Exploratory Data Analysis

K Iwasaki

September 15, 2017

Contents

Overview

Data exploration serves some important objectives. For this exploration, we focus on understanding the data-set in order to decide

1. What algorithm we select based the dataset
2. What variables we use for algorithm training and prediction
3. What transformation are required for each variables based on the choice of algorithm

Note that since we have spend significant amount of time for data cleaning and feature creation in the preprocessing phase, this data exploration doesn't cover these items.

Set-up

Before diving into detailed analysis, it is good to start with a high level picture. In this section, we look at summary statistics to see distribution of each variable, check a variable category (such as category/continuous) for variables, and validate NA values in each column.

A few things to notice at this point:

- Most of the columns are categorical variables except money_raised, year founded, latitude, longitude and etc.
- Industry labels are a dummary variable in which 1 is assinged for a company if it belongs a industry and 0 otherwise.
- There are NA values in some columns. We have actually noticed this in the data preprocessing stage.
- Target variable for the algorithm is CompanyName because what we want to predict(recommend) is which startup matches user's interests. This is a multiclass classification problem.

```r
drops = c("title", "link", "excerpt", "Company", "money_raised", "linkedin_link",
          "Company_at_Linkedin", "Specialties", "Website", "Location",
          "zip_code","State",
          "Description", "Also.viewed", "Industry")
data = data[, !(names(data) %in% drops) ]

# nrow(data)
# ncol(data)

summary(data[, 1:13 ])
```

```
##      published_at  funding_round money_raised_float
##  1/14/2016 :  3            : 12   Min.   : 10.00
##  7/29/2015 :  3   Series B: 48   1st Qu.: 15.00
##  11/15/2016:  2   series C:  1   Median : 25.00
##  11/3/2015 :  2   Series C:152   Mean   : 41.17
##  12/11/2013:  2   Series D: 14   3rd Qu.: 45.00
##  2/11/2008 :  2   Series E:  5   Max.   :793.50
##  (Other)   :218
##                                              CompanyName     CompanySize
##  2U                                                :  1   51-200   :101
##  3D Robotics                                       :  1   201-500  : 64
##  aCommerce - Ecommerce Solutions for Southeast Asia:  1   Nov-50   : 27
##  Affle                                             :  1   1001-5000: 21
##  App Annie                                         :  1   501-1000 : 14
##  Appear Here                                       :  1   10,001+  :  2
##  (Other)                                           :226   (Other)  :  3
##     Founded                 City    address_check          Country
##  Min.   :1939   San Francisco:49   False: 59     United States :178
##  1st Qu.:2007                :46   True :173     United Kingdom: 17
##  Median :2010   New York     :27                 Germany       :  8
##  Mean   :2009   Mountain View:11                 Canada        :  4
##  3rd Qu.:2012   San Mateo    : 8                 India         :  4
##  Max.   :2017   Boston       : 6                 Singapore     :  4
##                 (Other)      :85                 (Other)       : 17
##     latitude       longitude
##  Min.   :25.78   Min.   :-122.67
##  1st Qu.:37.44   1st Qu.:-122.39
##  Median :37.78   Median :-121.95
##  Mean   :38.47   Mean   :-103.87
##  3rd Qu.:40.74   3rd Qu.: -77.28
##  Max.   :47.62   Max.   : -71.04
##  NA's   :46      NA's   :46
##                          Industry_consolidated
##  Internet                           :68
##  Computer Software                  :48
##  Information Technology and Services:25
```

```
##  Financial Services                   :15
##  Consumers Goods & Services           :14
##  Infrastructure                       :10
##  (Other)                              :52
##  spc_Logistics.and.Supply.Chain
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1034
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

```r
# check columns 1:13. Columns 13: have same format.
str(data[, 1:13 ])
```

```
## 'data.frame':    232 obs. of  13 variables:
##  $ published_at                  : Factor w/ 209 levels "1/11/2010","1/14/2016",..: 189 176 160 158
##  $ funding_round                 : Factor w/ 6 levels "","Series B",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ money_raised_float            : num  45 39 40 48 90 20.2 29 32 36 20 ...
##  $ CompanyName                   : Factor w/ 232 levels "2U","3D Robotics",..: 25 29 185 126 39 127
##  $ CompanySize                   : Factor w/ 8 levels "10-Jan","10,001+",..: 4 7 7 7 4 4 7 7 7 7 ...
##  $ Founded                       : num  2013 2013 2015 2011 2013 ...
##  $ City                          : Factor w/ 68 levels "","Arlington",..: 1 1 36 1 55 1 52 36 52 23
##  $ address_check                 : Factor w/ 2 levels "False","True": 1 1 2 1 2 1 2 2 2 2 ...
##  $ Country                       : Factor w/ 23 levels "Belgium","Brazil",..: 8 22 23 22 23 7 23 23
##  $ latitude                      : num  NA NA 40.7 NA 37.4 ...
##  $ longitude                     : num  NA NA -74 NA -122 ...
##  $ Industry_consolidated         : Factor w/ 16 levels "Computer & Network Security & Hardware",..:
##  $ spc_Logistics.and.Supply.Chain: int  0 0 0 1 0 0 0 0 0 0 ...
```

```r
# show columns with na
na = lapply(data, function(x) sum(ifelse(is.na(x) | x == "" | x == "not found", TRUE, FALSE)))
na[na > 0]
```

```
## $funding_round
## [1] 12
##
## $City
## [1] 46
##
## $latitude
## [1] 46
##
## $longitude
## [1] 46
```

Univariate Analysis

Investigate distribution of key variables that we are interested in using for recommendation engines. If a variable is extremely skewed, we might need to consider transformation. In this analysis, we focus on 1) year founded, 2) funding round, 3) money raised, 4) company size, 4) country and 5) headquarter location.

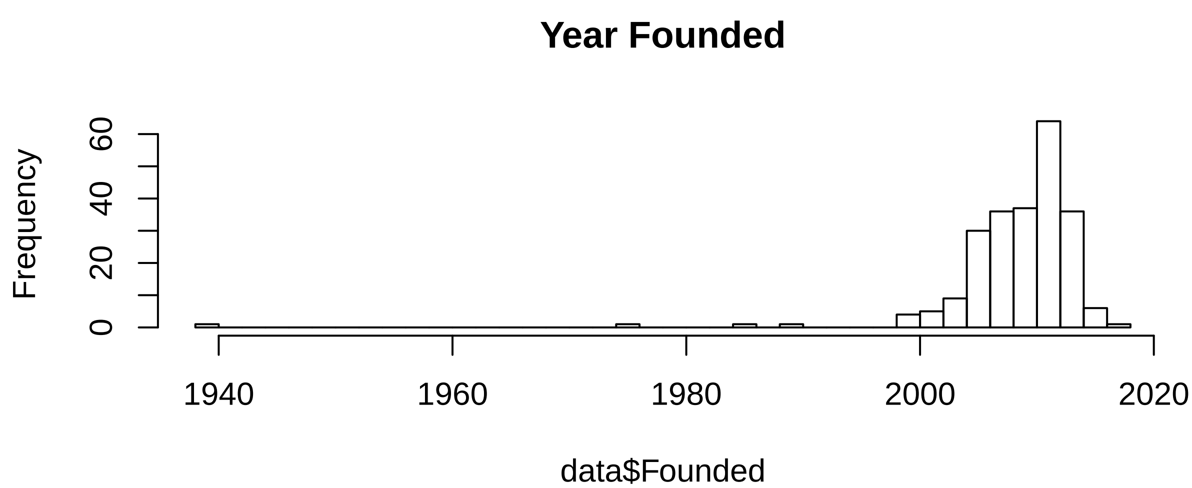
- Year Founded

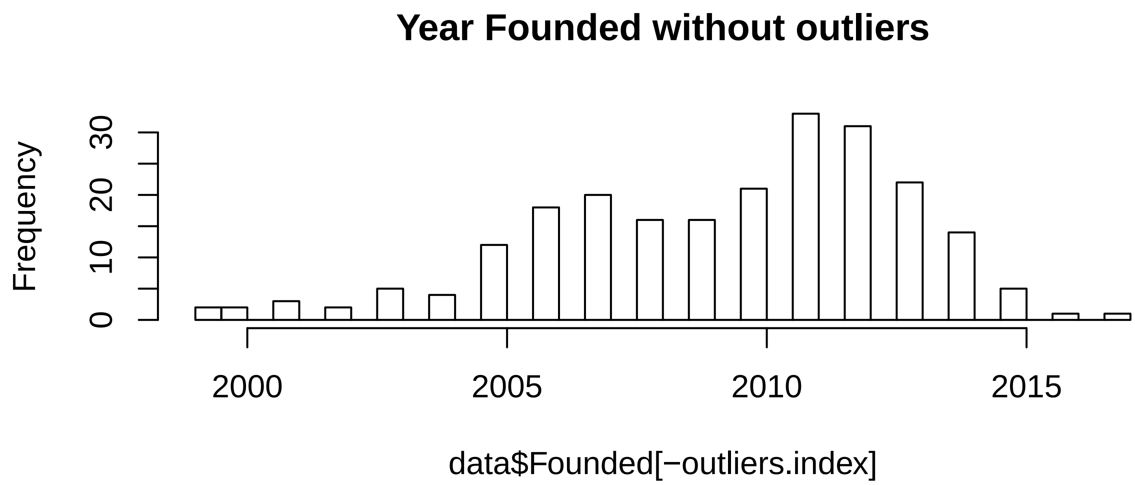
Unexpectedly, there are some companies founded before 1990. Given this recommendation engine focuses on "startups", we migth need to exlude the outliers who founded before 1990. without the outliers, the distribution is close to normal distribution.

```r
hist(data$Founded, main = "Year Founded", breaks = 50)

# check who are the outliers
outliers = data[data$Founded <= 1990,c("CompanyName","CompanySize", "Founded", "funding_round", "money_
outliers
```

```
##                        CompanyName CompanySize Founded funding_round
## 143                       Like.com      Nov-50    1986      Series C
## 146                   Ticketmaster 5001-10,000    1976      Series C
## 152 La Jolla Pharmaceutical Company      51-200    1989      Series C
## 214                Hillshire Brands 5001-10,000    1939      Series B
##     money_raised_float
## 143                 32
## 146                 25
## 152                 12
## 214                 24
```

```r
# store index of outliers
outliers.index = as.numeric(rownames(outliers))

# plot without the outliers
hist(data$Founded[-outliers.index], main = "Year Founded without outliers", breaks = 50)
```



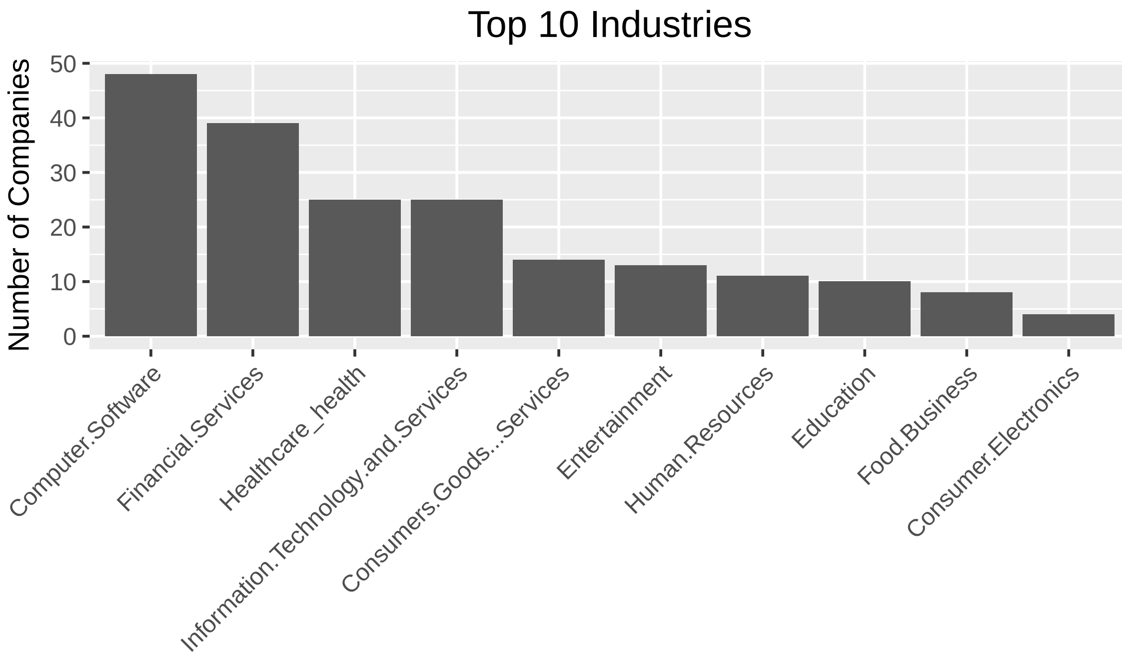


- Industry

We assigned multiple industry labels for each company in the preprocessing phase. This is because cross-industry nature of startups. For example, Netflix is Internet company and at the same time entertainment and media company. The multi-labeling should help our choice of algoritms to incorporate user inputs better by recognizing multiple industry selection as well.

Back to the dataset, top 10 industries are typical industries for startups. There is no surprise.

```
# store column sum in a list
counts = data[, 32:46] %>%
  summarise_each(funs(sum))

# transpose the dataframe for barplot
counts.T = data.frame(total = t(counts), industries = rownames(t(counts) ))

# plot in a descending order
counts.T[1:10,] %>%
  ggplot(.,aes(x = reorder(industries, -total), y = total )) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Top 10 Industries") +
  theme(plot.title = element_text(hjust = 0.5, size=14)) +
  theme(axis.title.x=element_blank()) +
  ylab("Number of Companies")
```



- Funding round

As intended, most companies are in Series B and Series C. Need to merge "series C" and "Series C".

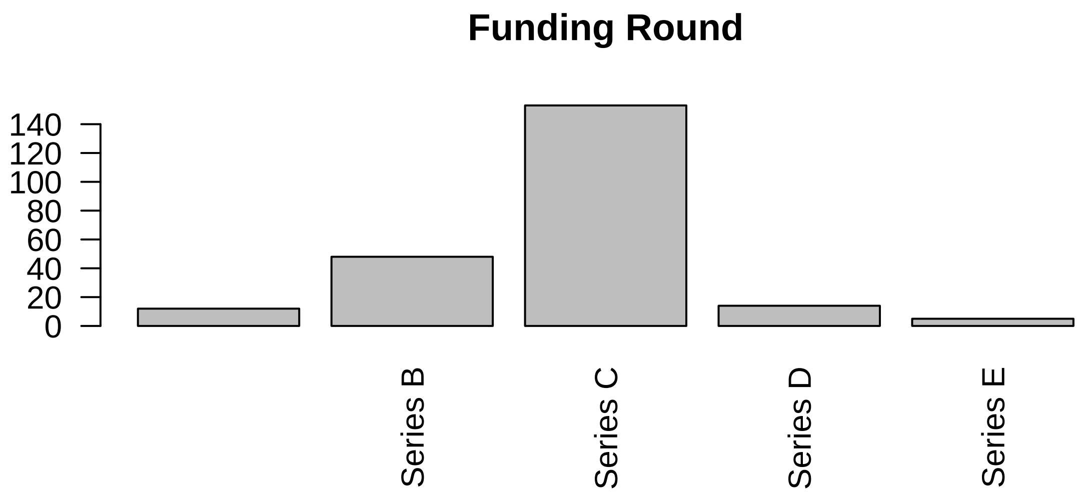
```
# counts = table(data$funding_round)
# counts

data$funding_round[data$funding_round == "series C"] = "Series C"

# remove the level does not occur ("series C")
data$funding_round = factor(data$funding_round)

counts = table(data$funding_round)
counts
```

```
##
##          Series B Series C Series D Series E
##       12       48      153       14        5
```

```
prop.table(table(data$funding_round))
```

```
##
##               Series B   Series C   Series D   Series E
## 0.05172414 0.20689655 0.65948276 0.06034483 0.02155172
```

```
barplot(counts, main = "Funding Round", las = 2)
```



- Money raised

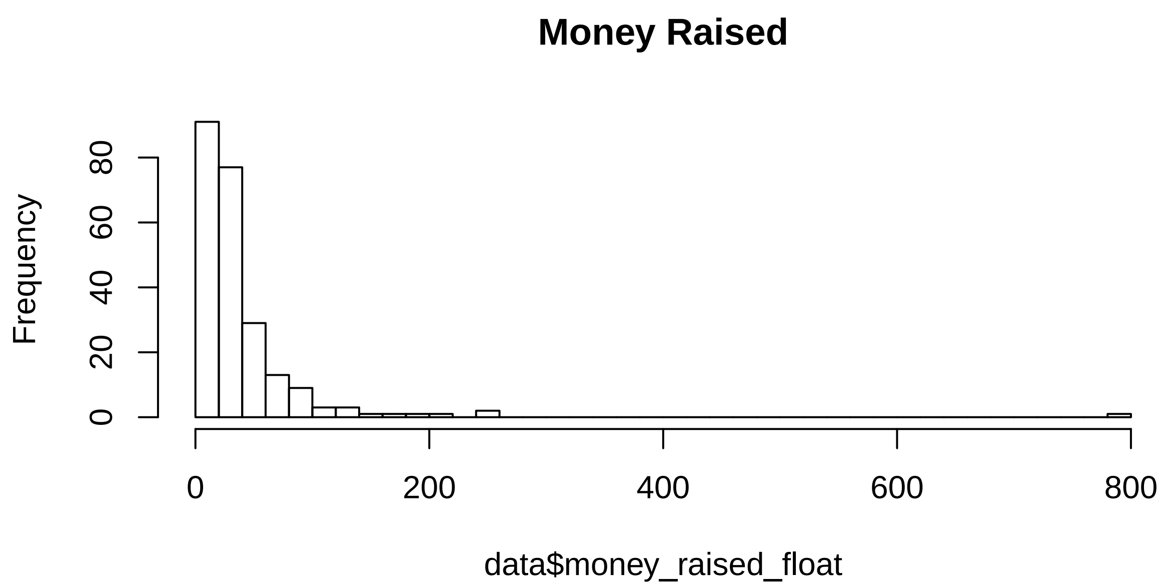
The distribution is skewed to the right as most of companies raised money under $100M. We observe some outliers: Magic Leap, Pivotal, GitHub, and Opendoor.com. Unlike the outliers in the year founded variable, we don't consider removing this data-set because they are still within a definition of startup.

```
summary(data$money_raised_float)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   15.00   25.00   41.17   45.00  793.50
```

```
hist(data$money_raised_float, breaks = 40, main = "Money Raised")

# check the outliers
data[data$money_raised_float > 200, c("CompanyName", "funding_round",
                                      "CompanySize","money_raised_float")]
```

```
##      CompanyName funding_round CompanySize money_raised_float
## 28    Magic Leap      Series C   1001-5000              793.5
## 157      Pivotal      Series C   1001-5000              253.0
## 173       GitHub      Series B    501-1000              250.0
## 230 Opendoor.com      Series D     201-500              210.0
```



- Number of Employees

The disribution is close to normal distribution with a peak at "51-200". But it has outliers as same to other variables: there are two companies with more than 10,000 employees. We don't normally call them startups with that size. I suspect this is because of M&A. These companies might have been purchased by the large corporation and their company size reflect their acquirers.

```
# counts = table(data$CompanySize)
# counts

# str(data$CompanySize)

# clean up - factors
data$CompanySize = revalue(data$CompanySize, c("Nov-50"="11-50", "10-Jan"="1-10"))

# clean up - the level orders
data$CompanySize = factor(data$CompanySize, levels = c(
  "1-10", "11-50","51-200","201-500", "501-1000", "1001-5000","5001-10,000","10,001+"))

counts = table(data$CompanySize)
counts
```

```
##
##        1-10       11-50      51-200     201-500    501-1000   1001-5000
##           1          27         101          64          14          21
## 5001-10,000     10,001+
##           2           2
```

```
barplot(counts, main = "Company size", las=2)

# check the outliers
data[data$CompanySize == "10,001+", c("CompanyName", "Founded", "CompanySize","money_raised_float")]
```

```
##                    CompanyName Founded CompanySize money_raised_float
## 171 eXelate, A Nielsen Company    2007     10,001+                 12
## 189                  Delhivery    2011     10,001+                 85
```

- Country

Since I collected startups from TechCrunch, the US-based news outlet, it turns out 77% startup in the dataset are based in the US. This might also be because the US produces the largest number of startups.

```r
counts = table(data$Country)
counts
```

```
##
##        Belgium         Brazil         Canada          China        Denmark
##              1              1              4              1              1
##         France        Germany          India           iran         Israel
##              1              8              4              1              1
##          Italy          Japan          Korea    New Zealand         Norway
##              1              1              1              1              1
##         Poland         Russia      Singapore         Sweden       Thailand
##              1              1              4              1              1
##         Turkey United Kingdom  United States
##              1             17            178
```

```r
prop.table(counts)
```

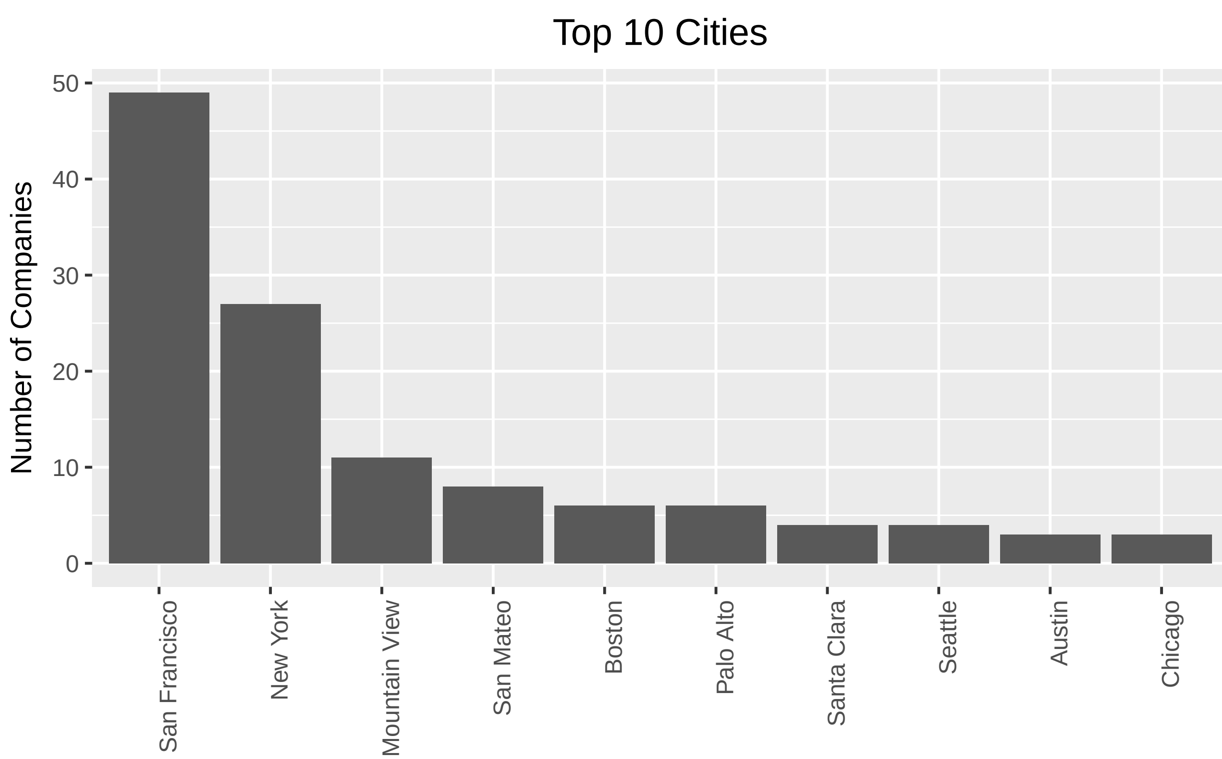
```
##
##        Belgium         Brazil         Canada          China        Denmark
##    0.004310345    0.004310345    0.017241379    0.004310345    0.004310345
##         France        Germany          India           iran         Israel
##    0.004310345    0.034482759    0.017241379    0.004310345    0.004310345
##          Italy          Japan          Korea    New Zealand         Norway
##    0.004310345    0.004310345    0.004310345    0.004310345    0.004310345
##         Poland         Russia      Singapore         Sweden       Thailand
##    0.004310345    0.004310345    0.017241379    0.004310345    0.004310345
##         Turkey United Kingdom  United States
```

```
##    0.004310345    0.073275862    0.767241379
```

- City

The location of startups comes with no suprise. The ranking tops San Francisco, then New York, Moutain View, San Mateo, and Boston.

```
detach(package:plyr)
data %>%
  group_by(City) %>%
  summarize(n = n()) %>%
  arrange(desc(n)) %>%
  filter(City != "") %>%
  slice(1:10) %>%
  ggplot(., aes(x = reorder(City, -n), y = n)) +
  geom_bar(stat = "identity") +
  ggtitle("Top 10 Cities") +
  theme(plot.title = element_text(hjust = 0.5, size=14)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme(axis.title.x=element_blank()) +
  ylab("Number of Companies")
```



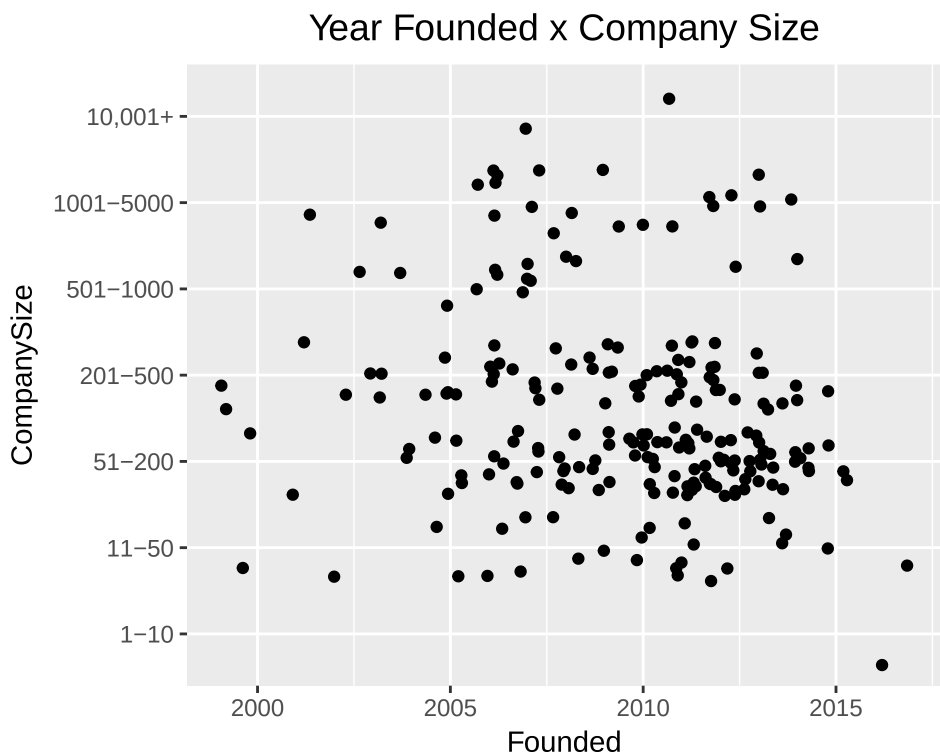
Bivariate Analysis

Analyze the relationship between two variables. Usually it serves two purposes: 1) look at the association between independent variables (explanatory variable) and target variable (in our case, CompanyName) in order to select variables to include in a model to build 2) look at the associations among independent variables to remove highly correlated variables from the inputs for the model. Since our problem is extreme multiclass classification problem where each example has different target variable (company name), we don't run the former analysis described above. Instead, we focus on the latter analysis. Specifically, we look at the relationships for the following variable combinations:

- Company size x Year Founded
- Company size x Money Raised
- Funding round x Money raised
- Location x Money Raised
- Location x Specific industries

- Company_size x Year_founded

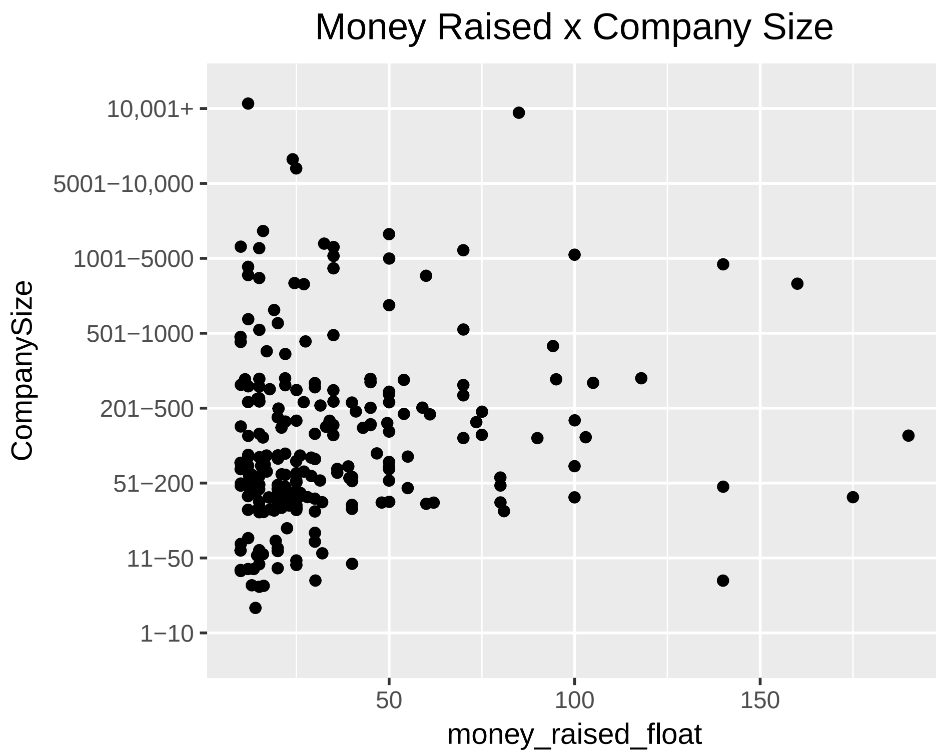
```r
# remove outliers (companies founded before 1990)
outliers = data[data$Founded <= 1990,]
outliers.index = as.numeric(rownames(outliers))

# plot
ggplot(data[-outliers.index,], aes(x= Founded, y = CompanySize)) +
  geom_jitter() +
  ggtitle("Year Founded x Company Size") +
  theme(plot.title = element_text(hjust = 0.5, size=14))
```



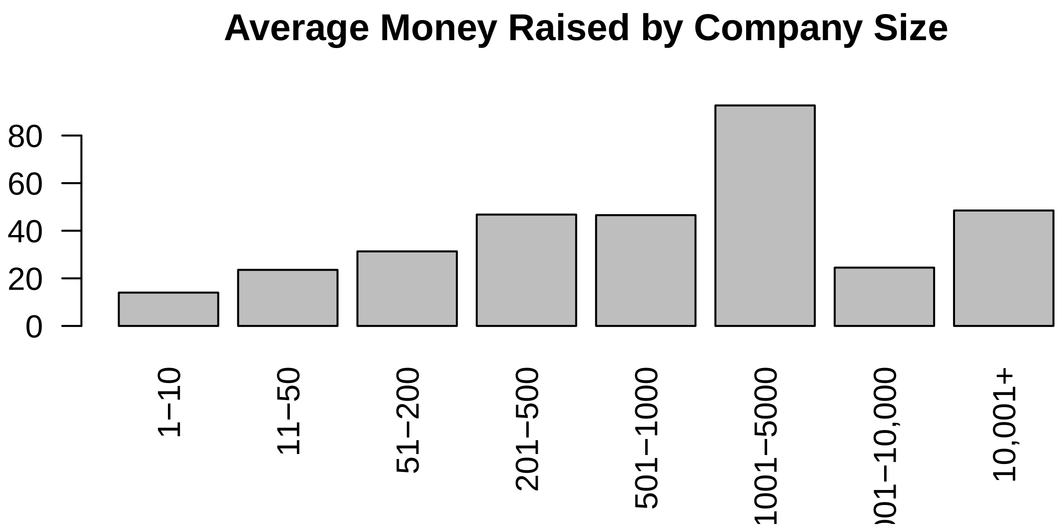
- Company_size x Money_raised

```r
# remove outliers
outliers = data[data$money_raised_float > 200, ]
```

```r
outliers.index = as.numeric(rownames(outliers))

# plot
ggplot(data[-outliers.index,], aes(x= money_raised_float, y = CompanySize)) +
  geom_jitter() +
  ggtitle("Money Raised x Company Size") +
  theme(plot.title = element_text(hjust = 0.5, size=14))
```

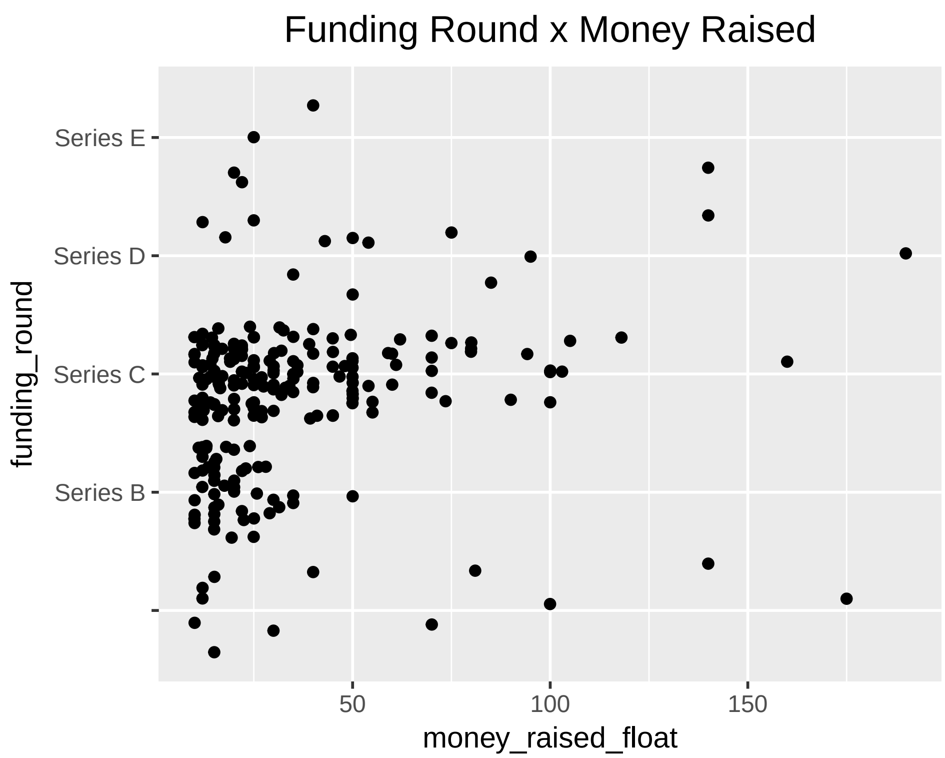



```r
money = data %>%
  group_by(CompanySize) %>%
  summarize(mean = mean(money_raised_float), sd = sd(money_raised_float))

counts = money$mean
names(counts) = money$CompanySize
barplot(counts, las = 2, main = "Average Money Raised by Company Size")
```



- Funding_round x Money_raised

```
# remove outliers
outliers = data[data$money_raised_float > 200, ]
outliers.index = as.numeric(rownames(outliers))

# plot
ggplot(data[-outliers.index,], aes(x= money_raised_float, y = funding_round)) +
  geom_jitter() +
  ggtitle("Funding Round x Money Raised") +
  theme(plot.title = element_text(hjust = 0.5, size=14))
```

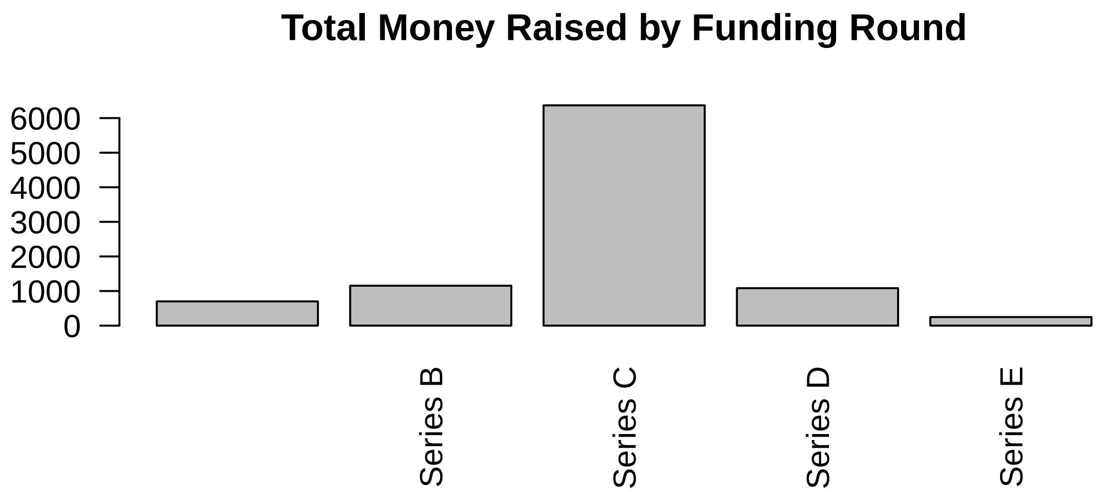


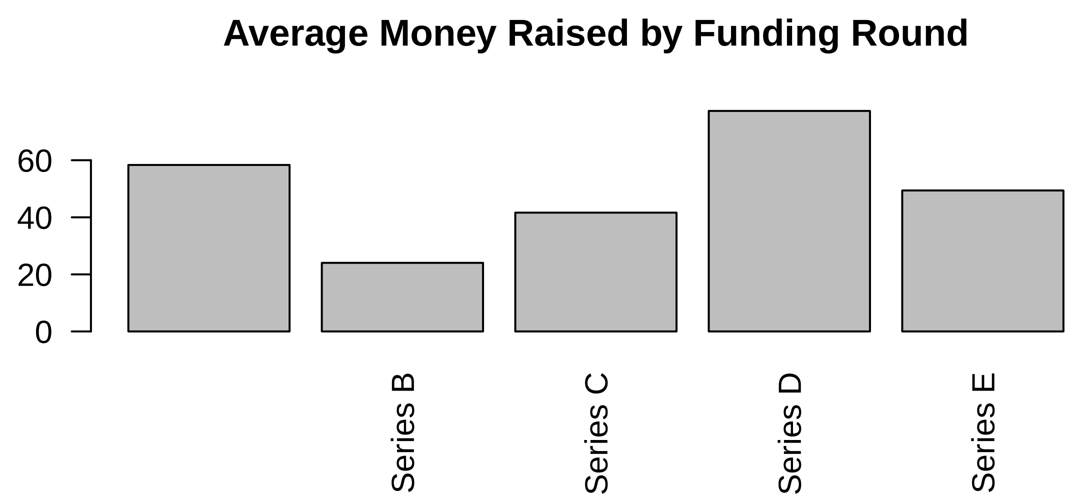
The table below shows mean and standard deviaiton of money raised for companies in each funding round. It makes sense that the mean increases as funding round progresses. Series E has lower mean than Series D. This might be because Series E is more of extension of Series D to sustain funding and not a funding round to drive a company to next level. Also note that stand devidations are quite larger for each round.

```
par(mfrow=c(2,1))

money = data %>%
  group_by(funding_round) %>%
  summarize(sum = sum(money_raised_float), mean = mean(money_raised_float), sd = sd(money_raised_float))
money
```

```
## # A tibble: 5 × 4
##   funding_round    sum     mean       sd
##          <fctr>  <dbl>    <dbl>    <dbl>
## 1                700.0 58.33333 55.75161
## 2      Series B 1153.8 24.03750 34.28641
## 3      Series C 6369.2 41.62876 68.57037
## 4      Series D 1081.8 77.27143 62.11565
## 5      Series E  247.0 49.40000 51.25232
```

```
counts = money$sum
names(counts) = money$funding_round
barplot(counts, las = 2, main = "Total Money Raised by Funding Round")

counts = money$mean
names(counts) = money$funding_round
barplot(counts, las = 2, main = "Average Money Raised by Funding Round")
```

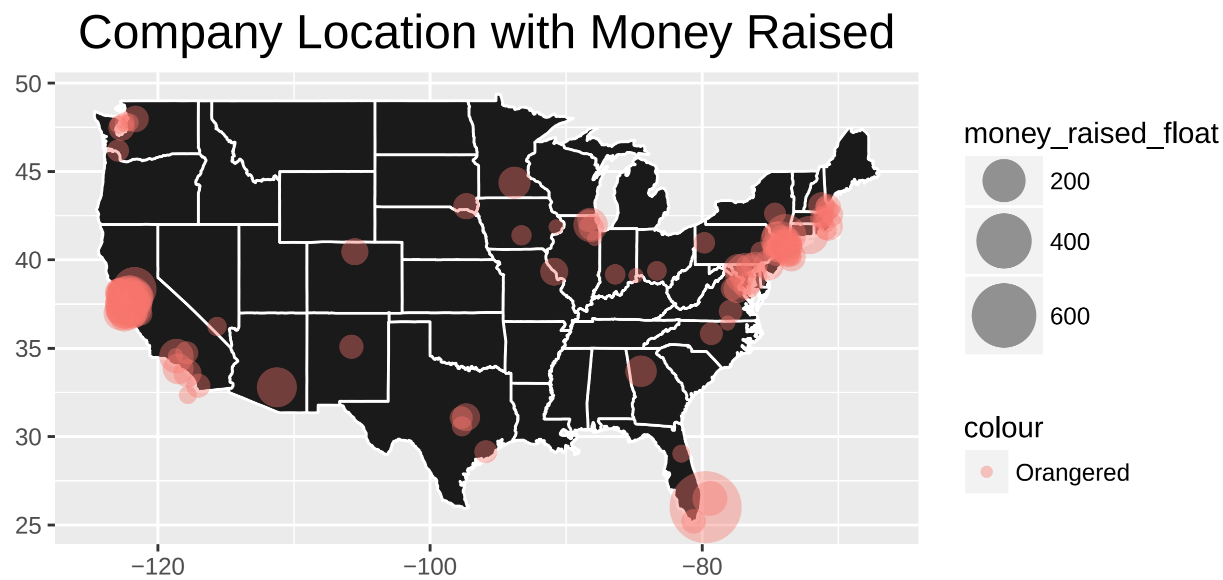




- Location x Money_Raised

```
# Set the map
all_states <- map_data("state")

# plot
ggplot() + geom_polygon(data = all_states, aes(x=long, y = lat, group = group),
                        colour="white", fill="grey10") +
  coord_fixed(1.3) +
  geom_jitter(data = data, mapping = aes(
    x = longitude, y = latitude, color = "Orangered" ,size = money_raised_float),
           alpha = 0.4, width = 0.7, height = 0.7) +
  scale_size(range = c(2, 12)) +
  ggtitle("Company Location with Money Raised") +
  theme(plot.title = element_text(hjust = 0.5, size=18)) +
```

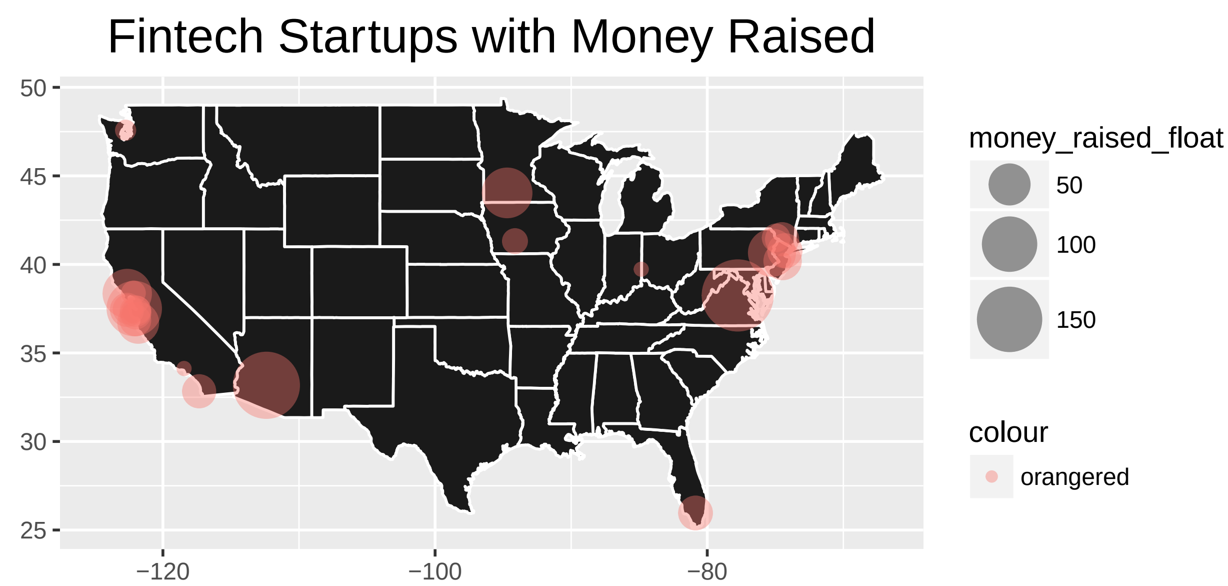
```
  labs(x=NULL, y=NULL) +
  theme(panel.border = element_blank())
```



- Location x Money_Raised for FinTech Startups

```
# Set the map
all_states <- map_data("state")

# plot
ggplot() + geom_polygon(data = all_states, aes(x=long, y = lat, group = group),
                        colour="white", fill="grey10") +
  coord_fixed(1.3) +
  geom_jitter(data = data[data$Financial.Services == 1,], mapping = aes(
    x = longitude, y = latitude, color = "orangered" ,size = money_raised_float),
             alpha = 0.4, width = 0.7, height = 0.7) +
  scale_size(range = c(2, 12)) +
  ggtitle("Fintech Startups with Money Raised") +
  theme(plot.title = element_text(hjust = 0.5, size=18)) +
  labs(x=NULL, y=NULL) +
  theme(panel.border = element_blank())
```

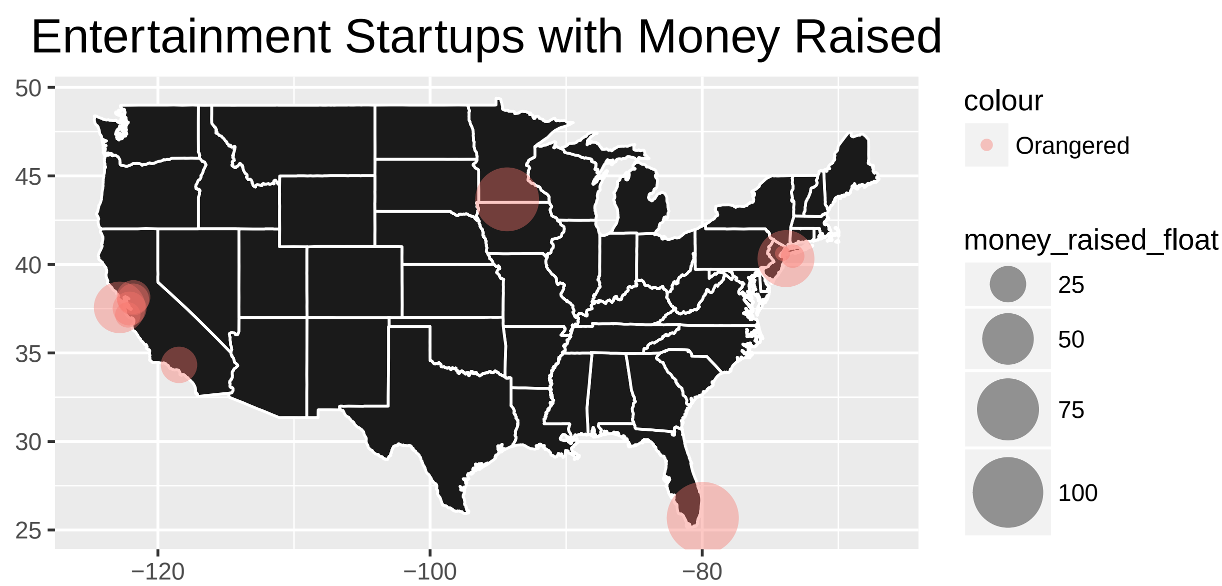
```
## Warning: Removed 13 rows containing missing values (geom_point).
```



- Location x Money_Raised for Entertainment Startups

```r
# Set the map
all_states <- map_data("state")

# plot
ggplot() + geom_polygon(data = all_states, aes(x=long, y = lat, group = group),
                        colour="white", fill="grey10") +
  coord_fixed(1.3) +
  geom_jitter(data = data[data$Entertainment == 1,], mapping = aes(
    x = longitude, y = latitude, color = "Orangered" ,size = money_raised_float),
             alpha = 0.4, width = 0.7, height = 0.7) +
  scale_size(range = c(2, 12)) +
  ggtitle("Entertainment Startups with Money Raised") +
  theme(plot.title = element_text(hjust = 0.5, size=18)) +
  labs(x=NULL, y=NULL) +
  theme(panel.border = element_blank())
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



Conclusion

Follow up the objectives set at the beginning of this data exploration.

- Algorithm Selection

Choose K-Nearest Neighbor for this problem because this is a multiclass classification problem where each example of training data has different target label (CompanyName). Also, the sample size is quite small for now. This is good for KNN.

- Feature Selection

KNN requires dimension reduction because it doesn't handle multidimensional space well. Thus, we need to select some variables that really matters for the model to make predictions. I selected following variables based on bi-variate analysis: money_raised, company_size, location, year_founded, and industry.

- Feature Transformation

KNN assumes each feature to be a numerical variable and scaled properly because KNN is a distance-based algorithm which calculate distance between each test sample and each training test example. It's also important to note that KNN is sensitive to outliers because again it's distance-based algorithm. Given these in mind, we need to make following feature transformations for each variables we selected as the model inputs.

- Money_raised - log transformation for outlier handling and min-max transformation
- Company size - Binning and min-max transformation
- Location - latitude and longitude of cities and min-max transformation for them
- Year Founded - removal of outliers and min-max transformation
- Industry - dummy variable transformation (already done)