

# AC209a Data Science Project: Data Science with User Ratings and Reviews

Andrew Ross, Sophie Hilgard, Reiko Nishihara, Nick Hoernle

November 4, 2016

## Data Source

We downloaded the data from the ‘Yelp Dataset Challenge’ ([https://www.yelp.com/dataset\\_challenge](https://www.yelp.com/dataset_challenge)). The data contains in total 2.7M reviews from 687K users for 86K businesses. Business data consist of 15 features including ID, category of business (e.g., fast food, restaurant, nightlife, etc.), city, full address, operation hours, latitude, longitude, review count, and stars earned. User data consist of 11 features including average stars, compliments, elite, number of fans, IDs of his/her friends, name, review count, vote categories, and the month start a yelp review.

## Data Exploration

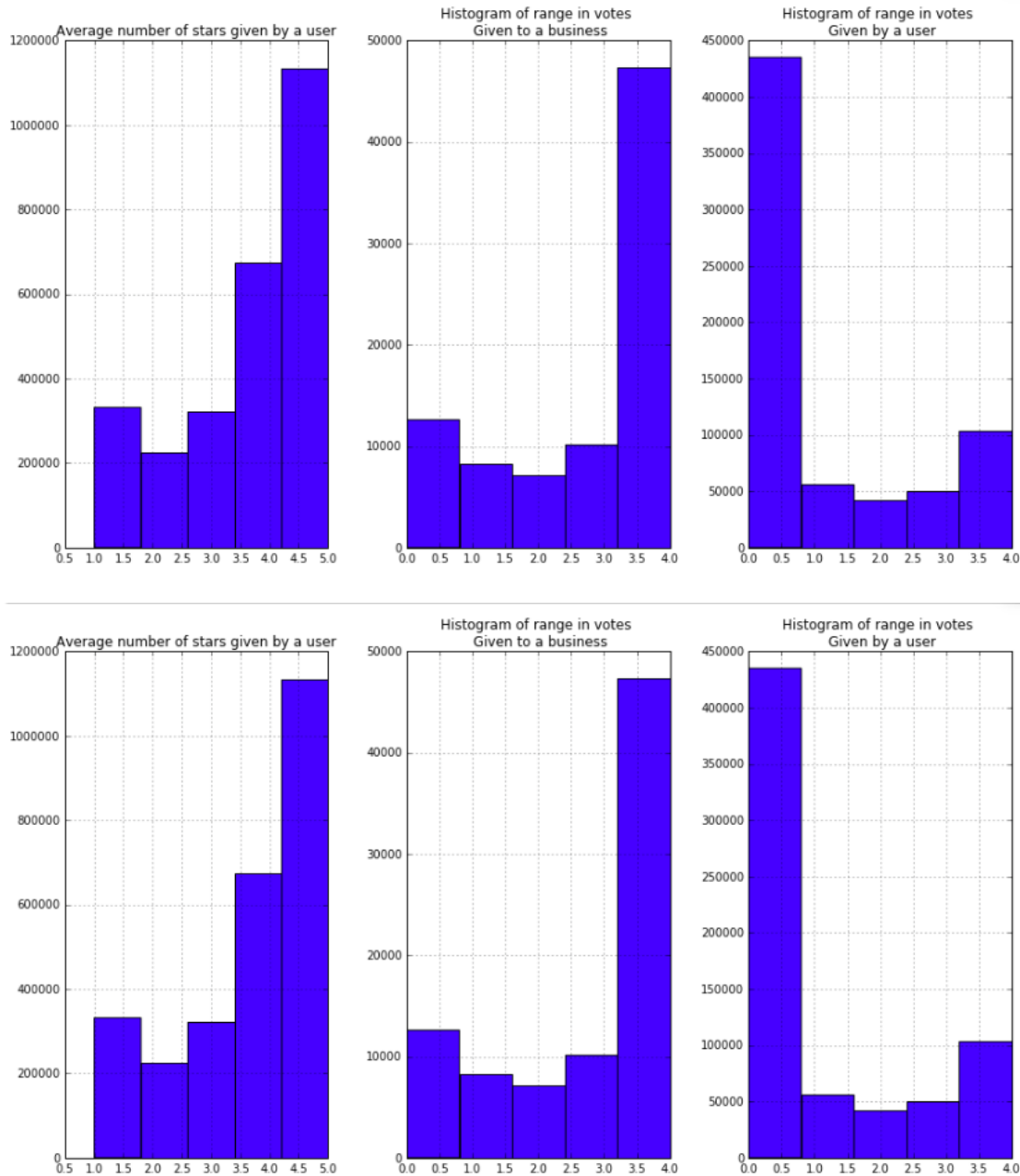
We here show descriptive summary statistics about the data and results from univariate analyses.

The table of the users data shows that median value of stars was 3.9 (inter-quartile range, IQR, was 3.2 - 4.6), median value of fans was 0 (IQR, 0 - 0), and median review counts was 5 (IQR, 2 - 17). This indicates that most reviewers do not have fans and wrote a small number of reviews.

Figures for the count of reviews per user and count of reviews per business show a unimodal distribution with fatter tail in the right side. Most reviewers tend to write a small number of reviews, and therefore most businesses got a small number of reviews. A question that arises from this plot: for the businesses that receive a high number of ratings, are the ratings more or less positive than the businesses that receive a low number of ratings? The same question can be asked of the users.

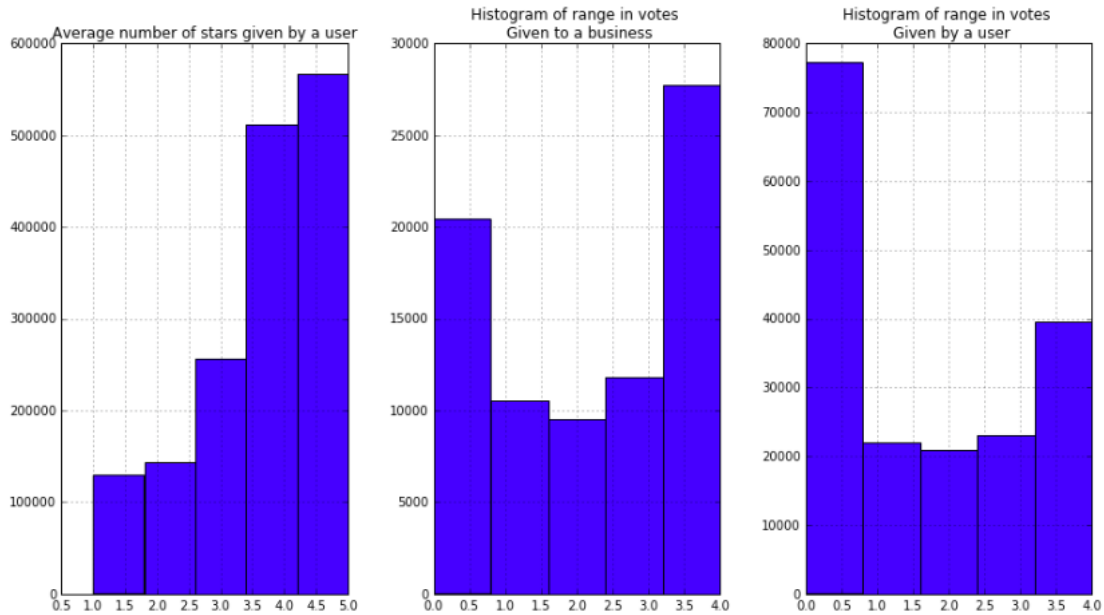
Now, we plotted average numbers of stars by the number of reviews of users according to the number of reviews in a business. Figures show a modest trend of having higher stars when a business was reviewed by many people. However, we don’t see clear relationship among the number of reviews given by a user (or to a business) and the average rating that the business has (or user gives). We can further corroborate this result by analysing the standard deviation in the reviews that a business receives (and a user gives).

We see that users typically rate businesses highly (with a clear mode being 5 stars). However, we also note that businesses have a high range of votes (a mode of 4 indicates that users differ in



their opinions). The figure also shows that users did not change their rating schema based on the business, and the majority of users rate all businesses within 1 point of each other.

Using a prior that assumes businesses will not be enjoyed (we wish to overcompensate for the average high ratings of users), we are able to build a more comprehensive likelihood and thus filter the review somewhat into a more confident top and bottom grouping. The ratings alone are unreliable and we rather wish to include the number of times a business has been rated in a particular manner to calculate the likelihood that this is a favorable (or not) business). For example, if a business is rated poorly twice, this is an unreliable statistic and we do not wish to penalise the



Businesses dataframe shape (85901, 15)

	attributes	business_id	categories	city	full_address	hours	latitude	longitude	name	neighbor
0	{'Accepts Credit Cards': True, 'Noise Level': ...}	0	['Fast Food', 'Restaurants']	Dravosburg	4734 Lebanon Church Rd\nDravosburg, PA 15034	{'Monday': {'close': '21:00', 'open': '11:00'}...	40.354327	-79.900706	Mr Hoagie	[]
1	{'Accepts Credit Cards': True, 'Price Range': ...}	1	['Nightlife']	Dravosburg	202 McClure St\nDravosburg, PA 15034	{}	40.350553	-79.886814	Clancy's Pub	[]

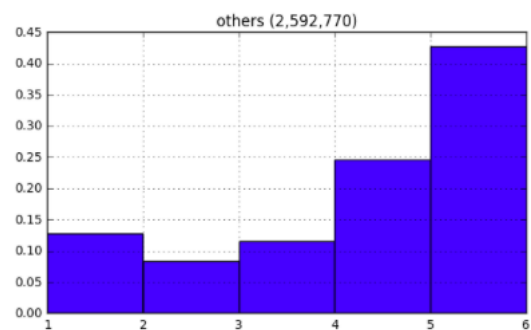
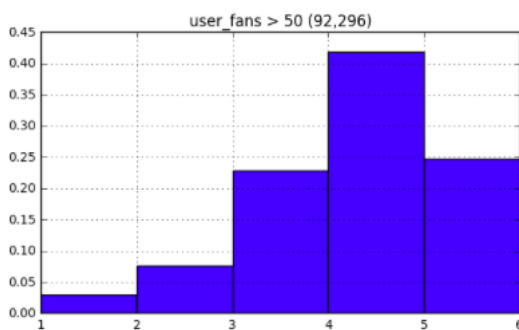
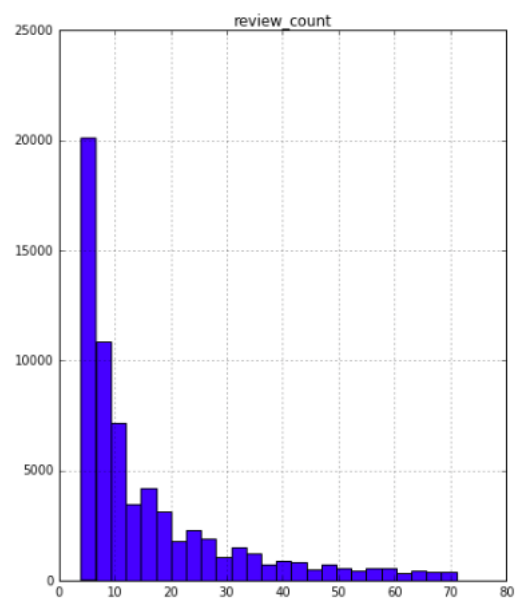
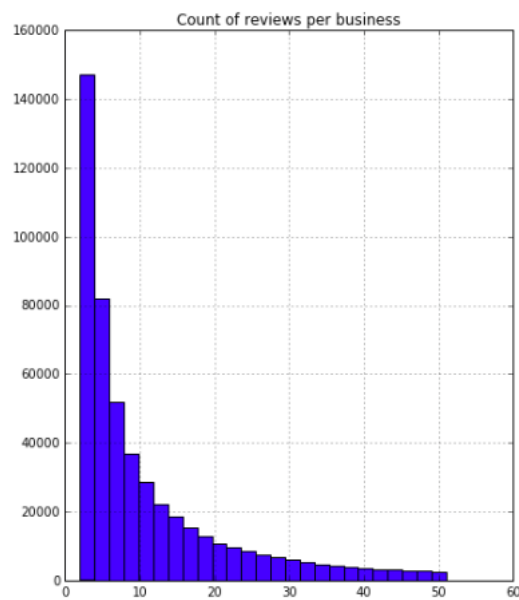
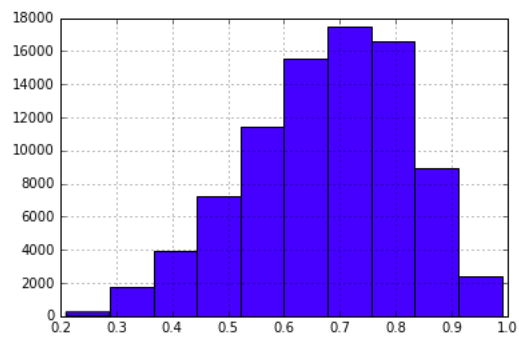
```
businesses.drop(['latitude', 'longitude', 'business_id'], axis=1).describe()
```

	open	review_count	stars
count	85901	85901.000000	85901.000000
mean	0.852272	34.352359	3.694852
std	0.354832	108.677591	0.946045
min	False	3.000000	1.000000
25%	1	5.000000	3.000000
50%	1	10.000000	4.000000
75%	1	26.000000	4.500000
max	True	6200.000000	5.000000

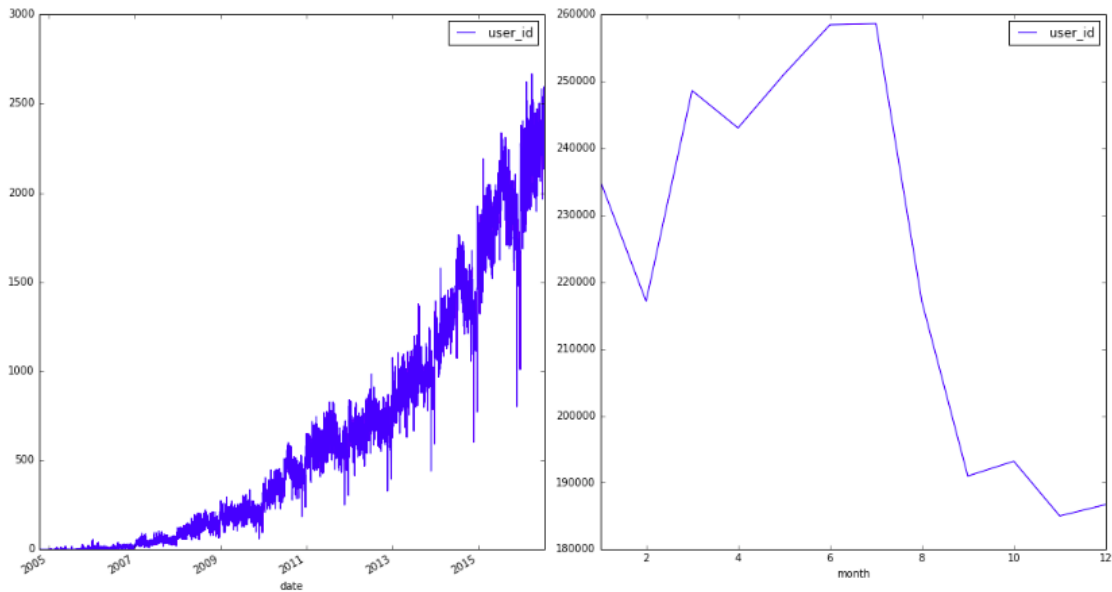
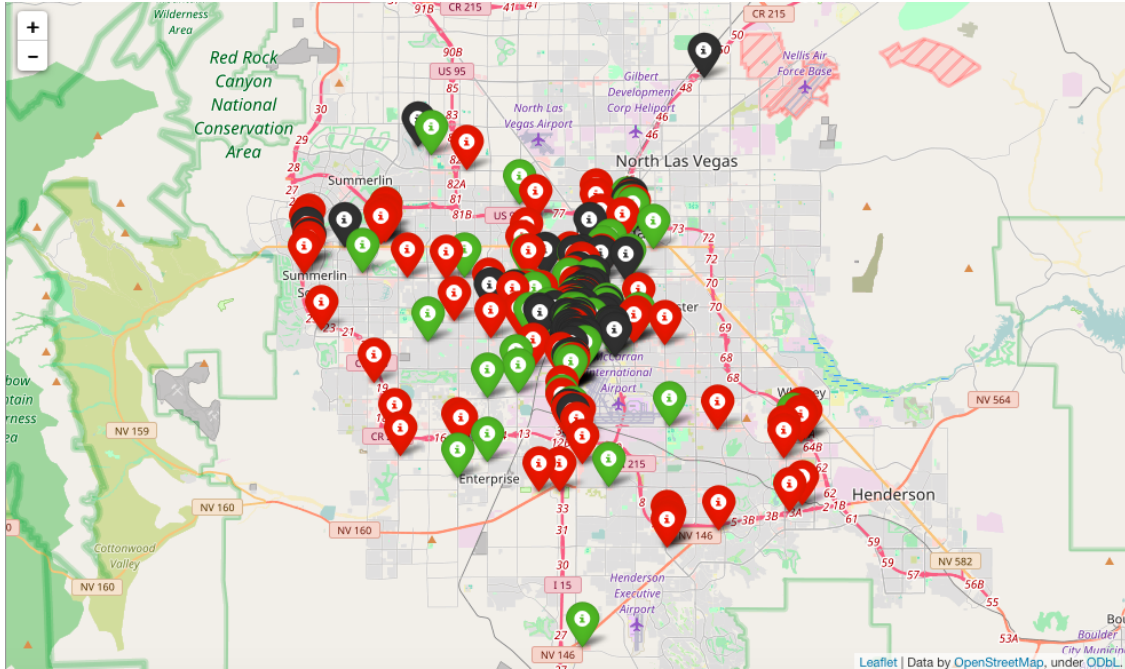
business from a median likelihood too dramatically. On the contrary, if a business is rated poorly 100 times, we are confident that users rate the business poorly and we wish to have a low likelihood

```
businesses_percentage_rating = businesses_sum_reviews.apply(sample_posterior, axis=1, args=(a, b,
n_samples))
businesses_percentage_rating.hist(bins=10)
```

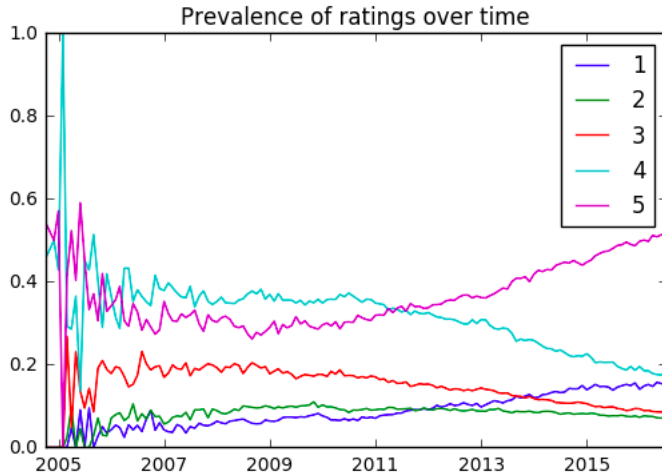
<matplotlib.axes.\_subplots.AxesSubplot at 0x115df7c50>



for enjoyment. In the figure, we show a distribution that corresponds to a probability that this is a good business (i.e., users will like it). On the low end of the scale we have a small number of very

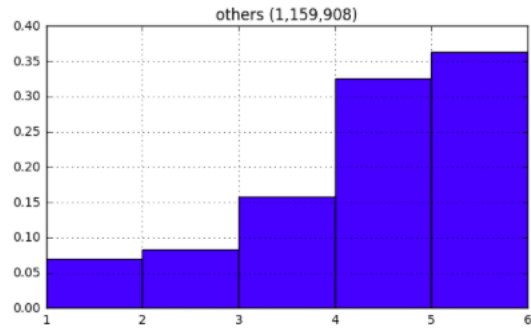
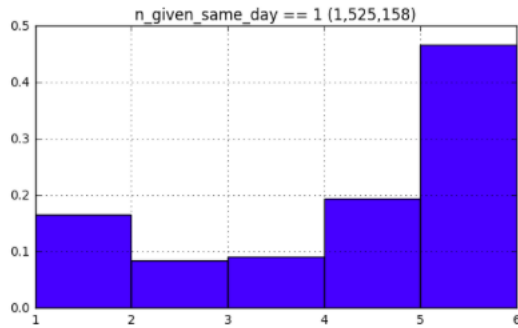


poor businesses. We note that these businesses will have a large number of poor ratings. Similarly, on the high end, we see that there is a relatively small number of highly rated businesses. These businesses require a high number of very positive ratings to be rated into this category. We note that we used a fairly aggressive prior assumption that all businesses have a median probability for being liked that is less than 2.5. This is to overcompensate for the generally high ratings. We now have a more normally distributed dataset.



```
compare_star_distribution(reviews, 'n_given_same_day == 1')
```

```
1 star % for n_given_same_day == 1 is 0.16 vs. 0.07 for others (difference of 81.9%)
2 star % for n_given_same_day == 1 is 0.08 vs. 0.08 for others (difference of 1.0%)
3 star % for n_given_same_day == 1 is 0.09 vs. 0.16 for others (difference of -54.9%)
4 star % for n_given_same_day == 1 is 0.19 vs. 0.33 for others (difference of -50.5%)
5 star % for n_given_same_day == 1 is 0.47 vs. 0.36 for others (difference of 24.8%)
```

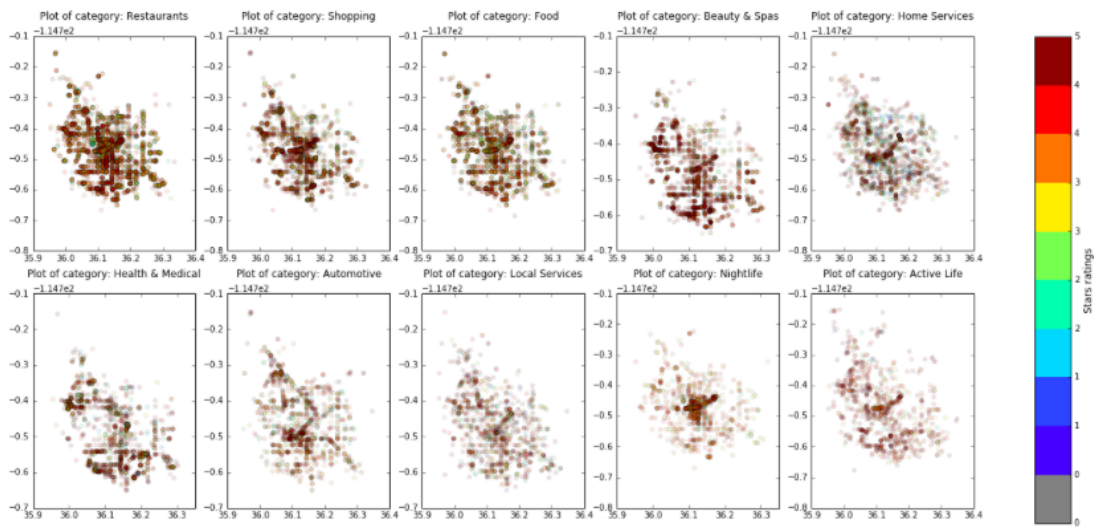


## Time series analysis

Now, we incorporated time axis on the data and conducted simple analyses. First, we noticed that yelp review has become more popular over time, and possibly, June and July are the most popular months.

In earlier years (i.e., before 2008), the average stars varied very widely from 1 to 5 stars. In contrast, after 2009 onward, the average stars converged to around 4. This is mainly because we have more reviews over time and the mean value became less variable. The rating did not differ according to the month reviewed.

When we analyzed the number of repeated reviews by a user, we found that majority of reviews reviewed once.



Users dataframe shape (686556, 11)

	average_stars	compliments	elite	fans	friends	name	review_count	type	user_id	votes	yelping_since
0	4.14	{'plain': 25, 'writer': 9, 'cute': 15, 'photos...	[2005, 2006]	69	[1, 2, 3, 5, 93, 12, 99, 464, 1025, 1298, 1388...	Russel	108	user	0	{'cool': 246, 'useful': 282, 'funny': 167}	2004-10
1	3.67	{'plain': 970, 'writer': 346, 'cute': 204, 'ph...	[2005, 2006, 2007, 2008, 2009, 2010, 2011, 201...	1345	[0, 2, 3, 4, 5, 6, 8, 9, 12, 95, 97, 187, 465,...	Jeremy	1292	user	1	{'cool': 12091, 'useful': 15242, 'funny': 8399}	2004-10

```
users.drop('user_id', axis=1).describe()
```

	average_stars	fans	review_count
count	686556.000000	686556.000000	686556.000000
mean	3.746704	1.290100	25.757102
std	1.086832	11.501621	83.755973
min	0.000000	0.000000	0.000000
25%	3.230000	0.000000	2.000000
50%	3.920000	0.000000	5.000000
75%	4.600000	0.000000	17.000000
max	5.000000	3549.000000	10897.000000