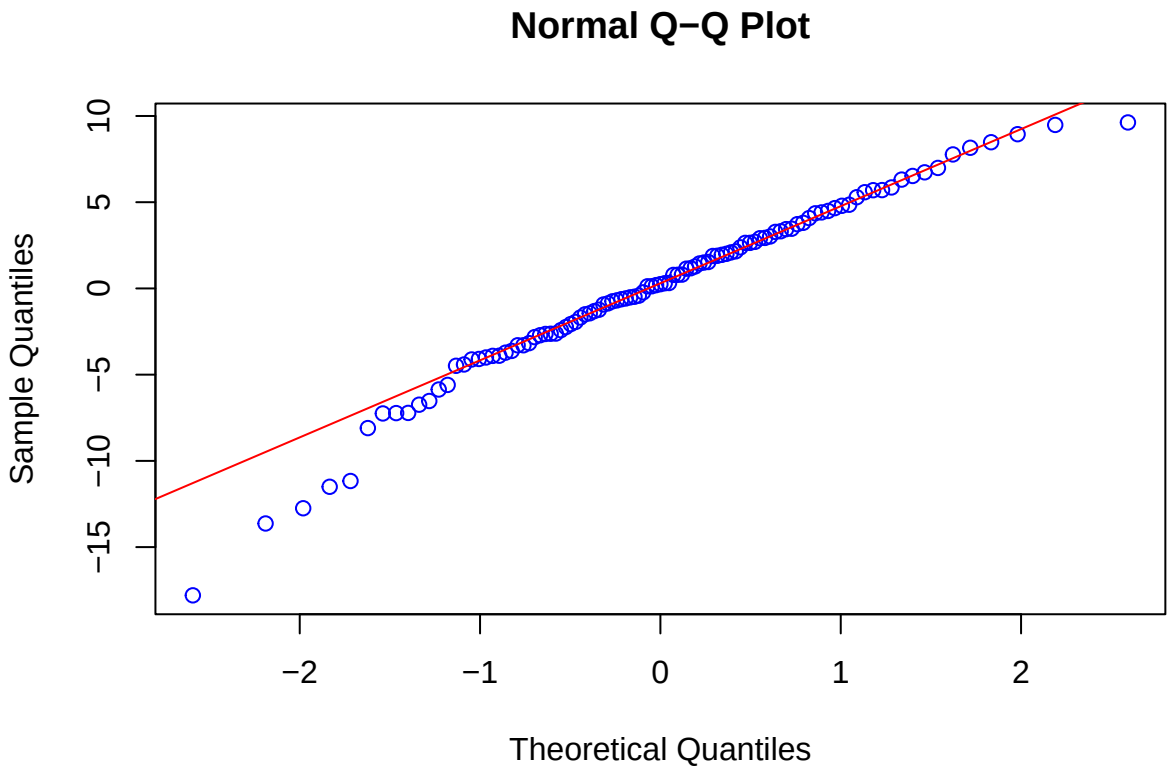
```
plot(lm(Success_Rate ~ Number_Projects, data = monthgroup),
     which = 1, col="blue")
```



From this graph we can see that the line that runs through all the points is roughly a straight line, with a bit of variation. We cannot say conclusively that the errors are independent from this.

Now to test that the residuals are normally distributed we can look at the Q-Q plot, what we want to see is a relatively straight line that does not deviate from the marked line down the diagonal ("Qqnorm Function - Rdocumentation" n.d.).

```
qqnorm(resid(lm(Success_Rate ~ Number_Projects,
                data = monthgroup)), col = "blue")
qqline(resid(lm(Success_Rate ~ Number_Projects,
                data = monthgroup)), col = "red")
```



From this we see that our model follows the straight line for the most part, although it tails off on the left. Although it is not perfect it would seem that the errors are roughly normally distributed, with a few outliers.

Conclusion and Future Steps

We found the statistically significant negative correlation between the number of projects launched in a month and their success rate to be -0.7457 with outliers removed. As the number of projects launched increases, the chance of success will decrease, for every one extra project launched in the month, the success rate will drop by approximately 0.0026%. The number seems small, but there will be a noticeable decrease when there is a large volume of projects launched in a month. Kickstarters may want to consider better timing for the launch of the project to improve the chance of success.

Through regression analysis and hypothesis testing, we have found both the intercept and slope of our model are statistically significant. It is a good fit for us to understand the general negative relationship between the number of projects launched in a given month and their success rates, but the residual vs. fitted graph showed us errors are not perfectly independent. The current model has shown us useful insights, but it is not enough to capture the relationship between number of project launches and success rates fully.

For our next steps to improve the model, we think the model will have more accurate predictions by including more predictor variables such as project category, goal of funding, and duration of the project. In this approach, the model will give us a more comprehensive understanding of what factors influence success beyond only the number of projects launched in a month. We could also dig into the outliers and trends, identifying the cause of the outliers will improve the predictive power of our model, and trends over time can provide us with better predictions of future outcomes based on the historical data.

References

"Kickstarter Projects." 2017. https://www.kaggle.com/datasets/kemical/kickstarter-projects.

"Plot.lm: Plot Diagnostics for an Lm Object." n.d. *R Documentation*. Accessed October 8, 2024. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/plot.lm.

"Qqnorm Function - Rdocumentation." n.d. Accessed October 8, 2024. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/qqnorm.

"Unname Function - Rdocumentation." n.d. Accessed October 8, 2024. https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/unname.