Relationship Between Number of Reviews and Amount of Travel From Yelp Data set

Jacob Townson

November 15, 2015

Introduction

I thought it would be interesting to try to find out if there was a correlation between the number of reviews a person had made on Yelp with the amount they have traveled. So my question was: do more frequent reviewers travel more than non-frequent reviewers? I found the question to be interesting because it raised my curiosity of whether or not these people travel purely for the sake of making reviews on certain organizations. So my hypothesis was that yes, there would be a correlation and frequent reviewers on Yelp would probably travel more; and their reviews would be more spread out based on city, longitude, and latitude.

Methods and Data

To begin, the data had to be read into R. To download the data, simply go to this link: data. This data needs to be put in the working directory, then read into the global environment in R. Next, a massive amount of data cleaning had to be done. I knew I needed to separate the data by user, the number of cities traveled, and also distance based on latitude and longitude. In order to do this, I ran the (rather large) code chunk seen in **Figure 1** . The final product of this code is the user_travel data frame (warning: those wanting to recreate the project, this takes a long time to run). This data frame shows us each user, how many cities they traveled to, and a the radius at which they traveled based on latitude and longitude.

Here I will explain step by step what exactly is happening in **Figure 1**. To make the full_data data frame, I organized the collection of all of the businesses traveled to based on user and city. Then in travel, I narrowed the data frame to only contain things necessary to show where the users traveled to. Then the final for loop in the chunk shows how the main data frame was made. It was done by taking each user, counting how many cites they visited, then using the distance formula on their largest range of area traveled based on latitude and longitude. I realize that the distance formula here isn't exact because it doesn't account for the curve of the Earth, but I figured since we were only dealing with the US, it wouldn't affect the calculation too much.

Now that we have our data, we can make the model to see how the variables relate. We can make an exploratory model to see how the variables relate to each other. To do this, we will simply use the lm function in R.

```
city.predict <- lm(cities~reviews, data = user_travel)
distance.predict <- lm(distance~reviews, data = user_travel)
```

Then we can view the residual plots for these models in order to see if they fit as a linear regression model. These plots can be found in **Figure 2**. As can be seen, in the city case the data does not seem to fit linearly. The distance case is similar. In fact, they looks much more quadratic when plotting the reviews vs the cities traveled to and distance. So I figured I would add in the squared and cubed values to the data to try and fit it more appropriately, then use the stepAIC function to find which model is the most appropriate for this situation (**Figure 3**).

Results

Now that we have run this code and found the best model for the relationship between cities and reviews, and distance and reviews, let's see the P values of the variables as chosen by the stepAIC function. The following tables are of the city regression model and the distance regression model respectively.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.3152	0.0018	725.14	0.0000
reviews	0.0034	0.0000	115.66	0.0000
reviewssq	-0.0000	0.0000	-41.67	0.0000
reviewscu	0.0000	0.0000	27.69	0.0000

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.1210	0.0031	38.85	0.0000
reviews	0.0076	0.0001	149.40	0.0000
reviewssq	-0.0000	0.0000	-46.15	0.0000
reviewscu	0.0000	0.0000	28.35	0.0000

As we can see, these P values are surprisingly small. Also, it is important to note that the stepAIC function included all of the review variables (normal, squared, and cubed) in the model. To see how well this worked, let's look at the residuals one more time. These can be seen in **Figure 4**. As we can see in these plots, the residuals look extremely similar to their initial model counterparts. In fact, they almost look identical other than the fact that the new models seem to have a bit smaller of a window. Based on the extremely small P values though, I believe it is time to move on to discussion.

Discussion

After using the stepAIC function and also adding in the squared and cubed values for the number of reviews per user, it can be concluded that reviews and the amount of travel are not correlated. Even though the P values are small, implying that the values are heavily correlated, the residuals seem to show otherwise. The line for the residuals vs the fitted seem to be all right based purely on the line, but the data is so scattered it would lead us to believe differently. Then the Q-Q plots both seem to imply a quadratic relationship, but then when adding in squared and cubed values, not much of a difference seems to be made. It almost seems as though there may be no correlation at all, and we were simply lucky to get the results that we did.

In retrospect, I wish I had more time to do the project. If I had more time, I would add other variables to the model to see if maybe they had more to do with the amount of travel a user makes rather than purely the reviews. I think this would make for interesting future work.

Appendix

Figure 1

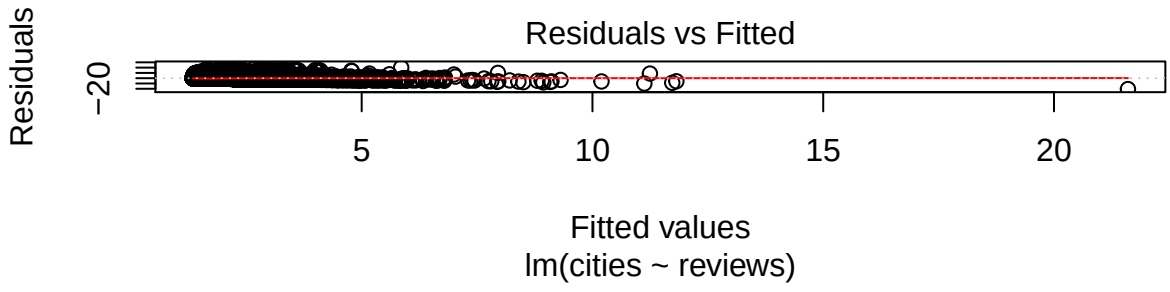
```
users <- user_data[user_data$review_count != 0,] #remove users with 0 reviews
votes <- users$votes$funny + users$votes$useful + users$votes$cool
users <- select(users, review_count, user_id, fans, average_stars)
users <- mutate(users, votes = votes)
```

```r
businesses <- data.frame(business_data$business_id, business_data$city, business_data$latitude, business
colnames(businesses) <- c("business_id", "city", "latitude", "longitude")

reviews <- data.frame(review_data$user_id, review_data$review_id, review_data$business_id)
colnames(reviews) <- c("user_id", "review_id", "business_id")

full_data <- left_join(reviews, businesses, by = "business_id")
full_data <- left_join(full_data, users, by = "user_id")
full_data <- full_data[complete.cases(full_data),] #get rid of NA's

rm(business_data, businesses, review_data, reviews, user_data) #cleaning up environment
travel <- full_data[,-c(2,3,7,8,9,10), drop = FALSE]
travel <- travel %>% group_by(user_id, city) %>% mutate(revCount = n())
travel2 <- travel %>% filter(row_number() == 1)

user <- c(0)
cities <- c(0)
distance <- c(0)
reviews <- c(0)
user_travel <- data.frame(user,cities,distance, reviews)
for(i in (1:length(users$user_id))){
    total_cities <- travel2[travel2$user_id == users$user_id[i],]
    user <- users$user_id[i]
    cities <- length(total_cities$city)
    distance <- sqrt(((max(total_cities$latitude)-min(total_cities$latitude))^2)+
      (max(total_cities$latitude)-min(total_cities$latitude))^2)
    reviews <- users$review_count[i]
    temp <- data.frame(user,cities,distance, reviews)
    user_travel <- rbind(user_travel, temp)
}
user_travel = user_travel[-1,]
```

Figure 2

City Data

```r
plot(city.predict, which=1:2, labels.id = '')
```



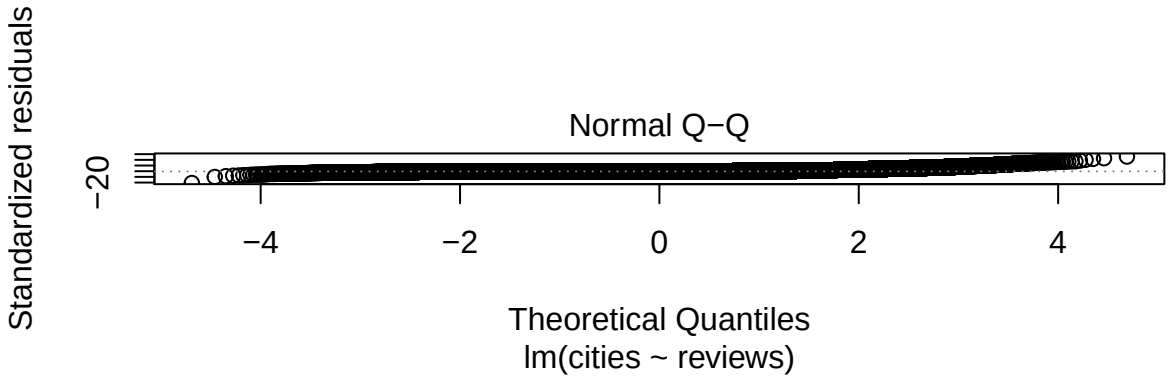


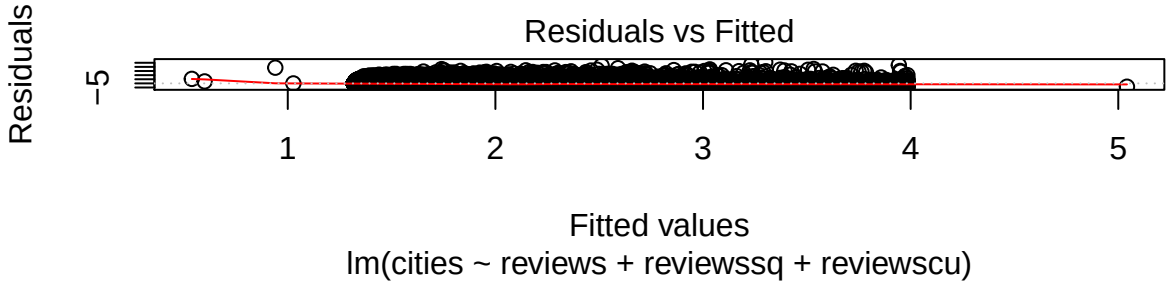
Figure 3

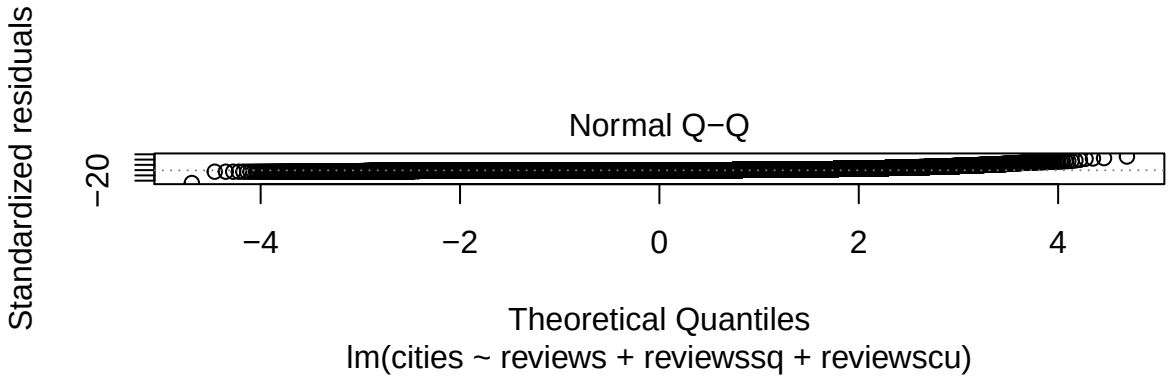
```
reviewssq <- (user_travel$reviews)^2
reviewscu <- (user_travel$reviews)^3
user_travel <- cbind(user_travel, reviewssq, reviewscu)

city.predict2 <- lm(cities~reviews+reviewssq+reviewscu, data = user_travel)
distance.predict2 <- lm(distance~reviews+reviewssq+reviewscu, data = user_travel)

city.predict2 <- (stepAIC(city.predict2, scope = list(lower = ~reviews)))
distance.predict2 <- (stepAIC(distance.predict2, scope = list(lower = ~reviews)))
```

Figure 4

City Data

```
plot(city.predict2, which=1:2, labels.id = '')
```





Distance Data

```
plot(distance.predict2, which=1:2, labels.id = '')
```