Yelp Dataset Challenge

By: Joshua Chang, Jae Choi, Edward Kang, Rachel Pang

Yelp Dataset Challenge

Yelp has had 8 rounds of of dataset challenges. This year, they are on their 9th round and are providing everyone with their largest dataset ever.

We’ve decided to take on this challenge which contained millions of observation across 5 separate datafiles.

The Datafiles

Business - Business ID, Business name, neighborhood, address, city, state, postal code, latitude, longitude, star rating, review count, open/closed, attributes, categories, hours, type.

User - User ID, name, account age, review count, # of friends (social media interaction), review count, User's Elite status, review vote ratings, overall average star score of reviews, compliments received, type

Review - Review ID, User ID, star rating, date, review text, review votes (useful, funny, cool), type.

Check in - Time, Business ID, type.

Tip - Business ID, User ID, text, date, likes, type

YelpBusinessData	144072 obs. of 16 variables
YelpCheckInData	125532 obs. of 3 variables
YelpReviewData	4153150 obs. of 10 variables
YelpTipData	946600 obs. of 6 variables
YelpUserData	1029432 obs. of 23 variables

The Data

yelp_academic_dataset_business.json

```
{
    "business_id":"encrypted business id",
    "name":"business name",
    "neighborhood":"hood name",
    "address":"full address",
    "city":"city",
    "state":"state -- if applicable --",
    "postal code":"postal code",
    "latitude":latitude,
    "longitude":longitude,
    "stars":star rating, rounded to half-stars,
    "review_count":number of reviews,
    "is_open":0/1 (closed/open),
    "attributes":["an array of strings: each array element is an attribute"],
    "categories":["an array of strings of business categories"],
    "hours":["an array of strings of business hours"],
    "type": "business"
}
```

yelp_academic_dataset_review.json

```
{
    "review_id":"encrypted review id",
    "user_id":"encrypted user id",
    "business_id":"encrypted business id",
    "stars":star rating, rounded to half-stars,
    "date":"date formatted like 2009-12-19",
    "text":"review text",
    "useful":number of useful votes received,
    "funny":number of funny votes received,
    "cool": number of cool review votes received,
    "type": "review"
}
```

The Data

yelp_academic_dataset_user.json


```
{
    "user_id":"encrypted user id",
    "name":"first name",
    "review_count":number of reviews,
    "yelping_since": date formatted like "2009-12-19",
    "friends":["an array of encrypted ids of friends"],
    "useful":"number of useful votes sent by the user",
    "funny":"number of funny votes sent by the user",
    "cool":"number of cool votes sent by the user",
    "fans":"number of fans the user has",
    "elite":["an array of years the user was elite"],
    "average_stars":floating point average like 4.31,
    "compliment_hot":number of hot compliments received by the user,
    "compliment_more":number of more compliments received by the user,
    "compliment_profile": number of profile compliments received by the user,
    "compliment_cute": number of cute compliments received by the user,
    "compliment_list": number of list compliments received by the user,
    "compliment_note": number of note compliments received by the user,
    "compliment_plain": number of plain compliments received by the user,
    "compliment_cool": number of cool compliments received by the user,
    "compliment_funny": number of funny compliments received by the user,
    "compliment_writer": number of writer compliments received by the user,
    "compliment_photos": number of photo compliments received by the user,
    "type":"user"
}
```


yelp_academic_dataset_checkin.json


```
{
    "time":["an array of check ins with the format day-hour:number of check ins from hour to hour+1"],
    "business_id":"encrypted business id",
    "type":"checkin"
}
```

Objectives

Our objectives after careful observation include:

(1) focusing on a general analysis of eateries/restaurants

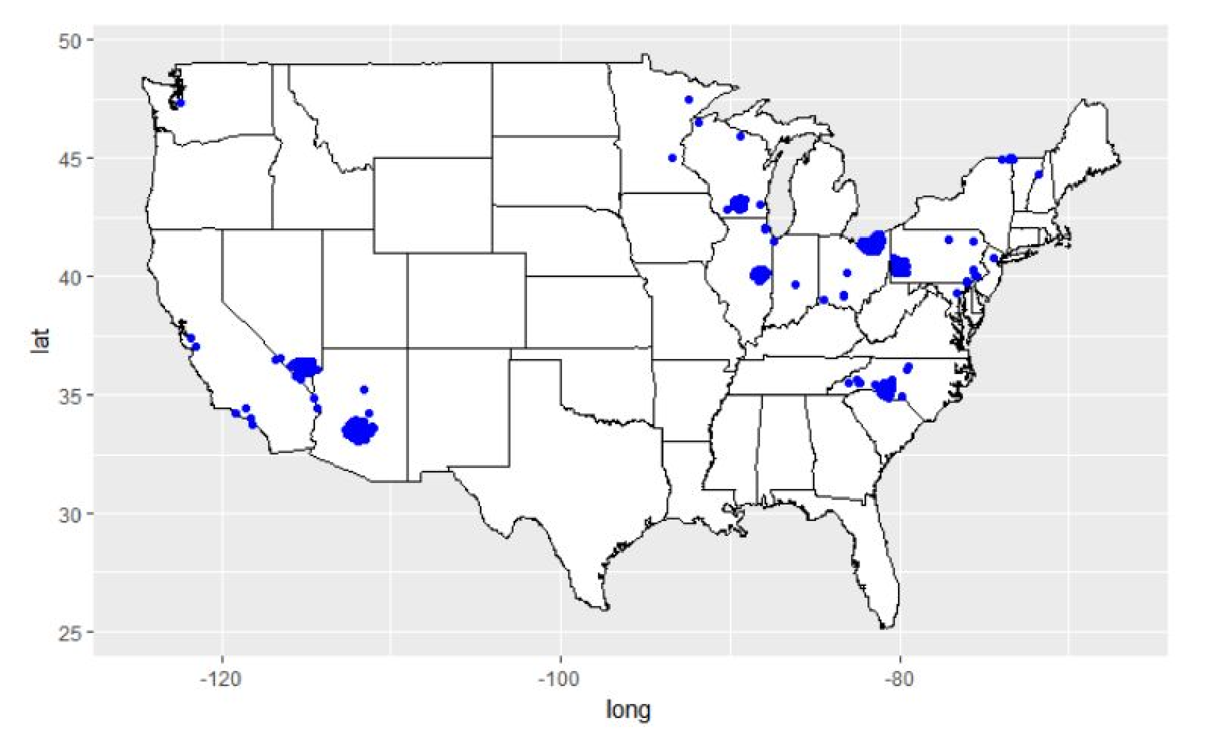
(2) joining the reviews, tips, and business datasets to analyze the expertise/credibility of users

(3) text mining in order to find good and bad reviews

For the most part we want to focus on businesses that are located in America, our area of interest, which will be applied to the restaurant businesses and to the reviews text mining.

Businesses in America

	state	count
1	AZ	43492
2	NV	28214
3	NC	10177
4	OH	9966
5	PA	8091
6	WI	3899
7	IL	1556
8	SC	498
9	NY	13
10	VT	1



Top 5 States in Our Dataset

	state	count
1	AZ	43492
2	NV	28214
3	NC	10177
4	OH	9966
5	PA	8091

Arizona
Nevada
North Carolina
Ohio
Pennsylvania

ANOVA test to test mean average star rating in each State's businesses

P-value <2e-16 which is less than alpha= 0.05

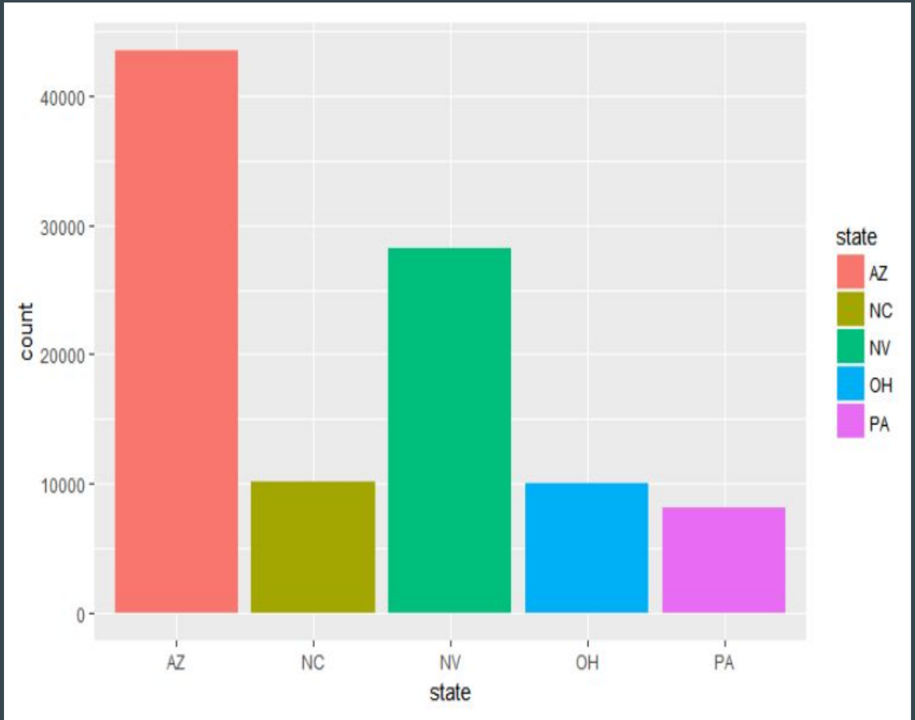
```
'data.frame':   15537 obs. of  6 variables:
 $ long     : num  -87.5 -87.5 -87.5 -87.5 -87.6 ...
 $ lat      : num  30.4 30.4 30.4 30.3 30.3 ...
 $ group    : num  1 1 1 1 1 1 1 1 1 1 ...
 $ order    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ region   : chr  "alabama" "alabama" "alabama" "alabama" ...
 $ subregion: chr  NA NA NA NA ...
                   Df Sum Sq Mean Sq F value Pr(>F)
as.factor(state)     4    335   83.70   82.74 <2e-16 ***
Residuals        99935 101098    1.01
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Top 5 Most Number of Businesses in America

Tukey HSD for multiple comparisons of mean average ratings per State

$`as.factor(state)`

	diff	lwr	upr	p adj
NC-AZ	-0.144616826	-0.17482805	-0.114405604	0.0000000
NV-AZ	-0.017115985	-0.03808905	0.003857077	0.1699750
OH-AZ	-0.149491659	-0.17996095	-0.119022369	0.0000000
PA-AZ	-0.089558446	-0.12277607	-0.056340818	0.0000000
NV-NC	0.127500841	0.09577643	0.159225255	0.0000000
OH-NC	-0.004874833	-0.04353937	0.033789708	0.9969999
PA-NC	0.055058380	0.01419299	0.095923768	0.0022127
OH-NV	-0.132375675	-0.16434594	-0.100405404	0.0000000
PA-NV	-0.072442461	-0.10704205	-0.037842875	0.0000001
PA-OH	0.059933213	0.01887667	0.100989755	0.0006530



Interpretations of Results

After we performed the ANOVA test, we got a p-value equal to <2e-16, which is less than alpha = 0.05 which tells us that at least one of the states had a significantly different mean average star rating than the others. We performed the Tukey HSD test to test for multiple comparisons of means and found that all the combinations of states except those of Nevada and Arizona and Ohio and Pennsylvania had significantly different mean average star ratings.

Attributes Of Restaurants

First... Filter Businesses in America to only Restaurants in America.

Attributes include:

“Ambiance:{romantic: false, classy: false, hipster: true...”

“BusinessAcceptsCreditCard: True”

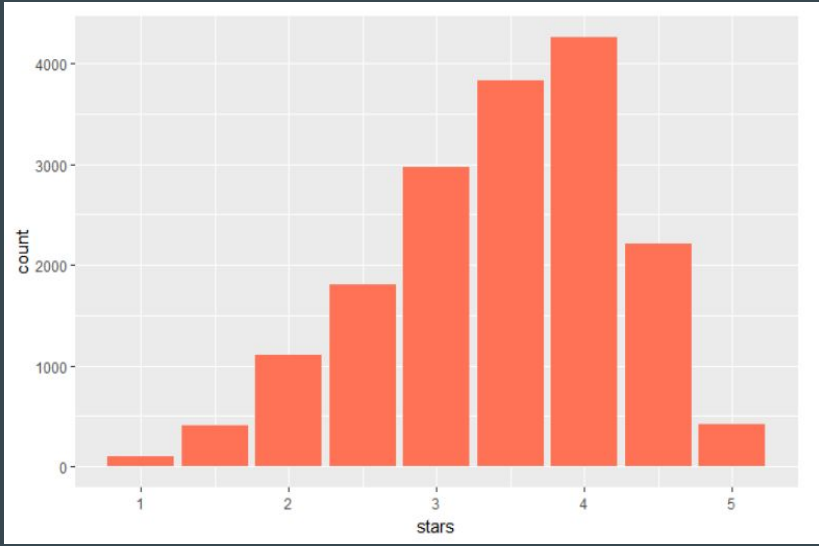
Attribute focus: “Alcohol: none”, “Alcohol: full_bar”, “Alcohol: beer_wine”

17,112 do not serve alcohol

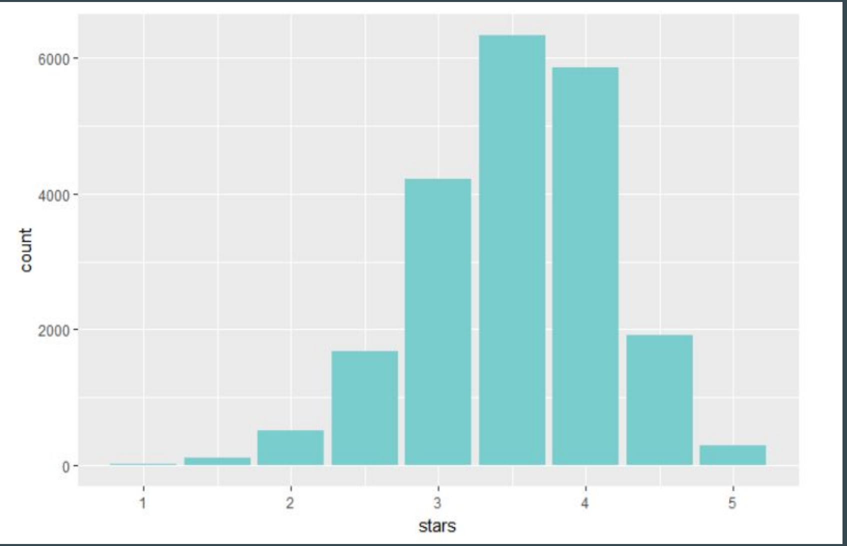
20,925 serve alcohol

Alcoholic vs. NonAlcoholic Restaurant Ratings with Barplots

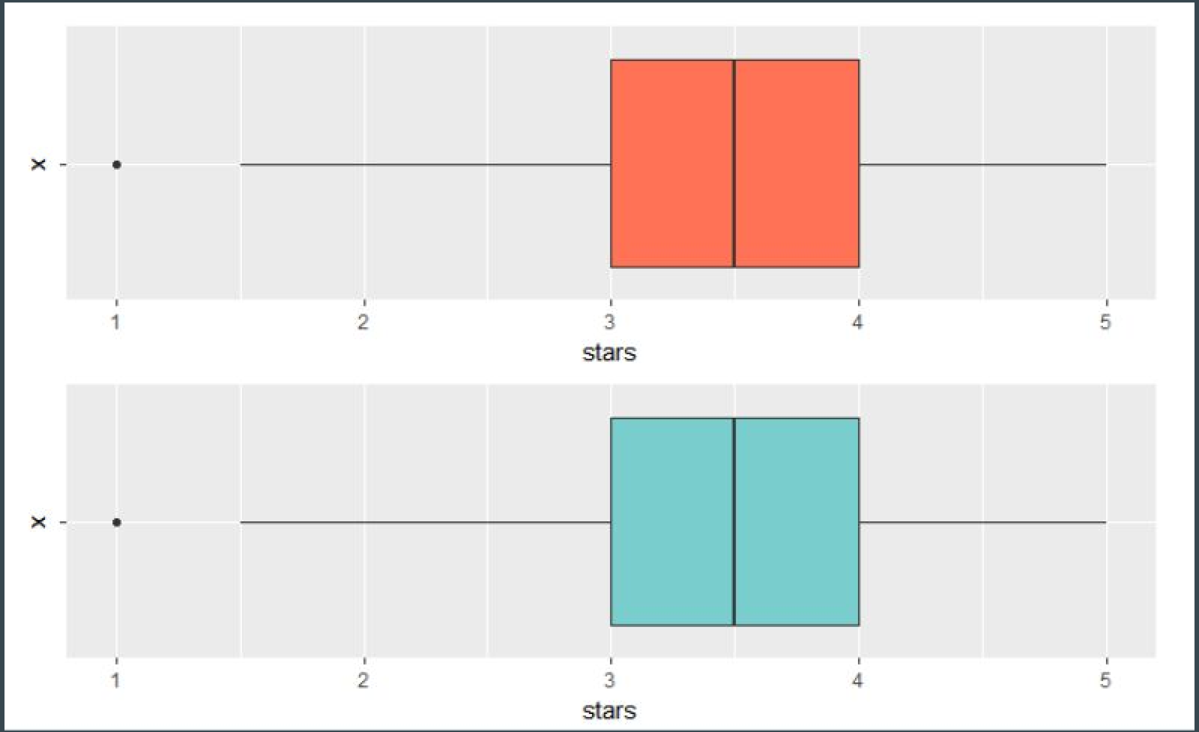
Serves Alcohol



Does not serve alcohol



Alcoholic vs. NonAlcoholic Restaurant Ratings with Boxplots



Text Mining Categories

In order to see which ethnic restaurants were in the top 5 and bottom five, we had to do a little text mining to see which categories were prominent in our observations

```
tidyRestaurants <- BusAmericaRestaurants %>%
    unnest_tokens(category, categories)
```

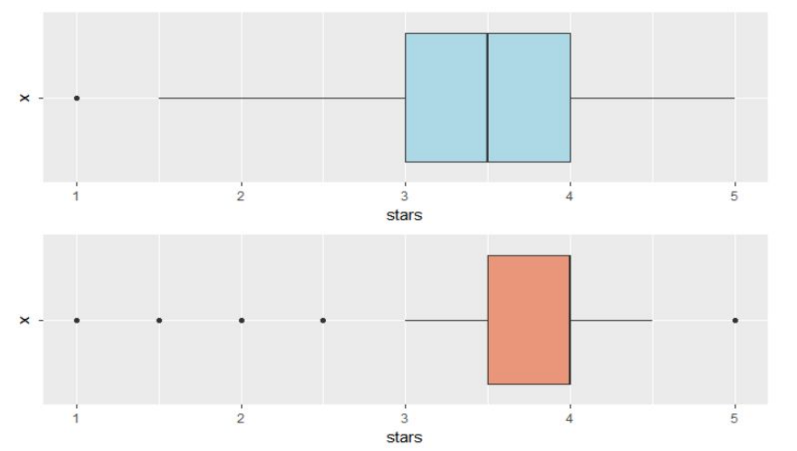
The Top 5 Ethnic Food Groups: American, Italian, Mexican, Chinese, Japanese

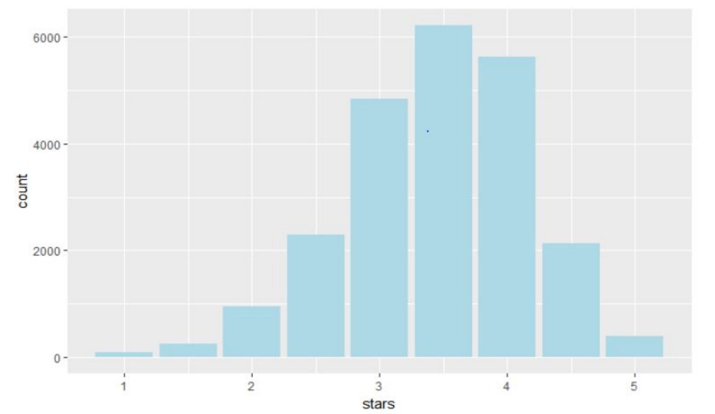
The Bottom 5 Ethnic Food Groups (from those that appeared at least 100 times):
African, Persian, Iranian, Lebanese, Taiwanese

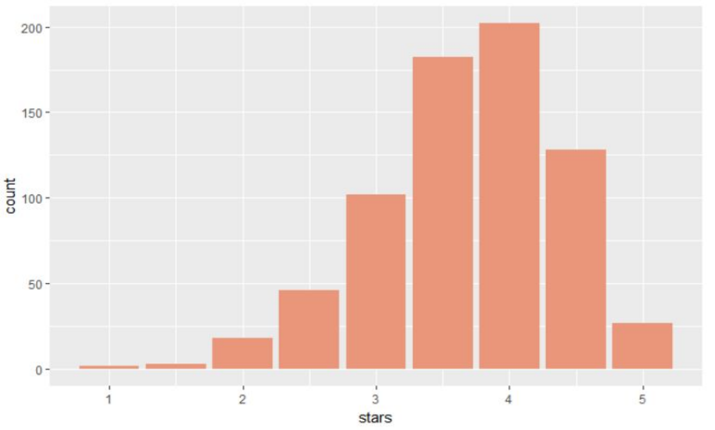
	category	n
1	restaurants	48486
2	food	17791
3	bars	11408
4	american	9351
5	nightlife	6334
6	traditional	5312
7	fast	5250
8	pizza	5229
9	sandwiches	5220
10	new	4922
11	italian	4118
12	burgers	3868
13	mexican	3688
14	chinese	3611

Means Comparison: Top 5 & Bottom 5 Ethnic Food Groups Ratings

```
        Welch Two Sample t-test

data:  ethnictop5$stars and ethniclow5$stars
t = -9.2401, df = 754.99, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3002060 -0.1949973
sample estimates:
mean of x mean of y
 3.441835  3.689437
```







Difference of Mean Average between Top 5 Ethnic Food Groups

```
                       Df Sum Sq Mean Sq F value Pr(>F)    
as.factor(category)     4    122  30.429   60.62 <2e-16 ***
Residuals           22818  11454   0.502                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = stars ~ as.factor(category), data = ethnictop5)

$`as.factor(category)`
                         diff         lwr         upr     p adj
chinese-american  -0.119867596 -0.15773644 -0.08199875 0.0000000
italian-american   0.097745933  0.06159793  0.13389393 0.0000000
japanese-american  0.124691844  0.07760273  0.17178096 0.0000000
mexican-american  -0.007917434 -0.04550000  0.02966513 0.9787749
italian-chinese    0.217613529  0.17354855  0.26167850 0.0000000
japanese-chinese   0.244559440  0.19115130  0.29796757 0.0000000
mexican-chinese    0.111950162  0.06670092  0.15719940 0.0000000
japanese-italian   0.026945911 -0.02525617  0.07914799 0.6223907
mexican-italian   -0.105663367 -0.14948257 -0.06184417 0.0000000
mexican-japanese  -0.132609278 -0.18581481 -0.07940374 0.0000000
```

Interpretation of Results

- We wanted to test if there was a difference in means of average star ratings for each of the top five ethnic groups. We performed the ANOVA test to test whether or not the population means were equal to each other. The p-value we observed, <2e-16, was less than alpha = 0.05 so we concluded that at least one of the population means were different from each other.
- After performing the post hoc test, the Tukey HSD test, we saw that only the Mexican-American and Japanese-Italian combinations had mean average star ratings that were not different from each other at 0.05 significance level.

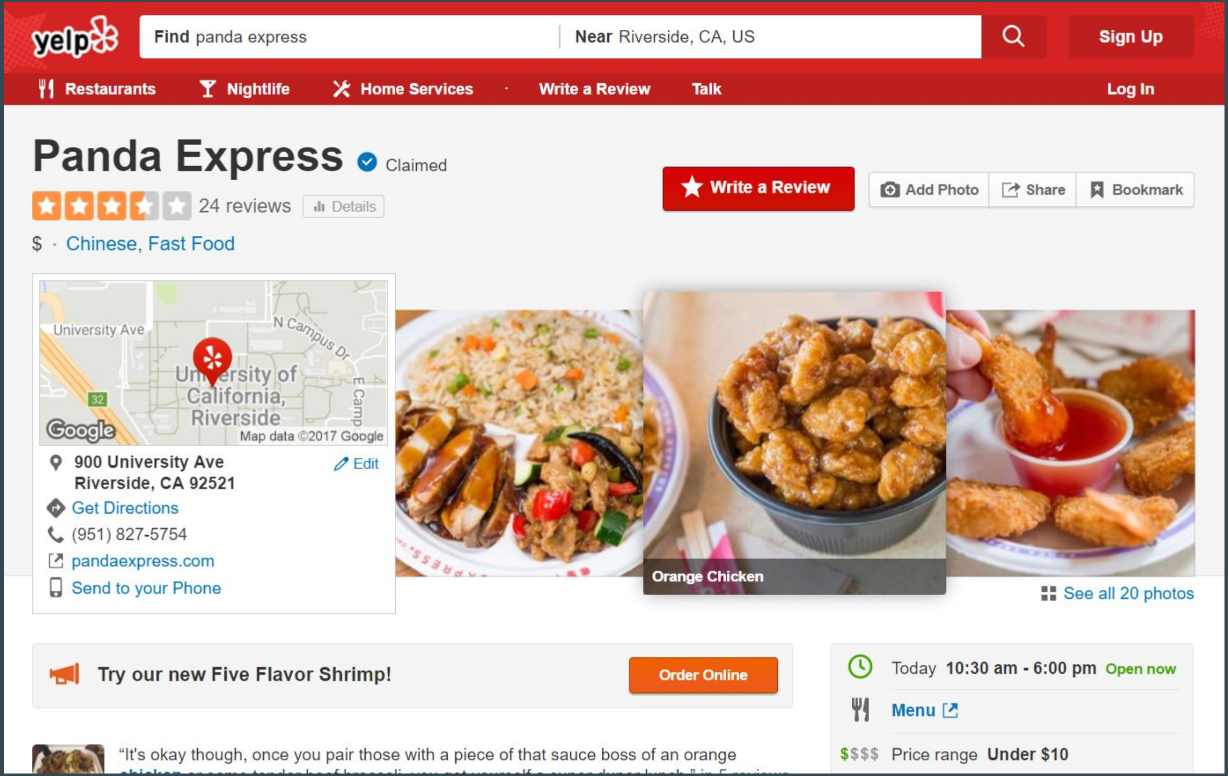
Difference of Means between bottom 5 ethnic restaurants

```
                     Df Sum Sq Mean Sq F value Pr(>F)
as.factor(category)   4    5.9  1.4825   3.035 0.0169 *
Residuals           705  344.3  0.4884
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = stars ~ as.factor(category), data = ethniclow5)

$`as.factor(category)`
                          diff        lwr         upr     p adj
iranian-african    -0.10442834 -0.3357989  0.12694226 0.7312425
lebanese-african   -0.07241119 -0.3002524  0.15543003 0.9081171
persian-african    -0.10442834 -0.3357989  0.12694226 0.7312425
taiwanese-african  -0.27409844 -0.5001386 -0.04805831 0.0084875
lebanese-iranian    0.03201715 -0.1945294  0.25856373 0.9952710
persian-iranian     0.00000000 -0.2300958  0.23009582 1.0000000
taiwanese-iranian  -0.16967010 -0.3944052  0.05506502 0.2366783
persian-lebanese   -0.03201715 -0.2585637  0.19452942 0.9952710
taiwanese-lebanese -0.20168725 -0.4227871  0.01941258 0.0929397
taiwanese-persian  -0.16967010 -0.3944052  0.05506502 0.2366783
```

Interpretation of the Results

- We then went on to perform another ANOVA test in the bottom five ethnic restaurants. Similarly we got a p-value less than alpha = 0.05 and concluded at least one of the population means were significantly different from each other.
- We performed the Tukey HSD test once again and observed that only the combination of Taiwanese and African restaurants had mean average star ratings that were different from each other at 0.05 significance level.

Review Accuracy and User Credibility



Review Accuracy and User Credibility

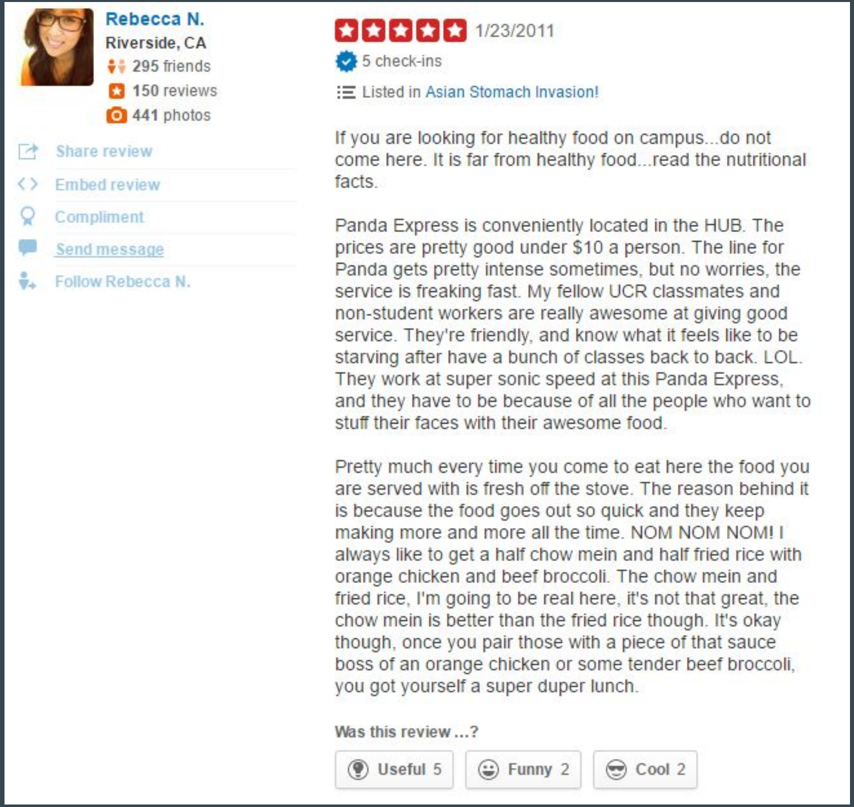

Andrew M.
Imperial Beach, CA
65 friends
14 reviews
6 photos
Share review
Embed review
Compliment
Send message
Follow Andrew M.

2/15/2012
2 check-ins

One of the only places I seek out and after eating at I swear never to eat at again.

This happens several times a month.

Was this review ...?
Useful
Funny 4
Cool


Rebecca N.
Riverside, CA
295 friends
150 reviews
441 photos
Share review
Embed review
Compliment
Send message
Follow Rebecca N.

1/23/2011
5 check-ins
Listed in Asian Stomach Invasion!

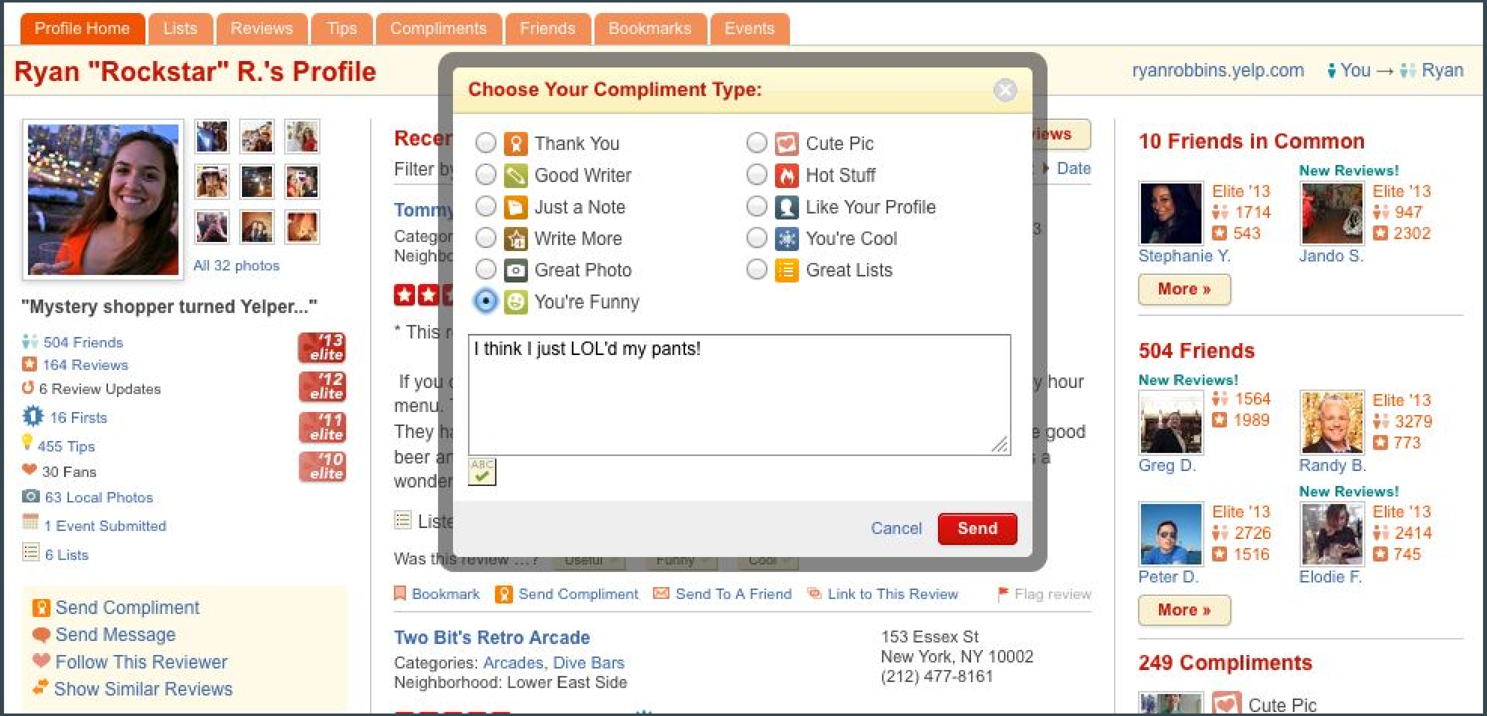
If you are looking for healthy food on campus...do not come here. It is far from healthy food...read the nutritional facts.

Panda Express is conveniently located in the HUB. The prices are pretty good under $10 a person. The line for Panda gets pretty intense sometimes, but no worries, the service is freaking fast. My fellow UCR classmates and non-student workers are really awesome at giving good service. They're friendly, and know what it feels like to be starving after have a bunch of classes back to back. LOL. They work at super sonic speed at this Panda Express, and they have to be because of all the people who want to stuff their faces with their awesome food.

Pretty much every time you come to eat here the food you are served with is fresh off the stove. The reason behind it is because the food goes out so quick and they keep making more and more all the time. NOM NOM NOM! I always like to get a half chow mein and half fried rice with orange chicken and beef broccoli. The chow mein and fried rice, I'm going to be real here, it's not that great, the chow mein is better than the fried rice though. It's okay though, once you pair those with a piece of that sauce boss of an orange chicken or some tender beef broccoli, you got yourself a super duper lunch.

Was this review ...?
Useful 5
Funny 2
Cool 2

Review Accuracy and User Credibility



Review Accuracy and User Credibility

```
YelpBusinessReviewJoint <- YelpBusinessReviewJoint %>%
  mutate(stardeviation = abs(YelpAvg_stars - stars))
```

```
YelpReviewAccuracy <- YelpBusinessReviewJoint %>%
  group_by(user_id) %>%
  summarise(reviewcount = n(), ReviewAccuracy = mean(stardeviation))
```

```
YelpUserExpert <- filter(YelpReviewAccuracy, ReviewAccuracy <= 0.75 ) %>%
                  mutate(Credibility = "Expert", Credibility_Binary = 1)

YelpUserFair075 <- filter(YelpReviewAccuracy, ReviewAccuracy > 0.75)
YelpUserFair <- filter(YelpUserFair075, ReviewAccuracy <= 1.5) %>%
                  mutate(Credibility = "Fair", Credibility_Binary = 0)

YelpUserPoor <- filter(YelpReviewAccuracy, 1.5 < ReviewAccuracy) %>%
                  mutate(Credibility = "Poor", Credibility_Binary = 0)

YelpUserCredibility <- rbind(YelpUserExpert, YelpUserFair, YelpUserPoor) %>%
                  arrange(desc(reviewcount))
```

Review Accuracy and User Credibility

user_id <chr>	reviewcount <int>	ReviewAccuracy <dbl>	Credibility <chr>	Credibility_Binary <dbl>	year <chr>	month <chr>	day <chr>	useful <int>
8RcEwGrFlgkt9WQ35E6SnQ	42	0.6904762	Expert	1	2009	11	06	51
hWDybu_KvYLSdEFzGrniTw	972	0.5889918	Expert	1	2009	03	08	16936
Xwnf20FKuikiHcSpcEbpKQ	148	0.6013514	Expert	1	2011	06	10	1259
CxDOIDnH8gp9KXzpBHJYXw	3291	0.5478578	Expert	1	2009	11	09	1143
kS1MQHYwIfD0462PE61IBw	51	0.5882353	Expert	1	2007	08	25	434
XYSDrlef7g4Gmp3lNFVO6A	171	0.8684211	Fair	0	2007	07	19	15717
nzsv-p1O8gCfP3XijfQrlw	131	0.7709924	Fair	0	2005	04	12	1595
wZPizeBxMAyOSl0M0zuCjg	51	0.5196078	Expert	1	2008	08	27	1541
U4INQZOPSUaj8hMjLlZ3KA	1012	0.8147233	Fair	0	2008	01	31	406
m07sy7eLtOjVdZ8oN9JKag	54	0.6759259	Expert	1	2006	07	22	114

1-10 of 211,199 rows | 1-9 of 33 columns

Previous 1 2 3 4 5 6 ... 100 Next

```
YelpUserEnhanced <- separate(YelpUserEnhanced, yelping_since, into = c("year", "month", "day"), sep="-")
```

Review Accuracy and User Credibility

compliment_photos <int>	compliment_list <int>	compliment_funny <int>	compliment_plain <int>	review_count <int>
76	14	94	209	7519
674	26	1378	1763	7125
82	1	273	487	6252
995	92	1278	3126	5596
101	73	728	524	4312
108	64	909	1075	4053
276	30	2365	1717	3992
87	16	513	556	3734
463	51	738	629	3632
1301	198	2422	1082	3546

1-10 of 211,199 rows | 10-14 of 33 columns

Previous 1 2 3 4 5 6 ... 100 Next

Review Accuracy and User Credibility

fans <int>	type <chr>	compliment_note <int>	funny <int>	compliment_writer <int>	compliment_cute <int>	average_stars <dbl>	compliment_more <int>
243	user	130	70	66	4	3.48	23
337	user	608	14304	814	10	3.53	290
204	user	195	959	73	1	3.33	38
508	user	1123	1643	400	60	3.28	175
393	user	227	21	197	14	3.79	85
325	user	630	2496	226	26	3.92	67
648	user	1160	1621	934	106	3.44	194
304	user	200	564	191	26	3.60	34
607	user	413	276	591	9	3.87	71
1425	user	640	55	2062	144	3.66	264

1-10 of 211,199 rows | 16-23 of 33 columns

Previous 1 2 3 4 5 6 ... 100 Next

Review Accuracy and User Credibility

friends <chr>
['NQcerFt8bdU3mu7QtxPUuw', 'lxB8uybfnHxuALjnArSCtg', 'eiiXoxbl2nzTy3VTsOd-Qw', 'XQsOfX0Xwk6C7aolDS7-Ww', 'FpB44ccQnPntnZQmWTTKDA', 'xD7...
['yxkTcHsWMh4uq3kIRowqGw', '7GIcGERUfVvOx_TNYomGcA', 'AACF348wJKAB3_Lt4A9IZg', 'ywYAclrxKjtlo_ZSfj5v2w', 'BUB_t_Rvzs1yPEzZipkWjw', 'Mt1PCN...
['OZWskTOKCWvYejhu63gFmQ', '-bMKjy4pd_0UXa4hWjCdvQ', 'oi-EOvREMolVejD9jPiO-g', 'oFsK3Ki_qY8iQn5eEMSztA', 'EMeATph-8T_JA3qH36pxuw', 'etut-rq...
['oZyiiNdA-I5zk_EHUmGRMQ', 'qpFCY_wj8_G-HOV8MxlaHw', '9EWR2_AULBcXID8ptXX_kA', 'Cy7k3jFH2LWct-LC294v_g', 'VI3WYpyaWGj-jtz3CkMP2Q', 'ma620...
['_c50WkYV9C2XieD8blXBUA', '5rmEJbcwwo5GhTi0AWpqWA', 'dfFNnnfCl3Kr_zMPRKrm_A', 'cjotiAHFSbFuvbxTWmGi1Q', 'jy_RGAeAvNeCv5BxZsUWFA', '1GQ...
['62V-Kyj1MD0IVGgZYkxy7w', 'KKpfY2xlz0E2ORXooKiz-w', 'DEFk299pEBqKgUgQ703V9g', 'rTyn3YHLhXylJwRfqeRxEg', 'yyYK8k2Ikga72nZ4cSBf-A', 'ADD7s6...
['JnGtgOPpkjyWOvWM0SYEXg', '2OrENo4Nwnqg6NxlZmnzvw', 'DBHCFW3mSmmOEpONHVu1rQ', 'KC333dkVGOZLjG8WFpyrpg', 'QvyujLggHLd7KR3uYKo3V...
['nH0fwGMkOAfFuzvWwh34gQ', 'z6F5QFaQ90g5ztZc5jTYsQ', 'JnGtgOPpkjyWOvWM0SYEXg', 'vqKLzj_SGdnBEnG2wSuGyg', '_KCaYj9WxrTAxzwtgMAZlw', 'u...
['eKBLRdDIMIQ7eE_3BobRPQ', 'xdQzGzNu3nlUEvOGPW1tYw', '3DIHjlfL8T-AlHvuuomFQQ', 's1kbaGxgMFUOeOy7_Jq4zg', 'NQffx45eJaeqhFcMadKUQA', 'giuW...
['fWJok1wljTdXP7VylDBYoA', 'dLIYdcLKQFlCCIjegqFqJQ', 'LaNXfiMDjMib2NhK4BE7-Q', '7uvfAKXezRXrLWNo1VZWbA', '7M_6FojGZNbT3ZJt_wrV-A', 'KltM7KJix...

1-10 of 211,199 rows | 24-24 of 33 columns

Previous 1 2 3 4 5 6 ... 100 Next

```
%>% mutate(numElite_Years = floor(nchar(elite) / 8),
           num_Friends = floor(nchar(friends) / 26),
           accountage_days = (2017 - as.numeric(year))*365 + (12 - as.numeric(month))*30 +
          as.numeric(day))
```

Review Accuracy and User Credibility

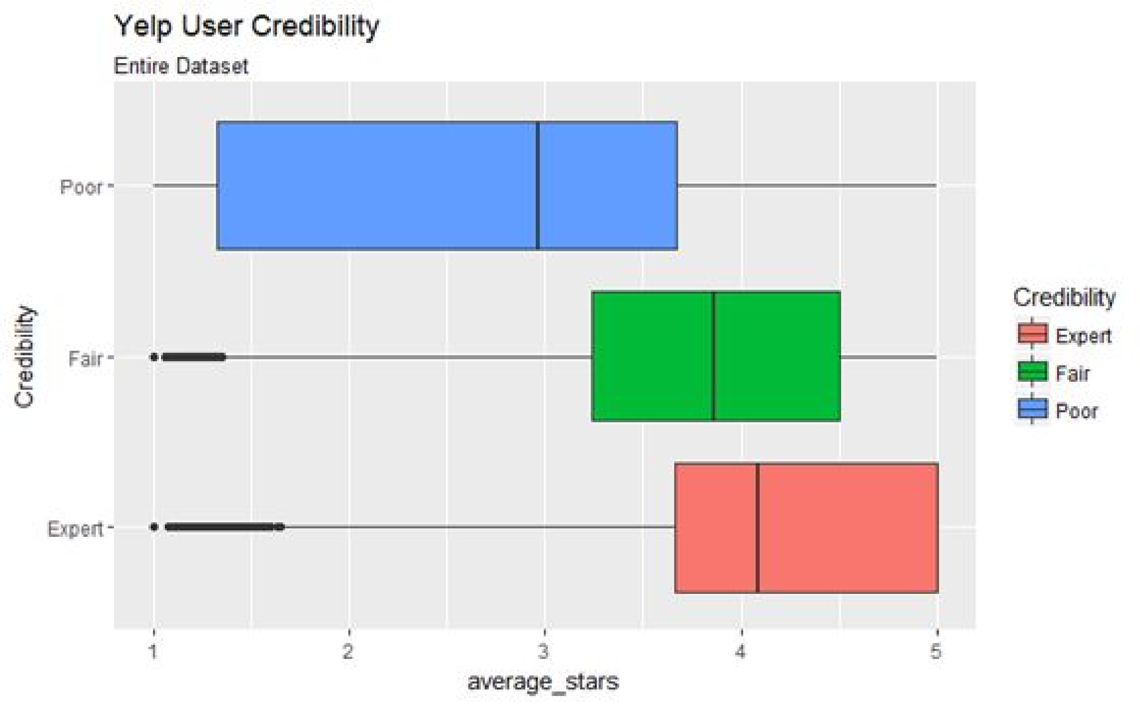
cool <int>	name <chr>	compliment_profile <int>	compliment_cool <int>	numElite_Years <dbl>	num_Friends <dbl>	accountage_days <dbl>	tipcount <int>
43	Dan	6	94	3	939	2956	20
14350	Bruce	218	1378	8	1480	3198	58
986	Kenneth	12	273	0	2350	2380	4
877	Jennifer	32	1278	9	469	2959	1337
23	Rob	23	728	5	1896	3795	1
61941	Neal	53	909	11	1376	3819	4
1386	Anita	111	2365	13	1739	4632	1
1756	Jess	33	513	9	3285	3432	1
315	Michael	38	738	7	2801	3646	194
93	Ed	144	2422	12	4574	4187	33

1-10 of 211,199 rows | 26-33 of 33 columns

Previous 1 2 3 4 5 6 ... 100 Next

Review Accuracy and User Credibility

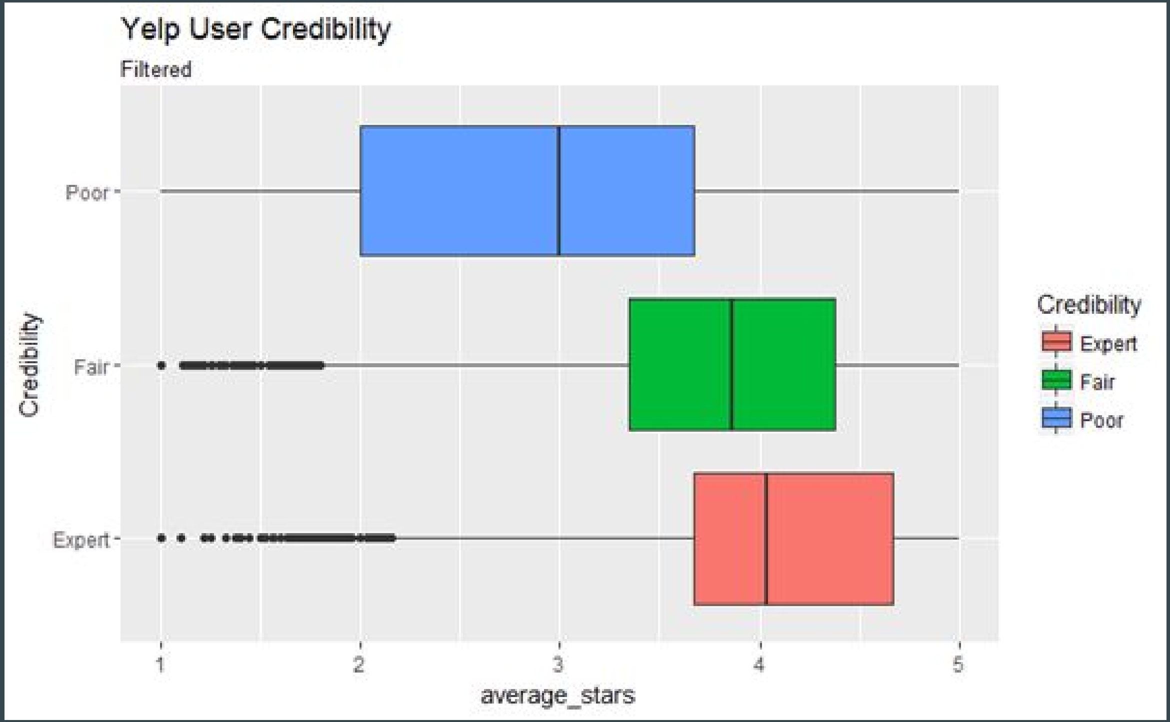
Unfiltered User Data



Review Accuracy and User Credibility

Filtered Data

(tipcount)

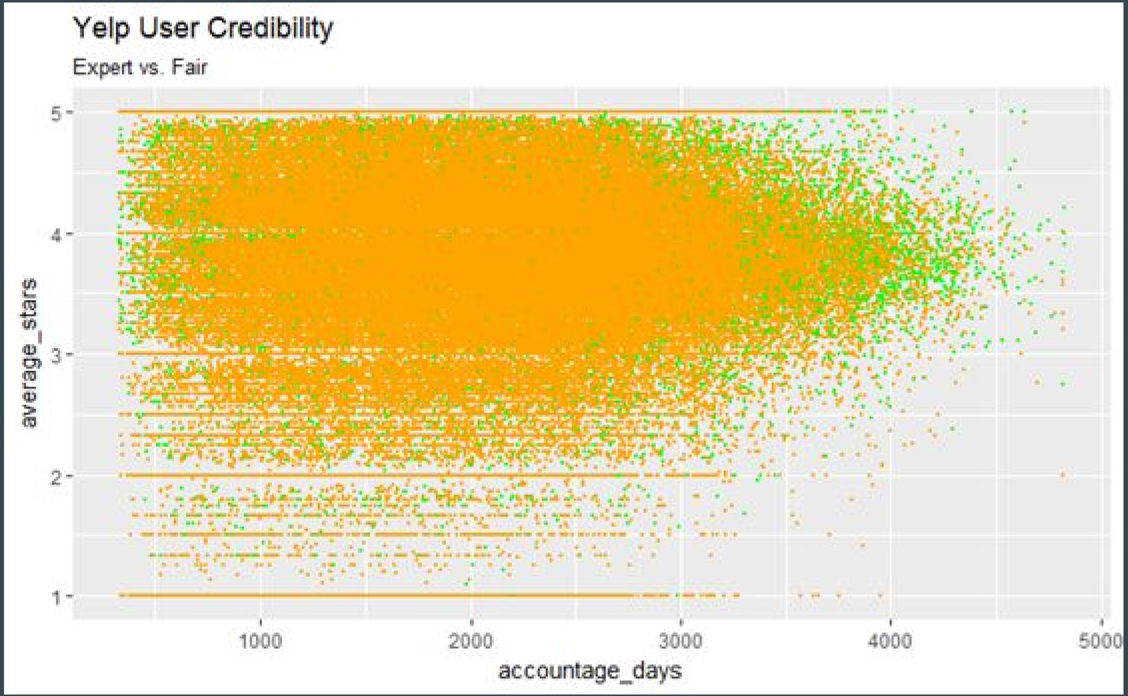


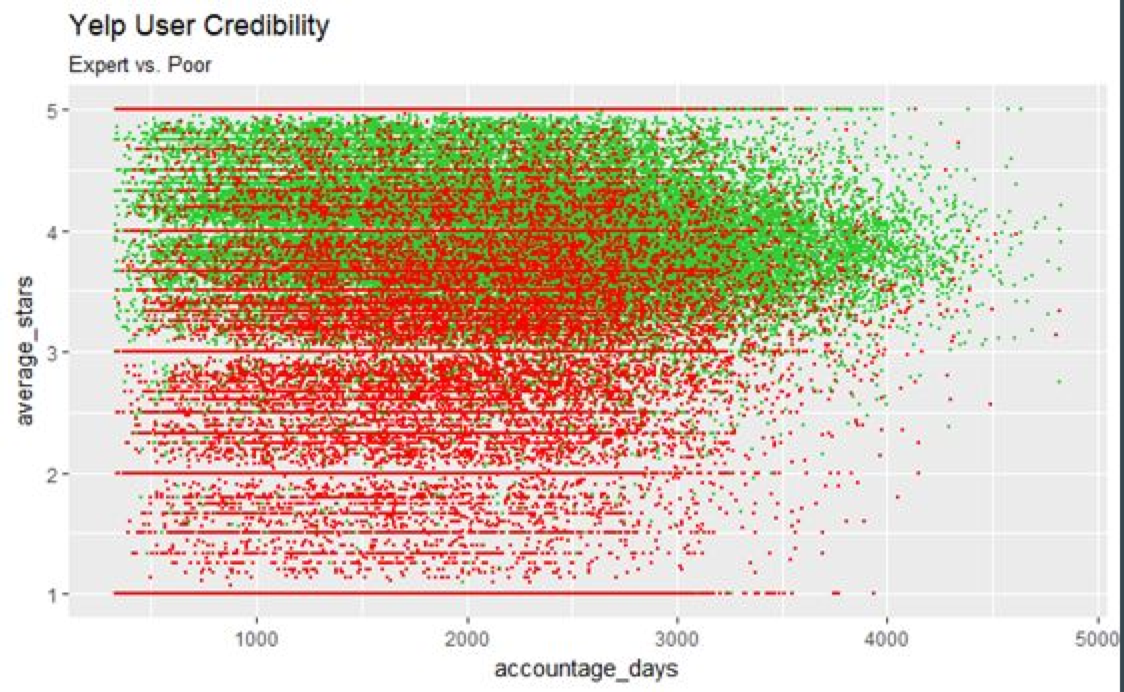
Two-Sample T-Tests for each User Class' Ratings

```
        Welch Two Sample t-test

data:  YelpUserExpert$average_stars and YelpUserFair$average_stars
t = 75.423, df = 175060, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.2643052 0.2784085
sample estimates:
mean of x mean of y 
 4.085716  3.814359 
```

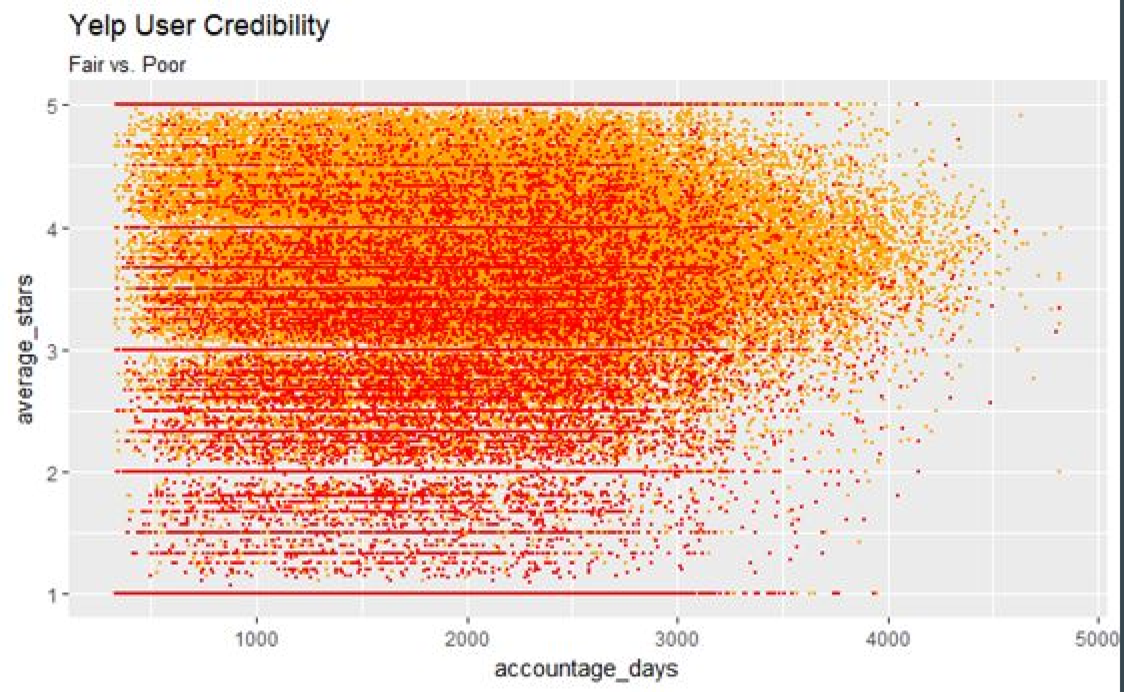
```
        Welch Two Sample t-test

data:  YelpUserExpert$average_stars and YelpUserPoor$average_stars
t = 160.95, df = 35818, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.211734 1.241611
sample estimates:
mean of x mean of y 
 4.085716  2.859044 
```

```
        Welch Two Sample t-test

data:  YelpUserFair$average_stars and YelpUserPoor$average_stars
t = 125.71, df = 35488, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.9404201 0.9702105
sample estimates:
mean of x mean of y 
 3.814359  2.859044 
```


Yelp User Credibility
Expert vs. Fair
average_stars
5
4
3
2
1
1000
2000
3000
4000
5000
accountage_days



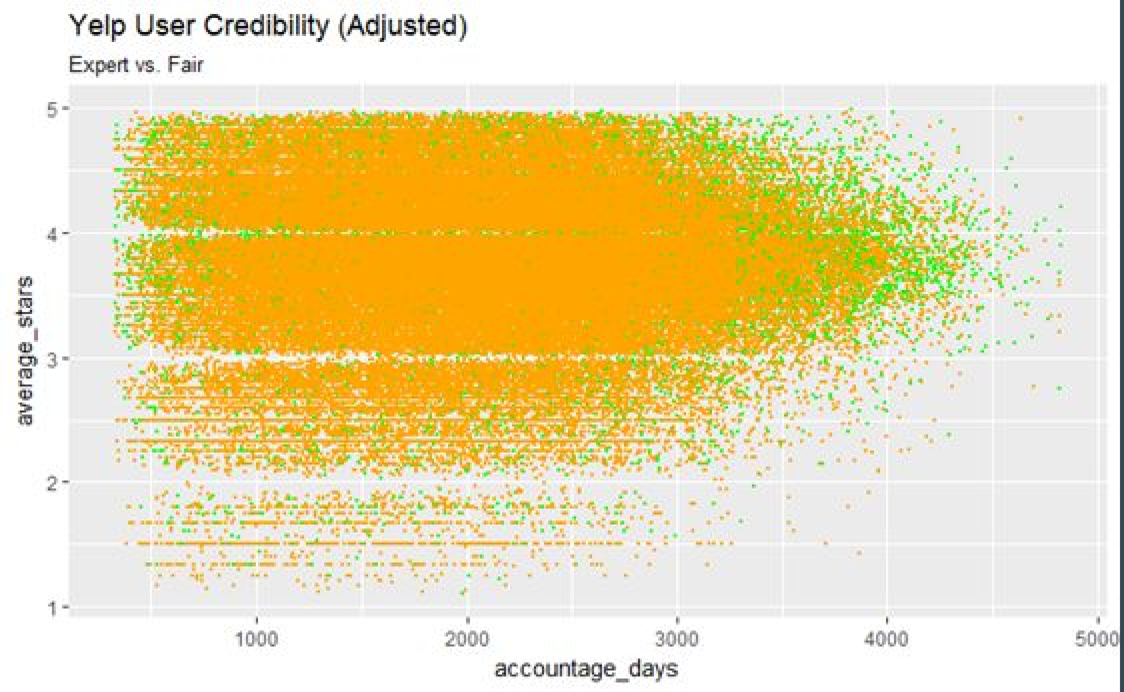
Yelp User Credibility
Expert vs. Poor
average_stars
5
4
3
2
1
1000
2000
3000
4000
5000
accountage_days



Yelp User Credibility
Fair vs. Poor
average_stars
5
4
3
2
1
1000
2000
3000
4000
5000
accountage_days

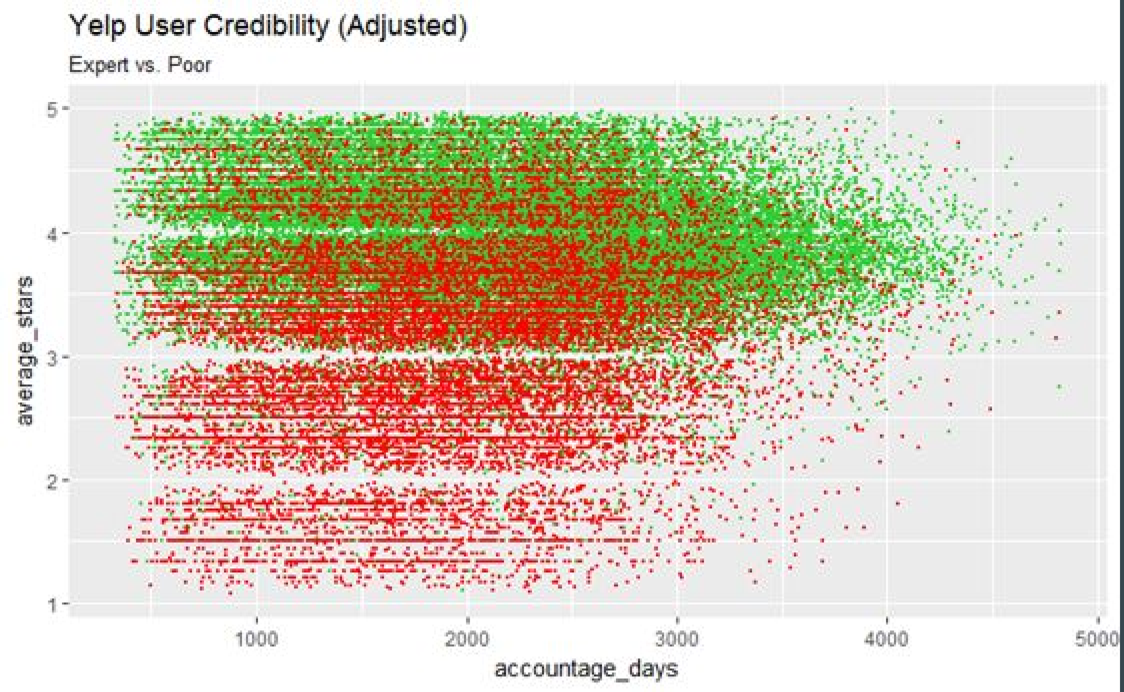
Cleaned Data

No whole # averages



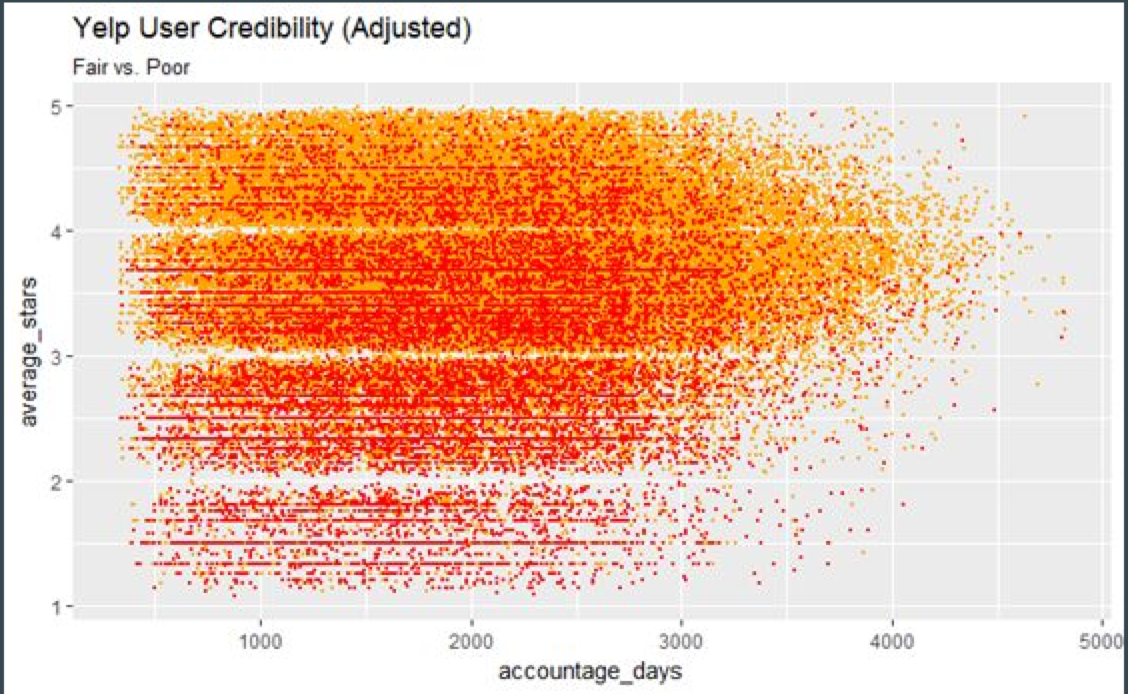
Cleaned Data

No whole # averages



Cleaned Data

No whole # averages

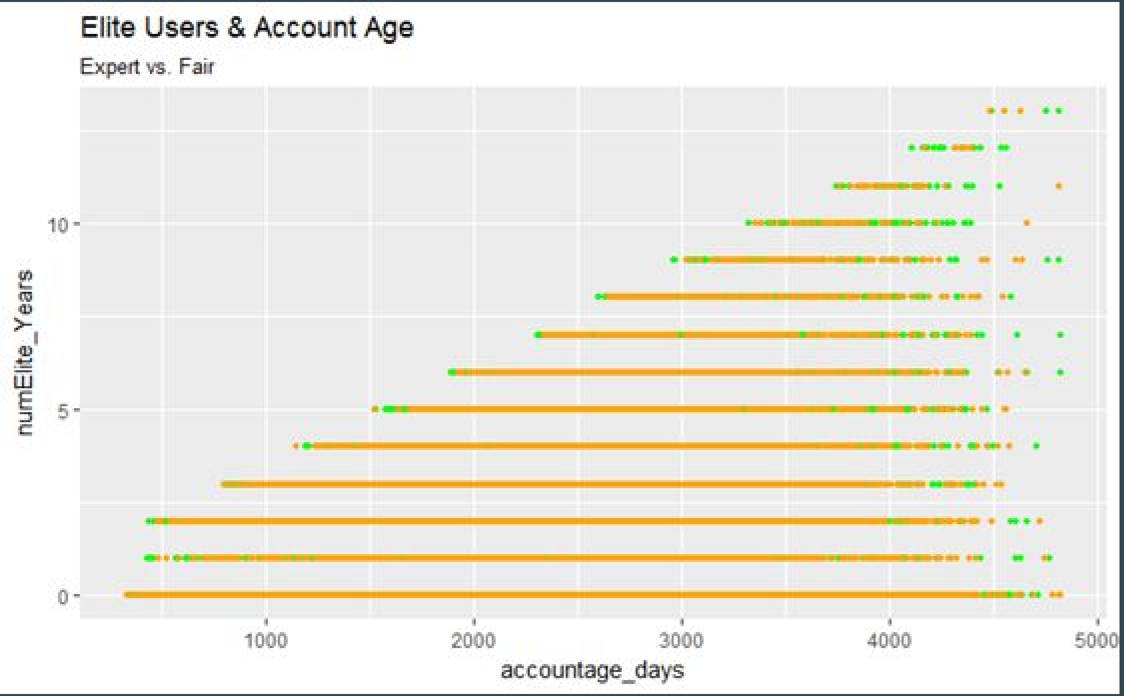


Review Accuracy and User Credibility

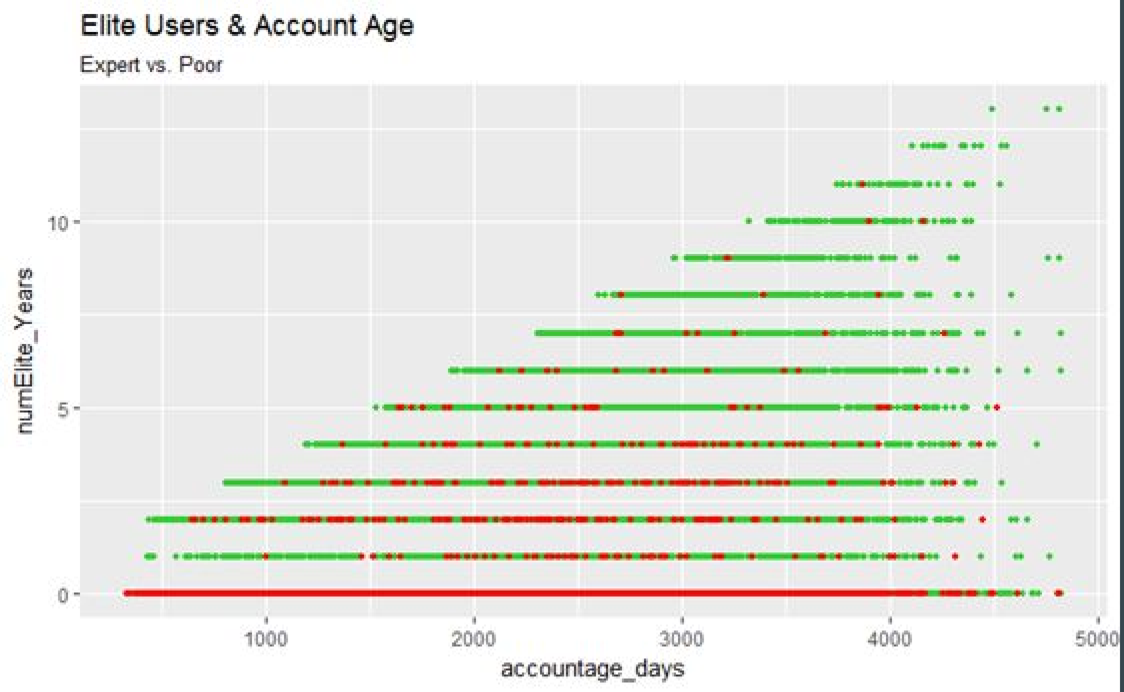
Account age in relation to Number of Elite Status Years

```
Call:
lm(formula = numElite_Years ~ accountage_days, data = YelpUserFiltered)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.5731 -0.5012 -0.2256  0.0462 11.5711 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -4.904e-01  6.383e-03  -76.82   <2e-16 ***
accountage_days  4.279e-04  3.139e-06  136.32   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.14 on 211197 degrees of freedom
Multiple R-squared:  0.08087,	Adjusted R-squared:  0.08087 
F-statistic: 1.858e+04 on 1 and 211197 DF,  p-value: < 2.2e-16
```

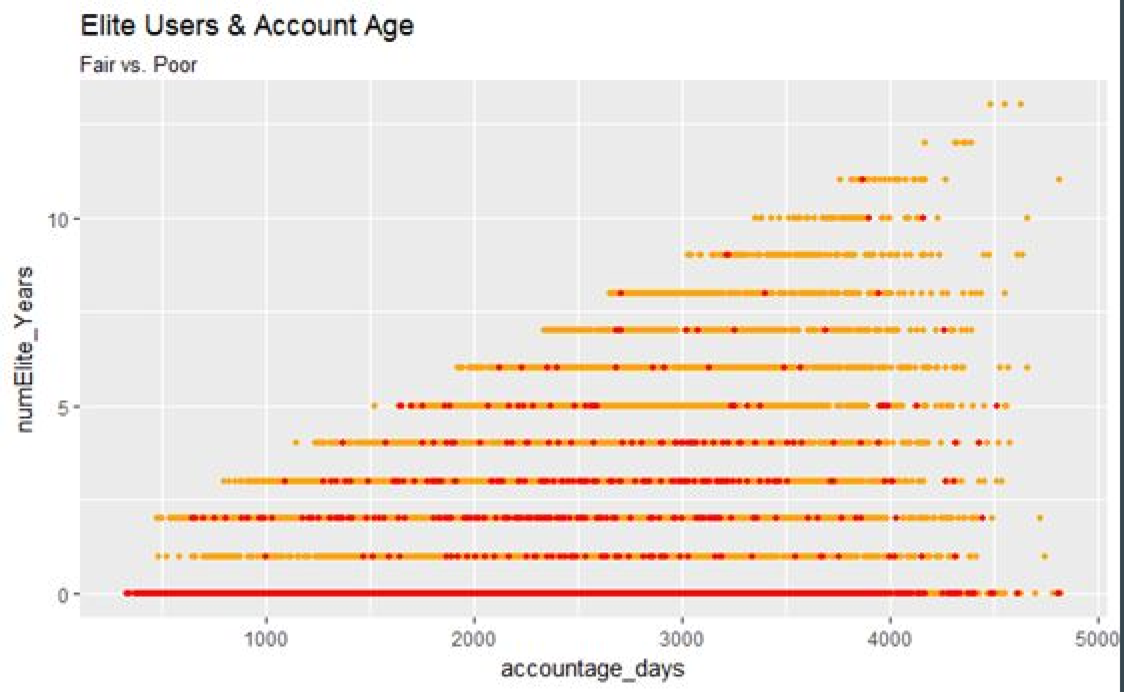
Account age in relation to Number of Elite Status Years



Account age in relation to Number of Elite Status Years



Account age in relation to Number of Elite Status Years

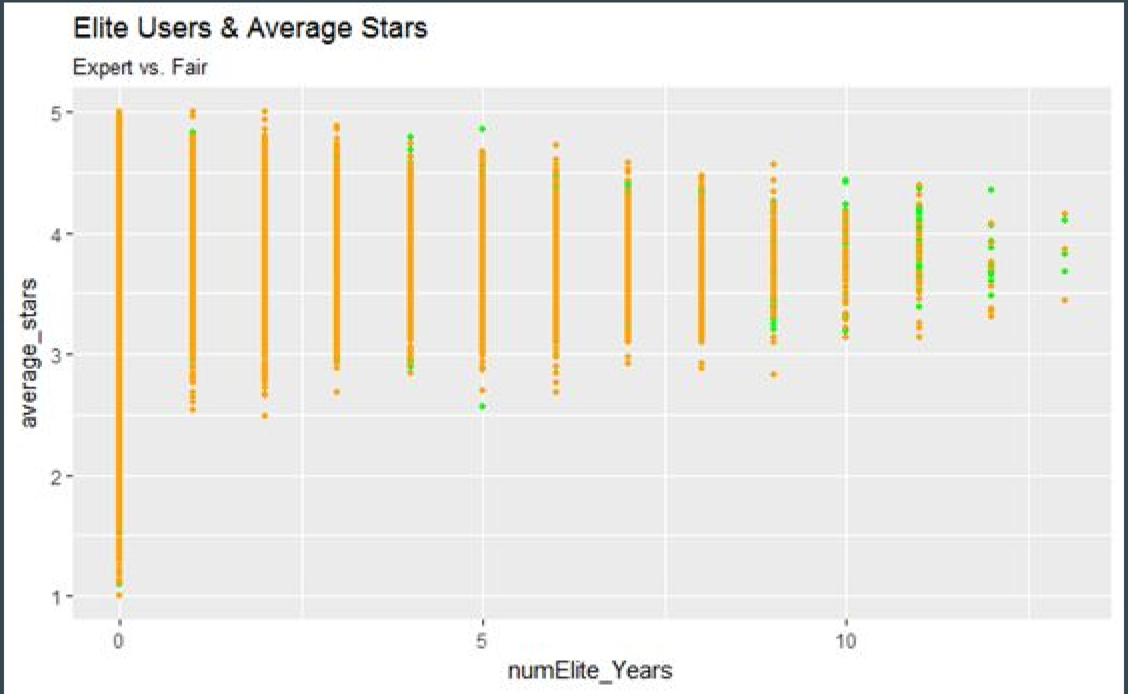


Review Accuracy and User Credibility

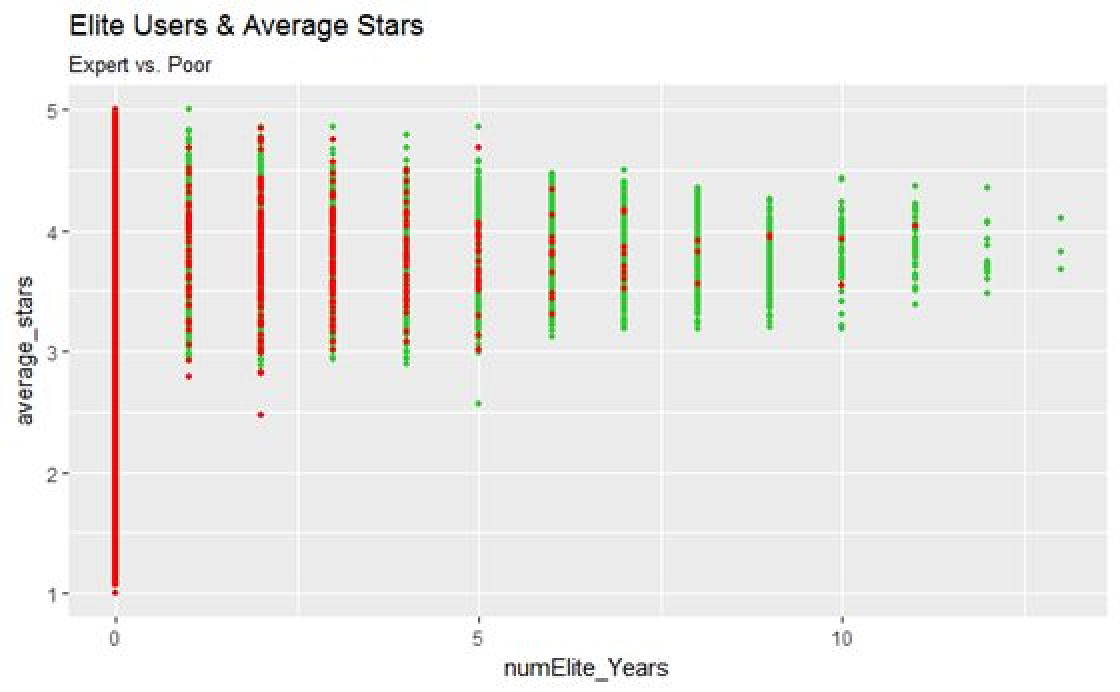
Number of Elite Status Years in relation to Average User Stars

```
Call:
lm(formula = average_stars ~ numElite_Years, data = YelpUserFiltered)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.78035 -0.45035  0.09965  0.64965  1.21965 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    3.780354   0.002094 1805.16  < 2e-16 ***
numElite_Years 0.007037   0.001704    4.13 3.62e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.931 on 211197 degrees of freedom
Multiple R-squared:  8.077e-05, Adjusted R-squared:  7.604e-05 
F-statistic: 17.06 on 1 and 211197 DF,  p-value: 3.622e-05
```

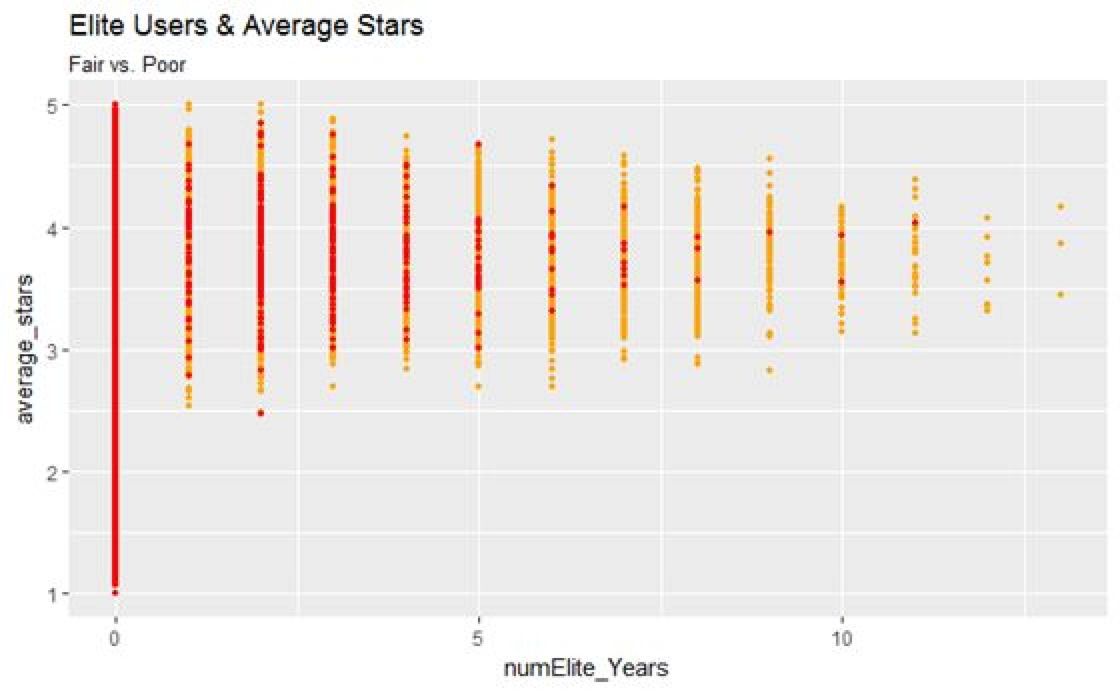
Number of Elite Status Years in relation to Average User Stars



Number of Elite Status Years in relation to Average User Stars



Number of Elite Status Years in relation to Average User Stars



Which factors influence Expert Credibility?

```
Model: binomial, link: logit

Response: as.factor(Credibility_Binary)

Terms added sequentially (first to last)


                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                            211198     274531
reviewcount        1    45.87   211197     274485 1.266e-11 ***
useful             1   200.72   211196     274285 < 2.2e-16 ***
compliment_photos  1    15.37   211195     274269 8.831e-05 ***
compliment_list    1    25.13   211194     274244 5.348e-07 ***
compliment_funny   1   146.13   211193     274098 < 2.2e-16 ***
compliment_plain   1     9.08   211192     274089  0.002582 **
fans               1   392.51   211191     273696 < 2.2e-16 ***
compliment_note    1     0.13   211190     273696  0.715972
funny              1     6.07   211189     273690  0.013724 *
compliment_writer  1     3.49   211188     273687  0.061851 .
compliment_cute    1    22.42   211187     273664 2.190e-06 ***
compliment_more    1     5.42   211186     273659  0.019876 *
compliment_hot     1     0.00   211185     273659  0.977193
cool               1     1.42   211184     273657  0.234188
compliment_profile 1     0.71   211183     273657  0.400064
compliment_cool    0     0.00   211183     273657
numElite_Years     1  1356.60   211182     272300 < 2.2e-16 ***
num_Friends        1     0.03   211181     272300  0.862831
tipcount           1     1.37   211180     272299  0.241455
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Model: binomial, link: logit

Response: as.factor(Credibility_Binary)

Terms added sequentially (first to last)


                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                            211198     274531
reviewcount        1    45.87   211197     274485 1.266e-11 ***
useful             1   200.72   211196     274285 < 2.2e-16 ***
compliment_photos  1    15.37   211195     274269 8.831e-05 ***
compliment_list    1    25.13   211194     274244 5.348e-07 ***
compliment_funny   1   146.13   211193     274098 < 2.2e-16 ***
compliment_plain   1     9.08   211192     274089  0.002582 **
fans               1   392.51   211191     273696 < 2.2e-16 ***
funny              1     6.04   211190     273690  0.013992 *
compliment_writer  1     3.48   211189     273687  0.062065 .
compliment_cute    1    22.50   211188     273664 2.100e-06 ***
compliment_more    1     5.49   211187     273659  0.019110 *
cool               1     1.43   211186     273657  0.231800
numElite_Years     1  1354.55   211185     272303 < 2.2e-16 ***
num_Friends        1     0.00   211184     272303  0.972784
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Analysis of Deviance Table
```

Analysis of Deviance Table to Compare Models

```
Model 1: as.factor(Credibility_Binary) ~ reviewcount + useful + compliment_photos +
    compliment_list + compliment_funny + compliment_plain + fans +
    compliment_note + funny + compliment_writer + compliment_cute +
    compliment_more + compliment_hot + cool + compliment_profile +
    compliment_cool + numElite_Years + num_Friends + tipcount
Model 2: as.factor(Credibility_Binary) ~ reviewcount + useful + compliment_photos +
    compliment_list + compliment_funny + compliment_plain + fans +
    funny + compliment_writer + compliment_cute + compliment_more +
    cool + numElite_Years + num_Friends
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1    211180     272299
2    211184     272303 -4   -4.161   0.3847
```

Factors that contribute whether or not a Yelper is of Expert Credibility:

Review Count, Useful, # of Fans, Funny, Compliment: Photos, List, Funny, Plain, Writer, Cute Pic, More

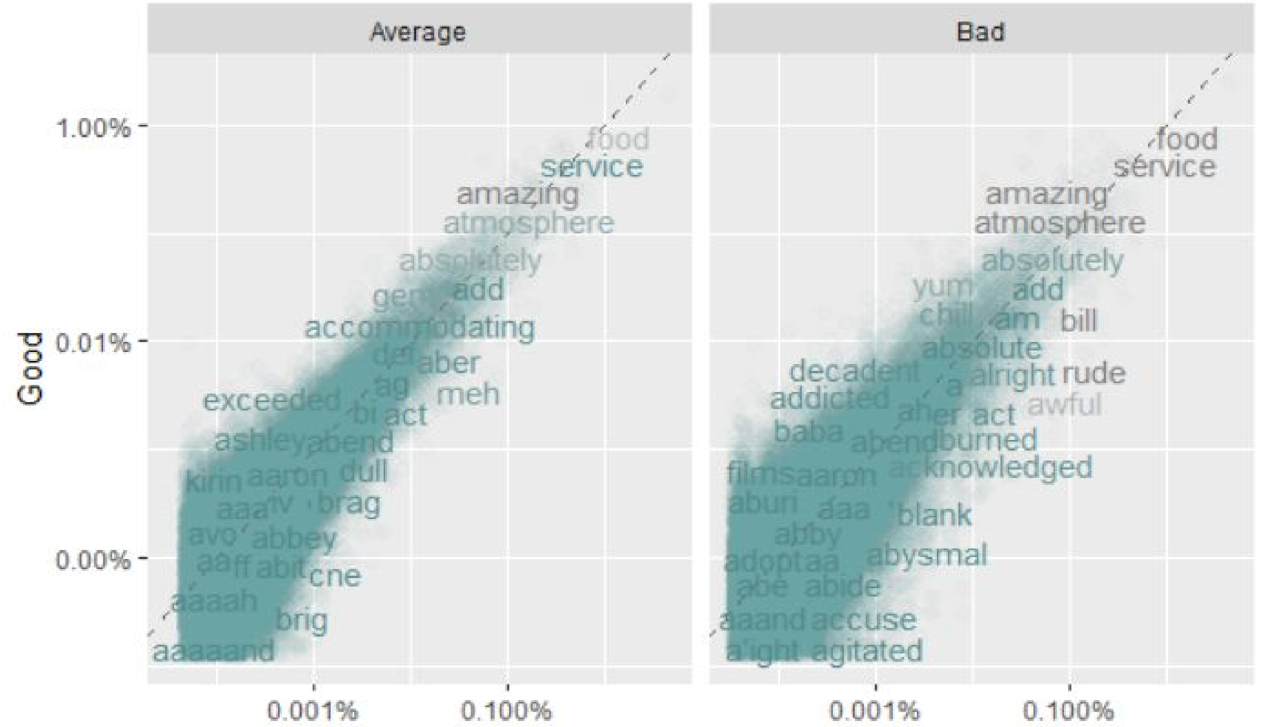
Word Cloud of Bad reviews

Word Cloud of Good Reviews

Visualization of word counts

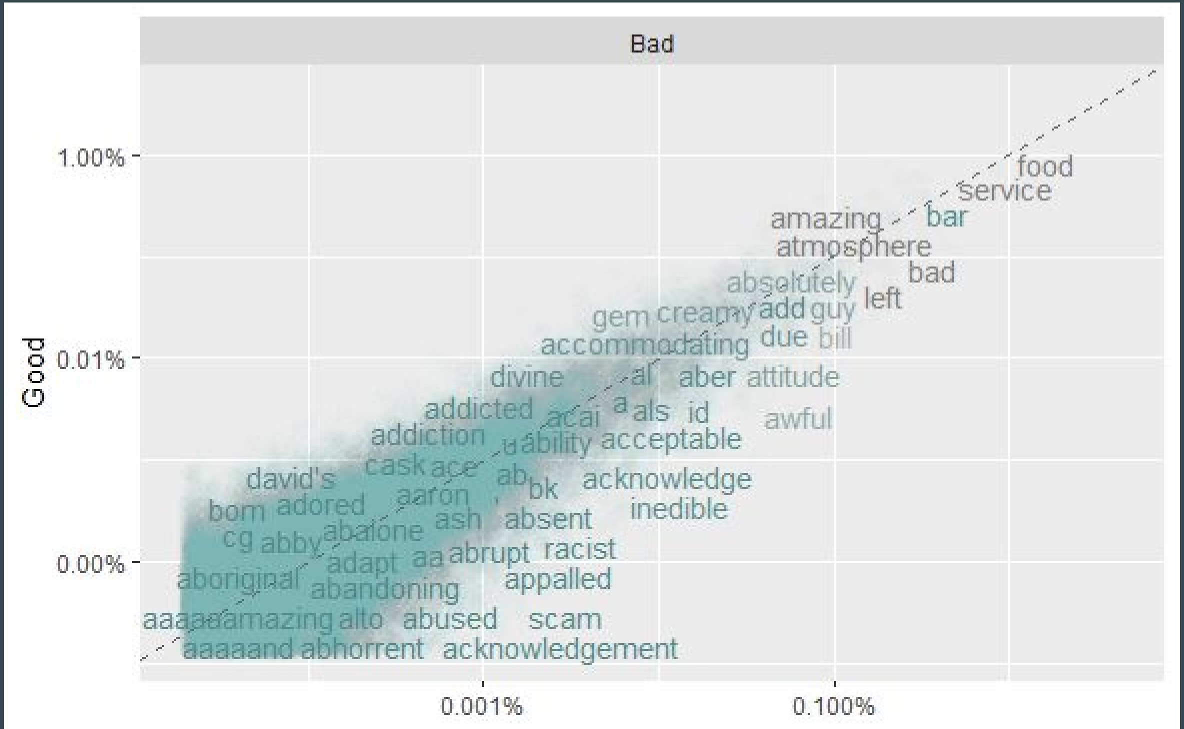
Good Reviews vs Average Reviews

Good Reviews vs Bad Reviews



What my plots looked like only cleaning data using stop words.

Visualization of word counts



Creating a classification model from the review

- Use Bag of Words to generate features from all of our review examples.

Example: only two reviews exist in our sample.

- Review 1: The hot dog looked very appealing on this day
- Review 2: The dog looked very appealing on this great day

Approach

- "Tokenize" each word of the reviews into tokens.
- Create our feature space from these tokens.
- Counts of each of these features becomes how many many times it appears in each review.

Creating a classification model from the review

- Use Bag of Words to generate features from all of our review examples.

Example: only two reviews exist in our sample.

- Review 1: The hot dog looked very appealing on this day
- Review 2: The dog looked very appealing on this great day

Our Document Term Matrix

The	hot	dog	looked	very	appealing	on	this	great	day
1	2	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1

Problem : High Dimensionality

The	hot	dog	looked	very	appealing	on	this	great	day
1	2	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1

Even a simple sentence will make us work in 10-D

Our sample we are working with has 113,369 unique words. Our feature space has 113,369 dimensions.

Solution :
Logistic Regression with L1 Penalization

Logistic Regression Code

```{r}
library(glmnet)
NFOLDS = 4
glmnet_classifier = cv.glmnet(x = dtm_train, y = train[['positive']],
                              family = 'binomial',
                              # L1 penalty
                              alpha = 1,
                              # interested in the area under ROC curve
                              type.measure = "auc",
                              nfolds = NFOLDS,
                              thresh = 1e-3,
                              maxit = 1e3)

plot(glmnet_classifier)

print(paste("max AUC =", round(max(glmnet_classifier$cvm), 4)))

```

Features: Count of each word within the bag of words
Label : TRUE(>= 4 stars) FALSE (<= 3 stars)

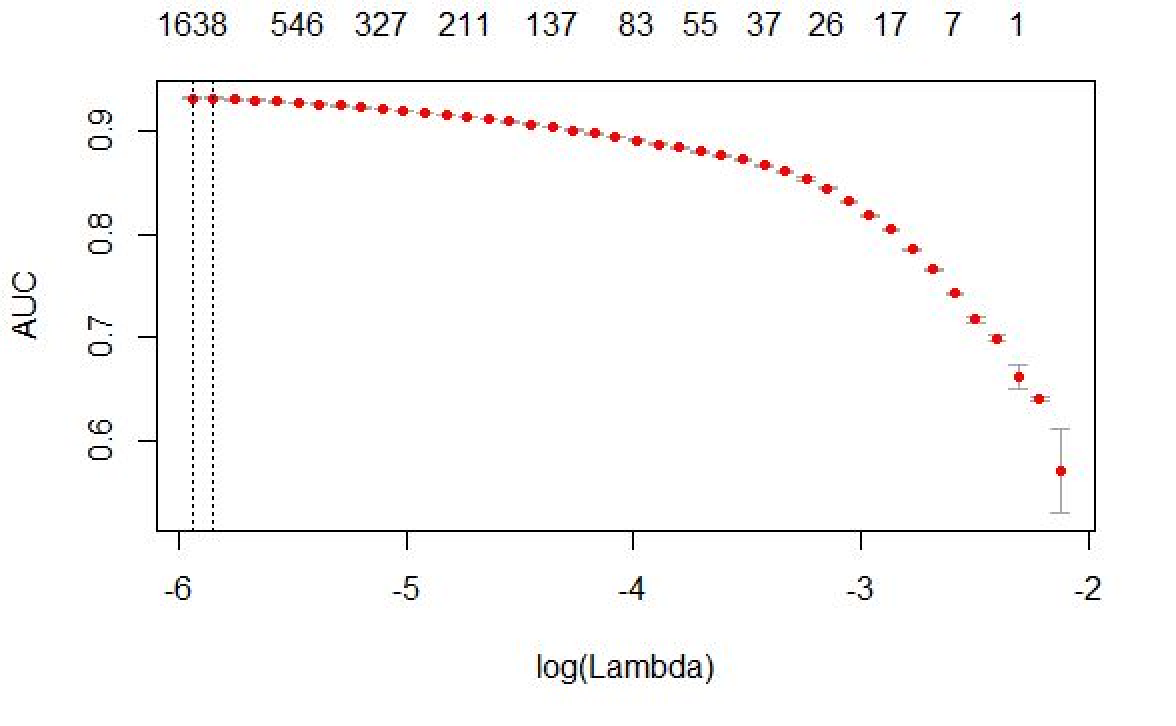
Logistic Regression ROC Curve

Max AUC

= 0.932

Test set accuracy

= 0.933




Thanks
WARNING:
YELP critic