Data Background Analysis

This report analyzes a series of tables from the Yelp database.

Yelp is a U.S.-based application that provides ratings for restaurants. The database contains a total of six datasets, covering business information, business hours, business registration, user information, review information, and tip information.

From the datasets, we can see that the data collection spans from 2004 to 2017. All businesses included in the datasets are located in the US, and the data also contains information about already closed businesses. In addition to basic information such as location, business type, and operating hours, the dataset provides yearly ratings of businesses. For users, Yelp mainly includes information related to reviews and user ratings. One dedicated table records detailed review content, ratings, and the associations between reviews, businesses, and users.

Data Interpretation

The Yelp database contains six tables: **yelp_business**, **yelp_business_hours**, **yelp_checkin**, **yelp_review**, **yelp_tip**, and **yelp_user**. The meaning of the data in each table is explained as follows:

yelp_business, a total of 174,567 records, mainly containing business basic information:

business_id	Business ID
name	Business name
neighborhood	Business neighborhood
address	Business address
city	Business city
state	Business state
postal_code	Business postal code
latitude	Business latitude
longitude	Business longitude
stars	Business rating
review_count	Number of reviews
is_open	Whether the business is still open
categories	Business categories

yelp_business_hours, a total of 174,567 records, mainly containing business opening hours:

business_id	Business ID
monday	Monday business hours
tuesday	Tuesday business hours
wednesday	Wednesday business hours
thursday	Thursday business hours
friday	Friday business hours
saturday	Saturday business hours
sunday	Sunday business hours

yelp_checkin, a total of 146,350 records, mainly containing business check-in information:

business_id	Business ID
weekday	Check-in weekday
hour	Check-in hour
checkins	Number of check-ins

yelp_review, a total of 5,261,668 records, mainly containing review information:

review_id	Review ID
user_id	User ID
business_id	Business ID
stars	Rating given to the business
date	Review date
text	Review content
useful	Number of “useful” votes
funny	Number of “funny” votes
cool	Number of “cool” votes

yelp_tip, a total of 1,098,324 records, mainly containing tip information:

text	Tip content
date	Tip date
likes	Number of likes
business_id	Business ID
user_id	User ID

yelp_user, a total of 1,326,100 records, mainly containing user information:

user_id	User ID
name	User name
review_count	Number of reviews posted by the user
yelping_since	Date the user registered
friends	Number of friends
useful	Number of “useful” votes received
funny	Number of “funny” votes received
cool	Number of “cool” votes received
fans	Number of fans
elite	Years the user was awarded “elite” status
average_stars	User’s average star rating
compliment_hot	Number of compliments for being “hot”
compliment_more	Number of compliments for “more”
compliment_profile	Unknown
compliment_cute	Number of compliments for being “cute”
compliment_list	Unknown
compliment_note	Number of compliments for notes
compliment_plain	Number of compliments for being “plain”
compliment_cool	Number of compliments for being “cool”
compliment_funny	Number of compliments for being “funny”
compliment_writer	Number of compliments for being a “writer”
compliment_photos	Number of compliments for user’s photos

Problem Setup

To analyze this online sales event and propose improvements for the next one, we examine the data from three perspectives:

1. **User perspective**

   (1) How many users are there in total? Since account creation, how many reviews does a non-elite user post per month on average, and how many does an elite user post per month on average?

   (2) How many times—and what proportions—are reviews voted useful, funny, or cool?

   (3) For reviews with many useful votes, are they more often positive or negative? What are the characteristics of their lengths?

   (4) What is the relationship between the number of fans and elite status?

   (5) What is the average user review rating? How are restaurant star ratings distributed under each review rating?

2. **Restaurant perspective**

   (1) What is the average star rating of restaurants? How many restaurants are there at each star level?

(2) How are a restaurant's star rating, number of reviews, and average review score related?

(3) Which states have the most restaurants? Which states have a higher share of five-star restaurants? How does this relate to location?

(4) For closed restaurants, what patterns exist in operating days and star ratings?

3. **Platform perspective**

(1) What is the trend of the number of registered users and number of reviews by year?

(2) What is the annual retention rate for all users?

Data Preprocessing

Before conducting data analysis, the first step is to clean the data. Each table is examined in turn:

1. **yelp_business**

First, we filter out abnormal values using the following code:

```sql
-- Select all columns from the 'yelp_business' table
SELECT *
FROM `yelp_business`
-- Filter the results where 'name' is NULL
-- or 'state' is NULL
-- or 'stars' is less than 0 or greater than 5
WHERE
    name IS NULL
    OR state IS NULL
    OR stars < 0
    OR stars > 5;
```

No results were returned, which means all values are valid:

business_id	name	neighborho...	address	city	state	postal_code	latitude	longitude	stars	review_count	is_open	categories
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. **yelp_business_hours**

We remove businesses that are closed all week (not open on any day) and businesses without operating hours information using the code below:

```
SELECT *
FROM `yelp_business_hours`
WHERE
    monday <> "None"
    OR tuesday <> "None"
    OR wednesday <> "None"
    OR thursday <> "None"
    OR friday <> "None"
    OR saturday <> "None"
    OR sunday <> "None";
```

A selection of the cleaned data results is shown below:

business_id	monday	tuesday	wednesday	thursday	friday	saturday	sunday
__1uG7MLxWGFIv2fCGPiQQ	7:0-11:0	14:0-18:0	14:0-18:0	14:0-18:0	7:0-11:0	8:0-9:0	None
__3I-DDkqM9XjLH1cJl3VA	8:0-19:0	8:0-19:0	8:0-19:0	8:0-19:0	8:0-19:0	8:0-19:0	8:0-19:0
__8j8yhsmE98wNWHJNyAgw	11:30-22:0	11:30-22:0	11:30-22:0	11:30-22:0	11:30-22:0	12:30-22:0	12:30-22:0
__aKnGBedQ51_hEc3D9ARw	6:0-21:0	6:0-21:0	6:0-21:0	6:0-21:0	6:0-22:0	6:0-22:0	7:0-20:0
__bqGGnOjtY9eEhrZAUsgA	11:0-22:0	11:0-22:0	11:0-22:0	11:0-22:0	11:0-22:0	12:0-22:0	12:0-22:0
__briK1e5l1BCEwGYijhFg	None	None	None	None	17:0-23:0	6:0-17:0	6:30-17:0
__CQ2SE4NXFFjYfrB_TJ6w	9:0-19:0	9:0-19:0	9:0-19:0	9:0-19:0	9:0-18:0	10:0-15:0	None
__D6AVR_hLpW_bott0-upA	10:0-19:0	10:0-19:0	10:0-19:0	10:0-19:0	10:0-18:0	10:0-17:0	None
__FFoyg0XmJluBBNE0QP0w	8:30-18:30	8:30-18:0	8:30-18:30	8:30-18:0	8:30-18:30	9:0-14:30	None
__fMLrmv9M1_W4kBvR2VnQ	10:30-21:0	10:30-21:0	10:30-21:0	10:30-21:0	10:30-21:30	10:30-21:30	10:30-21:0
__fyRzU8kL6HkVV3wgxfmQ	15:0-0:0	15:0-0:0	15:0-0:0	15:0-2:0	15:0-3:0	11:0-3:0	11:0-0:0

3. **yelp_chechins**

We filter out abnormal values using the code below:

```
SELECT *
FROM `yelp_checkin`
WHERE
    weekday IS NULL
    OR hour IS NULL
    OR checkins IS NULL;
```

No results were returned, which means all values are valid:

business_id	weekday	hour	checkins
NULL	NULL	NULL	NULL

4. **yelp_review**

We filter out abnormal values using the code below:

```sql
-- Select all columns from the 'yelp_review' table
SELECT *
FROM yelp_review
-- Filter the results where 'review_id' is NULL
-- or 'business_id' is NULL
-- or 'user_id' is NULL
-- or 'date' is NULL
-- or 'stars' is less than 0 or greater than 5
WHERE
    review_id IS NULL
    OR business_id IS NULL
    OR user_id IS NULL
    OR date IS NULL
    OR stars < 0
```

No results were returned, which means all values are valid:

review_id	user_id	business_id	stars	date	text	useful	funny	cool

5. **yelp_tip**

   We filter out abnormal values using the code below:

```sql
-- Select all columns from the 'yelp_tip' table
SELECT *
FROM yelp_tip
-- Filter the results where 'text' is NULL
-- or 'business_id' is NULL
-- or 'user_id' is NULL
-- or 'date' is NULL
-- or 'likes' is less than 0
WHERE
    text IS NULL
    OR business_id IS NULL
    OR user_id IS NULL
    OR date IS NULL
    OR likes < 0;
```

No results were returned, which means all values are valid:

text	date	likes	business_id	user_id

6. **yelp_user**

   We filter out abnormal values using the code below:

```sql
-- Select all columns from the 'yelp_user' table
SELECT *
FROM `yelp_user`
-- Filter the results where any of the following conditions are met:
-- 'useful' is less than 0
-- 'funny' is less than 0
-- 'cool' is less than 0
-- 'fans' is less than 0
-- 'average_stars' is less than 0
-- 'average_stars' is greater than 5
WHERE
    useful < 0
    OR funny < 0
    OR cool < 0
    OR fans < 0
    OR average_stars < 0
    OR average_stars > 5;
```

No results were returned, which means all values are valid:

user_id	name	review_count	yelping_since	friends	useful	funny	cool	fans	elite	average_stars	compliment_...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Next, remove user records without name, reviews, or registration information, and generate a new cleaned table using the following code:

```sql
-- Select all columns from the 'yelp_user' table
-- Where 'name', 'review_count', and 'yelping_since' are not NULL
SELECT *
FROM `yelp_user`
WHERE
    name IS NOT NULL
    AND review_count IS NOT NULL
    AND yelping_since IS NOT NULL;
```

A selection of the cleaned table is shown below:

user_id	name	review_count	yelping_since	friends	useful	funny	cool
___fEWlObjtPaZ-pK0eq9g	Nerissa	7	2016-05-09	Ngwaot7XkD4g75hHBY3wnQ	0	0	0
___MTsBloH4jvybJ5DrTYw	TJ	10	2011-03-25	KN6I2UWhB-N2nXtaeLgNsg, R3ZzMoqN3FibA...	3	0	0
__-8WEg7xD0CwCDu04MzBA	Maria	23	2010-05-03	None	2	0	0
__-C_dSG8Ky98M6EVPYehA	Craig	8	2012-09-20	None	1	0	0
__-FcRiBzu8tqWYGofto1Q	Sonia	102	2015-09-28	NCkOlje6qX0aYjJTyejTiQ, HnqSBHIR4KFICk88...	7	4	168
__-Kt26YrtJxGdWs8FqKCg	Shane	7	2012-08-06	NkTDQLctnUSo5BRo_FVrig, j-iOqnQUOpk5HO...	0	0	0
__02xlXHMOZda_nPoBTnoQ	Richard	1	2014-06-30	8Vx-Bn3SuQZiUK9XhrPrng, 5xrHnV-8-TotHbT-...	0	0	0

Data Analysis

User Perspective

(1) How many users are there in total? Since account creation, how many reviews does a non-elite user post per month on average, and how many does an elite user post per month on average?

We compute the total number of users on Yelp using the following code:

```
SELECT COUNT(*) AS total_users FROM yelp_user;
```

The result shows that Yelp had 663050 users in total.

total_users
663050

Next, we find out the final date of the data statistics. Here, the latest registration time of the user is indicated:

```
SELECT DISTINCT(yelping_since)
FROM yelp_user
ORDER BY yelping_since DESC
LIMIT 1;
```

The result shows that the last date of the data statistics is December 11, 2017:

yelping_since
2017-12-11

Then count the average number of reviews per day for non-elite users using the following code:

```
SELECT AVG(sub.avg_review_day) AS avg_review_day_non_elite
FROM (
    SELECT user_id,
           DATEDIFF("2017-12-11", yelping_since) AS register_days,
           review_count / DATEDIFF("2017-12-11", yelping_since) AS avg_review_day
    FROM yelp_user
    WHERE elite = "None"
) sub;
```

The result is the following:

avg_review_day_non_elite
0.00961322

Thus, the average number of reviews posted by non-elite users per day is 0.00961 (approximately equal to 0.01 per day or 1 every hundred days).

The number of reviews posted by non-elite users each day is calculated in the same way using the code below:

```
SELECT AVG(sub.avg_review_day) AS avg_review_day_elite
FROM (
    SELECT user_id,
           DATEDIFF("2017-12-11", yelping_since) AS register_days,
           review_count / DATEDIFF("2017-12-11", yelping_since) AS avg_review_day
    FROM yelp_user
    WHERE elite <> "None"
) sub;
```

Thus, the average number of reviews posted by elite users per day is 0.0973, which is approximately 10.1 times that of non-elite users:

avg_review_day_elite
0.09725806

(2) How many times—and what proportions—are reviews voted useful, funny, or cool?

We count the number of 'useful', 'funny', and 'cool' votes non-elite users' comments received, the following code is used:

```
SELECT
    SUM(review_count) AS total_review_count,
    SUM(useful) / SUM(review_count) AS useful_rate,
    SUM(funny) / SUM(review_count) AS funny_rate,
    SUM(cool) / SUM(review_count) AS cool_rate
FROM yelp_user
WHERE elite = "None";
```

It was found that the numbers of 'useful', 'funny', and 'cool' votes are 0.58, 0.20, and 0.20 times the total number of votes, respectively:

total_review_count	useful_rate	funny_rate	cool_rate
8427347	0.5783	0.1977	0.1961

Next, we calculate the number of 'useful', 'funny', and 'cool' votes elite users' comments received in the same way:

```
SELECT
    SUM(review_count) AS total_review_count,
    SUM(useful) / SUM(review_count) AS useful_rate,
    SUM(funny) / SUM(review_count) AS funny_rate,
    SUM(cool) / SUM(review_count) AS cool_rate
FROM yelp_user
WHERE elite <> "None";
```

The result is as follows:

total_review_count	useful_rate	funny_rate	cool_rate
6861501	2.0706	1.1463	1.6270

Therefore, the numbers of 'useful', 'funny', and 'cool' votes are 2.07, 1.15 and 1.63 times that of the reviews they sent, and 3.58, 5.80 and 8.30 times that of non-elite users respectively. Elite users have visited far more restaurants and posted more reviews than non-elite users, and the probability of their reviews being praised is also higher. Generally, their reviews are better.

At the same time, it was found that regardless of the type of user, the probability of the review being voted as 'useful' is much higher than that of being given as 'funny' and 'cool', which also indicates that when users write reviews, rather than considering language style and word choice, more attention is paid to the content. When other users are looking at the reviews, they will also pay more attention to whether the reviews are useful.

(3) For reviews with many useful votes, are they more often positive or negative? What are the characteristics of their lengths?

We compute, separately, the number of reviews that received more than 1000, 500, and 200 'useful' votes, the average customer rating assigned to the restaurants in these reviews, as well as the average length of these reviews. The code and results are shown below:

```
SELECT
    COUNT(sub.text) as review_count,
    ROUND(AVG(sub.stars), 1) as avg_stars,
    ROUND(AVG(sub.len), 0) as avg_length
FROM (
    SELECT stars, text, LENGTH(text) as len, useful
    FROM yelp_review
    WHERE useful > 1000
) as sub;
```

There are 11 reviews with more than 1000 'useful' votes, giving restaurants an average rating of 1.3 stars, with an average length of 2204 characters:

review_count	avg_stars	avg_length
11	1.3	2204

For reviews with more than 500 'useful' votes, the code is as follows:

```
SELECT
    COUNT(sub.text) as review_count,
    ROUND(AVG(sub.stars), 1) as avg_stars,
    ROUND(AVG(sub.len), 0) as avg_length
FROM (
    -- Subquery to select relevant data from yelp_review
    SELECT stars, text, LENGTH(text) as len, useful
    FROM yelp_review
    WHERE useful > 500
) as sub;
```

There are 38 reviews with more than 500 'useful' votes, giving restaurants an average rating of 1.3 stars, with an average length of 1456 characters:

review_count	avg_stars	avg_length
38	1.3	1456

For reviews with more than 200 'useful' votes, the code is as follows:

```
SELECT
    COUNT(sub.text) as review_count,
    ROUND(AVG(sub.stars), 1) as avg_stars,
    ROUND(AVG(sub.len), 0) as avg_length
FROM (
    -- Subquery to select relevant data from yelp_review
    SELECT stars, text, LENGTH(text) as len, useful
    FROM yelp_review
    WHERE useful > 200
) as sub;
```

There are 138 reviews with more than 200 'useful' votes, giving restaurants an average rating of 1.8 stars, with an average length of 1342 characters:

review_count	avg_stars	avg_length
138	1.8	1342

From the above data, it can be seen that the length of the review is likely positively correlated with the number of 'useful' votes. Moreover, the higher the number of 'useful' votes a review receives, the lower its average restaurant rating tends to be. This suggests that compared to positive reviews, people often regard negative reviews as providing more valuable information about restaurants.

(4) What is the relationship between the number of fans and elite status?

First, we calculate the number of followers of non-elite users and elite users using the following code:

```
SELECT 'Average Users' AS user_type, SUM(fans)/COUNT(*) AS avg_fans
FROM yelp_user WHERE elite = "None"
UNION
SELECT 'Elite Users', SUM(fans)/COUNT(*)
FROM yelp_user WHERE elite <> "None";
```

The results are as follows:

user_type	avg_fans
Average Users	0.4747
Elite Users	21.4171

The average number of followers for each elite user is approximately 21, which is more than 45 times that of non-elite users. It can then be suspected that when Yelp selects elite users, the number of their followers can be an important indicator for reference. This is because the number of followers not only indicates that the user is highly dependent on the platform and provides more effective information, but also brings more traffic, which is exactly what the platform relies on to survive.

(5) What is the average user review rating? How are restaurant star ratings distributed under each review rating?

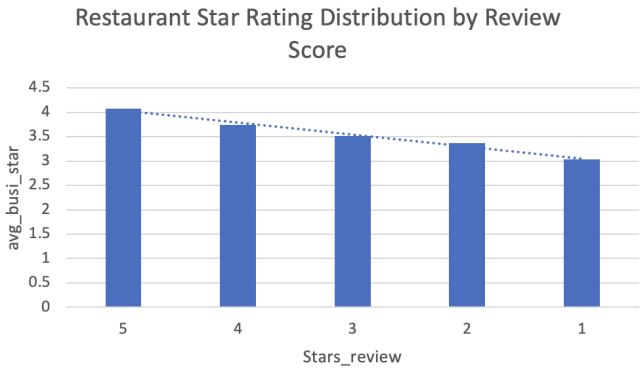
For the restaurants that users gave different ratings to, we calculate the average star ratings of these restaurants respectively. The codes are as follows:

```
SELECT yelp_review.stars,
       ROUND(AVG(yelp_business.stars), 2) as avg_busi_star
FROM yelp_review
LEFT JOIN yelp_business
ON yelp_review.business_id = yelp_business.business_id
GROUP BY yelp_review.stars
ORDER BY yelp_review.stars DESC;
```

The results are as follows: The first column stands for the user ratings, and the second stands for the average star rating of the restaurant under that rating:

stars	avg_busi_star
5	4.08
4	3.74
3	3.52
2	3.37
1	3.03

With user ratings as the horizontal axis and the overall average star rating of restaurants as the vertical axis, the chart was made as below:



It can be seen that the higher the star rating given by users, the higher the average rating of the restaurant, and it is approximately evenly distributed.

Restaurant Perspective

(1) What is the average star rating of restaurants? How many restaurants are there at each star level?

We compute the average ratings of all restaurants using the code below:

```
SELECT round(avg(stars),1) FROM yelp_business;
```

The result shows that the average rating of all restaurants is 3.6 stars:

round(avg(stars),1)
3.6

Since the rating range is from 1 to 5, with 0.5 as each rating, if the rating of a restaurant is roughly in a normal distribution, the average rating should be around 3 stars. The data shows that the average value is above 3, which indicates that most restaurants have received a relatively higher rating of 3 or above. This shows that the platform is not inclined to give restaurants a low rating.

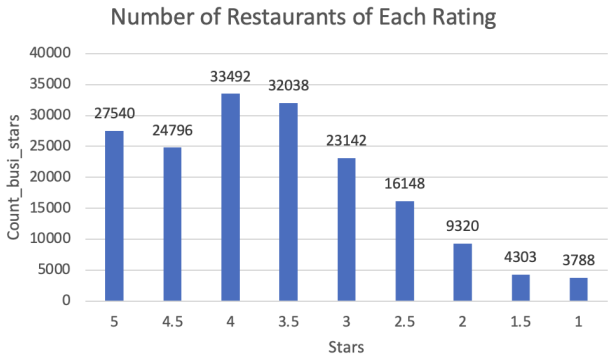
Next, we compute the number of restaurants of each rating:

```
SELECT stars, count(*) FROM yelp_business GROUP BY stars ORDER BY stars DESC;
```

The results are as follows: the first column stands for the star rating of the restaurant, and the second stands for the corresponding number of restaurants:

stars	count(*)
5.0	27540
4.5	24796
4.0	33492
3.5	32038
3.0	23142
2.5	16148
2.0	9320
1.5	4303
1.0	3788

With star ratings as the horizontal axis and the number of restaurants as the vertical axis, the chart was drawn as below:



This histogram roughly shows a right-skewed distribution, with more high-rating restaurants than low-rating ones, which is consistent with the information presented by the average rating. It can be seen that the number of restaurants with a score of 4.0 is the largest, followed closely by those with a score of 3.5. Most restaurants have achieved a level of customer satisfaction to a large extent.

It is worth noting that the number of 5-star restaurants is also significant, even exceeding the number of 4.5-star restaurants. On the one hand, users' ratings can affect a restaurant's star rating. When users have an excellent experience in a restaurant, they tend to give it a full score subjectively to show appreciation and pay less attention to the restaurant's shortcomings. In such cases, the restaurant's shortcomings are usually regarded as 'a minor flaw does not overshadow the overall quality' or 'distinctive features'. On the other hand, the platform is more inclined to promote 5-star restaurants than 4.5-star ones. This is beneficial for the platform to advertise the restaurants, as a full score is a powerful 'gimmick' in user promotion compared to 4.5-star ones. At the same time, restaurants promoted by the platform can also give back to the platform by offering exclusive discounts on Yelp, bringing a large amount of traffic to both the restaurants themselves and the platform.

(2) How does a restaurant's star rating relate to the number of reviews and average review rating?

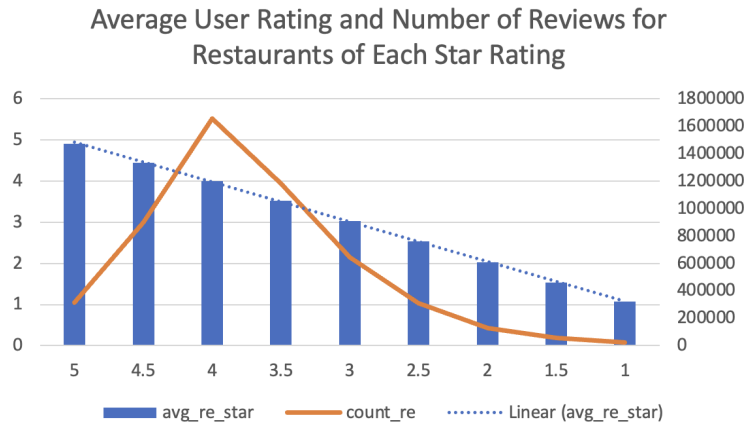
For the restaurants of each rating, we compute the average rating of user reviews and the number of reviews using the following code:

```
SELECT yelp_business.stars,
       ROUND(AVG(yelp_review.stars), 2) as avg_re_star,
       COUNT(review_id) as count_re
FROM yelp_business
RIGHT JOIN yelp_review
ON yelp_business.business_id = yelp_review.business_id
GROUP BY yelp_business.stars
ORDER BY yelp_business.stars DESC;
```

The results are as follows: the first column stands for the star rating of the restaurant, the second stands for the average rating of users, and the last column stands for the number of user reviews:

stars	avg_re_star	count_re
5.0	4.91	312992
4.5	4.45	901029
4.0	4.00	1654345
3.5	3.52	1181787
3.0	3.03	644282
2.5	2.53	309309
2.0	2.03	129061
1.5	1.54	58144
1.0	1.07	21859

With the restaurant's star rating as the horizontal axis and the average user rating and the number of user reviews as the vertical axes respectively, the combo chart was drawn as below:



It can be seen from the trend line that the average user rating decreases with the decrease of the restaurant's star rating, and it is roughly evenly distributed.

On the other hand, the trend curve of the number of user comments varying with the star rating of the restaurant roughly shows a right-skewed distribution, and reaches its peak at 4.0-star restaurants. This is partly because there are the most 4.0-star restaurants and they are the easiest to visit. On the other hand, it also indicates that 4.0-star restaurants can already meet the daily dining, restaurant exploration and social needs of most users. It can also be seen from the chart that the number of reviews for 5-star restaurants is relatively small, especially compared to the large base of 5-star restaurants. Based on the above analysis, this further indicates that the platform plays a significant role in the selection of 5-star restaurants. This might also be due to hard factors such as the generally high prices and limited capacity of five-star restaurants to accommodate people.

(3) Which states have the most restaurants? Which states have a higher share of five-star restaurants? How does this relate to location?

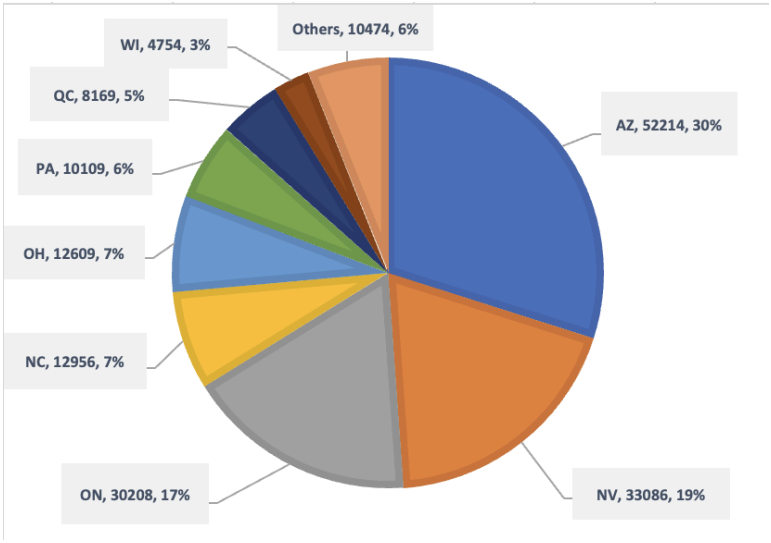
We compute the total number of restaurants in each state using the following code:

```
SELECT state, count(state)
FROM yelp_business
GROUP BY state
ORDER BY count(state) DESC
```

A selection of the results from highest to lowest is shown below:

state	count(state)
AZ	52214
NV	33086
ON	30208
NC	12956
OH	12609
PA	10109
QC	8169
WI	4754
EDH	3795
BW	3118
IL	1852
SC	679
MLN	208

Next, we plot the results as a pie chart:



The results are generally consistent with the states most frequently reviewed by users. The top two states account for 49% of all reviews: Arizona at 30% and Nevada at 19%. Next in proportion are North Carolina, Ohio, Pennsylvania, and others, which also align with the number of user reviews. For restaurants, establishing a presence in these states means potentially gaining more user attention, but also facing stronger competition.

Next, we calculate the proportion of restaurants at different star levels in each state. The code is as follows:

```sql
-- Calculate statistics related to star ratings for businesses grouped by state.
SELECT
    state,
    COUNT(state) as count_state,
    SUM(CASE stars WHEN 5.0 THEN 1 ELSE 0 END) as sum_5,
    (SUM(CASE stars WHEN 5.0 THEN 1 ELSE 0 END) / COUNT(state)) as rate_5,
    SUM(CASE stars WHEN 4.5 THEN 1 ELSE 0 END) as sum_4_5,
    (SUM(CASE stars WHEN 4.5 THEN 1 ELSE 0 END) / COUNT(state)) as rate_4_5,
    SUM(CASE stars WHEN 4.0 THEN 1 ELSE 0 END) as sum_4,
    (SUM(CASE stars WHEN 4.0 THEN 1 ELSE 0 END) / COUNT(state)) as rate_4,
    SUM(CASE stars WHEN 3.5 THEN 1 ELSE 0 END) as sum_3_5,
    (SUM(CASE stars WHEN 3.5 THEN 1 ELSE 0 END) / COUNT(state)) as rate_3_5,
    SUM(CASE stars WHEN 3.0 THEN 1 ELSE 0 END) as sum_3,
    (SUM(CASE stars WHEN 3.0 THEN 1 ELSE 0 END) / COUNT(state)) as rate_3,
    SUM(CASE stars WHEN 2.5 THEN 1 ELSE 0 END) as sum_2_5,
    (SUM(CASE stars WHEN 2.5 THEN 1 ELSE 0 END) / COUNT(state)) as rate_2_5,
    SUM(CASE stars WHEN 2.0 THEN 1 ELSE 0 END) as sum_2,
    (SUM(CASE stars WHEN 2.0 THEN 1 ELSE 0 END) / COUNT(state)) as rate_2,
    SUM(CASE stars WHEN 1.5 THEN 1 ELSE 0 END) as sum_1_5,
    (SUM(CASE stars WHEN 1.5 THEN 1 ELSE 0 END) / COUNT(state)) as rate_1_5,
    SUM(CASE stars WHEN 1.0 THEN 1 ELSE 0 END) as sum_1,
    (SUM(CASE stars WHEN 1.0 THEN 1 ELSE 0 END) / COUNT(state)) as rate_1
```

```
FROM
    yelp_business
GROUP BY
    state
ORDER BY
    count_state DESC;
```

The results are shown below, listing the number of restaurants by state, the number of restaurants at each star level, and their proportions:

state	count_sta...	sum_5	rate_5	sum_4_5	rate_4_5	sum_4	rate_4	sum_3_5	rate_3_5	sum_3	rate_3	sum_2_5	rate_2_5	sum_2	rate_2	sum_1_5	rate_1_5	sum_1	rate_1
AZ	52214	11698	0.2240	7441	0.1425	8928	0.1710	8207	0.1572	5995	0.1148	4632	0.0887	2765	0.0530	1275	0.0244	1273	0.0244
NV	33086	6631	0.2004	4924	0.1488	5978	0.1807	5426	0.1640	4040	0.1221	2827	0.0854	1790	0.0541	783	0.0237	687	0.0208
ON	30208	2219	0.0735	3268	0.1082	5954	0.1971	6724	0.2226	5258	0.1741	3332	0.1103	1869	0.0619	860	0.0285	724	0.0240
NC	12956	1893	0.1461	1706	0.1317	2422	0.1869	2445	0.1887	1767	0.1364	1313	0.1013	733	0.0566	368	0.0284	309	0.0238
OH	12609	1578	0.1251	1676	0.1329	2473	0.1961	2474	0.1962	1711	0.1357	1284	0.1018	750	0.0595	343	0.0272	320	0.0254
PA	10109	1310	0.1296	1561	0.1544	1934	0.1913	2011	0.1989	1372	0.1357	952	0.0942	534	0.0528	234	0.0231	201	0.0199
QC	8169	542	0.0663	1579	0.1933	2128	0.2605	1687	0.2065	1064	0.1302	621	0.0760	304	0.0372	161	0.0197	83	0.0102
WI	4754	673	0.1416	699	0.1470	993	0.2089	923	0.1942	570	0.1199	464	0.0976	226	0.0475	120	0.0252	86	0.0181
EDH	3795	229	0.0603	745	0.1963	1230	0.3241	836	0.2203	435	0.1146	198	0.0522	78	0.0206	33	0.0087	11	0.0029
BW	3118	330	0.1058	654	0.2097	799	0.2563	614	0.1969	421	0.1350	186	0.0597	78	0.0250	19	0.0061	17	0.0055
IL	1852	222	0.1199	265	0.1431	313	0.1690	338	0.1825	279	0.1506	193	0.1042	123	0.0664	70	0.0378	49	0.0265
SC	679	132	0.1944	85	0.1252	115	0.1694	111	0.1635	103	0.1517	64	0.0943	32	0.0471	19	0.0280	18	0.0265
MLN	208	12	0.0577	41	0.1971	58	0.2788	55	0.2644	16	0.0769	10	0.0481	9	0.0433	5	0.0240	2	0.0096
HLD	179	9	0.0503	31	0.1732	43	0.2402	42	0.2346	32	0.1788	16	0.0894	4	0.0223	0	0.0000	2	0.0112
NYK	152	17	0.1118	28	0.1842	36	0.2368	30	0.1974	20	0.1316	7	0.0461	6	0.0395	6	0.0395	2	0.0132
CHE	143	10	0.0699	27	0.1888	29	0.2028	26	0.1818	23	0.1608	20	0.1399	6	0.0420	2	0.0140	0	0.0000

From the chart, we can see that no state has a particularly high proportion of low-star restaurants. Apart from five-star restaurants, the proportions of other star levels fluctuate only slightly across states and are essentially unrelated to the total number of restaurants in a given state. Arizona and Nevada not only have more restaurants overall but also a higher proportion of five-star restaurants. This suggests that the selection of five-star restaurants is influenced by the state in which the business is located.

Arizona and Nevada are both located in the southwestern United States, with relatively large land areas, adjacent to California and Hawaii, while Yelp's headquarters is in California. Their coastal location, influenced by population flows from California, together with their proximity to Yelp's headquarters, may have contributed to the platform focusing on developing its business in these two states. As a result, they have more restaurants registered on Yelp and a higher number of five-star restaurants.

(4) For closed restaurants, what patterns exist in operating days and star ratings?

We compute the average weekly operating days and average star rating of the closed restaurants with the following codes:

```sql
SELECT ROUND(AVG(open_days.open_days), 1) as avg_open,
       ROUND(AVG(yelp_business.stars), 2) as avg_stars
FROM yelp_business
LEFT JOIN open_days
ON yelp_business.business_id = open_days.business_id
WHERE yelp_business.is_open = 0;
```

The results show that the average business days per week are approximately 4 days, and the average star rating of the restaurant is about 3.5:

avg_open	avg_stars
3.9	3.51

Both data is relatively low, which can then be inferred that the closure of a restaurant is slightly related to the number of business days and the average star rating, with the three factors influencing each other.

Platform

(1) What is the trend of the number of registered users and number of reviews by year?

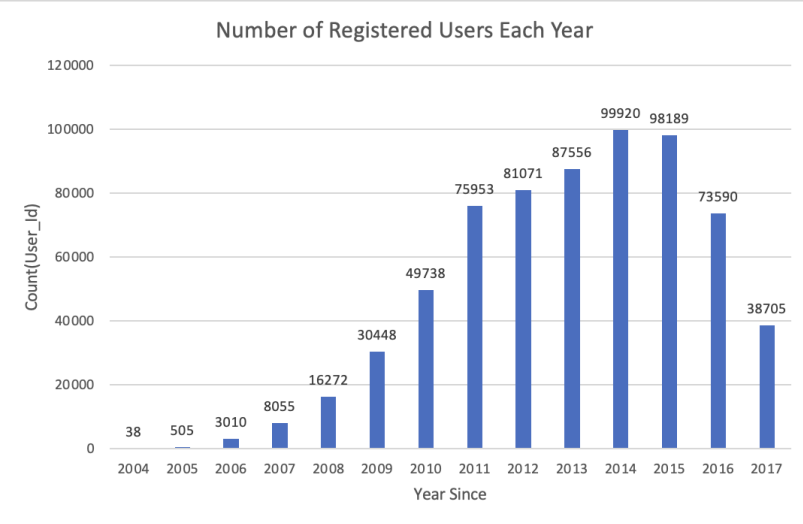
We compute the number of registered users each year using the following code:

```sql
SELECT YEAR(yelping_since) as year_since, COUNT(user_id)
FROM yelp_user
GROUP BY YEAR(yelping_since)
ORDER BY year_since ASC;
```

Results are as follows:

year_since	COUNT(user_id)
2004	38
2005	505
2006	3010
2007	8055
2008	16272
2009	30448
2010	49738
2011	75953
2012	81071
2013	87556
2014	99920
2015	98189
2016	73590
2017	38705

The histogram is drawn as follows:



It can be seen that the number of registered users was on the rise before 2014 and declined after then. In 2014, the year with the largest increase in the number of users, which was 99920. The growth rate of new users in 2015 was also similar, reaching 98189. The decline was significant after 2015. Although the statistics for 2017 are only up to December 11th, it is unlikely that there will be an explosive growth in new users in the last twenty days of the year. Therefore, it can be inferred that the number of registrations for the entire year will still be on a downward trend.

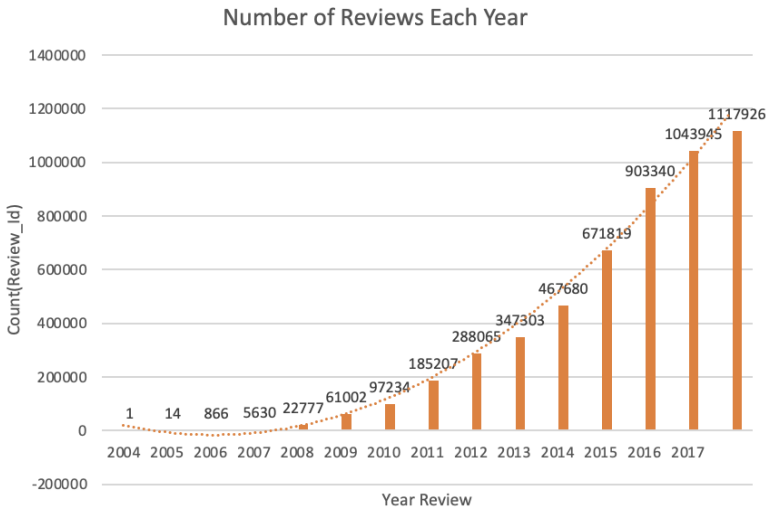
We then compute the total number of reviews posted by users on Yelp each year using the following code:

```
SELECT YEAR(date) as year_review, COUNT(review_id)
FROM yelp_review
GROUP BY YEAR(date)
ORDER BY year_review ASC;
```

Results are as follows:

year_review	COUNT(review_id)
0	1
2004	14
2005	866
2006	5630
2007	22777
2008	61002
2009	97234
2010	185207
2011	288065
2012	347303
2013	467680
2014	671819
2015	903340
2016	1043945
2017	1117926

With the year as the horizontal axis and the total number of reviews in that year as the vertical axis, the histogram and the trend line are drawn as follows:



Here, a power function (α=2) is used for fitting, which is more in line with the upward trend of the number of comments each year. It can be seen that the number of comments posted by users on the platform is gradually increasing, and the growth rate has not decreased with the increase of years. The usage of the platform by users is relatively good.

(2) What is the annual retention rate for all users?

For all users, the review year is used as the basis for determining retention. User IDs and the number of reviews per year are counted, with the code as follows:

```
ALTER TABLE yelp_review
  ADD INDEX idx_user_date (user_id, `date`);

SELECT
  user_id,
  SUM(`date` >= '2008-01-01' AND `date` < '2009-01-01') AS `2008_exist`,
  SUM(`date` >= '2009-01-01' AND `date` < '2010-01-01') AS `2009_exist`,
  SUM(`date` >= '2010-01-01' AND `date` < '2011-01-01') AS `2010_exist`,
  SUM(`date` >= '2011-01-01' AND `date` < '2012-01-01') AS `2011_exist`,
  SUM(`date` >= '2012-01-01' AND `date` < '2013-01-01') AS `2012_exist`,
  SUM(`date` >= '2013-01-01' AND `date` < '2014-01-01') AS `2013_exist`,
  SUM(`date` >= '2014-01-01' AND `date` < '2015-01-01') AS `2014_exist`,
  SUM(`date` >= '2015-01-01' AND `date` < '2016-01-01') AS `2015_exist`,
  SUM(`date` >= '2016-01-01' AND `date` < '2017-01-01') AS `2016_exist`,
  SUM(`date` >= '2017-01-01' AND `date` < '2018-01-01') AS `2017_exist`
FROM yelp_review
WHERE `date` >= '2008-01-01' AND `date` < '2018-01-01'
GROUP BY user_id;
```

Part of the results are shown below and saved as the **user_keep** table:

user_id	2008_exist	2009_exist	2010_exist	2011_exist	2012_exist	2013_exist	2014_exist	2015_exist	2016_exist	2017_exist
___DPmKJsBF2X6ZKgAeGqg	0	0	0	0	0	0	2	1	0	0
___fEWlObjtPaZ-pK0eq9g	0	0	0	0	0	0	0	0	1	0
___l9ZYdYGkZ6dMYxwJElQ	0	0	0	0	4	0	0	0	0	0
___MTsBloH4jvybJ5DrTYw	0	0	0	0	0	0	1	0	0	0
___QCazm0YrHLd3uNUPYMA	0	0	0	0	0	0	2	2	0	0
__-8WEg7xD0CwCDu04MzBA	0	0	0	0	1	0	3	3	4	2
__-C_dSG8Ky98M6EVPYehA	0	0	0	0	0	1	0	0	0	0
__-FcRiBzu8tqWYGofto1Q	0	0	0	0	0	0	0	0	1	0
__-FvQQtlLqqe5s4ClEzfA	0	0	0	0	0	0	0	1	0	2
__-jTbcqIU4pwDy4BZ9JIQ	0	0	0	1	0	0	0	0	0	0
__-Kt26YrtJxGdWs8FqKCg	0	0	0	0	1	2	1	0	2	0
__-XH6EoorFUcuyn1XqFUA	0	0	1	0	0	0	0	0	0	0
__-Xtxtb7z5YEag85AaHWw	0	0	0	0	0	1	0	1	0	0
__02xlXHMOZda_nPoBTnoQ	0	0	0	0	0	0	0	0	0	1
__05rytNjsye9MBhqB0DMA	7	0	0	0	0	0	0	0	0	0

Next, calculate the proportion of users who posted reviews in a given year and continued to post in the following year. This proportion is taken as the annual retention rate, with the code as follows:

```
WITH user_keep AS (
  SELECT
    user_id,
    SUM(`date` >= '2008-01-01' AND `date` < '2009-01-01') AS `2008_exist`,
    SUM(`date` >= '2009-01-01' AND `date` < '2010-01-01') AS `2009_exist`,
    SUM(`date` >= '2010-01-01' AND `date` < '2011-01-01') AS `2010_exist`,
    SUM(`date` >= '2011-01-01' AND `date` < '2012-01-01') AS `2011_exist`,
    SUM(`date` >= '2012-01-01' AND `date` < '2013-01-01') AS `2012_exist`,
    SUM(`date` >= '2013-01-01' AND `date` < '2014-01-01') AS `2013_exist`,
    SUM(`date` >= '2014-01-01' AND `date` < '2015-01-01') AS `2014_exist`,
    SUM(`date` >= '2015-01-01' AND `date` < '2016-01-01') AS `2015_exist`,
    SUM(`date` >= '2016-01-01' AND `date` < '2017-01-01') AS `2016_exist`,
    SUM(`date` >= '2017-01-01' AND `date` < '2018-01-01') AS `2017_exist`
  FROM yelp_review
  WHERE `date` >= '2008-01-01' AND `date` < '2018-01-01'
  GROUP BY user_id
)

SELECT
  ROUND(SUM(`2008_exist`>0 AND `2009_exist`>0)/NULLIF(SUM(`2008_exist`>0),0), 2) AS exist_2009,
  ROUND(SUM(`2009_exist`>0 AND `2010_exist`>0)/NULLIF(SUM(`2009_exist`>0),0), 2) AS exist_2010,
  ROUND(SUM(`2010_exist`>0 AND `2011_exist`>0)/NULLIF(SUM(`2010_exist`>0),0), 2) AS exist_2011,
  ROUND(SUM(`2011_exist`>0 AND `2012_exist`>0)/NULLIF(SUM(`2011_exist`>0),0), 2) AS exist_2012,
  ROUND(SUM(`2012_exist`>0 AND `2013_exist`>0)/NULLIF(SUM(`2012_exist`>0),0), 2) AS exist_2013,
  ROUND(SUM(`2013_exist`>0 AND `2014_exist`>0)/NULLIF(SUM(`2013_exist`>0),0), 2) AS exist_2014,
  ROUND(SUM(`2014_exist`>0 AND `2015_exist`>0)/NULLIF(SUM(`2014_exist`>0),0), 2) AS exist_2015,
  ROUND(SUM(`2015_exist`>0 AND `2016_exist`>0)/NULLIF(SUM(`2015_exist`>0),0), 2) AS exist_2016,
  ROUND(SUM(`2016_exist`>0 AND `2017_exist`>0)/NULLIF(SUM(`2016_exist`>0),0), 2) AS exist_2017
FROM user_keep;
```

The results are as follows:

exist_2009	exist_2010	exist_2011	exist_2012	exist_2013	exist_2014	exist_2015	exist_2016	exist_2017
0.30	0.31	0.31	0.28	0.30	0.33	0.33	0.32	0.32

From the chart, it can be seen that the annual retention rate for regular users has remained around 30% from 2009 to 2017.

Conclusion

1. **From the user perspective**

   For users, the most important question is often how to efficiently earn the title of elite user in order to enjoy more platform perks and additional benefits from exposure. Based on the analysis, users not only need to post more reviews, but also for each review, users should focus both on writing longer and increasing the informational content. Rather than polishing language style, it is more effective to provide information that is useful to the community. In particular, pointing out problems at otherwise highly rated restaurants tends to attract more attention from other users. Users residing in states such as Nevada or Arizona are more likely to be selected as elite users, as these states have a higher concentration of businesses with heavy customer traffic.

   For ordinary users who do not care about achieving elite status, the ultimate goal of using Yelp is to find good restaurants that meet their needs. Overall, restaurant star ratings and user review scores align fairly well, providing a reasonable reflection of quality that users can rely on. However, caution is needed with five-star restaurants, as their ratings may sometimes be inflated, requiring users to distinguish carefully based on authentic feedback.

2. **From the business perspective**

   For businesses that want to attract more customers, expanding into larger states such as Nevada, Arizona, North Carolina, Ohio, and Pennsylvania is a good strategy. Restaurants that want to become five-star may benefit most from focusing on Arizona and Nevada. From our analysis, five-star restaurants also tend to ensure high customer traffic and sufficient operating days. For restaurants aiming to avoid low ratings and pursue higher scores, geographical location is less important; instead, they should maximize the number of operating days per week and pay attention to maintaining a consistently high proportion of positive reviews.

3. **From the platform perspective**

   The platform continues to maintain strong appeal for existing users, with retention rate remaining healthy. However, new user registrations have been declining year by year, suggesting the need for interventions—for example, collaborating with restaurants during popular summer months when people go out more often, or promoting the app with discounts and campaigns. The decline in registrations may also signal that the market is

reaching saturation. Expanding into more states, or even more countries, could help extend the platform's reach.

For existing users, especially regular users who remain active, the platform could design campaigns tied to users' check-in time and activity patterns, offering opportunities to earn rewards through higher engagement, or special perks for returning users. At the same time, further enhancing the benefits of being an elite user could motivate more users to strive for elite recognition.