# Final_Project_Javier_Rojas

*Javier Rojas*

*12/16/2018*

Venture Capital Fund Flows from 1995 - Present: How has the venture capital funding landscpape changed over the past 20+ years? How about individual deal types? How active have individual investors been? How are these deals spread across the U.S in any particular year? Which business areas are the most popular for any year?

For my final project, I decided to delve into a dataset of ~200K entries, each entry representing a different startup investment. Data such as company names, investor names, company locations, funds raised, and business areas are included in the dataset. I created exploratory economic plots using functions in which a user can query from set parameters, a word cloud of business areas, and a K-means clustering global map of startups.

For next time I would change a text filter I use in Plot #6 where city names including accents were filtered out. This was due to character formatting in the database file, and the speed at which my computer can handle high-volume iterations in R.

## Load Libraries

```
library(dplyr)
```

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

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

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:lubridate':
##
##     here
```

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

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(httr)
```

```
##
## Attaching package: 'httr'
```

```
## The following object is masked from 'package:caret':
##
##     progress
```

```
library(jsonlite)
library(ggmap)
library(maps)
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:plyr':
##
##     ozone
```

```
library(scales)
```

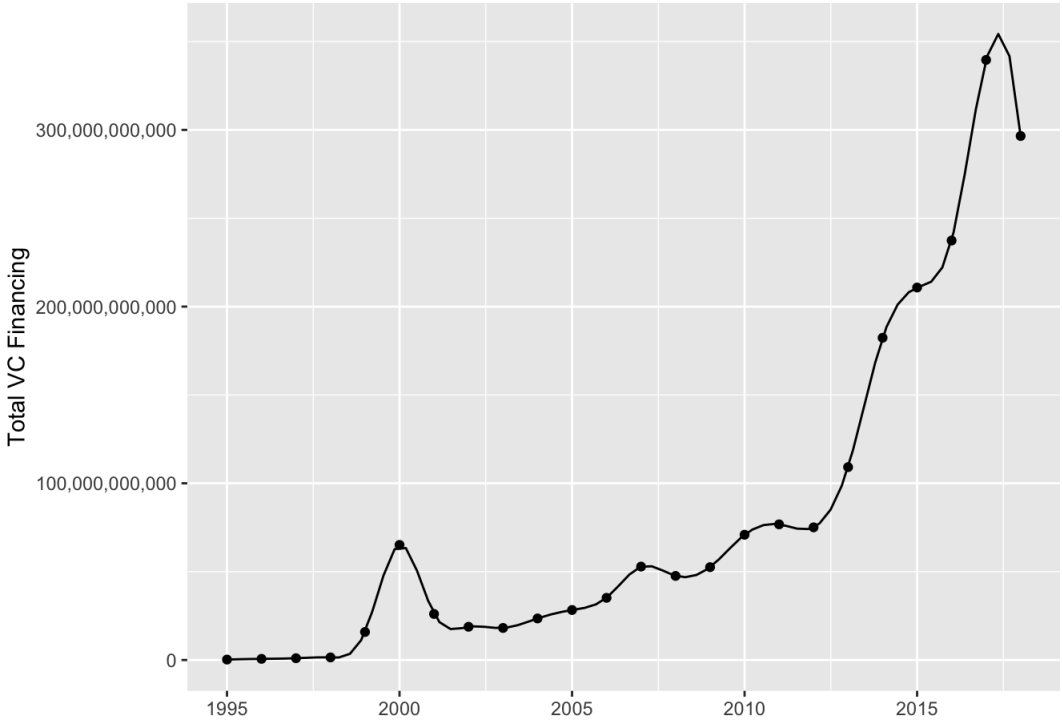
# Plot 1: Total Financing

```
setwd("/Users/jrojas/Desktop/Intro_Data_Sci")
crunchbase <- read.csv("crunchbase_export.csv", header = T,
                           stringsAsFactors = F)
get_summary_stats <- function(df){
  crunchbase$year <- year(crunchbase$announced_on)
  crunchbase$year <- as.numeric(crunchbase$year)

  cb <- filter(crunchbase, raised_amount_usd != "")
  cb$raised_amount_usd <- gsub(",","", cb$raised_amount_usd)



  cb$raised_amount_usd <- as.integer(cb$raised_amount_usd)

  cb <- filter(cb, !is.na(raised_amount_usd))

  cb <- filter(cb, year >= 1995)

  years <- unique(cb$year) %>% sort

  #total, average
  total_mkt_val <- rep(NA, length(years))
  avg_mkt_val <- rep(NA, length(years))

  #volatility of market value
  vol_mkt_val <- rep(NA, length(years))

  for (i in 1:length(years)){
    cy <- years[i]
    cur_df <- filter(cb, year == cy)
    total <- sum(cur_df$raised_amount_usd)
    avg <- mean(cur_df$raised_amount_usd)
    vol <- sd(cur_df$raised_amount_usd)
    total_mkt_val[i] <- total
    avg_mkt_val[i] <- avg
    vol_mkt_val[i] <- vol
  }

  summary.stat.df <- data.frame(cbind(years, total_mkt_val, avg_mkt_val,  vol_mkt_val))

  return(summary.stat.df)
}


summary.stats <- get_summary_stats(crunchbase)
```

```
## Warning in get_summary_stats(crunchbase): NAs introduced by coercion
```

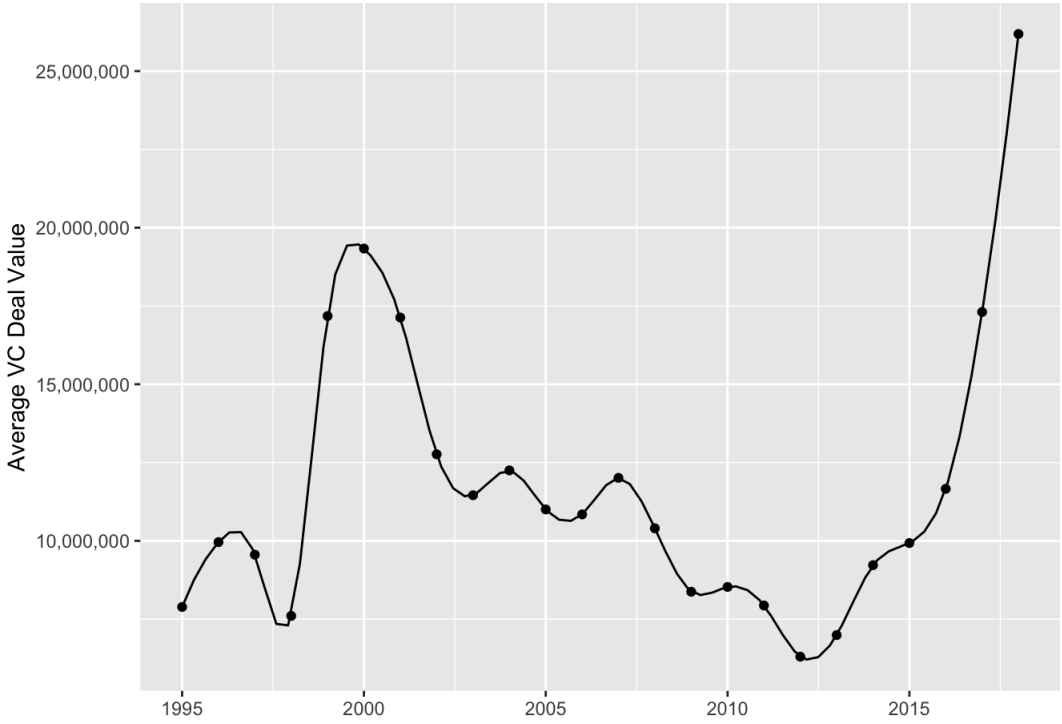
```
## Warning in get_summary_stats(crunchbase): NAs introduced by coercion to
## integer range
```

```
#data frames to plot total val, avg val, val volatility
val_df <- data.frame(cbind(summary.stats$years, summary.stats$total_mkt_val))
avg_df <- data.frame(cbind(summary.stats$years, summary.stats$avg_mkt_val))
vol_df <- data.frame(cbind(summary.stats$years, summary.stats$vol_mkt_val))

#splines
total.spline.df <- as.data.frame(spline(val_df))
avg.spline.df <- as.data.frame(spline(avg_df))
vol.spline.df <- as.data.frame(spline(vol_df))

total_financing_plot <- ggplot(summary.stats) + geom_point(data = summary.stats,
                                    aes(years, total_mkt_val)) +
                          scale_y_continuous(labels = comma) +
                          geom_line(data = total.spline.df, aes(x=x, y=y)) +
                          xlab("") + ylab("Total VC Financing")

avg_financing_plot <- ggplot(summary.stats) + geom_point(data = summary.stats,
                                  aes(years, avg_mkt_val)) +
                                  scale_y_continuous(labels = comma) +
                                  geom_line(data = avg.spline.df, aes(x=x, y=y)) +
                                  xlab("") + ylab("Average VC Deal Value")

vol_financing_plot <- ggplot(summary.stats) + geom_point(data = summary.stats,
                                  aes(years, vol_mkt_val)) +
                                  scale_y_continuous(labels = comma) +
                                  geom_line(data = vol.spline.df, aes(x=x, y=y)) +
                                  xlab("") + ylab("Deal Value Volatility")


total_financing_plot
```



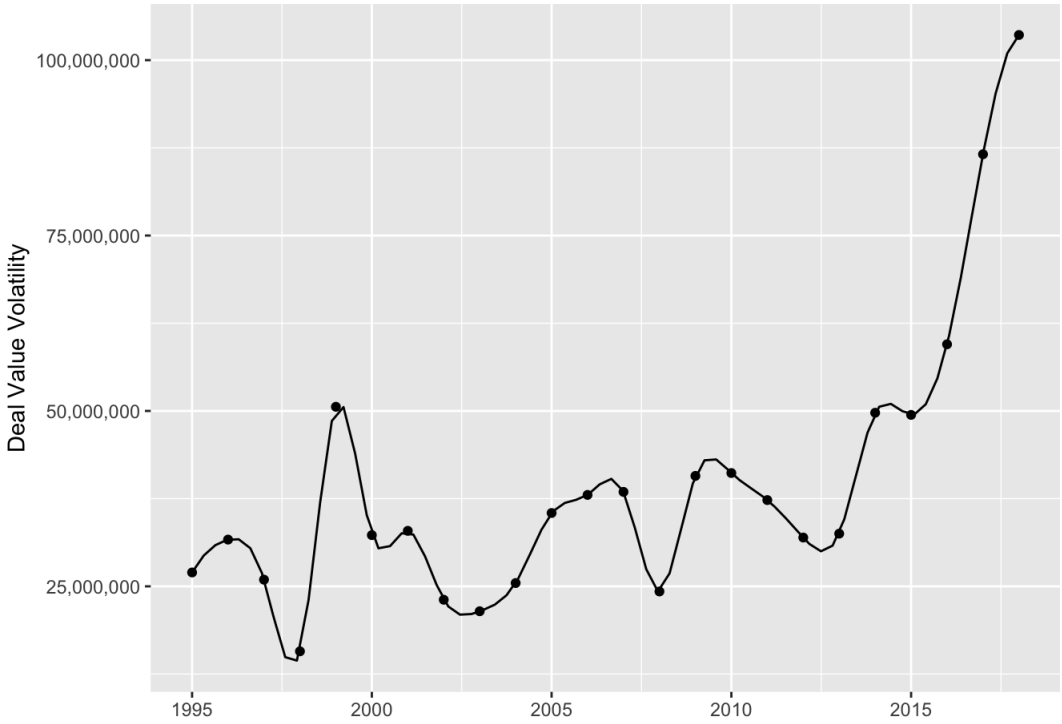

# Plot 2: Average Financing Deal Size

```
avg_financing_plot
```



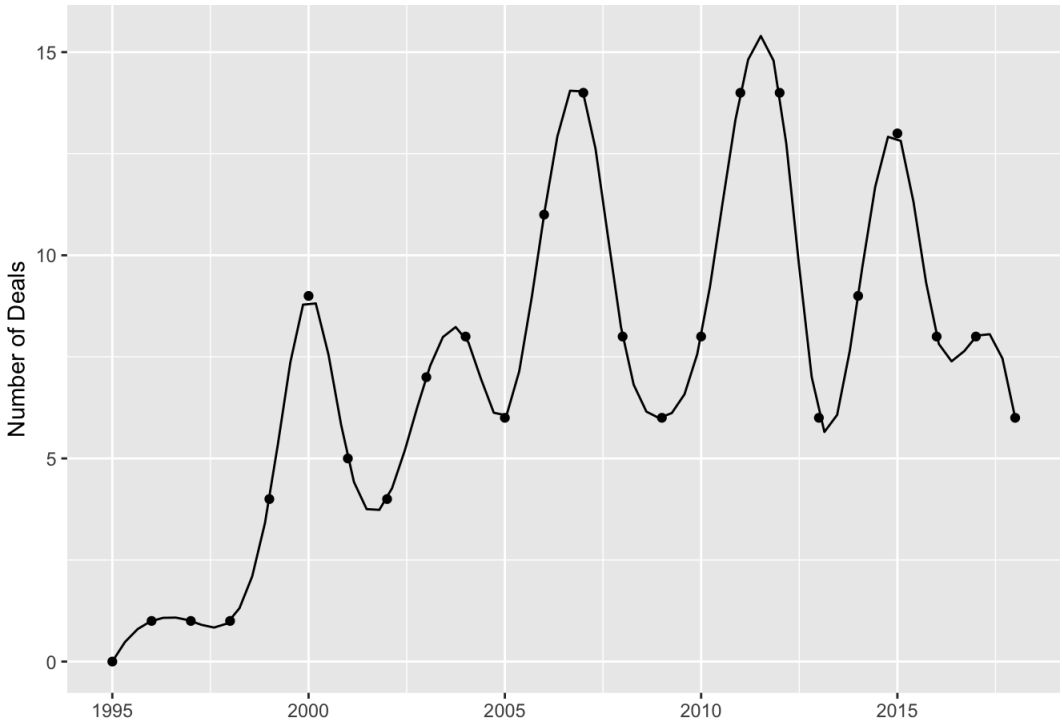

## Plot 3: Volatility of VC Deal Value

```
vol_financing_plot
```



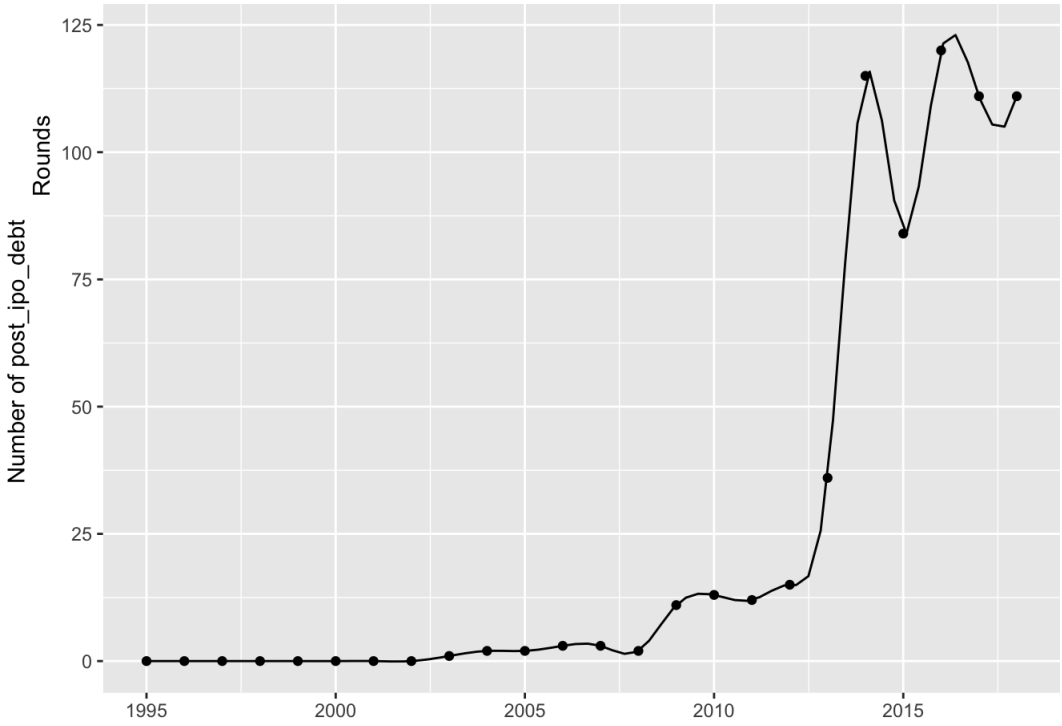

## Plot 4: Investor History

```
get_investor_history <- function(crunchbase, investor_name){
    crunchbase$year <- year(crunchbase$announced_on)
    crunchbase$year <- as.numeric(crunchbase$year)

    crunchbase$raised_amount_usd <- gsub(",","", crunchbase$raised_amount_usd)
    crunchbase$raised_amount_usd <- as.integer(crunchbase$raised_amount_usd)

    #plot years after 1995 since data before is sparse
    cb <- filter(crunchbase, year >= 1995)
    years <- unique(cb$year) %>% sort
    crunchbase$investor_names <- gsub("Lead - ", "", crunchbase$investor_names)
    plot.df <- data.frame(years=years)

    deals <- rep(NA, length(years))
    for (i in 1:length(years)){
      cy <- years[i]
      investor.annual <- filter(crunchbase, year == cy)
      v.names <- strsplit(investor.annual$investor_names, split = ",")  %>% unlist
      num <- sum(v.names == investor_name)
      deals[i] <- num
    }
    plot.df <- data.frame(cbind(plot.df, deals))

    return(plot.df)
}

investor.history <- get_investor_history(crunchbase, "Greylock Partners")

spline.investor <- as.data.frame(spline(investor.history))

investor_plot <- ggplot(investor.history) + geom_point(data =
                                                         investor.history,
                                                       aes(x=years, y=deals))
+ geom_line(data = spline.investor,                                    aes(x=spline.investor$x, y=
spline.investor$y)) + xlab("") + ylab("Number of Deals")

investor_plot
```




# Plot 4: Number of Deals for a Given Round

```
get_round_counts <- function(crunchbase, round_type){
  crunchbase$year <- year(crunchbase$announced_on)
  crunchbase$year <- as.numeric(crunchbase$year)

  #plot years after 1995 since data before is sparse
  cb <- filter(crunchbase, year >= 1995)
  years <- unique(cb$year) %>% sort
  round.df <- data.frame(years = years)

  deals <- rep(NA, length(years))
  for (i in 1:length(years)){
    cy <- years[i]
    round.yr.df <- filter(crunchbase, year == cy, funding_round_type == round_type)
    deal.num <- nrow(round.yr.df)
    deals[i] <- deal.num
  }
  round.df <- cbind(round.df,deals)


  names(round.df) <- c("years", round_type)
  return(round.df)
}


rounds <- get_round_counts(crunchbase, "post_ipo_debt")
deal_type <- names(rounds)[2]
spline.rounds <- as.data.frame(spline(rounds))

round_count_plot <- ggplot(rounds) + geom_point(data = rounds,
                                                aes(x=rounds$years,
                                                y=rounds[[deal_type]])) +
                                    xlab("") + ylab(paste("Number of ",
                                                          deal_type, "
                                                          Rounds", sep =
                                                            "")) +
                                    geom_line(data = spline.rounds,
                                              aes(x=spline.rounds$x,
                                                  y=spline.rounds$y))
round_count_plot
```



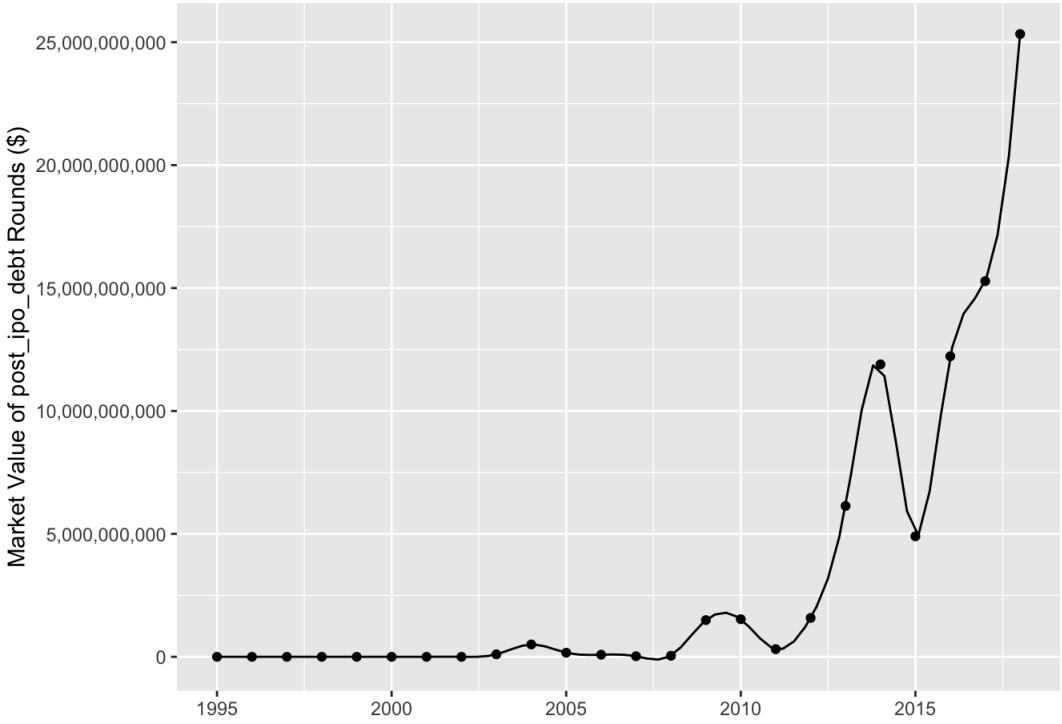

Plot 5: Market Value of Deals for a Given Round

```
get_round_mkt_val <- function(crunchbase, round_type){

  crunchbase$year <- year(crunchbase$announced_on)
  crunchbase$year <- as.numeric(crunchbase$year)
  crunchbase$raised_amount_usd <- gsub(",","", crunchbase$raised_amount_usd)
  crunchbase$raised_amount_usd <- as.integer(crunchbase$raised_amount_usd)

  cb <- filter(crunchbase, raised_amount_usd != "")
  cb <- filter(cb, !is.na(cb$raised_amount_usd))


  #plot years after 1995 since data before is sparse
  cb <- filter(cb, year >= 1995)
  years <- unique(cb$year) %>% sort
  round.df <- data.frame(years = years)

  deal.values <- rep(NA, length(years))
  for (i in 1:length(years)){
    cy <- years[i]
    round.yr.df <- filter(cb, year == cy, funding_round_type == round_type)
    deal.num <- sum(round.yr.df$raised_amount_usd)
    deal.values[i] <- deal.num
  }
  round.df <- cbind(round.df,deal.values)


  names(round.df) <- c("years", round_type)
  return(round.df)
}



round_values <- get_round_mkt_val(crunchbase, "post_ipo_debt")
```

```
## Warning in get_round_mkt_val(crunchbase, "post_ipo_debt"): NAs introduced
## by coercion
```

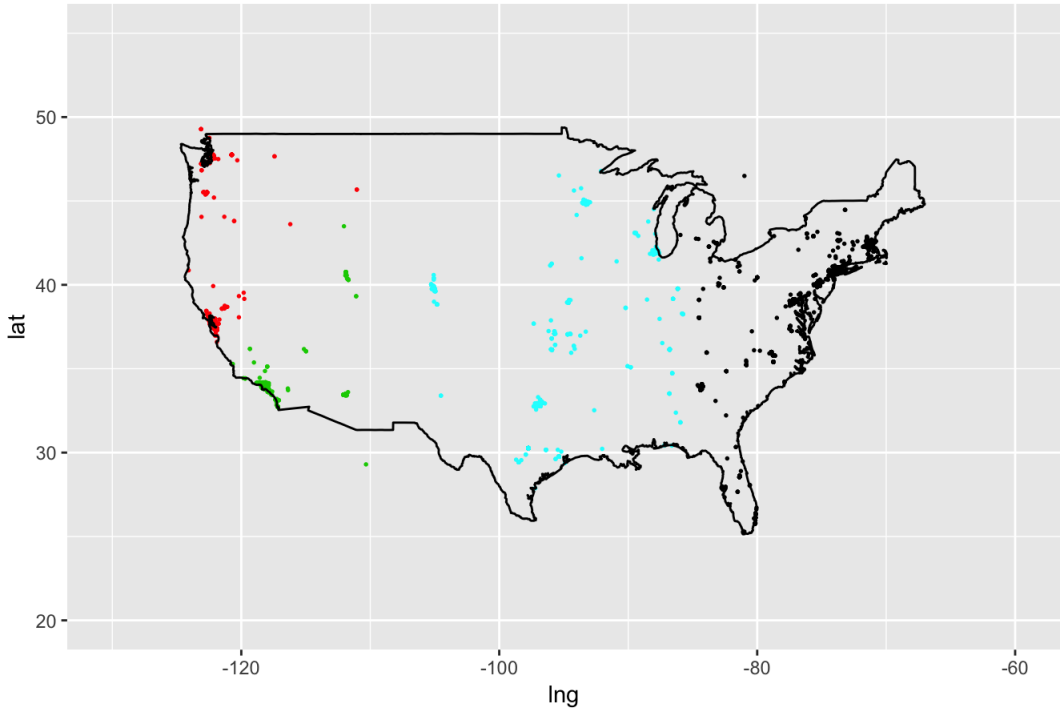
```
## Warning in get_round_mkt_val(crunchbase, "post_ipo_debt"): NAs introduced
## by coercion to integer range
```

```
spline.round.values <- as.data.frame(spline(round_values))

round_mkt_val_plot <- ggplot(round_values) + geom_point(data = round_values,                      aes(x=ro
und_values$years, y=round_values[[2]])) +                          xlab("") +
                     ylab(paste("Market Value of ",                                                names(r
ound_values)[2], " Rounds ($)", sep = "")) +
                     scale_y_continuous(labels = comma) +                              geom_line(data =
spline.round.values, aes(x=spline.round.values$x, y=spline.round.values$y))
round_mkt_val_plot
```




Plot 6: Geographic Clustering of Deals - Geocoding API

```
google_api <- "AIzaSyBo4XcR1L7zGovVabrycKrLpwzLenEIKqY"

get_usa_coordinates <- function(yr, api, df){

  df$year <- year(df$announced_on)
  df$year <- as.numeric(df$year)
  filt.df <- filter(df, year == yr, country_code == "USA")

  #filter out empty entries
  cities <- filt.df$city
  states <- filt.df$state_code
  cities <- cities[cities != ""]
  cities <- gsub(" ", "+", cities)

  lat <- rep(NA, length(cities))
  lng <- rep(NA, length(cities))

  for (i in 1:length(cities)){
    cc <- cities[i]
    cs <- states[i]
    url <- paste("https://maps.googleapis.com/maps/api/geocode/json?address=", cc, "&key=", api, sep = "")
    x <- fromJSON(url)
    latitude <- x$results$geometry$location$lat
    longitude <- x$results$geometry$location$lng
    lat[i] <- latitude
    lng[i] <- longitude
  }

  out <- as.data.frame(cbind(lat,lng))



  return(out)
}

get_kmeans <- function(df, centers, iter){
  kmeans <- kmeans(df, centers = centers, iter.max = iter, algorithm = "Hartigan-Wong")
  return(kmeans)
}

out <- get_usa_coordinates(2000, google_api, crunchbase)

#map for plot
usa <- map_data("usa")

#kmeans clustering
get_kmeans <- function(df, centers, iter){
  kmeans <- kmeans(df, centers = centers, iter.max = iter, algorithm = "Hartigan-Wong")
  return(kmeans)
}
#user selects centroids and max.iter
k <- get_kmeans(out, 5, 100)

f <- ggplot(out) + geom_point(data = out,
                              aes(x=lng,
                                  y=lat),
                              color = k$cluster,
                              size = 0.3) + geom_polygon(data = usa,
                                                         aes(x=long,
                                                             y = lat,
                                                             group = group),
                                                         fill = NA,
                                                         color = "black") +
  coord_fixed(1.3) + xlim(-130,-60) + ylim(20, 55)

f
```




# Plot 7: Word Cloud of Deal Business Areas

```
#word cloud of verticals

vertical_word_cloud <- function(crunchbase, year){
  crunchbase$year <- year(crunchbase$announced_on)
  crunchbase$year <- as.numeric(crunchbase$year)
  cb.year <- filter(crunchbase, year == year)
  cb.words <- strsplit(cb.year$company_category_list, split = ",")
  cb.words <- sort(unlist(cb.words))
  cb.words <- gsub("[[:punct:]]", "", cb.words)

  word.table <- as.data.frame(table(cb.words))
  total <- sum(word.table[2])

  word.table$weight <- word.table[2] / total

  names(word.table) <- c("words", "count")

  word.cloud <- wordcloud(word.table[[1]], word.table[[2]], max.words = 100)
  return(word.cloud)
}


cloud <- vertical_word_cloud(crunchbase, "2017")
```

# Information Technology