Yelp Data Wrangling Project

Rich Dean

22 October 2015

This document covers my initial exploration of the Yelp Dataset Challenge data. This dataset from the online review site Yelp, and is part of a data anlysis competition, now in its fifth run. The data contains (from the Yelp website):

- 1.6M reviews and 500K tips by 366K users for 61K businesses
- 481K business attributes, e.g., hours, parking availability, ambience.
- Social network of 366K users for a total of 2.9M social edges.
- Aggregated check-ins over time for each of the 61K businesses

Initialisation

To start, I'll import my libraries, include files, and set globals variables. After that, I'll load in the data (from cache if available).

```
library(jsonlite)
library(ggmap)
library(ggplot2)
library(GGally)
library(RColorBrewer)
library(tidyr)
library(dplyr)
# Include my standard function library
source(file="~/R_code/common/rrfuncs.R")
# Optionally clear my environment - makes things clearer
# rm(list=ls())
data_dir = c('~/datasets/yelp/')
```

```
if (file.exists(paste0(data_dir,'yelp_data.Rd')))
{
  load(paste0(data_dir,'yelp_data.Rd'))
} else {
  # The data is in a line-by-line JSON format, rather than fully-compliant JSON.
  # Use 'stream_in' from jsonlite
  y.business <- stream_in(file(paste0(data_dir,
                                      "yelp_academic_dataset_business.json")))
  y.checkin <- stream_in(file(paste0(data_dir,
                                     "yelp_academic_dataset_checkin.json")))
  y.review <- stream_in(file(paste0(data_dir,
                                    "yelp_academic_dataset_review.json")))
  y.tip <- stream_in(file(paste0(data_dir,"yelp_academic_dataset_tip.json")))
  y.user <- stream_in(file(paste0(data_dir,"yelp_academic_dataset_user.json")))
  # Cache the R data
  save(list=c('y.business','y.user','y.review','y.tip','y.checkin'),
       file=paste0(data_dir,'yelp_data.Rd'))
}
```

Now that the data is loaded, I'll quickly verify the Yelp statements about the dataset size, and take a quick look at the structure of the **y.business** data frame.

```
data.frame(rbind(
    c(Stat='Reviews',Rows=nrow(y.review)),
    c('Tips', nrow(y.tip)),
    c('Users', nrow(y.user)),
    c('Businesses', nrow(y.business)))
  )
```

```
##         Stat    Rows
## 1    Reviews 1569264
## 2       Tips  495107
## 3      Users  366715
## 4 Businesses   61184
```

```
# Use max.level, as this is a coplex nested frame
str(y.business, max.level=1)
```

```
## 'data.frame':    61184 obs. of  15 variables:
##  $ business_id  : chr  "vcNAWiLM4dR7D2nwwJ7nCA" "UsFtqoBl7naz8AVUBZMjQQ" "cE27W9VPgO88Qxe4ol6y_g" "H
##  $ full_address : chr  "4840 E Indian School Rd\nSte 101\nPhoenix, AZ 85018" "202 McClure St\nDravosb
##  $ hours        :'data.frame':   61184 obs. of  7 variables:
##  $ open         : logi  TRUE TRUE FALSE TRUE TRUE TRUE ...
##  $ categories   :List of 61184
##   .. [list output truncated]
##  $ city         : chr  "Phoenix" "Dravosburg" "Bethel Park" "Pittsburgh" ...
##  $ review_count : int  9 4 5 3 11 15 5 4 3 8 ...
##  $ name         : chr  "Eric Goldberg, MD" "Clancy's Pub" "Cool Springs Golf Center" "Verizon Wireles
##  $ neighborhoods:List of 61184
##   .. [list output truncated]
##  $ longitude    : num  -112 -79.9 -80 -80.1 -79.9 ...
##  $ state        : chr  "AZ" "PA" "PA" "PA" ...
##  $ stars        : num  3.5 3.5 2.5 3.5 4.5 4 1.5 4 2.5 3.5 ...
##  $ latitude     : num  33.5 40.4 40.4 40.4 40.4 ...
##  $ attributes   :'data.frame':   61184 obs. of  38 variables:
##  $ type         : chr  "business" "business" "business" "business" ...
```

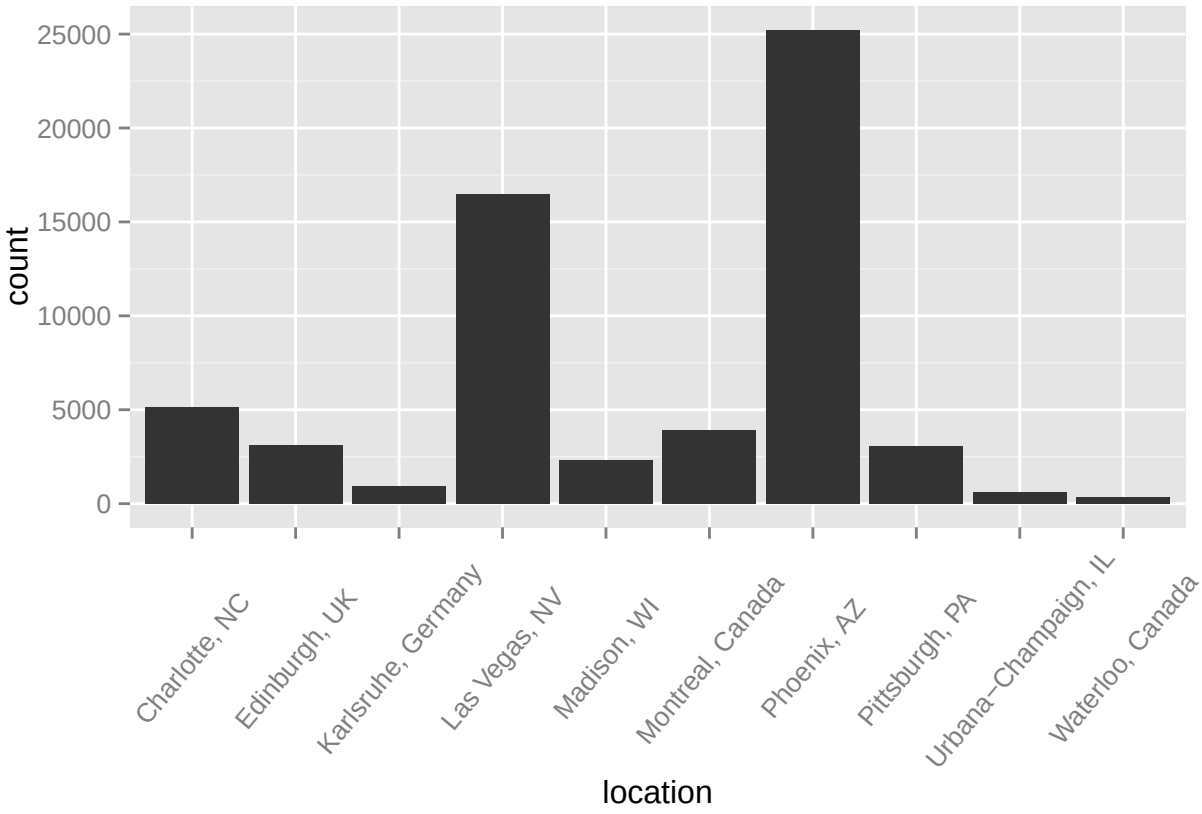
Geography

I know that the businesses are located in 10 distinct areas. I have their names, and the **latitude**/**longitude** are in the **y.business** object. I'll use k-means clustering to label the groups, which will allow for some comparative geo-analysis. I have the list of actual locations which the data should be from, so i will seed the kmeans with the lat/long of those ten locations.

```
cities<-c('Edinburgh, UK', 'Karlsruhe, Germany', 'Montreal, Canada',
          'Waterloo, Canada', 'Pittsburgh, PA', 'Charlotte, NC',
          'Urbana-Champaign, IL', 'Phoenix, AZ', 'Las Vegas, NV', 'Madison, WI')
region.centres<-geocode(cities, source='google')
set.seed(43046721)
myclus<-kmeans(y.business[,c('longitude','latitude')],region.centres)
y.business$location<-cities[myclus$cluster]
table(y.business$location, y.business$state)
```

```
##
##                          AZ    BW    CA   EDH   ELN   FIF   HAM    IL
##   Charlotte, NC           0     0     0     0     0     0     0     0
##   Edinburgh, UK           0     0     0  2971    10     4     1     0
##   Karlsruhe, Germany      0   934     0     0     0     0     0     0
##   Las Vegas, NV           0     0     2     0     0     0     0     0
##   Madison, WI             0     0     0     0     0     0     0     0
##   Montreal, Canada        0     0     0     0     0     0     0     0
##   Phoenix, AZ         25230     0     1     0     0     0     0     0
##   Pittsburgh, PA          0     0     0     0     0     0     0     0
##   Urbana-Champaign, IL    0     0     0     0     0     0     0   627
##   Waterloo, Canada        0     0     0     0     0     0     0     0
##
##                         KHL    MA   MLN    MN    NC   NTH    NV    NW
##   Charlotte, NC           0     0     0     0  4962     0     0     0
##   Edinburgh, UK           1     0   123     0     0     1     0     0
##   Karlsruhe, Germany      0     0     0     0     0     0     0     1
##   Las Vegas, NV           0     0     0     0     1     0 16485     0
##   Madison, WI             0     1     0     1     0     0     0     0
##   Montreal, Canada        0     0     0     0     0     0     0     0
##   Phoenix, AZ             0     0     0     0     0     0     0     0
##   Pittsburgh, PA          0     0     0     0     0     0     0     0
##   Urbana-Champaign, IL    0     0     0     0     0     0     0     0
##   Waterloo, Canada        0     0     0     0     0     0     0     0
##
##                          ON    OR    PA    QC    RP    SC   SCB    WA
##   Charlotte, NC           0     0     0     0     0   189     0     0
##   Edinburgh, UK           0     0     0     0     0     0     3     0
##   Karlsruhe, Germany      0     0     0     0    13     0     0     0
##   Las Vegas, NV           0     1     0     0     0     0     0     1
##   Madison, WI             0     0     0     0     0     0     0     0
##   Montreal, Canada        0     0     0  3921     0     0     0     0
##   Phoenix, AZ             0     0     0     0     0     0     0     0
##   Pittsburgh, PA          0     0  3041     0     0     0     0     0
##   Urbana-Champaign, IL    0     0     0     0     0     0     0     0
##   Waterloo, Canada      351     0     0     0     0     0     0     0
##
##                          WI   XGL
##   Charlotte, NC           0     0
##   Edinburgh, UK           0     1
##   Karlsruhe, Germany      0     0
##   Las Vegas, NV           0     0
##   Madison, WI          2307     0
##   Montreal, Canada        0     0
##   Phoenix, AZ             0     0
##   Pittsburgh, PA          0     0
##   Urbana-Champaign, IL    0     0
##   Waterloo, Canada        0     0
```

```
ggplot(data=y.business, aes(x=location)) +
  geom_histogram() +
  theme(axis.text.x=element_text(angle=50, size=10, vjust=0.5))
```



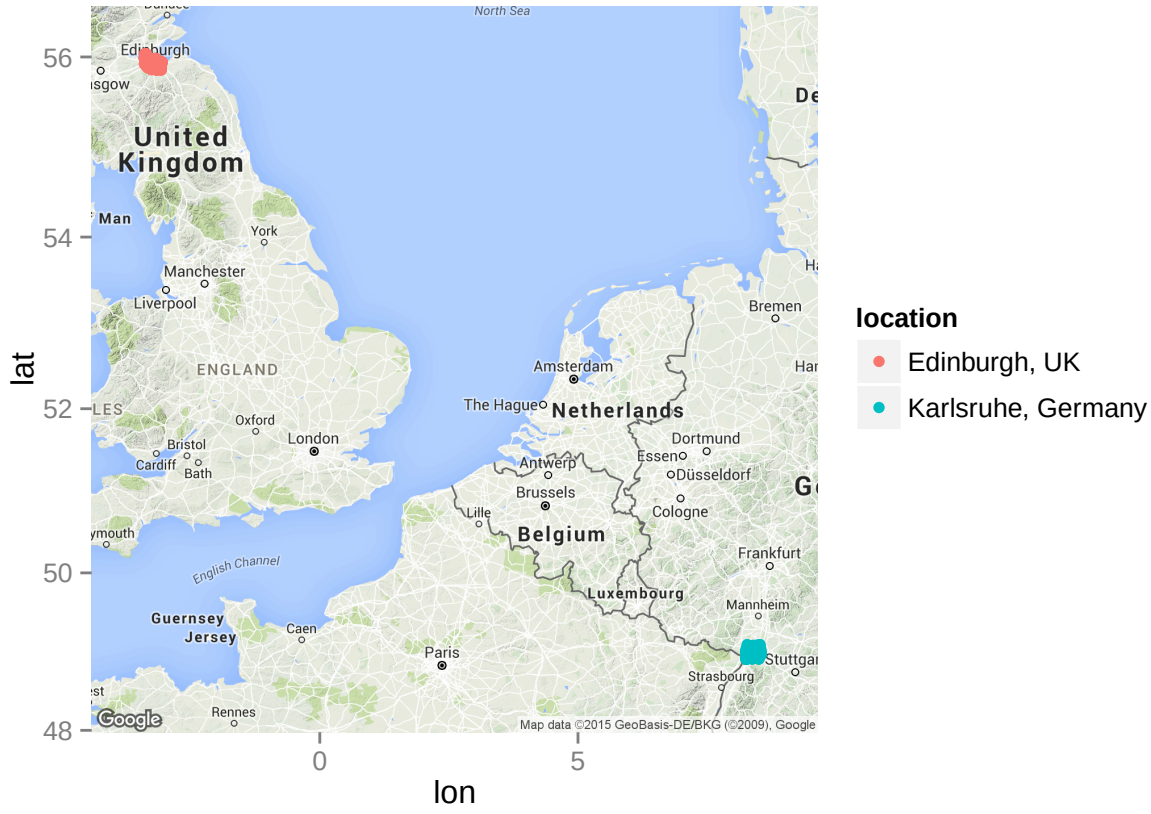
These results show a good grouping, where each state is entirely represented by one cluster, with the exception of two. Cluster 8 and 9 both contain CA - reasonable as both AZ and NV border it. Cluster 9 seems to contain a rogue SC entry:

```
subset(y.business, state=='NC' & location=='Las Vegas, NV')$full_address
```

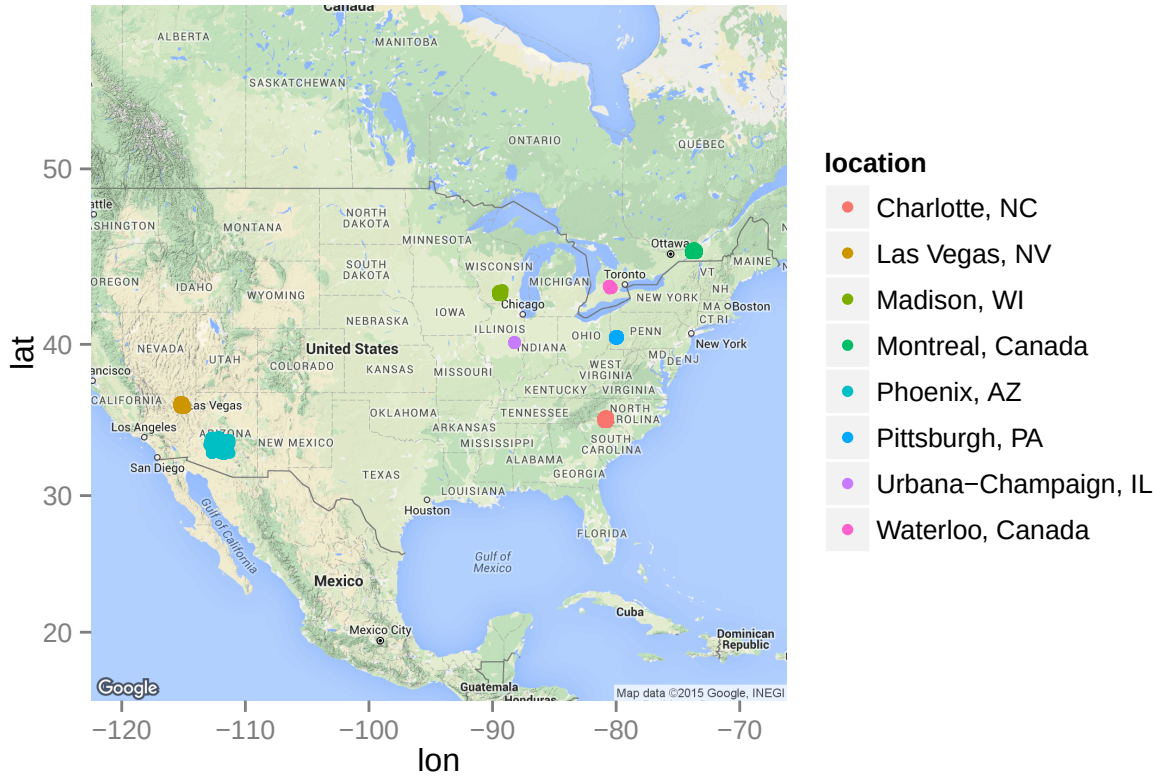
```
## [1] "3890 Blue Diamond Rd\nSuite C\nSouthwest\nLas Vegas, NC 89139"
```

It's a typo! Yelp must take the **state** value from the entered address. I'll do a pair of map plots - one for the US, and one for Europe - to show the distribution of the business, coloured by the cluster.

```
europe.centre <- colMeans(apply(myclus$centers[cities %in%
                            c('Edinburgh, UK','Karlsruhe, Germany'),],2,range))
map.europe <- get_map(location = europe.centre, zoom = 6)
map1<-ggmap(map.europe) +
    geom_point(aes(x=longitude, y=latitude, color=location),
               data=subset(y.business,
                           location %in% c('Edinburgh, UK','Karlsruhe, Germany')))

usa.centre <- colMeans(apply(myclus$centers[!cities %in%
                            c('Edinburgh, UK','Karlsruhe, Germany'),],2,range))
map.usa <- get_map(location = usa.centre, zoom = 4)
map2<-ggmap(map.usa) +
    geom_point(aes(x=longitude, y=latitude, color=location),
               data=subset(y.business,
                           !location %in% c('Edinburgh, UK','Karlsruhe, Germany')))

map1
```


North Sea
Edinburgh
United Kingdom
Man
York
Manchester
Liverpool
ENGLAND
Oxford
London
Bristol
Cardiff
Bath
English Channel
Guernsey
Jersey
Caen
Paris
Rennes
Amsterdam
The Hague
Netherlands
Antwerp
Brussels
Lille
Belgium
Dortmund
Essen
Düsseldorf
Cologne
Bremen
Frankfurt
Luxembourg
Mannheim
Strasbourg
Google
Map data ©2015 GeoBasis-DE/BKG (©2009), Google
56
54
52
50
48
0
5
lat
lon
location
Edinburgh, UK
Karlsruhe, Germany

```
map2
```


ALBERTA
MANITOBA
SASKATCHEWAN
ONTARIO
QUEBEC
MONTANA
NORTH DAKOTA
MINNESOTA
SOUTH DAKOTA
WISCONSIN
MICHIGAN
IDAHO
WYOMING
NEBRASKA
IOWA
ILLINOIS
NEVADA
UTAH
COLORADO
KANSAS
MISSOURI
INDIANA
OHIO
PENN
United States
CALIFORNIA
Las Vegas
Los Angeles
San Diego
ARIZONA
NEW MEXICO
OKLAHOMA
TEXAS
ARKANSAS
LOUISIANA
MISSISSIPPI
TENNESSEE
KENTUCKY
WEST VIRGINIA
VIRGINIA
NORTH CAROLINA
SOUTH CAROLINA
ALABAMA
GEORGIA
FLORIDA
Houston
Gulf of California
Gulf of Mexico
Mexico
Mexico City
Cuba
Dominican Republic
Guatemala
Ottawa
Toronto
Chicago
New York
Boston
MAINE
VT
NH
MA
CT
RI
NJ
DE
MD
Google
Map data ©2015 Google, INEGI
50
40
30
20
−120
−110
−100
−90
−80
−70
lat
lon
location
Charlotte, NC
Las Vegas, NV
Madison, WI
Montreal, Canada
Phoenix, AZ
Pittsburgh, PA
Urbana−Champaign, IL
Waterloo, Canada

The maps show that the clustering worked nicely, and also shows the geographic spread of the businesses in the dataset. Despite the scale of the USA map, you can see something of the dispersal of businesses within each cluster - the Phoenix AZ cluster is the largest of the groups on the map, while the Urbana-Champaign IL group is smallest.

I'll add a distance to the cluster center too - I can then do some analysis of the area covered by each cluster This is only for a comparative measure, and the distances /should/ be relatively short, so i won't go so far as a great circle calculation

```
t.distance<-y.business[,c('longitude','latitude')]-myclus$centers[
                                                        myclus$cluster,]
y.business$distance.from.centre<-sqrt(rowSums(t.distance**2))
tbl_df(y.business[,c('location','distance.from.centre')]) %>%
  group_by(location) %>%
  summarise(mn=mean(distance.from.centre), sd=sd(distance.from.centre))
```

```
## Source: local data frame [10 x 3]
##
##                location         mn         sd
##                   (chr)      (dbl)      (dbl)
## 1         Charlotte, NC 0.09622936 0.05922118
## 2         Edinburgh, UK 0.02252761 0.02907276
## 3    Karlsruhe, Germany 0.03526781 0.03735799
## 4         Las Vegas, NV 0.08950494 0.05578469
## 5           Madison, WI 0.07210386 0.04991131
## 6      Montreal, Canada 0.05166342 0.05228179
## 7           Phoenix, AZ 0.19152956 0.12064225
## 8        Pittsburgh, PA 0.04853231 0.03025525
## 9  Urbana-Champaign, IL 0.01798966 0.01288208
## 10     Waterloo, Canada 0.04142594 0.03097660
```

This shows that Urbana-Champaign has the tightest group of businesses (low mean and standard deviation), while Phoenix has the most dispersed, as seen on the map plots.

********** TODO MORE GEO STUFF!!!!! Maybe a multivariate plot of some kind? Deeper analysis of a single city?

Business Categories

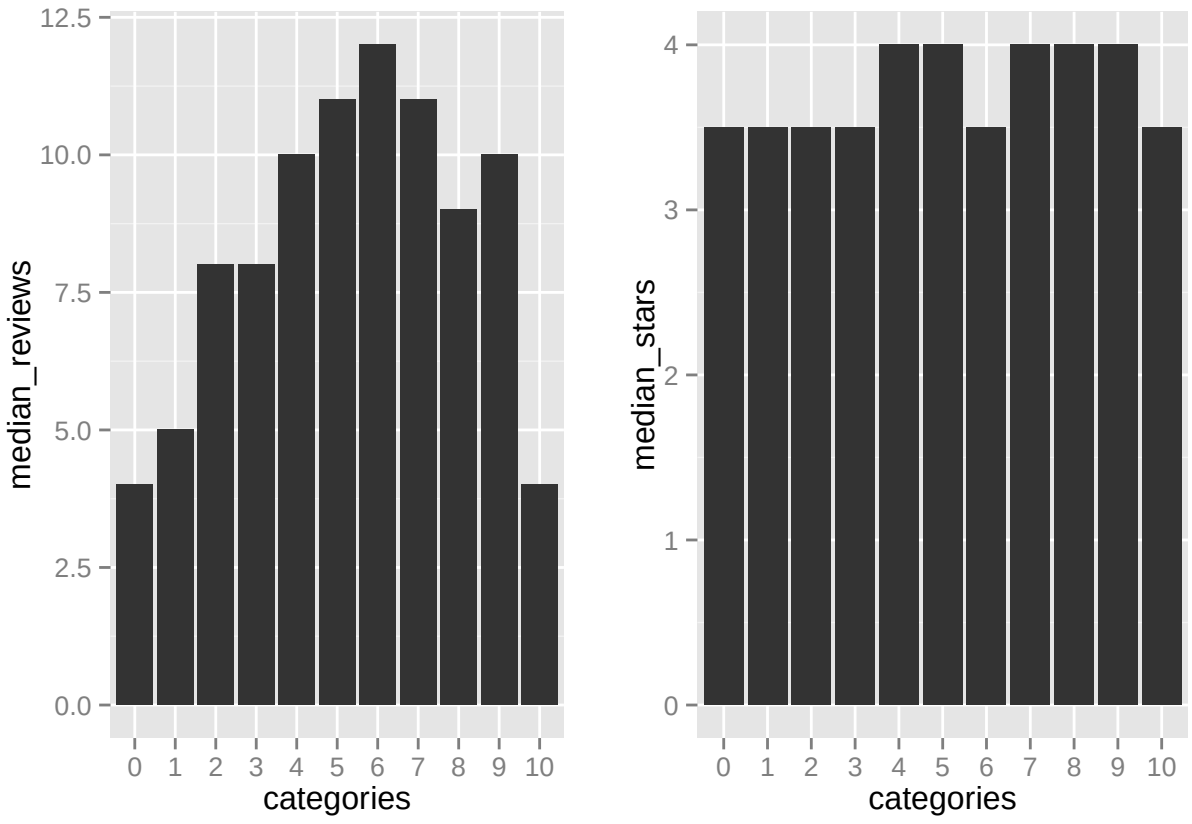
I'd also like to get a primary category for each business - this will also simplify analysis. The primary category list is taken from https://www.yelp.com/developers/documentation/v2/all_category_list.

```
yelp.primary.categories<-
  c('Active Life','Arts & Entertainment','Automotive','Beauty & Spas',
    'Education','Event Planning & Services','Financial Services','Food',
    'Health & Medical','Home Services','Hotels & Travel','Local Flavor',
    'Local Services','Mass Media','Nightlife','Pets','Professional Services',
    'Public Services & Government','Real Estate','Religious Organizations',
    'Restaurants','Shopping')

y.business$primary.categories<-
  lapply(y.business$categories,
         function (x) sort(intersect(x, yelp.primary.categories)))
length(unique(y.business$primary.categories))
```

```
## [1] 325
```

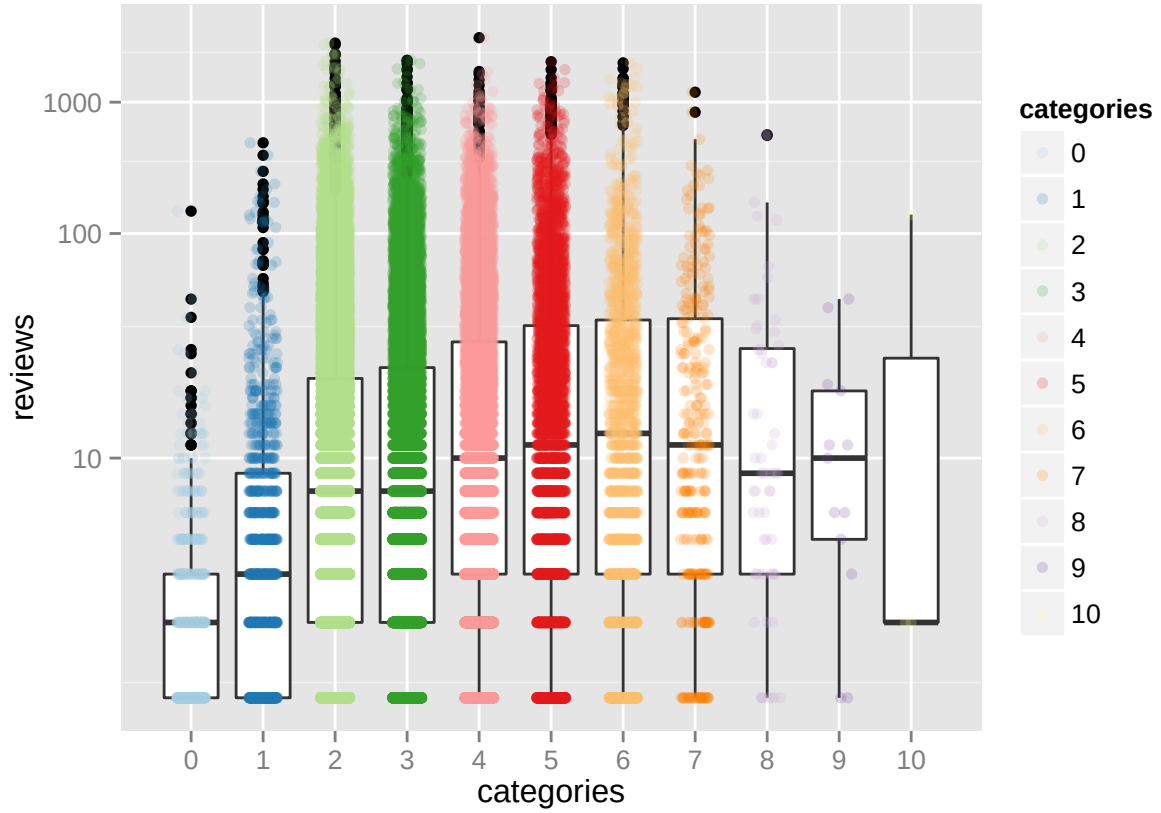
There are 325 combinations of primary categories! This makes any specific comparison of primary categories tricky, as businesses could belong to several primary categories. Taking an slightly different tack led me to think more deeply about the categories and review counts. Do businesses with more categories have more reviews and/or higher ratings?

```
cat.dist<-
  data.frame(categories=
               as.factor(
                 sapply(y.business$categories,
                        function(x) length(unlist(x)))),
             location=y.business$location,
             reviews=y.business$review_count,
             stars=y.business$stars)
# Build a second variable to hold summaries
cat.dist2<- cat.dist %>%
            group_by(categories) %>%
            summarise(median_reviews=median(reviews),
                      median_stars=median(stars),
                      n=n())
rr.multiplot(
  ggplot(data=cat.dist2, aes(x=categories)) +
    geom_histogram(aes(y=median_reviews), stat='identity'),
  ggplot(data=cat.dist2, aes(x=categories)) +
    geom_histogram(aes(y=median_stars), stat='identity'),
  cols=2)
```



There isn't a clear link between the ratings and number of categories, but there appears to be a clear link between categories and number of reviews

```
ggplot(data=cat.dist, aes(x=categories, y=reviews)) +
  coord_trans(ytrans='log10') +
  geom_boxplot() +
  geom_jitter(alpha=0.2, aes(color=categories),
              position=position_jitter(width=.2))+
  scale_colour_brewer('categories', palette='Paired') +
  scale_y_log10()
```



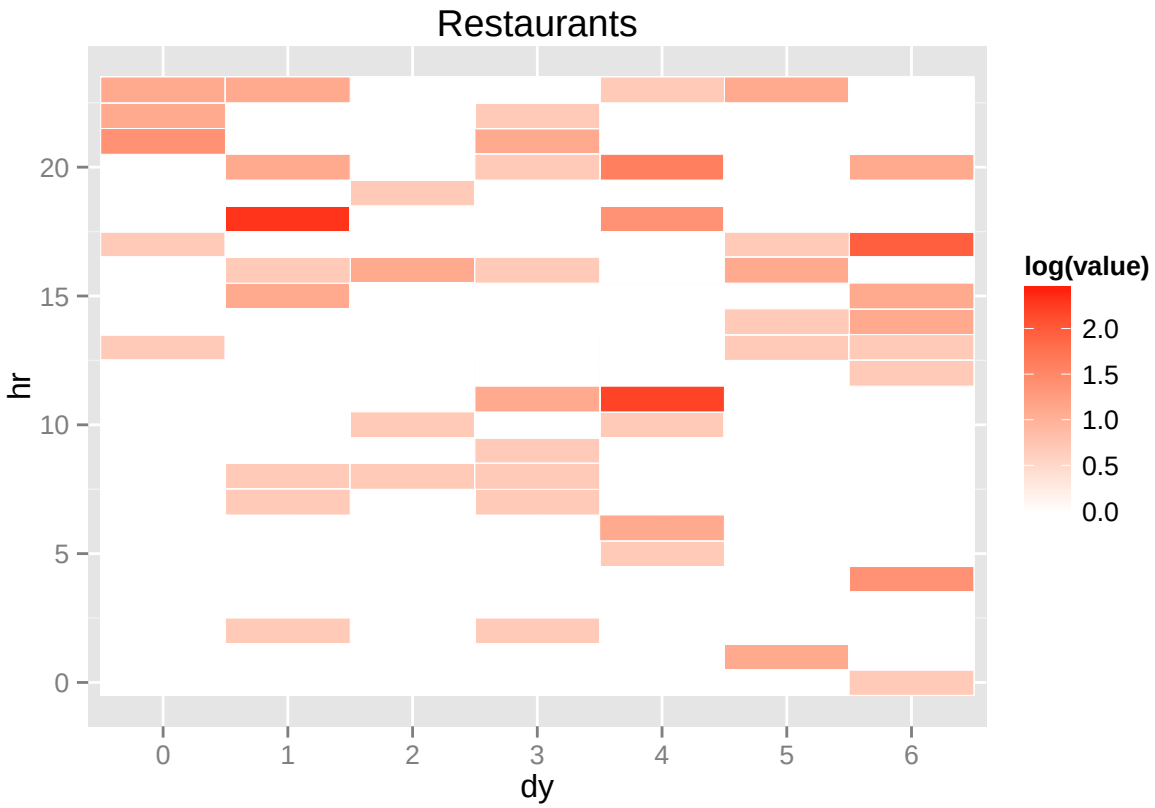
Using a log plot of reviews shows there are actually few businesses with very high numbers of reviews. The point density shows the bulk of businesses have between 2 and 5 categories, and less than 100 reviews

Checkins

I had hoped to perform more category-based analysis, but with multiple primary categories, I would have to sift the combinations and derive so kind of hierarchy to determine a single primary for each business. Instead, I chose to look into a different view of the data - checkins. This shows when businesses are visited, by week-hour. The **y.checkin** data frame contains my core data - I'll add some of the columns from **y.business** to provide a more useful single frame. I will use heatmaps to visualise this data.

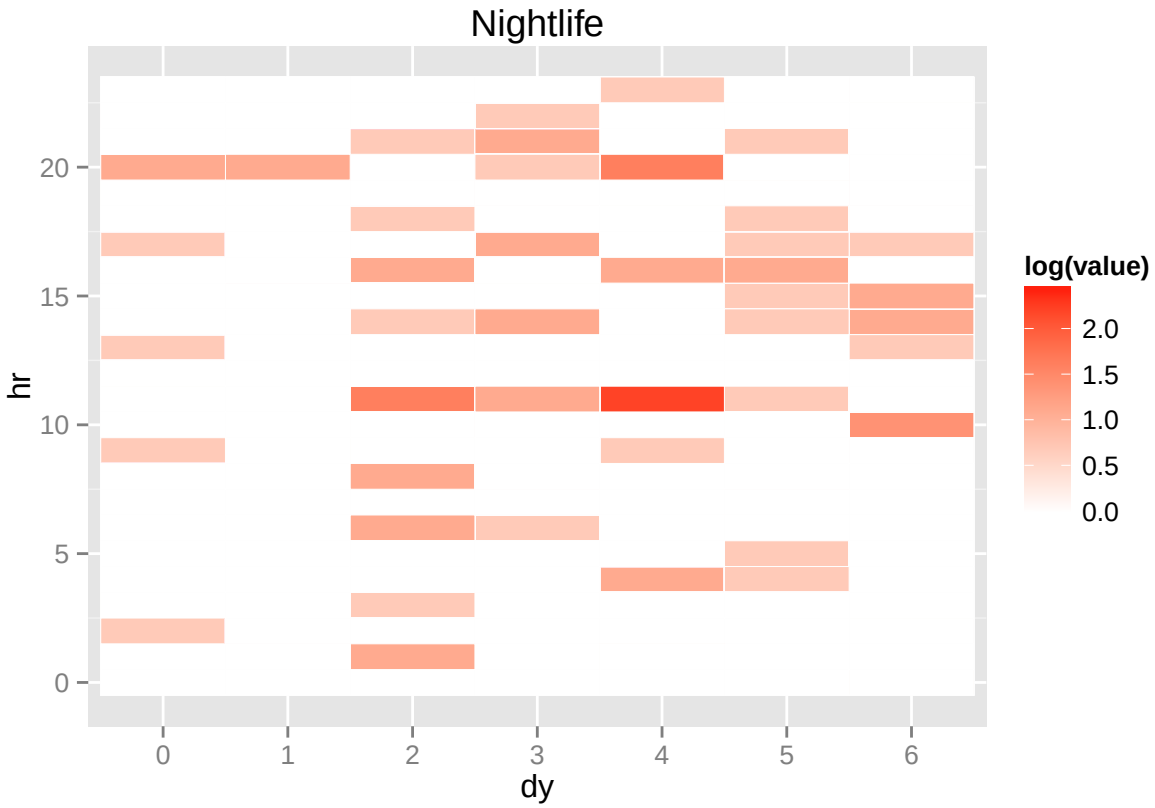
```
y.checkin<-data.frame(flatten(y.checkin))
y.checkin<-
  y.checkin %>%
  left_join(y.business[,c('business_id','location','review_count',
                          'stars', 'primary.categories')],
```

```
            by = 'business_id')
# Need to gather the checkin_info columns
# I'll split the check-in column names to give hour/day values
y.checkin.long <-
  y.checkin %>%
  gather(key, value, -location, -stars, -review_count, -type, -business_id,
         -primary.categories, na.rm=TRUE) %>%
  mutate(checkin.list= strsplit(as.character(key), '\\.')) %>%
  mutate(hr = as.integer(sapply(checkin.list, function (x) x[2])),
         dy = sapply(checkin.list, function (x) x[3])) %>%
  select(-key, -checkin.list, -type, business_id)

# Break the primary category list into columns
# This code from http://stackoverflow.com/questions/25347739/r-convert- \\
#                     factor-column-to-multiple-boolean-columns
lvl <- unique(unlist(y.checkin.long$primary.categories))
res <- data.frame(do.call(rbind,lapply(y.checkin.long$primary.categories,
                                       function(x) table(factor(x, levels=lvl)))
                          ),
                  stringsAsFactors=FALSE)
y.checkin.long<-cbind(y.checkin.long, res)


#  Lets look at restaurants, nightlife, and religious organisations
rr.heatmap <- function (l.criteria, loc_wrap = FALSE)
{
  p<-ggplot(subset(y.checkin.long, eval(parse(text=l.criteria))), aes(dy, hr)) +
    geom_tile(aes(fill = log(value)), colour = "white") +
    scale_fill_gradient(low = "white", high = "red")
  p + if (loc_wrap==TRUE) facet_wrap( ~ location ) else NULL
}

rr.heatmap('Restaurants>0&value<quantile(value,0.95)')+ggtitle('Restaurants')
```



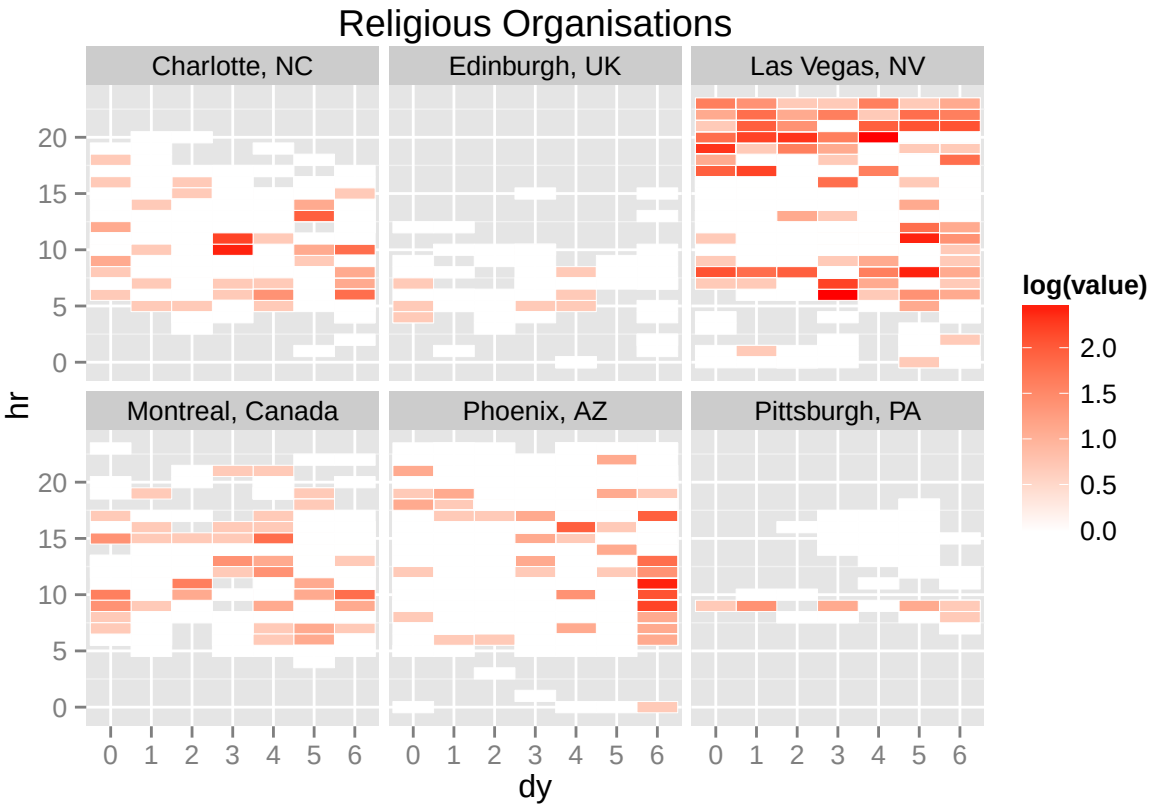
I would have expected to see a pattern in restaurant checkins - more frequent at lunchtimes and evenings - but there are no clear patterns to be seen in this plot. The *nightlife* category should clearly show a pattern...

```
rr.heatmap('Nightlife>0&value<quantile(value,0.95)')+ggtitle('Nightlife')
```



This very puzzling. The checkins seem to be widely scattered. A final look at these, taking Religious Organisations and splitting by location...

```
rr.heatmap('Religious.Organizations>0&value<quantile(value,0.95)', TRUE)+
  ggtitle('Religious Organisations')
```

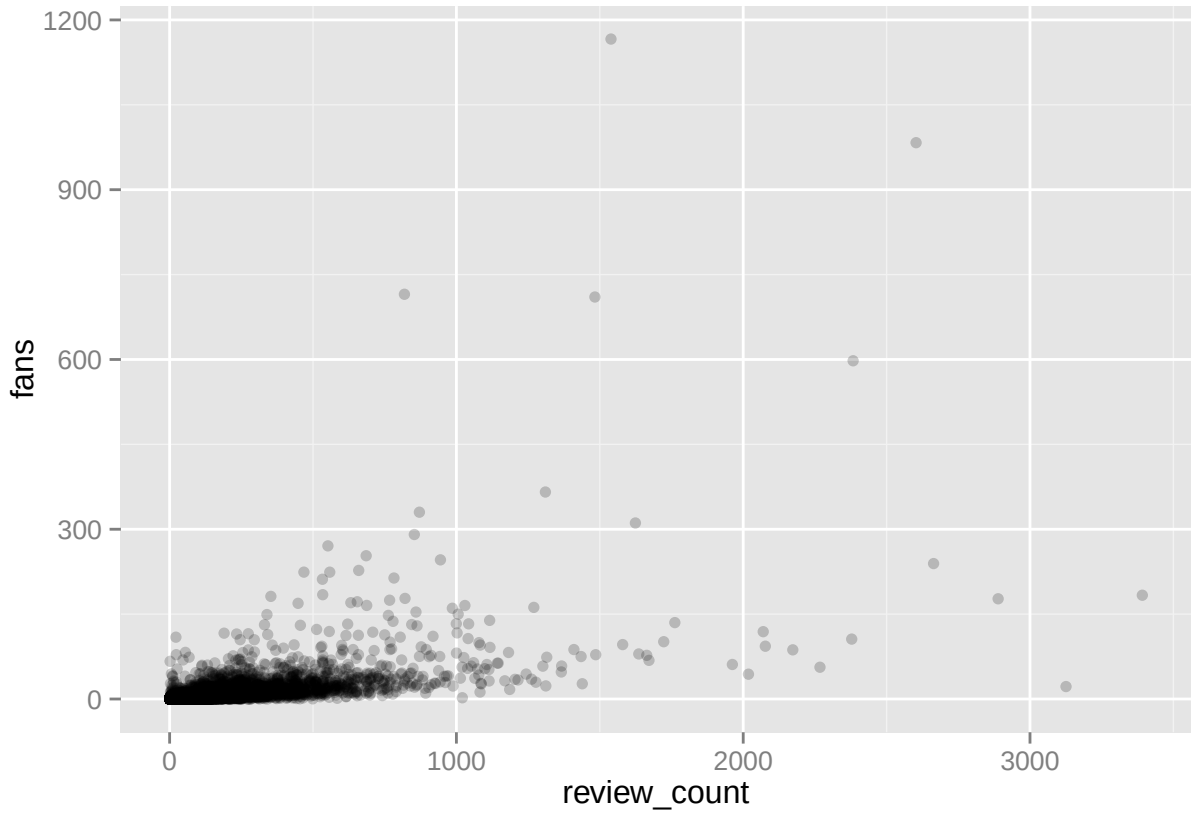


This plot does show a few patterns. Phoenix has a clear grouping of checkins on day 6, around 10am - morning on a Sunday seems a sensible time for visitng church. The Las Vegas distribution is clearly weighted towards the evening, and Saturday morning and lunchtime. My theory here would be that the checkins are mainly at wedding venues. Sadly the dataset doesn't contain business names, so this would be difficult to investigate further. It would appear that checkins are not always associated with actual presence at the business in question, which makes the data of limited real use.

Users

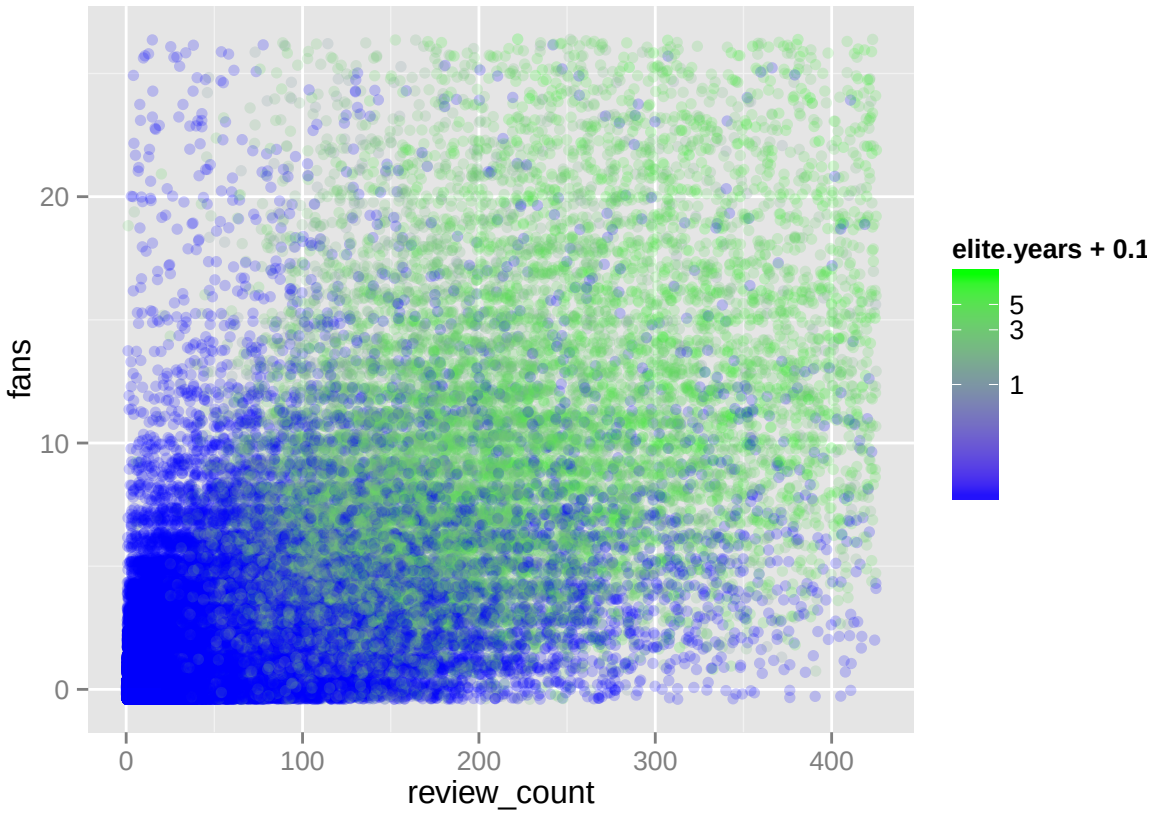
********** TODO Expand on the text Initial look at the user data

```
y.user$elite.years<-unlist(lapply(y.user$elite, length))

# Sample 50k rows to increase speed
ggplot(data=sample_n(y.user,50000), aes(x=review_count, y=fans)) +
  geom_point( alpha=0.2, position='jitter')
```



Improve - remove extreme outliers, log the scales, add the elite.years as a colour

```
ggplot(data=
       subset(y.user,
              review_count<quantile(review_count,0.99)
              & fans<quantile(fans,0.99)
              ),
       aes(x=review_count, y=fans)
       ) +
  geom_point(alpha=0.2,
             position='jitter',
             aes(color=elite.years+0.1)
             )+
  scale_colour_gradientn(
             colours=c('blue','green'),
             trans='log',
             breaks=c(1,3,5,10,15))
```



****** TODO Correlation between review count and fans ****** TODO Further user analysis - 2-3 more vis

Final Plots and Summary

****** TODO Select my 3 plots, make them perfect. Suggest: ****** TODO USA map, heat map of religion in Phoenix, one other TBD

Reflections

******* TODO 200 words Large multi-layered dataset. Primarily categorical/ text based. Well suited for a NLP/prediction project.