Springboard Capstone 3

Carlee Price

July 8, 2017

A STUDY ON THE ANGEL INVESTING LANDSCAPE AND OUTCOMES FOR INVESTORS

WHAT WE HAVE:

A Crunchbase data set that reports investments in seed-stage companies. This data is a sample, reflecting some portion of the total amount of activity in this space. It was published in 2014 and reflected activity in the space up to that point. We're going to select from the global, long-horizon data set those companies that received first funding in the US after 2007.

The goal here is to characterise for potential investors in this space, what the range of potential outcomes (returns on their invested capital) might be, and what proper portfolio construction looks like and means for those same returns.

PART I: WORKING WITH THE CRUNCHBASE DATA

LOAD & TRANSFORM

```r
#read in the file that includes all the company information for this database
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```r
allcompanies = read.csv("companies.csv")
```

```r
#keep only the companies that are US-based
companies <- subset(allcompanies, country_code == "USA")
#we also want to keep only the companies with first_funded_at dates of 2007 or later
#first cast this column as a date
companies$first_funding_at <- format(as.Date(companies$first_funding_at), "%Y/%m/%d")
#then screen for chosen date
```

```
companies <- subset(companies, first_funding_at >= '2007/01/01')
nrow(companies)
```

```
## [1] 25837
```

```
#check for completeness in the resulting set; list of rows that have missing values
nrow(companies[!complete.cases(companies),])
```

```
## [1] 4688
```

```
#what information has been captured for these companies?
colnames(companies)
```

```
##  [1] "ï..permalink"      "name"              "homepage_url"
##  [4] "category_list"     "market"            "funding_total_usd"
##  [7] "status"            "country_code"      "state_code"
## [10] "region"            "city"              "funding_rounds"
## [13] "founded_at"        "founded_month"     "founded_quarter"
## [16] "founded_year"      "first_funding_at"  "last_funding_at"
```

After subsetting the data, we have 25,837 rows of which 4,688 are incomplete. Some of this missing data won't bother us, but in the case of funding_total_usd, it is important. Our study focuses on companies that raise money from outsiders. Our conclusions will largely depend on what happens to the dollars these companies raise. If we don't know in a case (row) how may dollars are involved, it becomes difficult to draw conclusions. Below, we will remove any companies for which the funding total shows N/A.

DATA EXPLORATION

WIN/LOSS RATIO

Starting with the most simple outcome evaluation, let's consider how frequently companies are closed (return falls to zero, all money lost) versus acquired (return is cemented at some non-zero amount). What is the ratio of each within the total population of companies?

```
#transform funding column by stripping punctuation & converting to integer
companies$funding_total_usd <- as.numeric(gsub("[[:punct:]]", "", companies$funding_total_usd))
#strip out companies funded with too small a pool to be relevant for our audience
companies <- subset(companies, funding_total_usd > 100000)
nrow(companies)
```

```
## [1] 19401
```

```
#replace missing status fields
levels(companies$status)[1] <- "unknown"
table(companies$status)
```

```
##
##   unknown  acquired    closed operating
##       418      1592       859     16532
```

```
#find the ratios
table(companies$status)[2]/nrow(companies)
```

```
##   acquired
## 0.08205763
```

```
table(companies$status)[3]/nrow(companies)
```

```
##     closed
## 0.04427607
```

Counting exits (acquired) as a win (8.2%) and closeds as a loss (4.4%) looks encouraging; the ratio is > 1. Will the numbers be different if we look at just one vintage of companies? Let's look just at companies that were funded in 2007, not after. There are 1276 rows in this subset.

```
companies07 <- subset(companies, first_funding_at <= '2007/12/31')
nrow(companies07)
```

```
## [1] 1238
```

```
table(companies07$status)
```

```
##
##   unknown  acquired    closed operating
##         7       322       156       753
```

```
table(companies07$status)[2]/nrow(companies07)
```

```
##  acquired
## 0.2600969
```

```
table(companies07$status)[3]/nrow(companies07)
```

```
##    closed
## 0.1260097
```

Here we see the wins (26.0%) and losses (12.6%) are much higher; an even better ratio. It seems that spending more time in the market leads to more resolutions (closure or acquisition). Remember this data was captured in 2014.

DOLLAR-ON-DOLLAR RETURNS

Let's get more granular, and look not just at the outcome but at the degree of success. Closure of a business can only result in 100% loss but an exit can clearly generate > 100% returns. We're working towards a picture of total portfolio returns.

Add two new data sets from the same source: funding rounds (amounts raised) and acquisitions (exit values). Both will need to be tidied in a similar fashion to our companies table.

```
allrounds = read.csv("rounds.csv")
rounds <- subset(allrounds, company_country_code == "USA")
rounds$raised_amount_usd <- as.numeric(gsub("[[:punct:]]", "", rounds$raised_amount_usd))
allacquisitions = read.csv("acquisitions.csv")
acquisitions <- subset(allacquisitions, company_country_code == "USA")
acquisitions$price_amount <- as.numeric(gsub("[[:punct:]]", "", acquisitions$price_amount))
```

In the process of this exploration, some data-quality issues emerge. We address two of these here.

First, in acquisitions set: Riot Games was acquired by Tenecet for 400Mm, not the USD 4,000 noted.

```
#calling this by index rather than name is dangerous - consider changing
acquisitions$price_amount[5344] <- 400000000
```

Second, in companies set: Aptalis Pharma is listed as having raised a single round (173k seed capital) and then proceeding to exit for $2.9Bn. That would create an impressive return indeed for funders. The reality is that the company actually was spun out from the merger of two established pharmaceutical giants. At the date of formation had seven branded products in market, and a robust development pipeline. There was far more than USD 173k in value within the company. Because this is incorrect and because the actual nature of the enterprise disqualifies it from this (startup) data set, it comes out as well.

We want to add this information to our set of companies, created above, for which funding data exists.

Each of these tables has information useful to our returns analysis. We need to create a unique key for each, in this case namecity (since many of these companies share names with other, unrelated companies created in different places). https://smbrate.com/We can also be selective about bringing only useful rows into the joined table.

```
nrow(rounds)
```

```
## [1] 54313
```

```
#select just the columns we want to see, and just the rows meeting our "seed or angel funding" criteria
rounds2 <- subset(rounds, rounds$funding_round_type %in% c("seed", "angel") & raised_amount_usd >= 10000
#create namecity field for proper joining
rounds2$namecity <- paste(rounds2$company_name, rounds2$company_city)
#eliminate duplicates
rounds2 <- distinct(rounds2, company_name, company_city, .keep_all = TRUE)
```

Repeat formatting steps for companies and acquisitions tables.

```
#select just the columns from companies that we want to see in the new dataset
companies2 <- subset(companies, select = c(name, funding_total_usd, status, state_code, region, city, fu
#remove Aptalis Pharma
companies2 <- companies2[!companies2$name == "Aptalis Pharma",]
acquisitions2 <- subset(acquisitions, select = c(company_name, company_city, acquirer_name, acquired_at
#we can't simply join on name, since there are a number of unique companies that share a name.
#create a new column namecity that includes both the name of the company & the city of its founding
companies2$namecity <- paste(companies2$name, companies2$city)
acquisitions2$namecity <- paste(acquisitions2$company_name, acquisitions2$company_city)
#then join them
fullset <- merge(x = rounds2, y = companies2, by = "namecity", all.x = TRUE)
fullset <- merge(x = fullset, y = acquisitions2, by = "namecity", all.x = TRUE)
```

There are duplicate rows in this merged dataframe. If we screen using unique, we remove 12 rows that are perfectly identical. There are also companies in here which have been acquired twice (see: Forrst). Stakeholders may in this case be getting paid twice, but not necessarily. Further, the amount of the gain in the second case would be the differential between that and the first bid, rather than the entire amount. This gets tricky, but can be properly addressed only where both prices (acquisition 1 and acquisition 2) are disclosed.

Uniqueness for our case will include name + first funding + acquired date. Eliminates 18 rows.

```
nrow(fullset)
```

```
## [1] 6640
```

```
fullset <- distinct(fullset, name, first_funding_at, acquired_at, .keep_all = TRUE)
nrow(fullset)
```

```
## [1] 6166
```

As a result of the join, we have several repeating columns. Let's take them out.

```
drops <- c("company_name.x", "company_state_code", "company_region", "company_city.x", "company_city.y"
fullset <- fullset[ , !(names(fullset) %in% drops)]
```

And we have to filter out for repeats again. We'll also use this opportunity to correct several other faulty datapoints. We're left with 6647 companies in the study set.

```
fullset$acquired_at[fullset$name == "DataPad"] <- "2014-09-30"
fullset$acquired_at[fullset$name == "Modern Feed"] <- "2009-06-01"
fullset <- fullset[!fullset$name == "Buccaneer",]
fullset <- fullset[!(fullset$name == "Roost" & fullset$founded_year == 2013),]
fullset <- subset(fullset, !is.na(fullset$name))
```

Now we're ready to look at returns. Angels want to see exits, as that's the only way they're getting the money back, and the source of their returns. Our "status" field tells us which companies have been acquired and from that we got % exits (successes). We also want dollar-on-dollar % return, which requires acquisition price information (numerator) in addition to total funding information (denominator). Here we see another opportunity to remove corrupt data, specifically OhmData which is purported to have been funded for 185k and acquired for 3Mm just a month later.

We've also made the judgement to remove WhatsApp from the group. This transaction was an outlier to such an extent that it may unfairly impact our analysis & conclusions.

```
nrow(fullset)
```

```
## [1] 6085
```

```
#subset for where acquisition price information is disclosed
acquired <- subset(fullset, price_amount > 1, funding_total_usd >1)
acquired <- acquired[!acquired$name == "OhmData",]
acquired <- acquired[!acquired$name == "WhatsApp",]
acquired <- acquired[!acquired$name == "Medafor",]
nrow(acquired)
```

```
## [1] 90
```

```
#create a new column that shows total $ returned to investors against total $ raised
acquired$return <- acquired$price_amount/acquired$funding_total_usd
#use sum here instead of averaging the acquired$return column as it's more reflective
sum(acquired$price_amount)/sum(acquired$funding_total_usd, na.rm = TRUE)
```

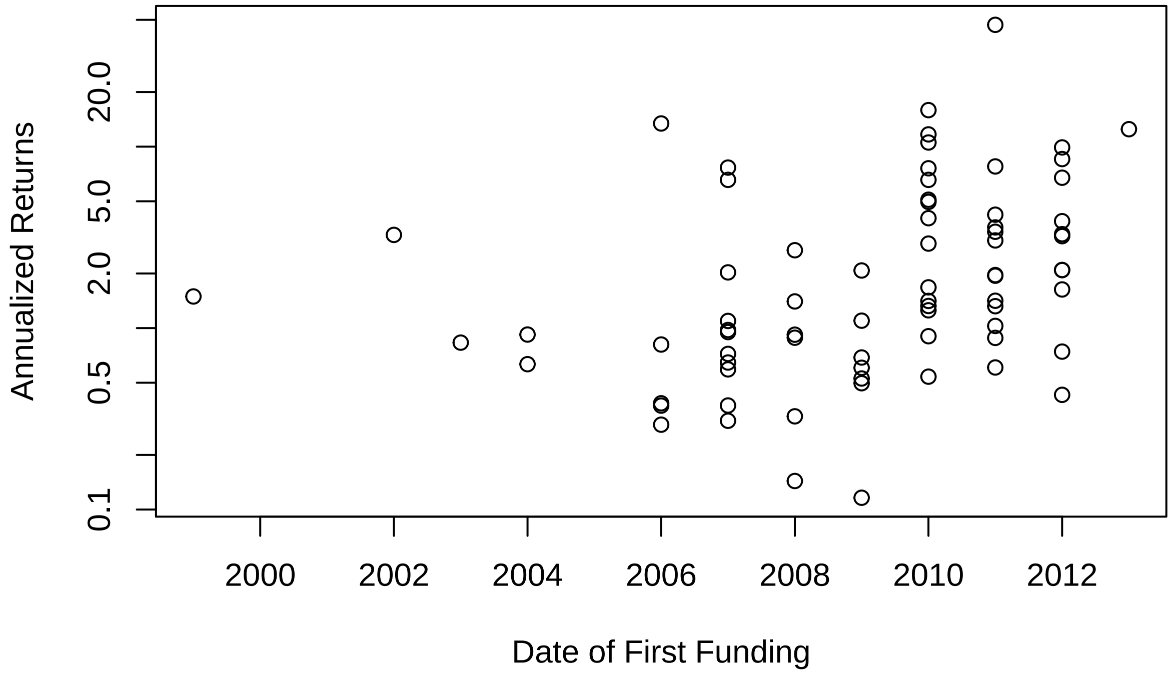
```
## [1] 8.245818
```

Only 90 of the 6085 companies described as "acquired" in our joined set include pricing information. Still a reasonable sample size, and the numbers are encouraging (8.24X return on total amount raised) but let's look closer. First, to annualize results.

Let's create a field that shows annualized returns by company, and then plot these against time to see if there are any interesting trends.

```
#first convert acquired_at to proper date format
acquired$acquired_at <- format(as.Date(acquired$acquired_at), "%Y/%m/%d")
#then to calculate how many days a company spent between funding and acquisition
acquired$age_at_acquisition <- (as.integer(as.Date(acquired$acquired_at) - as.Date(acquired$first_fundi
acquired <- subset(acquired, age_at_acquisition >= 1)
#whiche we then use to calculate annualized returns
acquired$annual_return <- (acquired$return ^ (365.25/as.integer(acquired$age_at_acquisition)) - 1)
plot(acquired$founded_year, acquired$annual_return, log = "y", ylab = "Annualized Returns", xlab = "Dat
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 8 y values <= 0 omitted
## from logarithmic plot
```



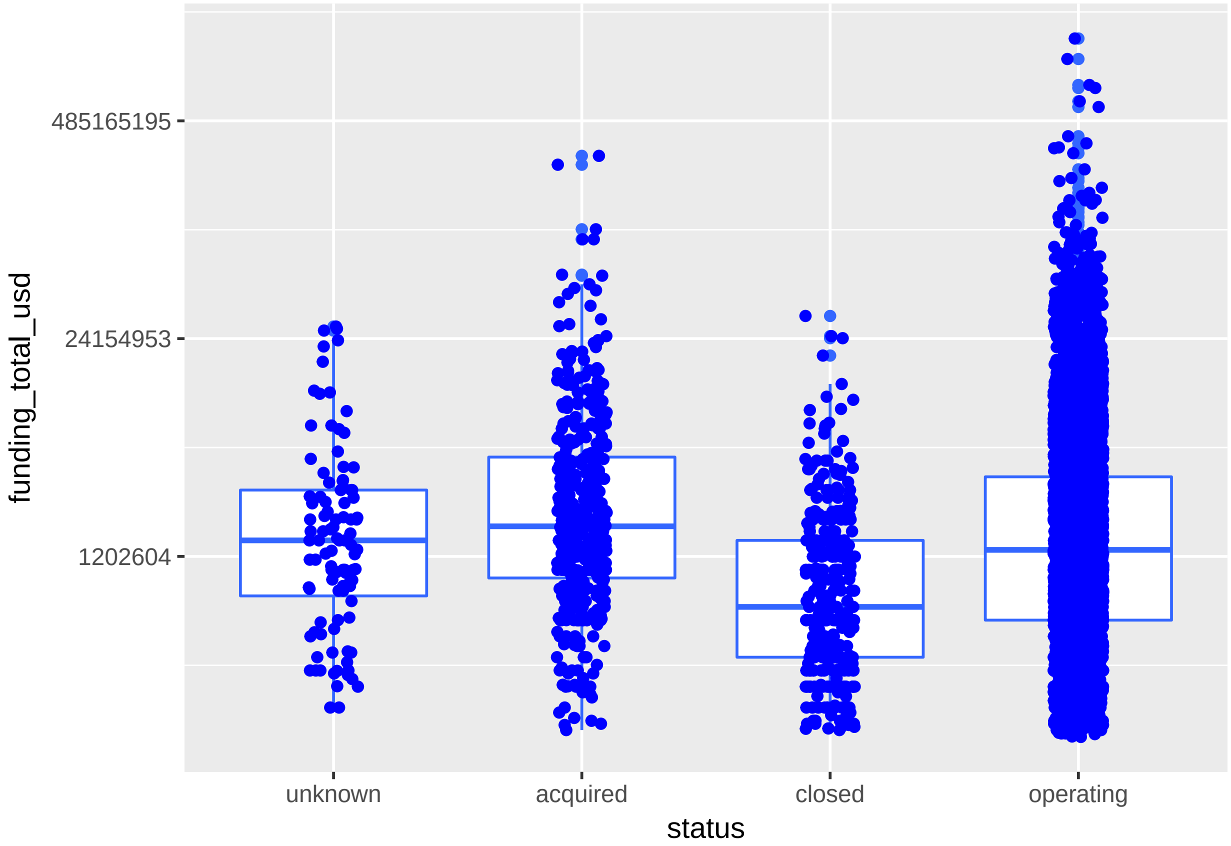
```
mean(acquired$annual_return)
```

```
## [1] 3.179139
```

```
(sum(acquired$price_amount)/sum(acquired$funding_total_usd))^(365.25/(mean(acquired$age_at_acquisition)
```

```
## [1] 1.106313
```

```
library(ggplot2)
#ggplot(fullset, aes(x=status, y=funding_total_usd)) + geom_boxplot() + scale_y_continuous(trans = "log
ggplot(fullset, aes(status, funding_total_usd)) + geom_boxplot(colour = "#3366FF") +
  geom_jitter(width = 0.1, colour = "blue") +
  scale_y_continuous(trans = "log")
```



We know what returns on our fully-documented transactions are, but in order to get a full picture of the space, we need to take into account the other transactions as well. For this we'll go back to our fullset and work to understand the metrics here. We can step closer to the truth on returns by keeping sum of disclosed prices in the numerator, but changing the denominator to all the funds put to work during this same period.

```
sum(acquired$price_amount)/sum(acquired$funding_total_usd)
```

```
## [1] 8.114933
```

```
sum(acquired$price_amount)/sum(fullset$funding_total_usd)
```

```
## [1] 0.2423171
```

This gives us a sense of the magnitude of effect including our incomplete cases in the analysis will have. We start by differentiating priced/acquired from unpriced/acquired

```
fullset$newstatus <- factor(fullset$status, levels = c(levels(fullset$status), "priced acquired"))
fullset$newstatus[which(fullset$status == "acquired" & fullset$price_amount >= 1)] <- "priced acquired"
```

It also may be useful to add an age column for the remaining companies also. When moving from age to return, however, we notice that the $ returned on disclosed acquisitions is below the total amount raised by this group. So returns are going to be negative. We must work to fill in some of these missing values.

```
end <- as.Date("12/31/14", "%m/%d/%y")
fullset$age <- ifelse(is.na(fullset$acquired_at), as.integer((end - as.Date(fullset$first_funding_at)))
                          as.integer(as.Date(fullset$acquired_at) - as.Date(fullset$first_funding_at)))
(sum(acquired$price_amount)/sum(fullset$funding_total_usd))^(365.35/(mean(fullset$age, na.rm = TRUE)))-
```

```
## [1] -0.3909465
```

```
#remove two rows that show negative ages for companies
fullset <- subset(fullset, age >= 1)
fullset %>%  group_by(status) %>%
          summarise(raised = sum(funding_total_usd), rounds = mean(funding_rounds), age = mean(age))
```

```
## # A tibble: 4 × 4
##      status      raised   rounds       age
##      <fctr>       <dbl>    <dbl>     <dbl>
## 1   unknown   342112681 1.969697  945.0404
## 2  acquired  2832191052 2.201878  908.7559
## 3    closed   509198575 1.566456 1836.6519
## 4 operating 33473608764 2.120755 1010.0462
```

INFERENCES AROUND MISSING DATA

STATUS ACQUIRED, PRICE MISING

First let's subset the companies acquired without disclosed price

```
table(fullset$status)
```

```
##
##   unknown  acquired    closed operating
##        99       426       316      5242
```

```
guess <- subset(fullset, fullset$status == "acquired" & is.na(fullset$price_amount), drop = TRUE)
nrow(guess)
```

```
## [1] 339
```

Let's have a crack at populating the "price_amount" column for these companies. We know that on average these companies raised 42% of what the "priced acquired" group did.

```
mean(guess$funding_total_usd)/mean(acquired$funding_total_usd)
```

```
## [1] 0.3971631
```

But there's no reason to believe that the returns generated on exit by this group are distributed differently than for our "price-disclosed" group. So we'll assign inferred-price amounts that result in a comparable distribution of returns for this group.

```
#THIS IS JUST MATH BUT I'M SO IMPRESSED WITH MYSELF I'M LEAVING IT IN
log(acquired$annual_return[1] + 1) * (acquired$age_at_acquisition[1]/365.25)
```

```
## [1] 1.624942
```

```
#is equal to
log(acquired$price_amount[1]/acquired$funding_total_usd[1])
```

```
## [1] 1.624942
```

```
#and as luck would have it,
exp((log(acquired$annual_return[1] + 1)) * (as.integer(acquired$age_at_acquisition[1])/365.25))
```

```
## [1] 5.078125
```

```
#is equal to
acquired$price_amount[1]/acquired$funding_total_usd[1]
```

```
## [1] 5.078125
```

```
#which further means that
acquired$funding_total_usd[1] * (exp((log(acquired$annual_return[1] + 1)) * (as.integer(acquired$age_at_
```

```
## [1] 6.5e+07
```

```
#is equal to
acquired$price_amount[1]
```

```
## [1] 6.5e+07
```

We need to break the returns into two subgroups: young companies (which are distributed toward higher returns) and old companies (lower returns). Failing to do so results in inferred prices that are unrealistic, because of a magnification that happens when applying high returns to long-lived companies.

```
#start by capturing the 87 documented returns from into two different lists
guessreturnold <- acquired$annual_return[which(acquired$age_at_acquisition >= mean(acquired$age_at_acqu
length(guessreturnold)
```

```
## [1] 36
```

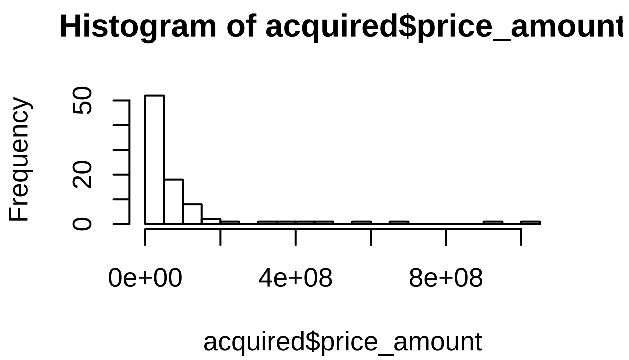
```
guessreturnyoung <- acquired$annual_return[which(acquired$age_at_acquisition < mean(acquired$age_at_acqu
length(guessreturnyoung)
```

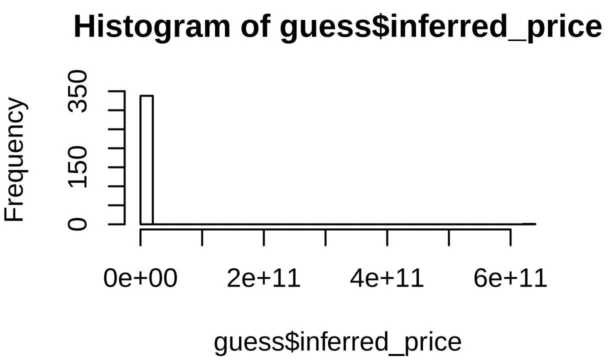
```
## [1] 53
```

```
#then we draw randomly (with replacement) from that vector to populate the 381 "acquired price unknown"
#this ensures that the distribution of returns between our "disclosed" and "undisclosed" groups are the
guess$inferred_return[which(guess$age >= mean(guess$age))] <- sample(guessreturnold, size = length(whic
guess$inferred_return[which(guess$age < mean(guess$age))] <- sample(guessreturnold, size = length(which
#we then work backward from the assigned annual return number to calculate the implied price amount (ma
guess$inferred_price <- guess$funding_total_usd * (exp((log(guess$inferred_return + 1)) * (as.integer(g
```

Let's have a quick look to see if our resulting inferred prices look correct-ish.

```
par(mfrow=c(2,2))
hist(acquired$price_amount, breaks = 25)
hist(guess$inferred_price, breaks = 25)
par(mfrow=c(1,1))
```





So, there's a $17Bn acquisition in there (and three others over a billion), which seems improbable on the basis that a transaction of this size could simply not be done on the down-low. We'll scale down these four deals by a factor of 10 which is entirely unscientific and terrible practice but I'm at a loss for alternatives.

```
guess$ inferred_price[which(guess$inferred_price > 650000000)] <- guess$inferred_price[which(guess$infe
```

Now that we've populated these rows with informed-ish prices, let's have a fresh look at returns.

```
sum(guess$inferred_price)/sum(guess$funding_total_usd)
```

```
## [1] 41.49271
```

STATUS OPERATING, POTENTIALLY ACQUIRABLE

We'd also like to make some inferences about our Operating companies, as some of these will go on to be acquired outside our visible horizon. To do this, we need to put our acquired and guess datasets together.

First let's look at the age structute of our entire (priced and unpriced) acquireds, first by recreating the dataset.

```
#change status to differentiate where we've made assumptions
guess$status <- "undisclosed acquired"
acquired$status <- "disclosed acquired"
#move our inferred prices into the price_amount column for guess
guess$price_amount <- guess$inferred_price
guess$newstatus <- NULL
guess$inferred_price <- NULL
acquired$newstatus <- NULL
colnames(guess)[18] <- "annual_return"
acquired$return <- NULL
```

```
colnames(acquired)[17] <- "age"
```

And now that the tables match, we merge them together. Since we've taken pains to ensure that the columns all match, we can use an rbind.

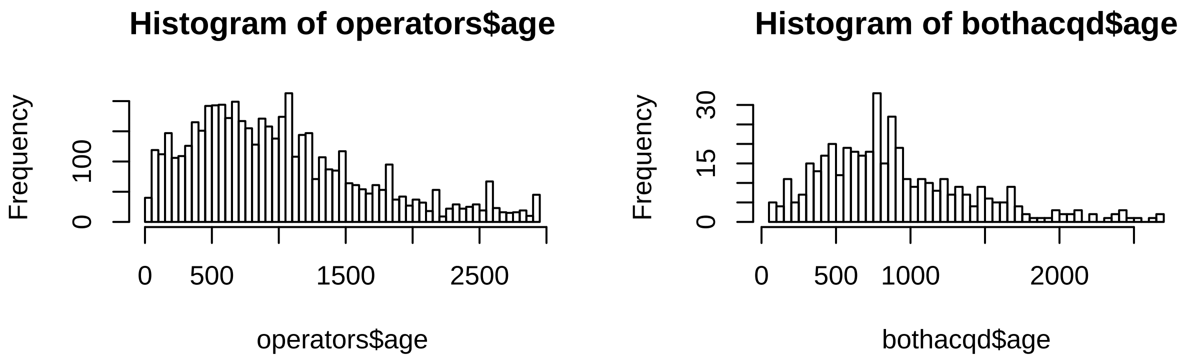
```
colnames(guess) == colnames(acquired)
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE TRUE TRUE
```

```
bothacqd <- rbind(acquired, guess)
```

Then comparing histograms.

```
operators <- subset(fullset, status == "operating")
par(mfrow=c(2,2))
hist(operators$age, breaks = 50)
hist(bothacqd$age, breaks = 50)
par(mfrow=c(1,1))
```



First let's look at the age structute of our acquired set:

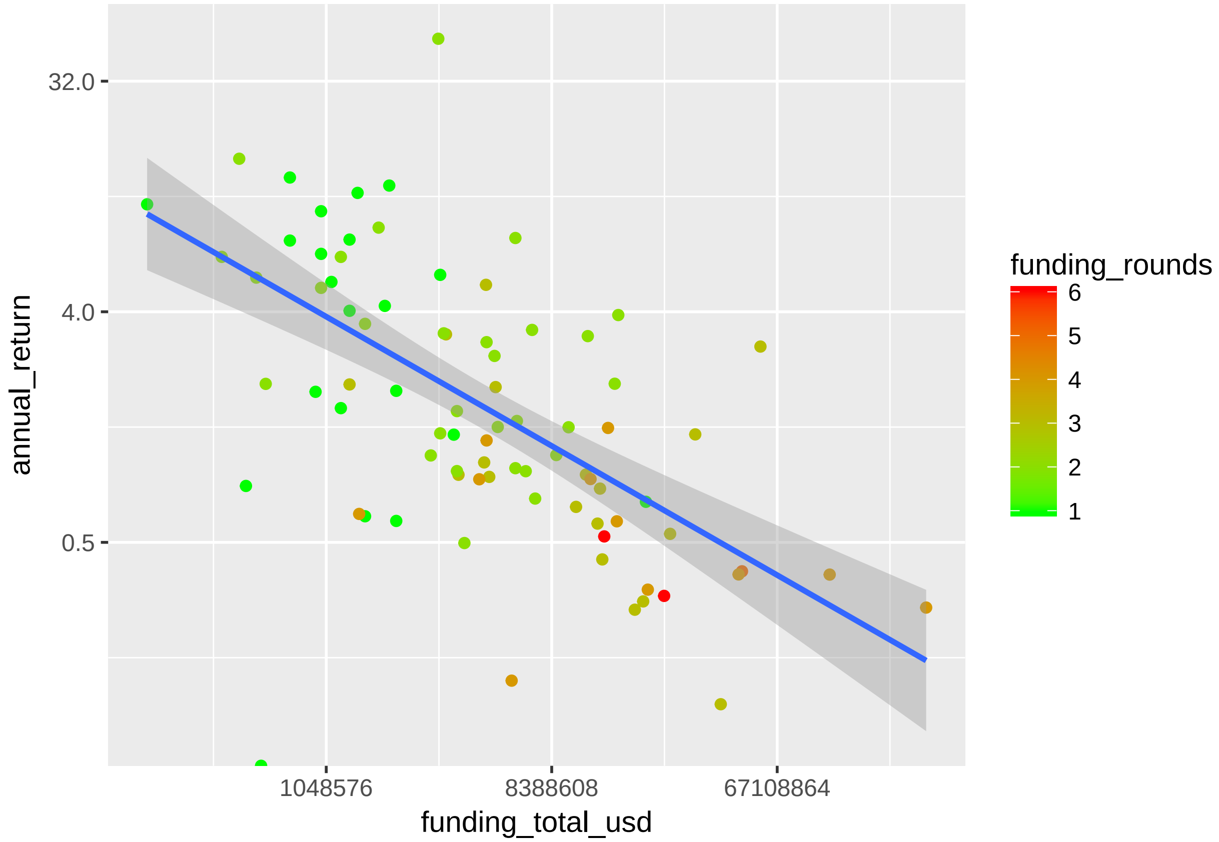
IDENTIFYING DRIVERS OF SUCCESS/FAILURE

DOES RAISING MORE MONEY LEAD TO BETTER RETURNS?

We can also use our returns column to draw some conclusions about returns relative to amounts raised. The companies we hear about in the press tend to be frequently in front of investors; they're cash-consumptive and generally percieved as fancy. But do these companies outperform their less capital-intensive peers? A regression analysis should give a clue.

```
ggplot(acquired, aes(funding_total_usd, annual_return)) +
 geom_point(aes(color = funding_rounds)) +
 scale_color_gradient(low = "green", high = "red") +
 geom_smooth(method ="lm") +
 scale_y_continuous(trans = "log2") +
  scale_x_continuous(trans = "log2")
```

```
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 8 rows containing non-finite values (stat_smooth).

## Warning: Removed 7 rows containing missing values (geom_point).
```

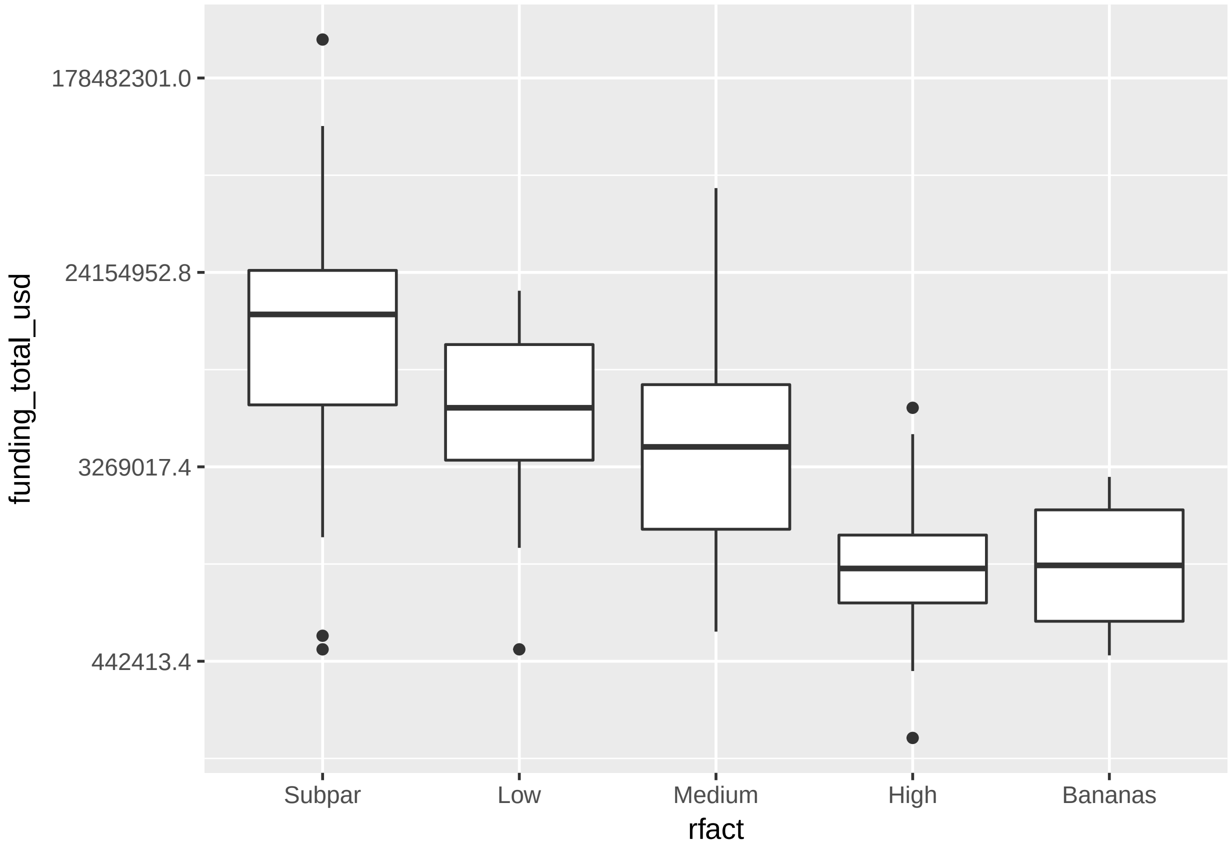


The slope here is negative, which tells us that companies which raise more tend to return less dollar-for-dollar. This may suprise some. You'll also notice that companies that raise a large number of rounds, struggle to generate returns

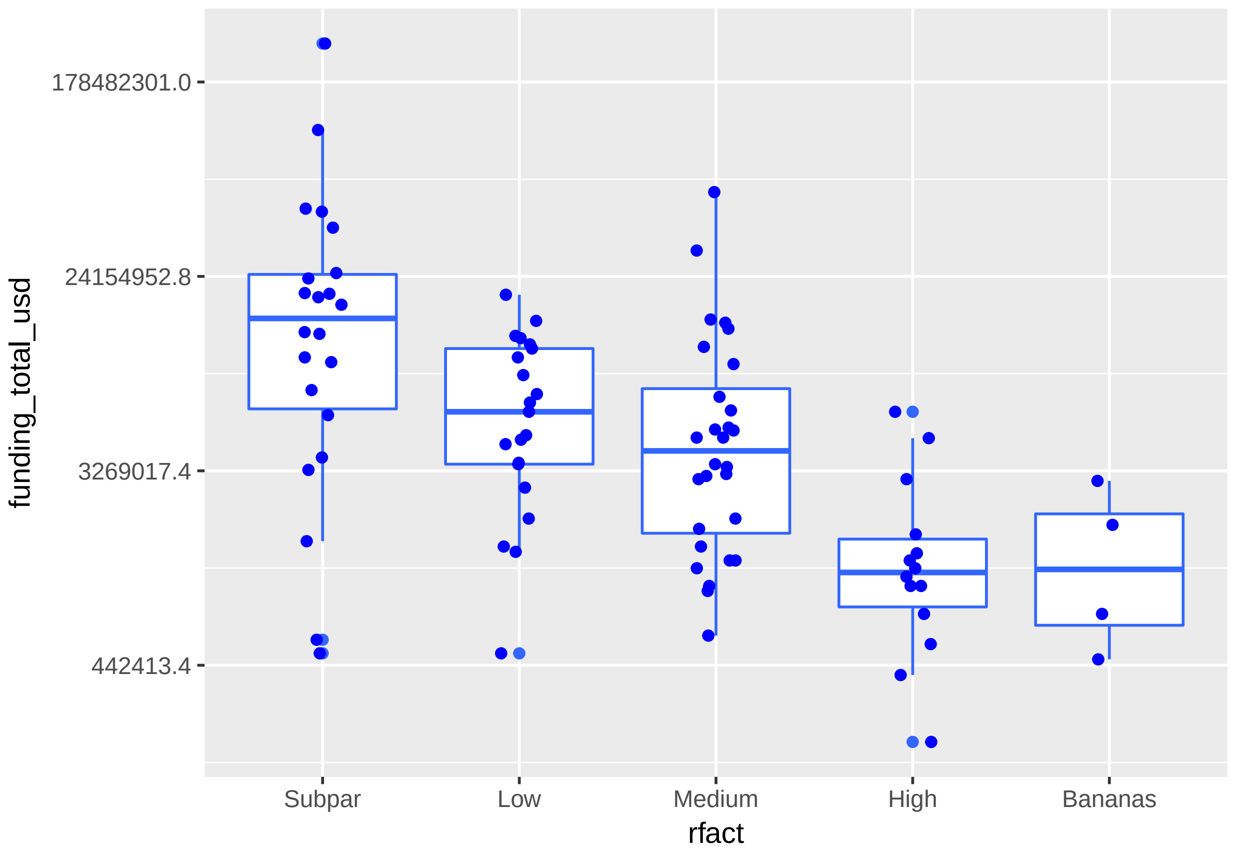
Next we factor these companies by returns, and look at the funding characteristics (amounts raised) by bucket.

```
#set breaks for each funding bucket, levels came about through trial & error
grp <- c(-1, .5575, 1.25, 5, 12, 60)
```

```
#set factors & assign names
acquired$rfact = cut(acquired$annual_return, breaks = grp,labels=c('Subpar','Low', 'Medium','High', 'Ba
#create a boxplot
#plot(x = acquired$rfact, y = acquired$funding_total_usd, log = 'y', ylab = "funds raised", xlab = "qua
ggplot(acquired, aes(x=rfact, y=funding_total_usd)) + geom_boxplot() + scale_y_continuous(trans = "log"
```



```
ggplot(acquired, aes(rfact, funding_total_usd)) + geom_boxplot(colour = "#3366FF") +
  geom_jitter(width = 0.1, colour = "blue") +
  scale_y_continuous(trans = "log")
```



Here again we can see that companies are most likely to appear generate "bananas" returns if they've raised less money. Likewise, those companies that have raised the most are in the lower-tier returns buckets. Interesting!

Is it a statistically significant relationship? Let's get the z score.

```
#there must be an easier way to do this
ftusd <- sd(acquired$funding_total_usd, na.rm = TRUE)
ftumn <- mean(acquired$funding_total_usd, na.rm = TRUE)
a <- subset(acquired, acquired$rfact == "Bananas")
amean <- mean(a$funding_total_usd)
(amean - ftumn)/(ftusd/(sqrt(nrow(a))))
```

```
## [1] -0.7047281
```

```
b <- subset(acquired, acquired$rfact == "Subpar")
bmean <- mean(b$funding_total_usd)
(bmean - ftumn)/(ftusd/(sqrt(nrow(b))))
```

```
## [1] 2.964359
```

Returns in the Subpar category are much more of an outlier than those in the Bananas category. Raising large amounts of money is very coincident with lower returns.

We can get into SQL-like queries by using the dplyr pipe operator. Let's summarise some statistics by factor.

```
#I don't know why this is no longer working
acquired %>%  group_by(rfact) %>%
          summarise(raised = mean(funding_total_usd), rounds = mean(funding_rounds), return = mean(annua
## # A tibble: 5 × 4
##     rfact   raised   rounds     return
##     <ord>    <dbl>    <dbl>      <dbl>
## 1  Subpar 32119315 3.454545  0.1284059
## 2     Low  7260748 2.523810  0.8327669
## 3  Medium  7866682 2.107143  2.4628154
## 4    High  1726857 1.500000  7.4962501
## 5 Bananas  1510547 1.500000 22.1810043
```

We can also have a look at how well these characteristics interact with each other.

Let's start by building a predictor model from the set of medium returns (5 - 15X):

```
#medset <- subset(bothacqd, rfact == "Medium")
#fit = lm(annual_return ~ funding_total_usd, medset) # Run a regression analysis
#par(mfrow=c(2,2)) # Change the panel layout to 2 x 2
#plot(fit)
#par(mfrow=c(1,1))
```

USING K-MEANS CLUSTERING TO DO STUFF