Yelp Dataset Draft 1-Copy

August 8, 2019

1 YELP ANALYSIS

There are five datasets: 'yelp_academic_dataset_checkin.json', 'yelp_academic_dataset_review.json', 'yelp_academic_dataset_business.json', 'yelp_academic_dataset_user.json', 'yelp_academic_dataset_tip.json'

What can we do to make the data manageable?

We can analyze subparts of this data by separating it by location or business category. The business category I am interested in is Food. The location I am interested in is San Francisco Bay Area. But with the data available, the location I will be focusing on is Arizona and potentially Nevada.

** Objectives of this Project **

1. Discover some interesting trends on business closures
2. Understand the market and economic factors in the location. And how it can influence the businesses.
3. Improve user experience by offering them restaurants in the area they may be interested in based on their past reviews
4. Learn how to do text analysis and use Naive Bayes classifier
5. Study the behavior of checkins and how businesses can utilize checkins to improve traction.
6. *action-oriented analyses*

** What are questions or topics we have for the Yelp dataset?**

- Does the overall sentiment for user's reviews correlate with the user's average_stars?
- Does the business location has an effect on total number of reviews for the business?
- What kind of categories in the Food section of Yelp are the most popular?
- Do Yelp users with no Yelp friends tend to post negative reviews?
- **Analyze the text of the reviews for specific categories such as service, food quality, and experience.**

  How do we quantify or qualify these categories to examine for in the text?

  - ** Predict ratings based on text of reviews **

- **Predict what kind of restaurants users would like based on past reviews**
- **Predict ratings based on similar businesses' rating and users' previous ratings**
- More reviews led to more stars, but what are other factors that led to higher rating besides popularity?
- Do checkin has an effect on the rating? For example, rate 5/5 for a free item.
- **The growth of food business in certain locations**
- Does opening of a business led to greater average housing value in the area?
- Do certain areas have more expensive restaurants than others? What kind of users eat at these places?
- **Why do businesses close? Does it affect the location?**
- Which category of business is popular?
- User acquistion
- Which metropolian area is popular for its food scene?

What are other data we can combine with this data?
We can use social media data, census tract data, transportation data, housing data, and etc.
Median Income Level of Neighborhoods.
Sources: https://www.phoenixopendata.com, https://www.phoenix.gov/nsdsite/Documents/lowmod-ct51.pdf, https://www.governing.com/gov-data/phoenix-gentrification-maps-demographic-data.html

Definitions Reviews with 3 stars are considered neutral. x>3 is considered positive. x < 3 is considered negative.

Challenges to our Questions and Expectations

I thought that we can filter by location and study specific locations but the yelp academic dataset is limited. In the dataset, it is only limited to 19 businesses in CA. It is difficult to understand growth and urban problems. Looking at the value counts of businesses in each state, AZ, NV, and ON have significant numbers of businesses we can study for location. Thus, if we continue with that area of study, AZ would be a good location pick.

There are lot of data in this package. There will be alot of features we do not need to evaluate in our study. The challenge is picking what features are important to answering our questions.

The location's latitude and longitude doesn't match up with the address or there could be multiple businesses in one latitude and longitude but it doesn't tell us if that is one physical location.

[1]:
```
import numpy as np
import pandas as pd
import json
import csv
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import datetime
import nltk
from nltk.corpus import stopwords
import re
from collections import Counter
from sklearn.feature_extraction.text import CountVectorizer
from wordcloud import WordCloud
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪accuracy_score
import tensorflow as tf
from tensorflow import keras
```

1.1 Data Summary and Statistics

** What is in the Yelp database? **

Business dataset has columns- address, business_id, categories, city, is_open, latitude, longitude, name, postal_code, review_count, stars, state

Business Attributes dataset has

Review dataset has business_id, user_id, stars, text, useful, funny, cool

User dataset has user_id, review_count, elite, friends, fans,

Checkins dataset has business_id, day, time, count

1.1.1 Reading the dataset

[2]:
```
# Reading the business dataset
# columns include address, attributes, business_id,
# categories, city, hours, is_open, latitude, longitude,
# name, postal_code, review_count, stars, state
business = pd.read_csv('yelp_business.csv')
businessattributes = pd.read_csv('yelp_business_attributes.csv')
# businesshours = pd.read_csv('yelp_business_hours.csv')

checkin = pd.read_csv('yelp_checkin.csv')
user = pd.read_csv('yelp_user.csv')
review = pd.read_csv('yelp_review.csv')
```

Business First

[3]:
```
# The total number of businesses: 174567
print(business.shape[0])
# The number of businesses for each state
#AZ = 52214, NV = 33086, ON = 30208
business['state'].value_counts()

# Found data on outside of US. We should exclude international businesses.
```

```
stateinitials = ['AL', 'AK', 'AZ', 'AR', 'CA',
                 'CO', 'CT', 'DE', 'FL', 'GA', 'HI',
                 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA',
                 'ME', 'MD', 'MA', 'MI', 'MN', 'MS',
                 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM',
                 'NY', 'NC', 'ND', 'OH', 'OK', 'OR', 'PA',
                 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VT',
                 'VA', 'WA', 'WV', 'WI', 'WY']
# Filter for only USA LOCATIONS
usbusiness = business[business['state'].isin(stateinitials)]

# Filter for Food and Restaurants Including Categories
usbusiness['categories'] = usbusiness['categories']
foodbusiness = usbusiness[usbusiness['categories'].apply(lambda x: 'Food' in␣
 ↪str(x) or 'Restaurant' in str(x))]

#number of businesses in the food section
#AZ = 13826, NV = 9263, OH = 6031, NC = 4969
# foodbusiness['state'].value_counts()
```

```
174567

/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:21:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
```

[4]:
```
# 27459 businesses that are open and 10375 businesses that are closed
foodbusiness['is_open'].value_counts()
```

[4]:
```
1    30966
0    10890
Name: is_open, dtype: int64
```

[5]:
```
# the food section includes supermarkets and nonrestaurant food related␣
 ↪businesses
nonrestaurantlist = ['Shopping', 'Shopping,',
                     'Services', 'Groceries', 'Event Planning',␣
 ↪'Convenience','Convenience Stores',
                    'Gas Station', 'Gas', 'Grocery', 'Station', 'Banks,']
# filter out sole food serving places
foodbusiness = foodbusiness[foodbusiness['categories'].apply(
    lambda x: not any(s in nonrestaurantlist for s in str(x).split(';')))]
#change categories into a list not a string
```

```
foodbusiness['categorieslist'] = foodbusiness['categories'].apply(lambda x:␣
 ↪str(x).split(';'))
# # number of food businesses for each state
# # AZ 12331, NV 8399, OH = 5535
# foodbusiness['state'].value_counts()
```

[6]:
```
#selecting only AR businesses
arizonafoodbusiness = foodbusiness[foodbusiness['state'] == 'AZ']
#selecting only NV businesses
nevadafoodbusiness = foodbusiness[foodbusiness['state'] == 'NV']
```

[8]:
```
#ethnic food categories
selectedfoodcategories = ['Vietnamese', 'French', 'Italian',
                  'Mexican', 'Chinese', 'Japanese', 'American (New)',
                   'American (Traditional)', 'American',
                  'Korean', 'Fast Food', 'Desserts', 'Thai',
                  'Mediterranean', 'Indian', 'Pizza',
                          'Ice Cream & Frozen Yogurt',
                          'Greek', 'Cuban', 'Creole/Cajun',
                          'Asian Fusion']
```

[9]:
```
restaurantgroups = ['Afghan',
'African', 'Senegalese', 'South African', 'American (New)', 'American␣
 ↪(Traditional)',
'Arabian','Argentine','Armenian','Asian Fusion','Australian','Austrian',
'Bangladeshi','Barbeque','Basque','Belgian','Brasseries','Brazilian',
'Breakfast & Brunch','Pancakes','British,Buffets',
'Bulgarian','Burgers','Burmese','Cafes','Themed Cafes','Cafeteria','Cajun/
 ↪Creole','Cambodian','Caribbean','Dominican','Haitian','Puerto␣
 ↪Rican','Trinidadian','Catalan','Cheesesteaks','Chicken Shop','Chicken␣
 ↪Wings','Chinese',
'Cantonese','Dim Sum','Hainan','Shanghainese','Szechuan','Comfort Food',
'Creperies','Cuban','Czech','Delis','Diners','Dinner␣
 ↪Theater','Eritrean','Ethiopian','Fast Food','Filipino','Fish &␣
 ↪Chips','Fondue','Food Court','Food␣
 ↪Stands','French','Mauritius','Reunion','Game Meat','Gastropubs','Georgian',
'German','Gluten-Free','Greek','Guamanian','Halal','Hawaiian','Himalayan/
 ↪Nepalese',
'Honduran','Hong Kong Style Cafe','Hot Dogs','Hot␣
 ↪Pot','Hungarian','Iberian','Indian','Indonesian','Irish','Italian','Calabrian','Sardinian','S
 ↪Belt Sushi','Izakaya','Japanese Curry','Ramen','Teppanyaki',
'Kebab','Korean','Kosher','Laotian','Latin␣
 ↪American','Colombian','Salvadoran','Venezuelan','Live/Raw␣
 ↪Food','Malaysian','Mediterranean','Falafel','Mexican',
'Tacos','Middle Eastern','Egyptian','Lebanese','Modern European','Mongolian',
```

```
'Moroccan','New Mexican Cuisine','Nicaraguan','Noodles','Pakistani','Pan␣
 ↪Asian','Persian/Iranian','Peruvian','Pizza','Polish','Polynesian','Pop-Up␣
 ↪Restaurants','Portuguese','Poutineries','Russian','Salad','Sandwiches','Scandinavian','Scotti
 ↪Food','Soup','Southern','Spanish','Sri Lankan','Steakhouses','Supper␣
 ↪Clubs','Sushi Bars','Syrian','Taiwanese','Tapas Bars',
'Tapas/Small␣
 ↪Plates','Tex-Mex','Thai','Turkish','Ukrainian','Uzbek','Vegan','Vegetarian','Vietnamese','Waf
'Wraps']
```

[10]:
```
print(foodbusiness['stars'].corr(foodbusiness['review_count']))
foodbusiness.groupby('stars').agg({'review_count': 'mean'})
```

```
0.1306777140216589
```

[10]:

stars	review_count
1.0	6.018727
1.5	14.533172
2.0	20.864827
2.5	34.255274
3.0	49.370585
3.5	78.045724
4.0	115.967449
4.5	94.888312
5.0	19.447035

Business Attributes Dataset

[11]:
```
#businessattributes of only food business
foodbusinessattributes = businessattributes[businessattributes['business_id'].
 ↪isin(foodbusiness['business_id'])]
#business attributes of only arizona food business
arizonabusinessattributes = businessattributes[businessattributes['business_id'].
 ↪isin(arizonafoodbusiness['business_id'])]
```

Business Hours Dataset

#businessattributes of only food business foodbusinesshours = businesshours[businesshours['business_id'].isin(foodbusiness['business_id'])] #business attributes of only arizona food business arizonabusinesshours = businesshours[businesshours['business_id'].isin(arizonafoodbusiness['business_id'])]

Checkin Dataset

Check-in Offers reward customers with a special offer when they "check-in" to your business on Yelp. By using the check-in feature on the Yelp mobile application, your customers broadcast to their friends on Yelp that they're at your business.

[12]:
```
# filter to include only the business id of the food business
foodcheckin = checkin[checkin['business_id'].isin(foodbusiness['business_id'])]
# filter only arizona food businesses
arizonafoodcheckin = checkin[checkin['business_id'].
 ↪isin(arizonafoodbusiness['business_id'])]
```

```
# sum of checkins per business for food business
sumfoodcheckin = foodcheckin.groupby('business_id').agg(
    {'checkins': 'sum'}).reset_index()
sumfoodcheckin = dict(zip(sumfoodcheckin['business_id'],
                          sumfoodcheckin['checkins']))
#adding it to the az and food df
arizonafoodbusiness['sumcheckin'] = arizonafoodbusiness['business_id'].
 ↪map(sumfoodcheckin)
foodbusiness['sumcheckin']= foodbusiness['business_id'].map(sumfoodcheckin)
```

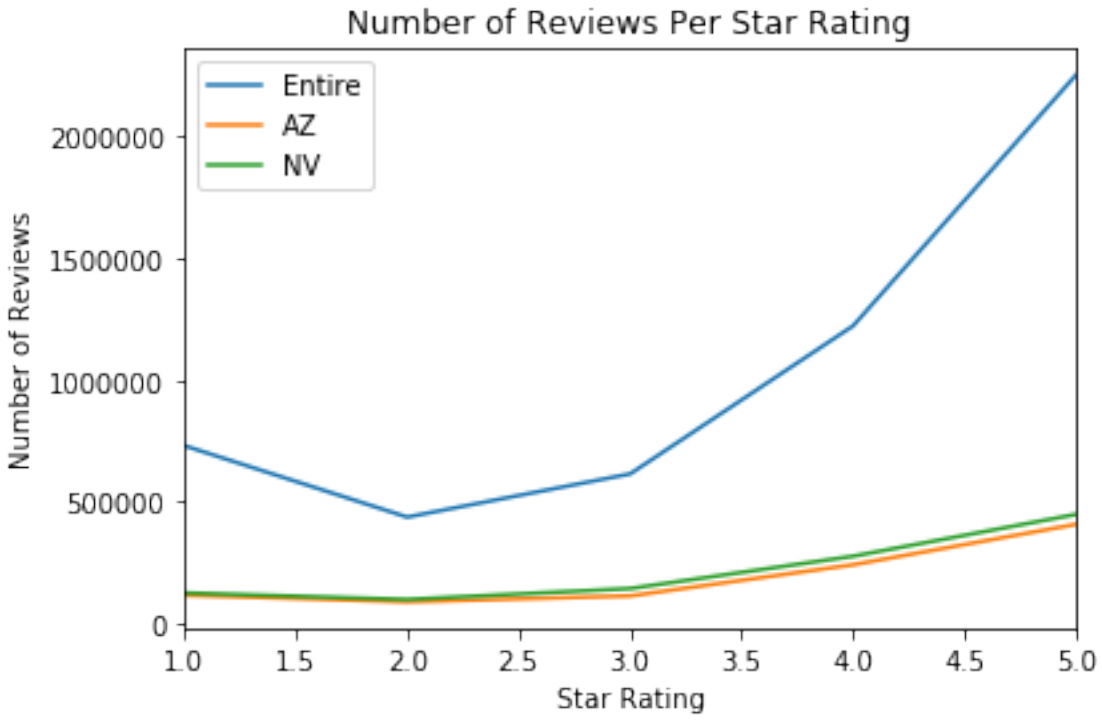
```
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:11:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  # This is added back by InteractiveShellApp.init_path()
```

Review Dataset

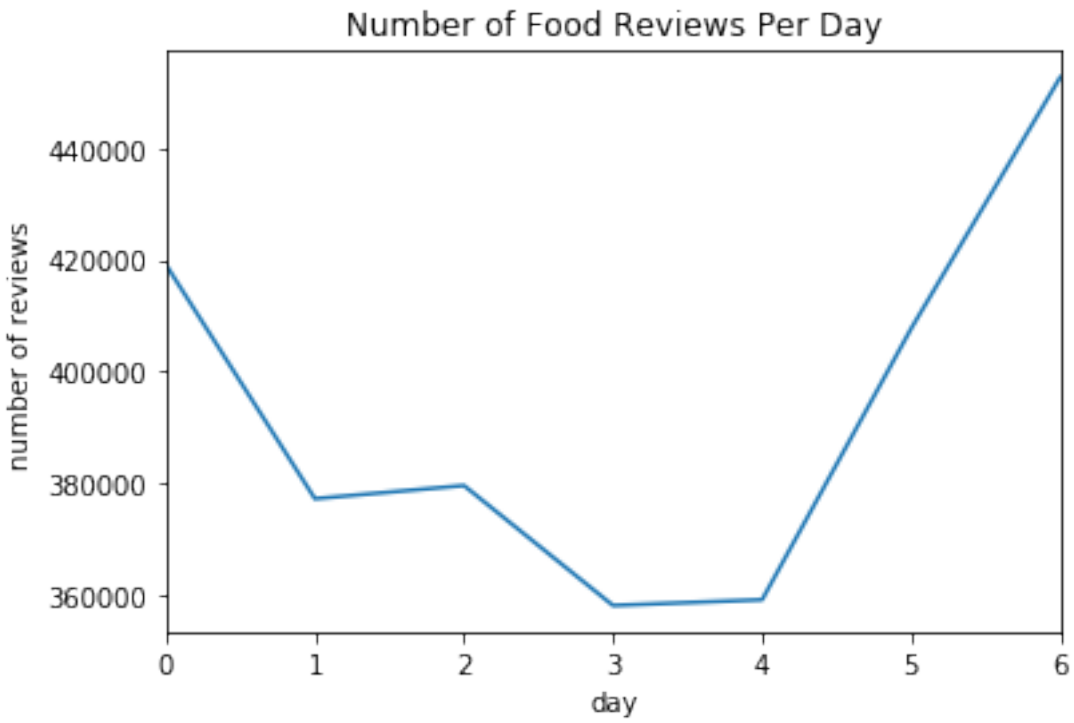
[13]:
```
# change the date into year, day
review['dateobject'] = review['date'].apply(lambda x: datetime.datetime.
 ↪strptime(x, '%Y-%m-%d'))
review['year' ] = review['dateobject'].apply(lambda h: h.year)
review['day'] = review['dateobject'].apply(lambda h: h.weekday())
# #include reviews of only food businesses in our study
foodreviews = review[review['business_id'].isin(foodbusiness['business_id'])]
#include reviews of only food businesses in Arizona
arizonafoodreviews = review[review['business_id'].
 ↪isin(arizonafoodbusiness['business_id'])]

# include reviews of only food businesses in Nevada
nevadafoodreviews = review[review['business_id'].
 ↪isin(nevadafoodbusiness['business_id'])]

# food reviews from users living in Arizona
userarizona = arizonafoodreviews['user_id'].unique()
```

[14]:
```
# number of reviews for each star rating for the entire dataset
review['stars'].value_counts().sort_index().plot()
# number of reviews for each star rating for Arizona
arizonafoodreviews['stars'].value_counts().sort_index().plot()
# number of reviews for each star rating for Nevada
nevadafoodreviews['stars'].value_counts().sort_index().plot()
plt.legend(['Entire', 'AZ', 'NV'])
plt.ylabel("Number of Reviews")
plt.xlabel('Star Rating')
plt.title("Number of Reviews Per Star Rating")
```

```
plt.show()
```



[15]:
```
foodreviews['day'].value_counts().sort_index().plot()
plt.ylabel('number of reviews')
plt.xlabel('day')
plt.title('Number of Food Reviews Per Day')
plt.show()
```



[16]:
```
#min year and max year review for business id
minyear = arizonafoodreviews.groupby('business_id').agg(
    {'year': np.min}).reset_index()
maxyear = arizonafoodreviews.groupby('business_id').agg(
    {'year': np.max}).reset_index()
minmaxyears = minyear.merge(maxyear, on = 'business_id', how = 'inner')
#dictionary of business's min and max year of reviews
minyear = dict(zip(minmaxyears['business_id'],
                   minmaxyears['year_x']))
maxyear = dict(zip(minmaxyears['business_id'],
                   minmaxyears['year_y']))
#add that to the business dataset
arizonafoodbusiness['minyear'] = arizonafoodbusiness['business_id'].map(minyear)
arizonafoodbusiness['maxyear'] = arizonafoodbusiness['business_id'].map(maxyear)
```

```
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:13:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  del sys.path[0]
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:14:
```

```
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
```

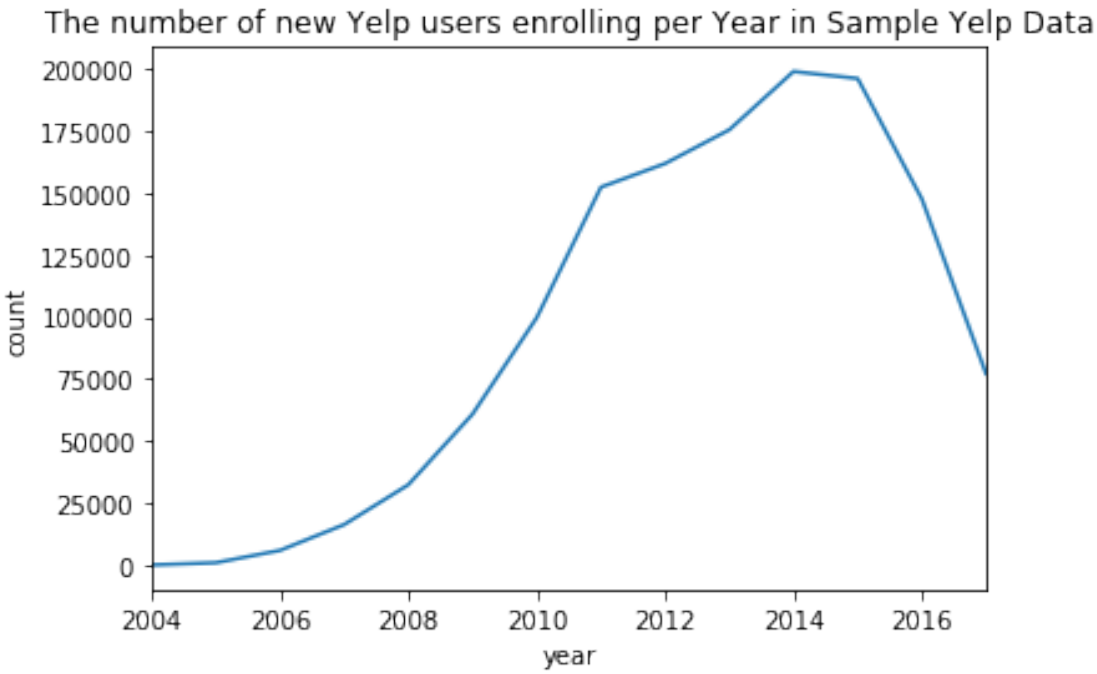
[17]:
```
#how many years of reviews on yelp
arizonafoodbusiness['diffyear']=arizonafoodbusiness['maxyear']-arizonafoodbusiness['minyear']
```

```
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
```

User Dataset

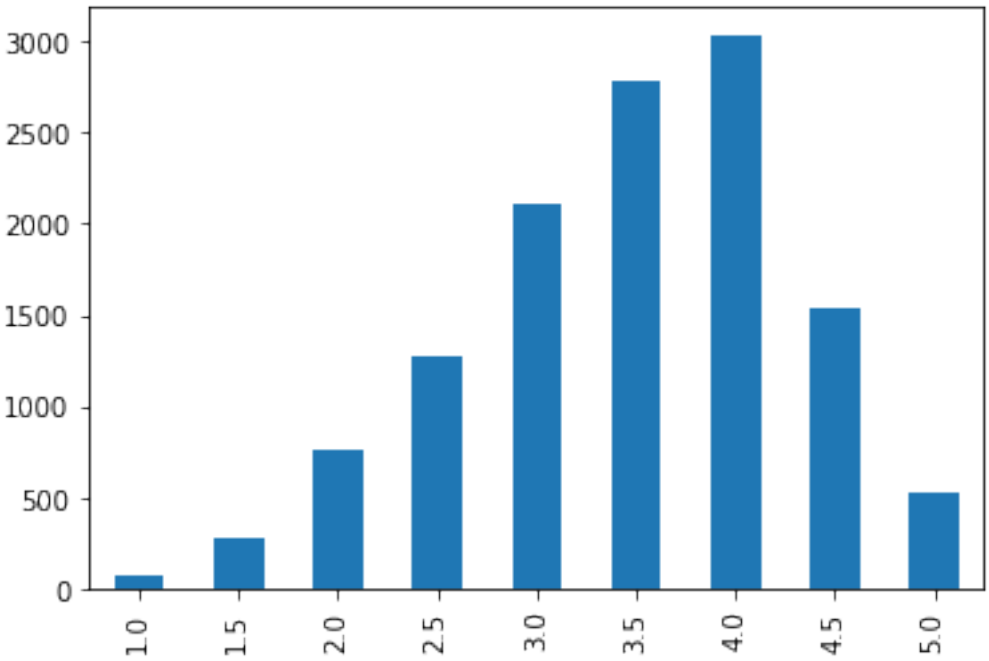
[18]:
```
# users profile from Arizona
arizonauserprofile = user['user_id'].isin(userarizona)
# elite users on Yelp
eliteusers = user[user['elite'] != 'None']
```

[19]:
```
#getting the start year for yelping since
user['startyear'] = user['yelping_since'].apply(lambda x: x[0:4])
user['startyear'].value_counts().sort_index().plot()
plt.title('The number of new Yelp users enrolling per Year in Sample Yelp Data')
plt.ylabel("count")
plt.xlabel('year')
plt.show()
```

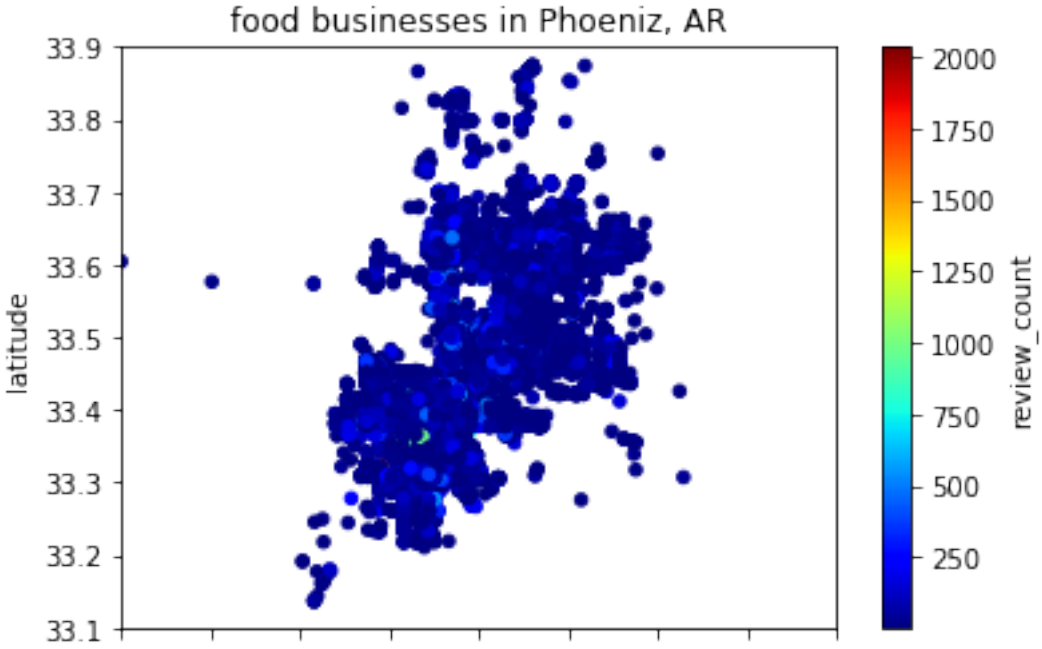


1.2 VISUALIZING BUSINESSES IN ARIZONA

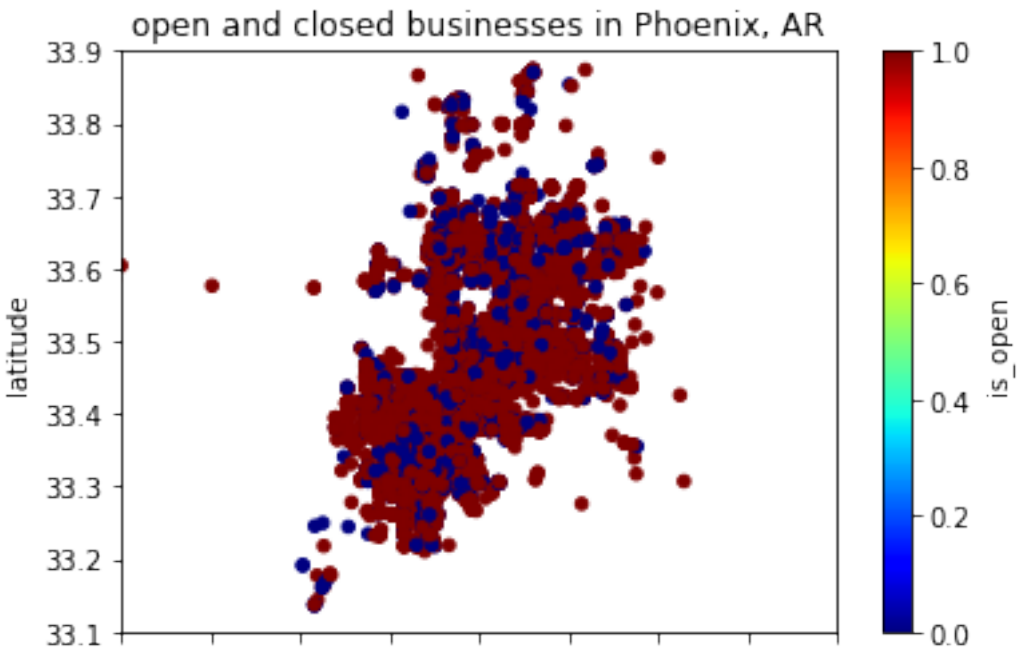
[20]:
```
arizonafoodbusiness['stars'].value_counts().sort_index().plot(kind = 'bar')
plt.show()
```



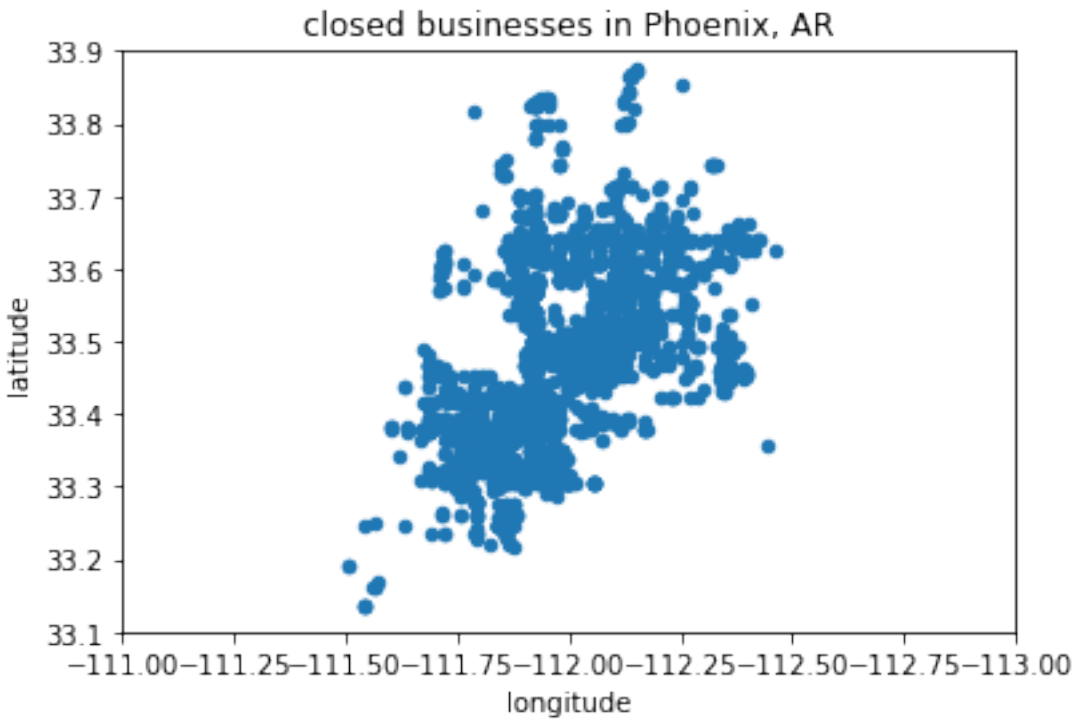
[21]:
```
#remove outlier - gZGsReGOVeX4uKViHTB9EQ in Arizona
arizonafoodbusiness = arizonafoodbusiness[arizonafoodbusiness['business_id'] !=␣
 ↪'gZGsReGOVeX4uKViHTB9EQ']
arizonafoodbusiness.plot(
    kind="scatter", x="longitude", y="latitude", c = 'review_count',
    cmap=plt.get_cmap("jet"), colorbar=True)
plt.title('food businesses in Phoeniz, AR')
plt.ylim(33.1, 33.9)
plt.xlim(-111,
        -113)
plt.show()
```

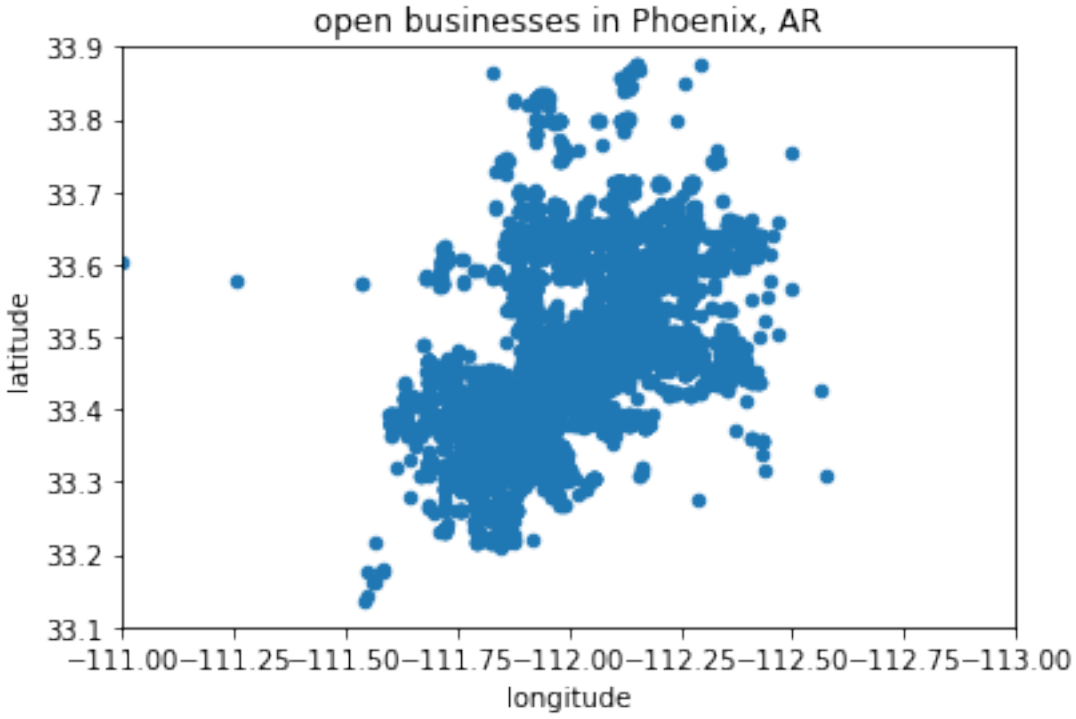


[22]:
```
# where businesses are closed and open
arizonafoodbusiness.plot(
    kind="scatter", x="longitude", y="latitude", c = 'is_open',
    cmap=plt.get_cmap("jet"), colorbar=True)
plt.title('open and closed businesses in Phoenix, AR')
plt.ylim(33.1, 33.9)
plt.xlim(-111,
        -113)
plt.show()
```



[23]:
```
closedazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 0]
closedazbusiness.plot(
    kind="scatter", x="longitude", y="latitude")
plt.title('closed businesses in Phoenix, AR')
plt.ylim(33.1, 33.9)
plt.xlim(-111,
        -113)
openazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 1]
openazbusiness.plot(
    kind="scatter", x="longitude", y="latitude")
plt.title('open businesses in Phoenix, AR')
plt.ylim(33.1, 33.9)
plt.xlim(-111,
        -113)
plt.show()
```

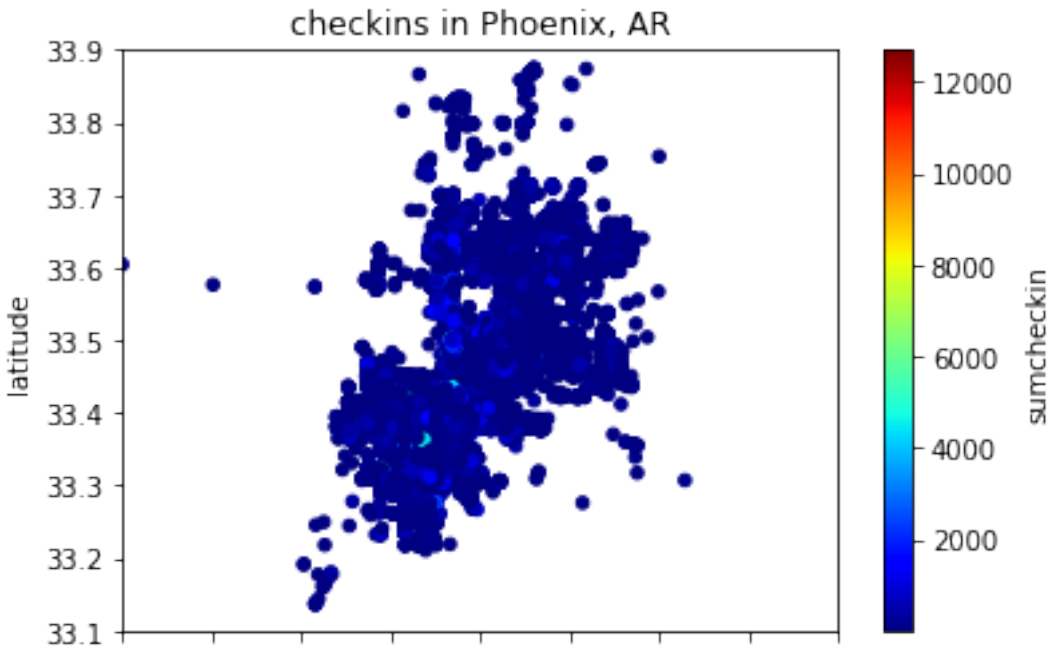

closed businesses in Phoenix, AR
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude


open businesses in Phoenix, AR
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

We see that there has been many closed businesses throughout Arizona metro area. Perhaps we can study it through the years. The time factor could help us visualize the progression of the demand in the market.

[24]:
```
# checkin values of the arizona food business
arizonafoodbusiness['sumcheckin'] = arizonafoodbusiness['business_id'].
 ↪map(sumfoodcheckin)
arizonafoodbusiness.plot(
    kind="scatter", x="longitude", y="latitude", c = 'sumcheckin',
    cmap=plt.get_cmap("jet"), colorbar=True)
plt.title('checkins in Phoenix, AR')
plt.ylim(33.1, 33.9)
plt.xlim(-111,
         -113)
plt.show()
```

```
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
```



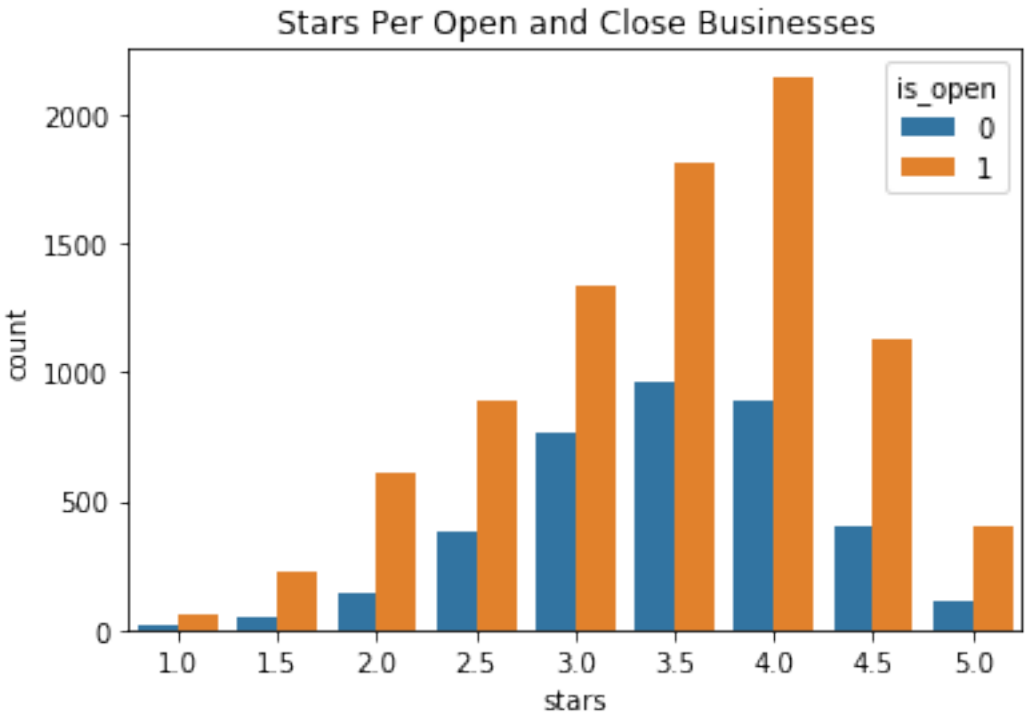
1.3 Closure of Business in Arizona

Common reasons cited for business failure include poor location, lack of experience, poor management, insufficient capital, unexpected growth, personal use of funds, over investing in fixed assets and poor credit arrangements.

[25]:
```
arizonafoodbusiness['is_open'].value_counts()
```

[25]:
```
1    8622
0    3725
Name: is_open, dtype: int64
```

[26]:
```
#average star for closed businesses and average star for opened businesses
azstars = arizonafoodbusiness.groupby(['is_open', 'stars']).count().
 ↪reset_index()[['is_open', 'stars', 'business_id']].rename(
columns = {'business_id': 'count'})
sns.barplot(x = 'stars', y = 'count', hue = 'is_open', data = azstars)
plt.title("Stars Per Open and Close Businesses")
sns.color_palette("PuBuGn_d")
plt.show()
```

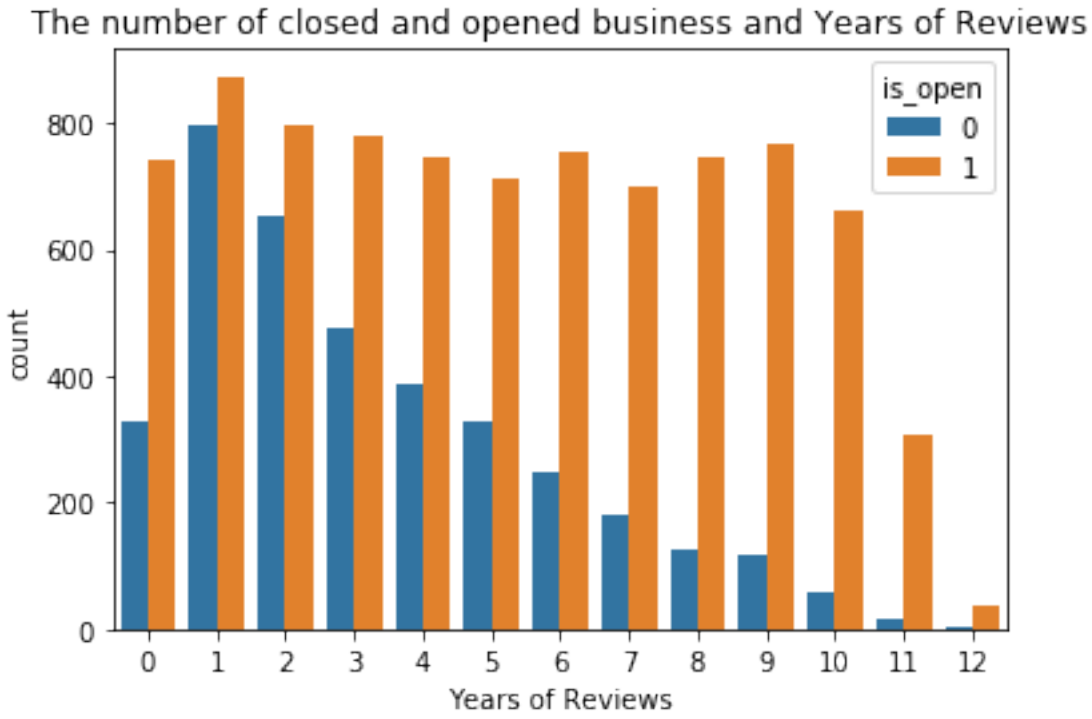


[27]:
```
arizonafoodbusiness.groupby('is_open').agg(
    {'review_count': 'mean'})
```

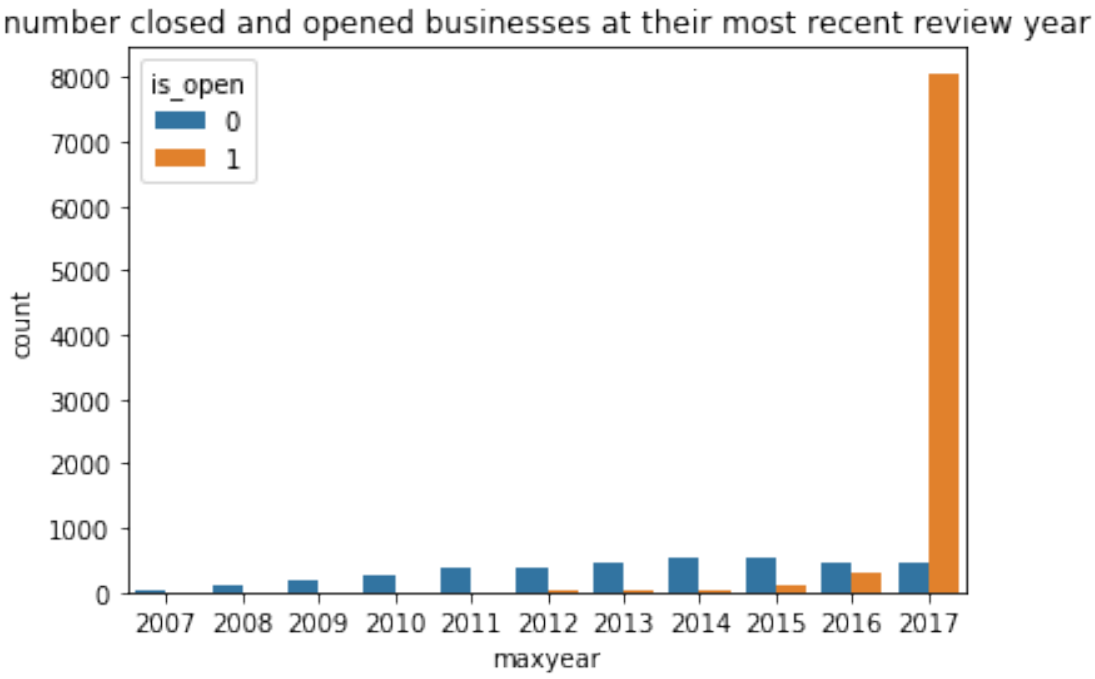
[27]:
```
          review_count
is_open
0            37.802148
```

```
1            96.557875
```

[28]:
```
countperdiffyear = arizonafoodbusiness.groupby(
    ['is_open', 'diffyear']).count().reset_index().rename(
    columns = {'business_id': 'count'})[['is_open', 'diffyear', 'count']]
sns.barplot(x = 'diffyear', y = 'count', hue = 'is_open', data =␣
 ↪countperdiffyear)
plt.title('The number of closed and opened business and Years of Reviews')
plt.xlabel('Years of Reviews')
plt.show()
```



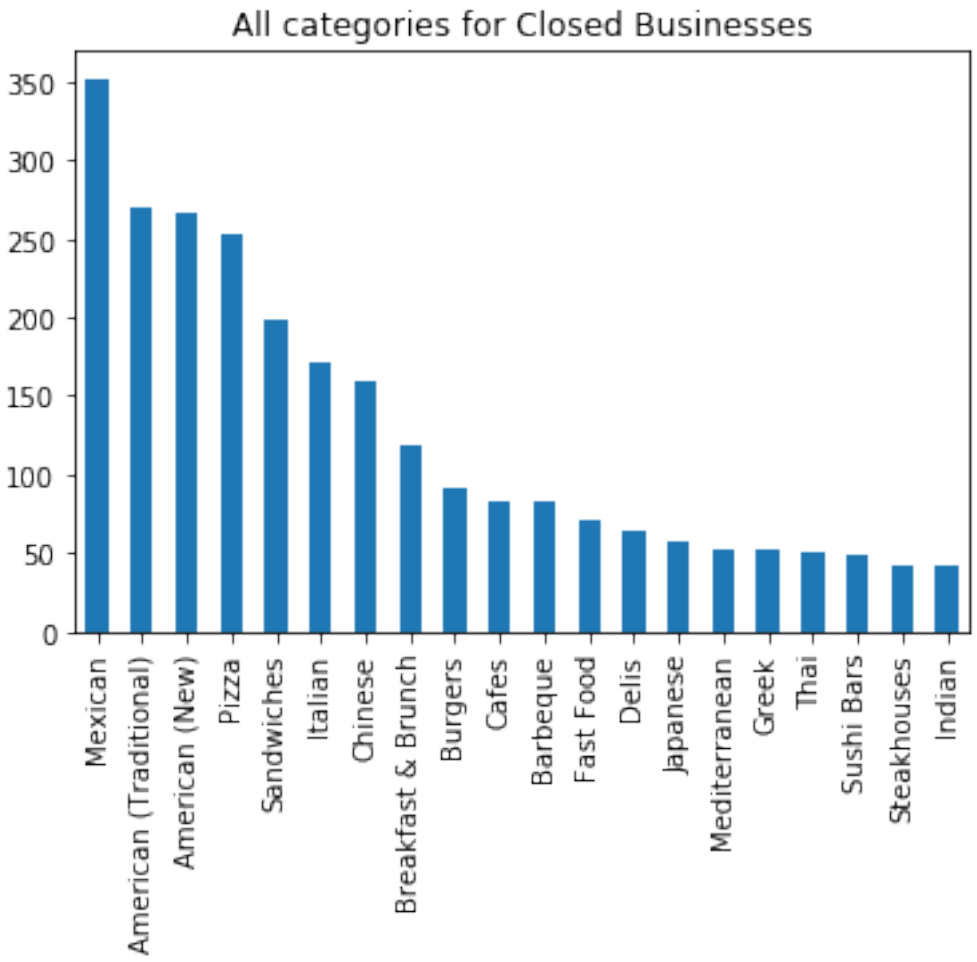
[29]:
```
maxyearo = arizonafoodbusiness.groupby(
    ['is_open', 'maxyear']).agg(
    {'business_id': 'count'}).reset_index().rename(
    columns = {'business_id': 'count'})
sns.barplot(x = 'maxyear', y = 'count', hue = 'is_open', data = maxyearo)
plt.title("number closed and opened businesses at their most recent review year")
plt.show()
```



What kind of businesses close?

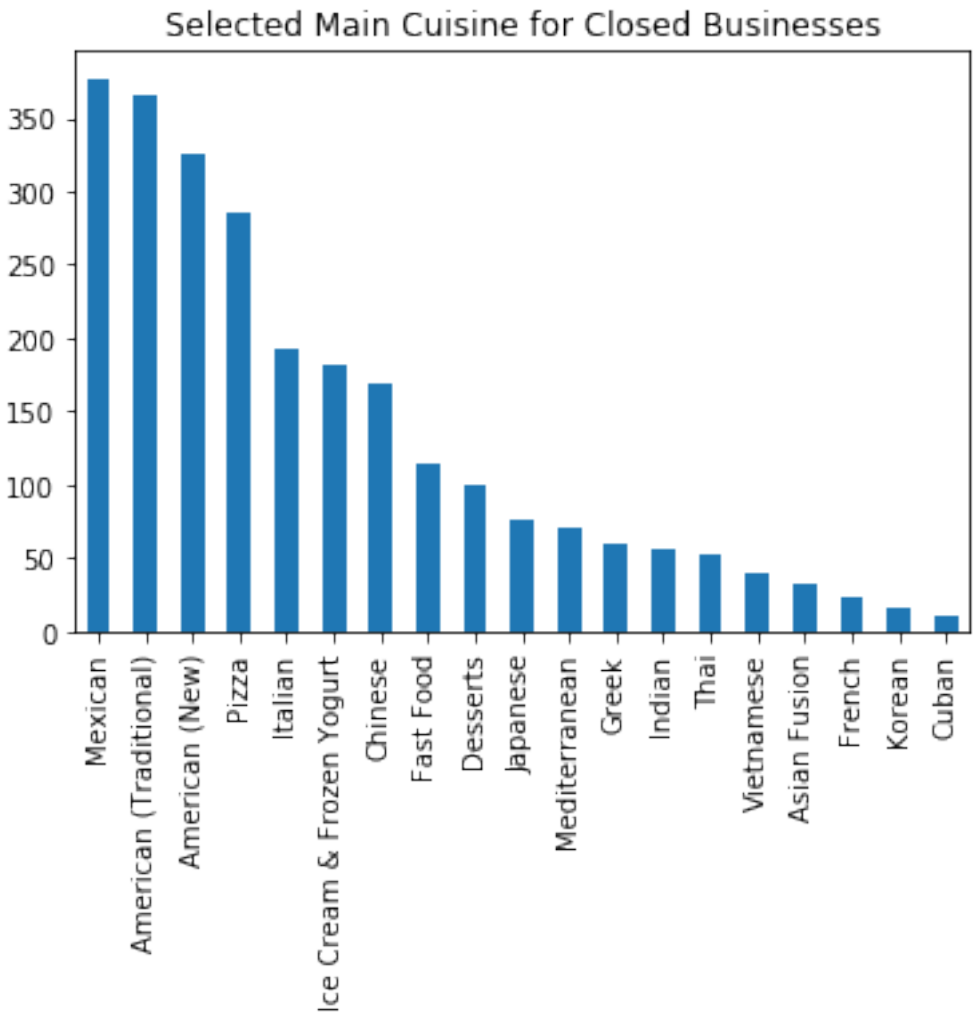
[30]:
```
closedazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 0]
closedazbusiness['maincuisine'] = closedazbusiness['categorieslist'].apply(
    lambda x: [a for a in x if a in restaurantgroups]).apply(
    lambda h: h[0] if len(h) != 0 else np.nan)
closedazbusiness['maincuisine'].value_counts()[0:20].plot.bar()
plt.title('All categories for Closed Businesses')
plt.show()
```

```
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
```



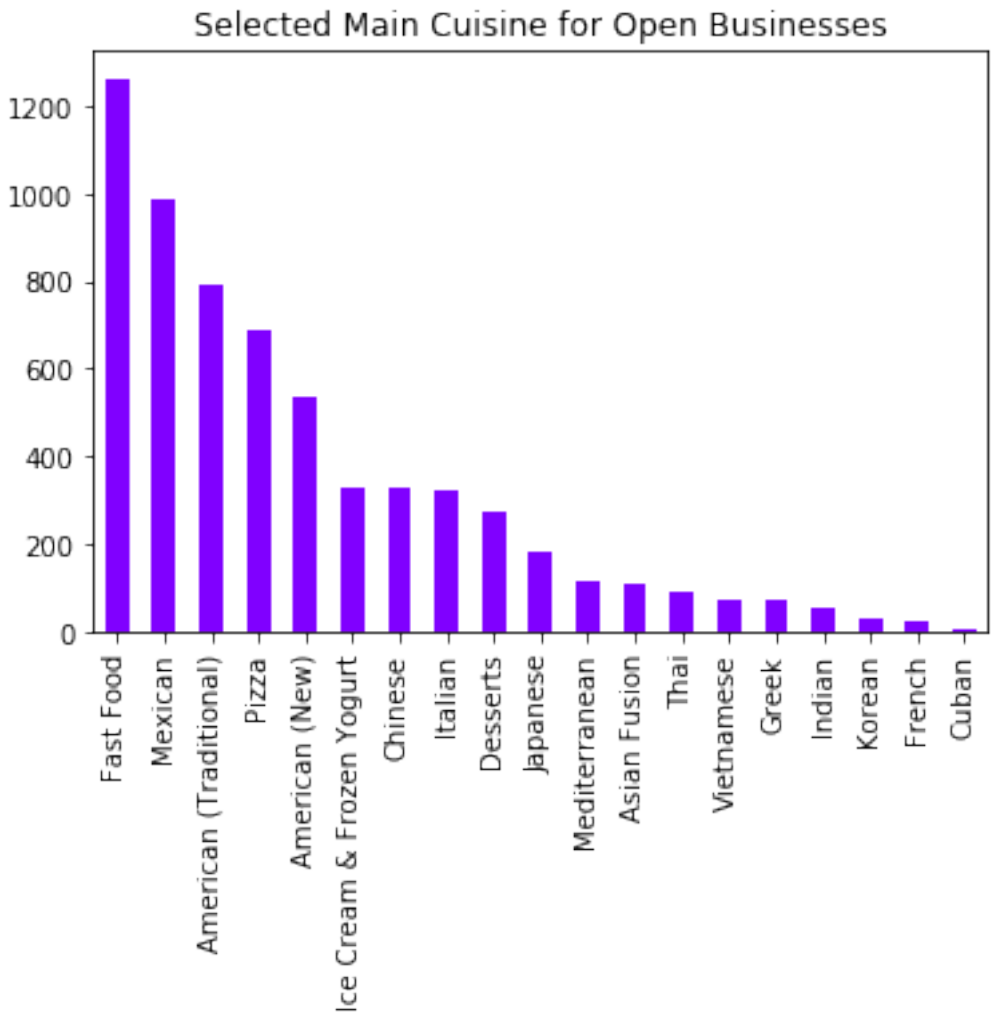
[31]:
```
closedazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 0]
closedazbusiness['maincuisine'] = closedazbusiness['categorieslist'].apply(
    lambda x: [a for a in x if a in selectedfoodcategories]).apply(
    lambda h: h[0] if len(h) != 0 else np.nan)
closedazbusiness['maincuisine'].value_counts().plot.bar()
plt.title('Selected Main Cuisine for Closed Businesses')
plt.show()
```

```
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
```



[32]:
```
openazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 1]
openazbusiness['maincuisine'] = openazbusiness['categorieslist'].apply(
    lambda x: [a for a in x if a in selectedfoodcategories]).apply(
    lambda h: h[0] if len(h) != 0 else np.nan)
openazbusiness['maincuisine'].value_counts().plot.bar(colormap = 'rainbow')
plt.title('Selected Main Cuisine for Open Businesses')
plt.show()
```

```
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
  after removing the cwd from sys.path.
```



The rating didn't make a difference in whether a business is open or closed. Perhaps we can study the sentiment of users.

The food businesses that are opened have on average more review_count than those that are closed. This could mean that younger start up businesses didn't fare so well so they closed their businesses early.

The number of closed businesses are higher when they have fewer years of reviews on yelp! This could either mean that they close during their startup or businesses close for other reasons besides startup especially the ones with more than 7 years on Yelp Reviews.

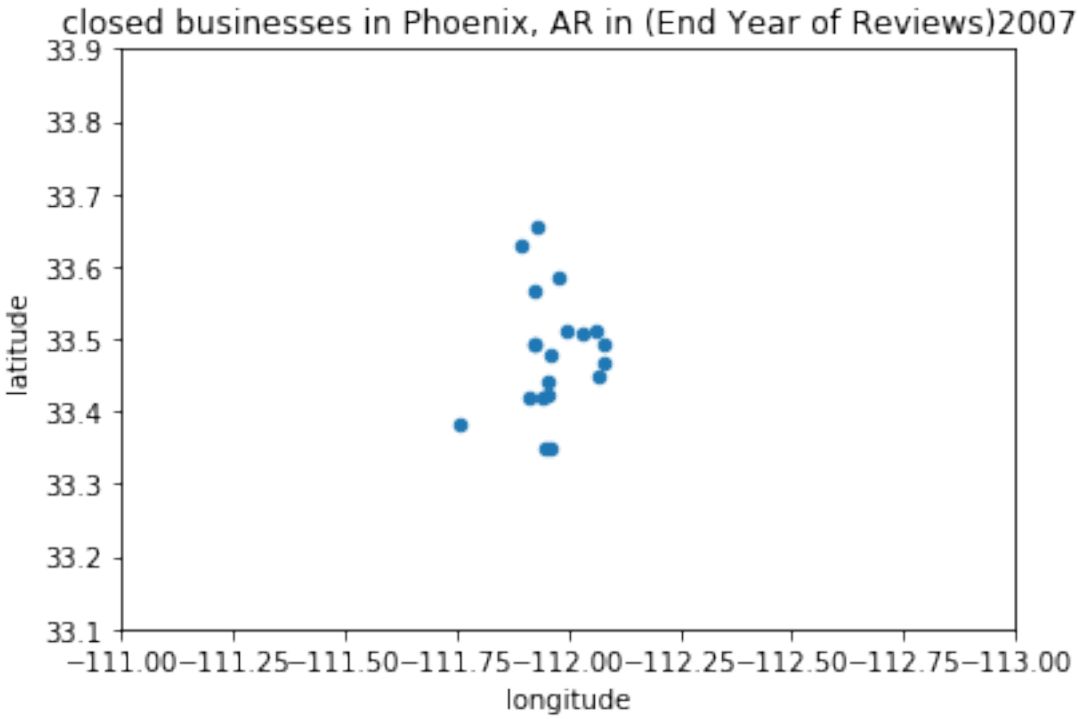
Fast food restaurants tend to stay open because they are part of a chain franchise so they got more funding and capital.

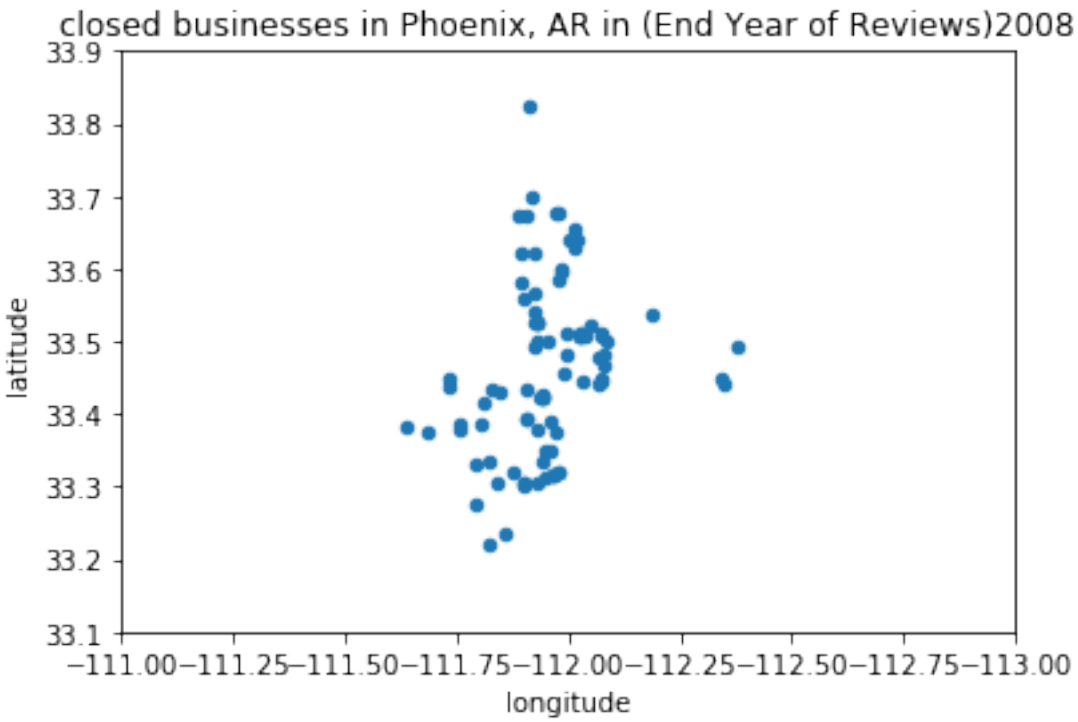
About 41 percent of Phoenix residents are of Hispanic descent(2018 BizJournal)

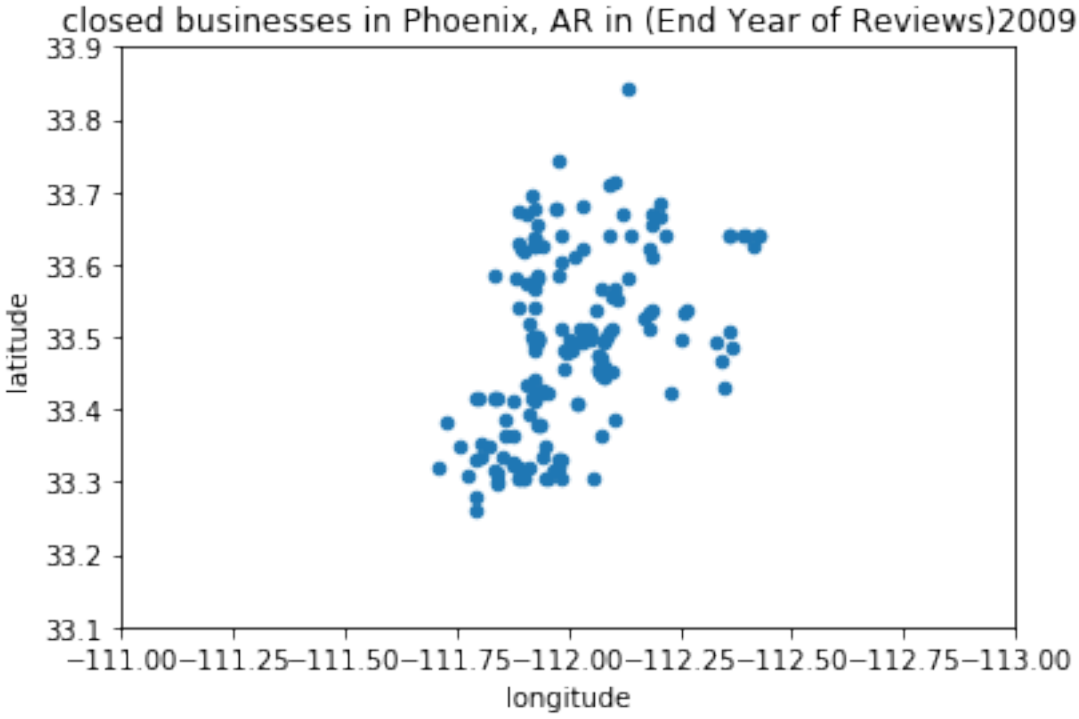
By the Year

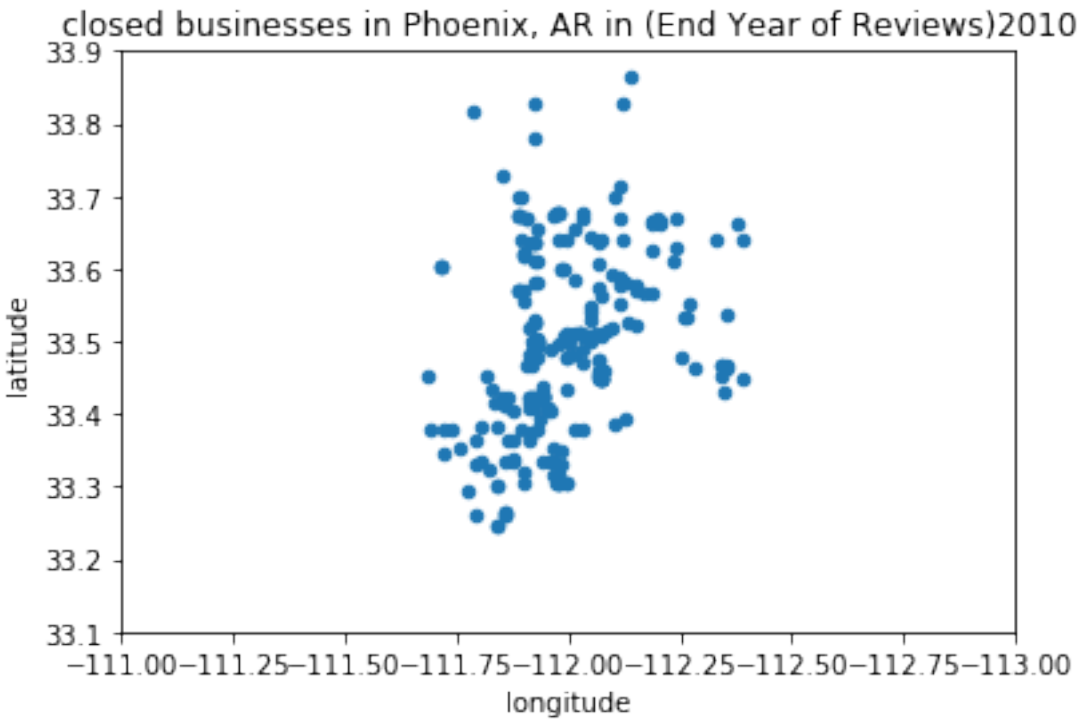
[33]:
```
closedazbusiness['maxyear'].unique()
for year in range(2007, 2018,1):
    dta = closedazbusiness[closedazbusiness['maxyear'] == year]
    dta.plot(
```

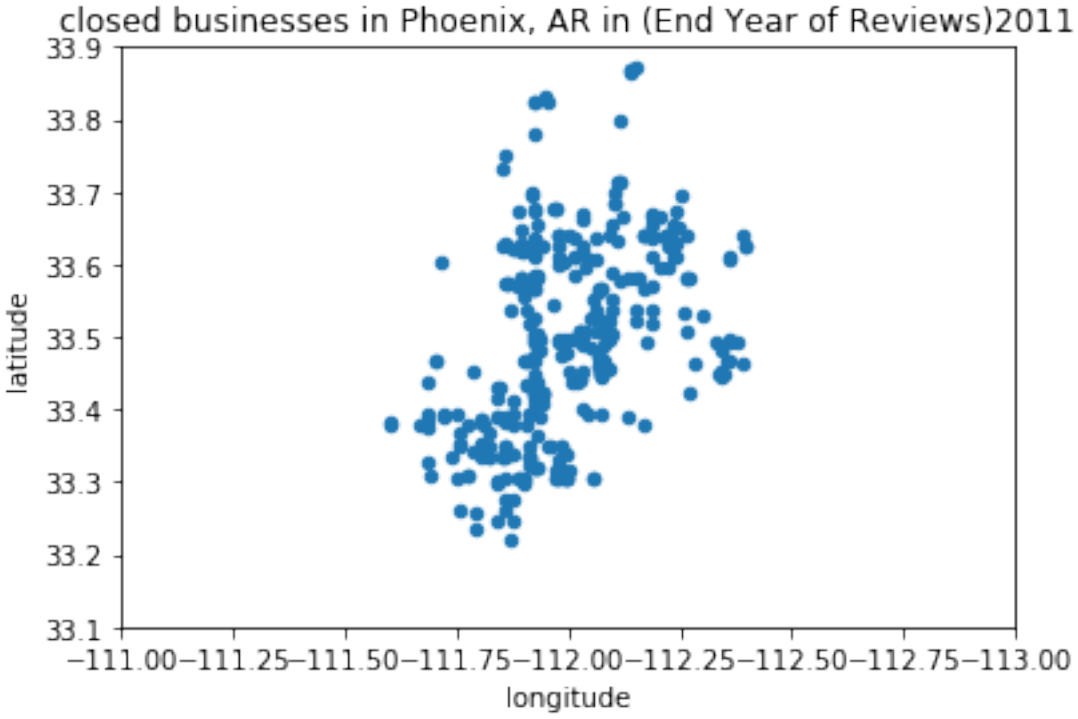
```
   kind="scatter", x="longitude", y="latitude")
   plt.title('closed businesses in Phoenix, AR in (End Year of Reviews){0}'.
↪format(year))
   plt.ylim(33.1, 33.9)
   plt.xlim(-111,
       -113)
   plt.show()
```

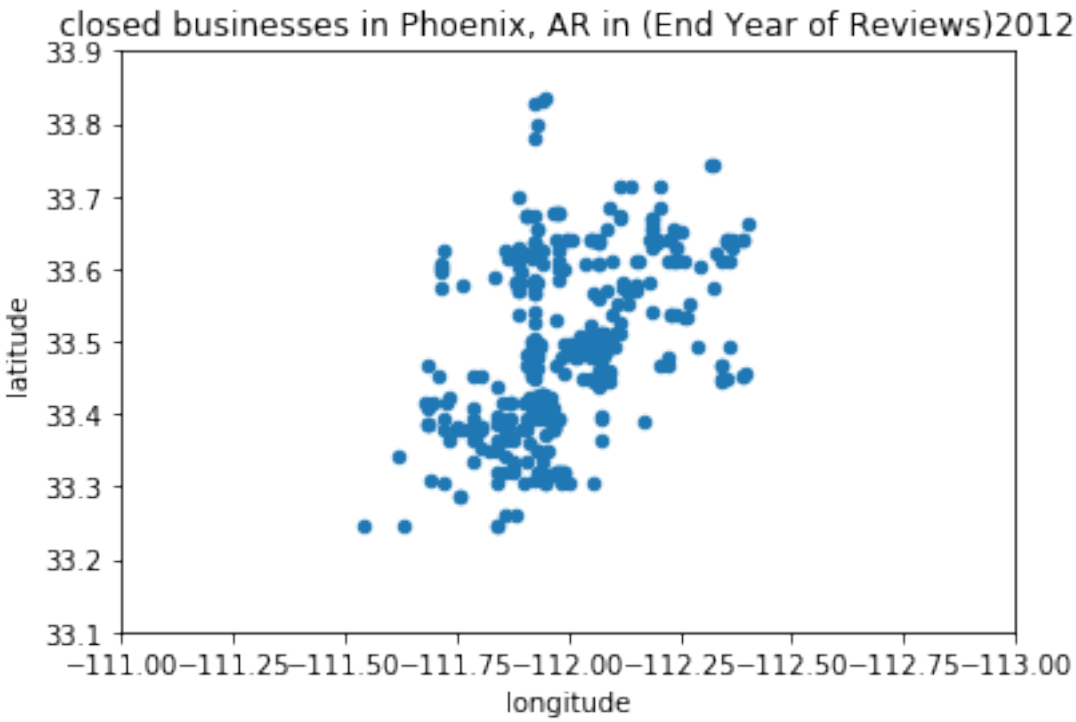


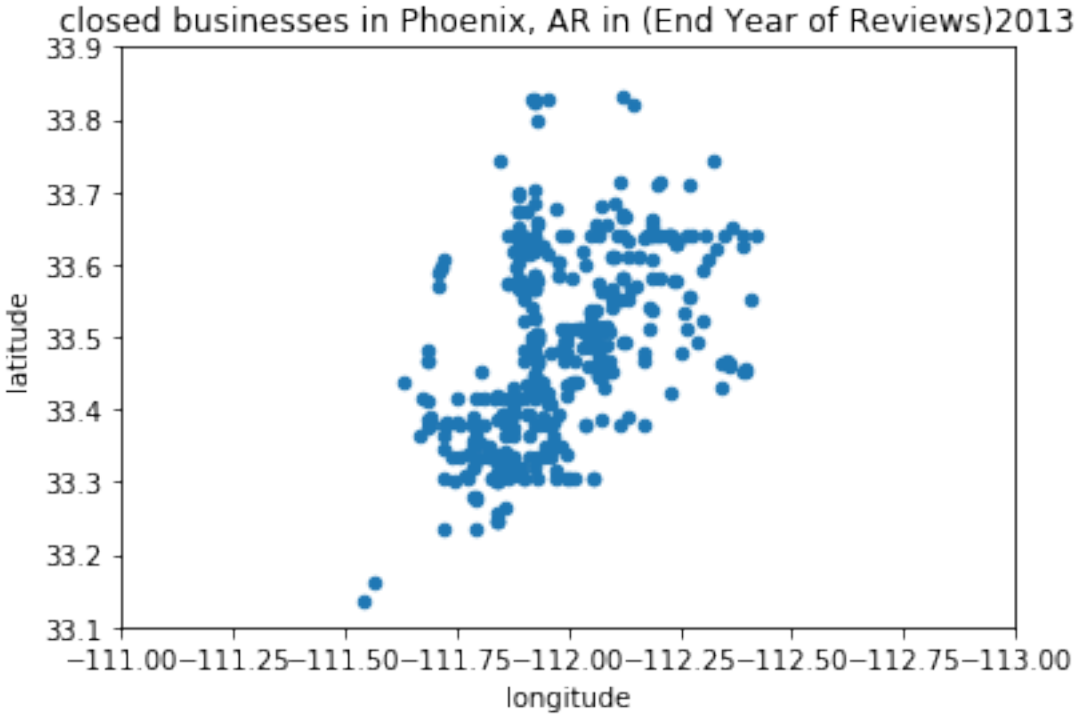

closed businesses in Phoenix, AR in (End Year of Reviews)2008
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

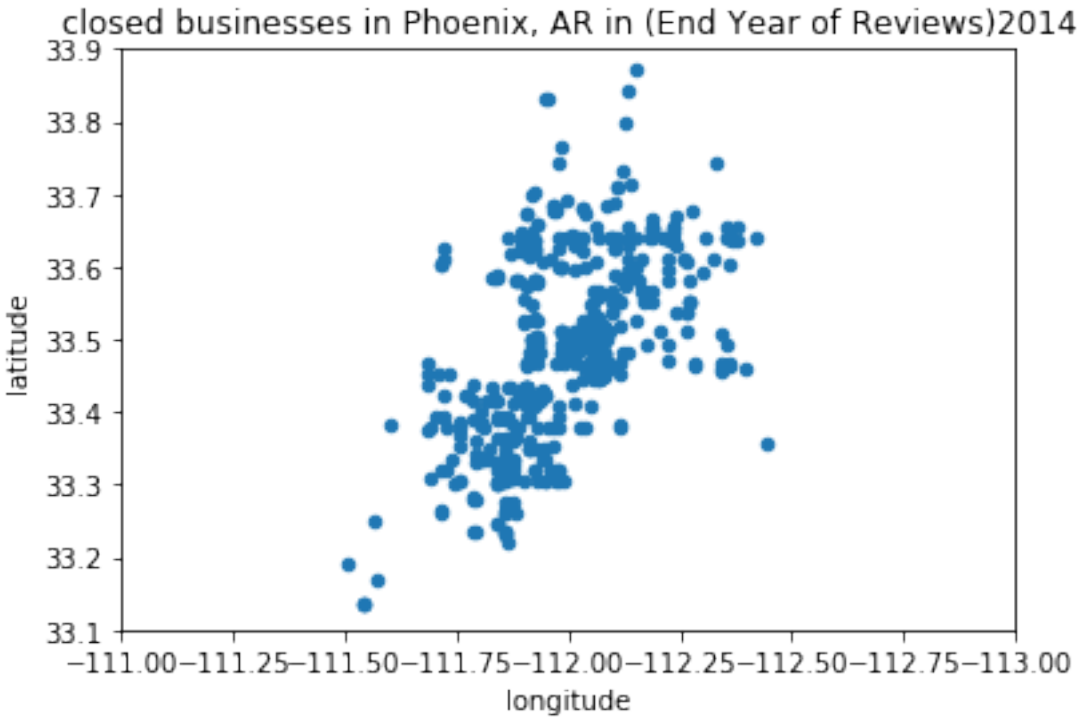

closed businesses in Phoenix, AR in (End Year of Reviews)2009
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

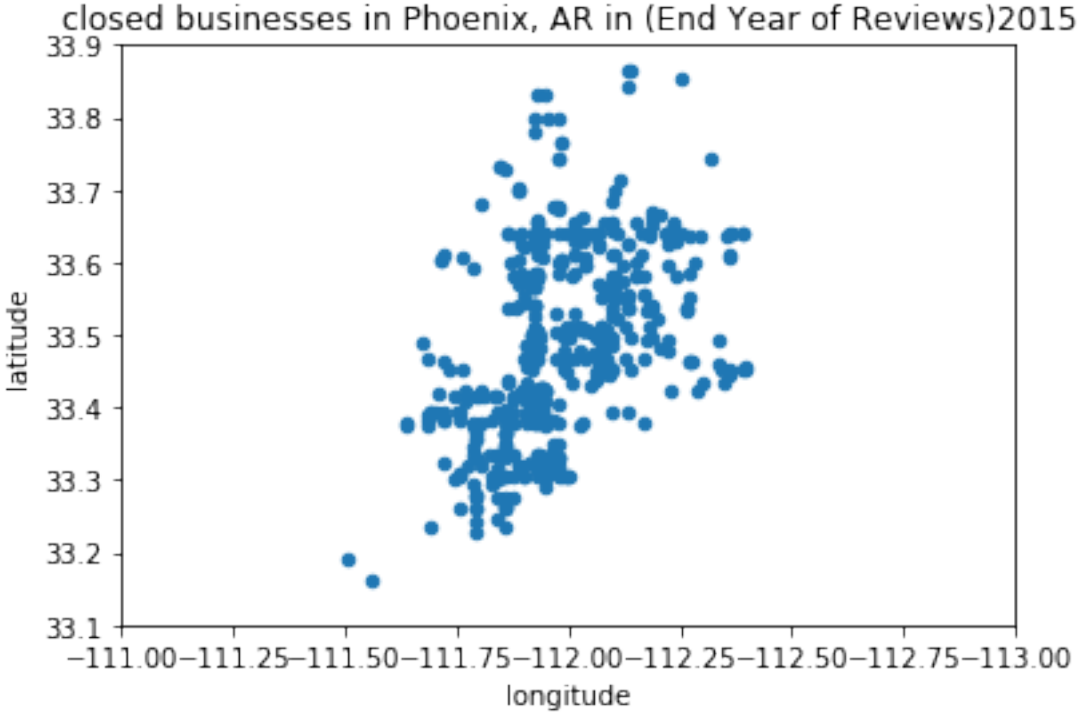

closed businesses in Phoenix, AR in (End Year of Reviews)2010
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

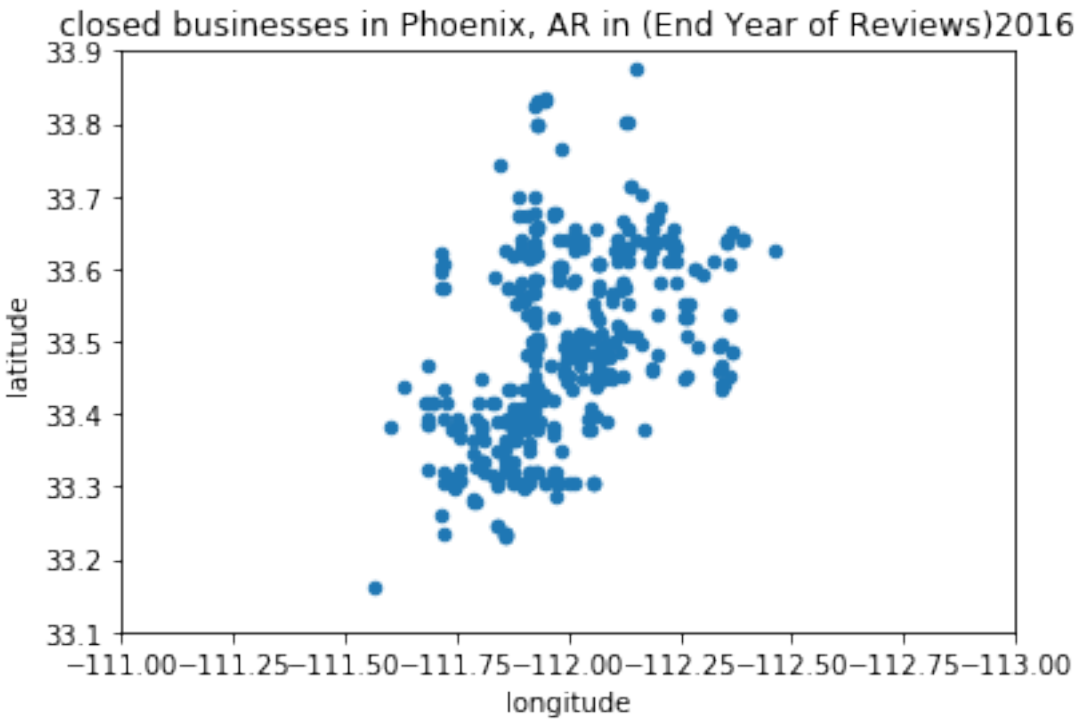

closed businesses in Phoenix, AR in (End Year of Reviews)2011
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

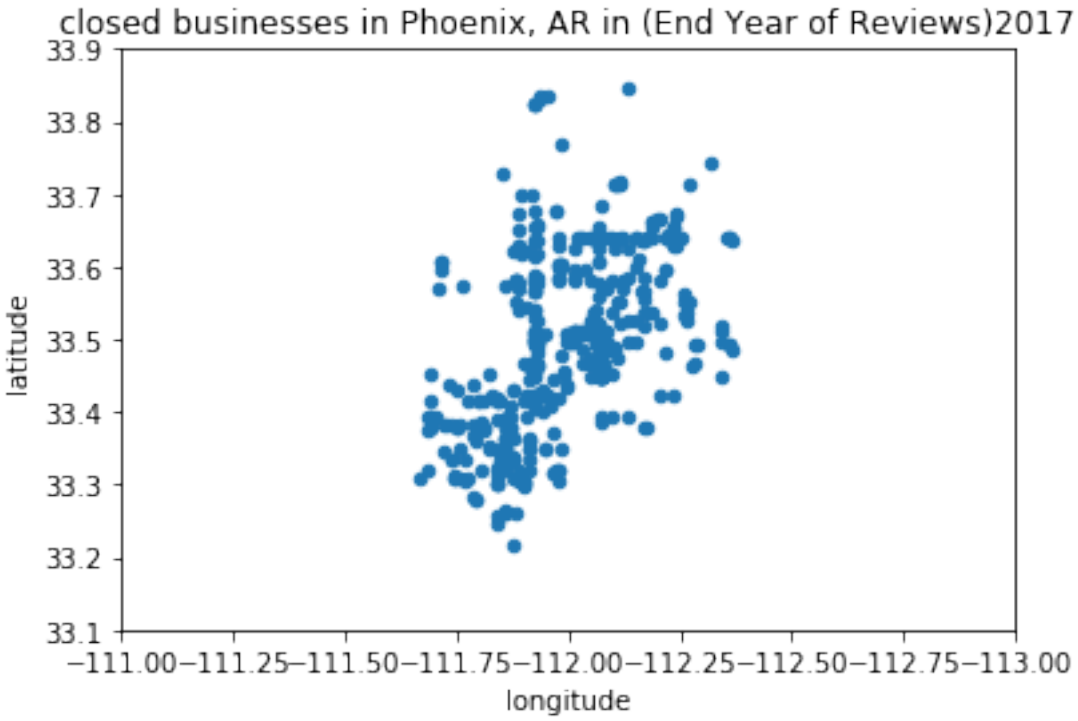

closed businesses in Phoenix, AR in (End Year of Reviews)2012
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude


closed businesses in Phoenix, AR in (End Year of Reviews)2013
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude


closed businesses in Phoenix, AR in (End Year of Reviews)2014
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude


closed businesses in Phoenix, AR in (End Year of Reviews)2015
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude





Opening New Businesses on Yelp

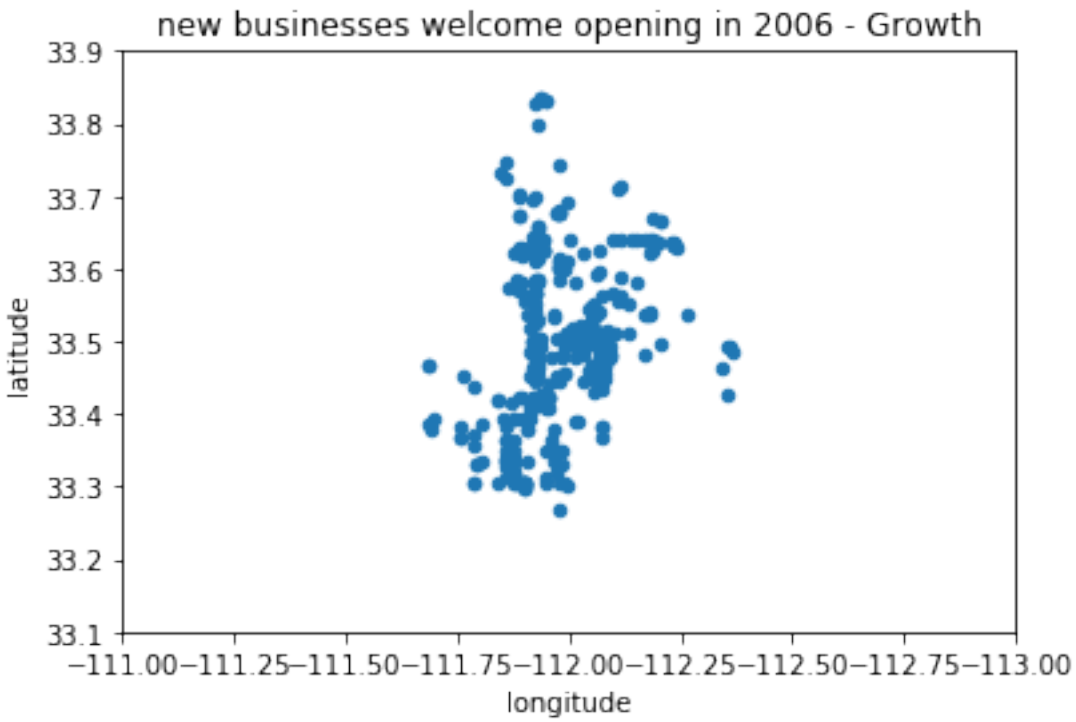
[34]:
```
arizonafoodbusiness['minyear'].unique()
```

[34]:
```
array([2011, 2012, 2008, 2016, 2006, 2009, 2007, 2010, 2017, 2013, 2014,
       2015, 2005])
```

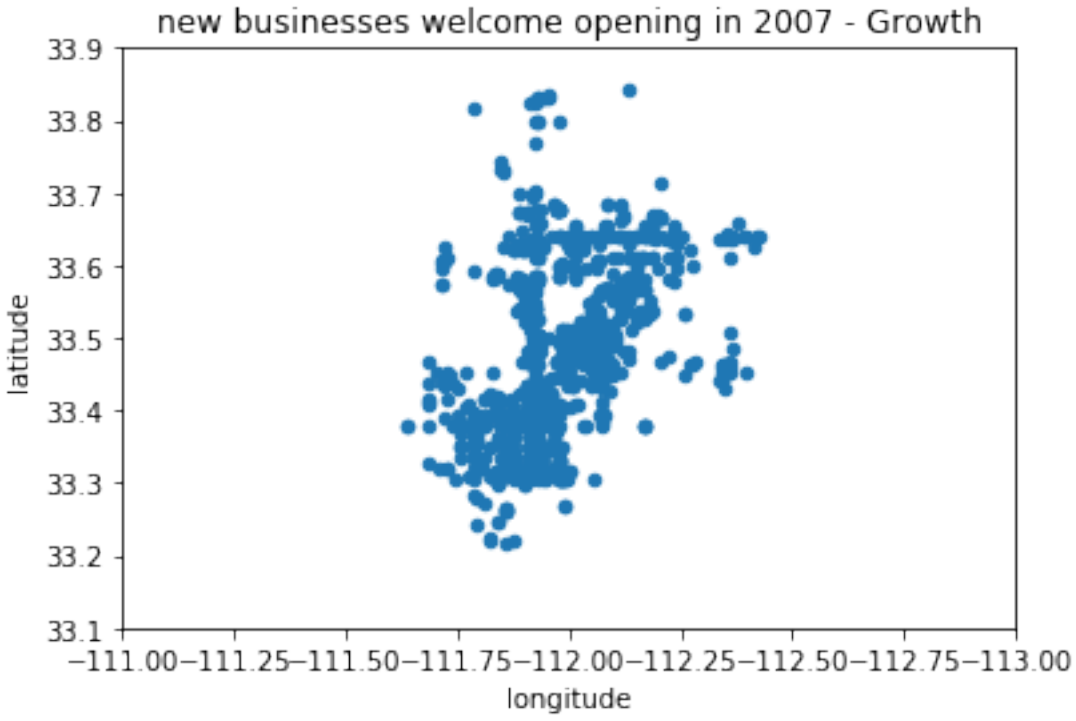
[35]:
```
for year in range(2005, 2018,1):
    dta = arizonafoodbusiness[arizonafoodbusiness['minyear'] == year]
    print(dta.shape[0])
    dta.plot(
    kind="scatter", x="longitude", y="latitude")
    plt.title('new businesses welcome opening in {0} - Growth'.format(year))
    plt.ylim(33.1, 33.9)
    plt.xlim(-111,
        -113)
    plt.show()
```

61

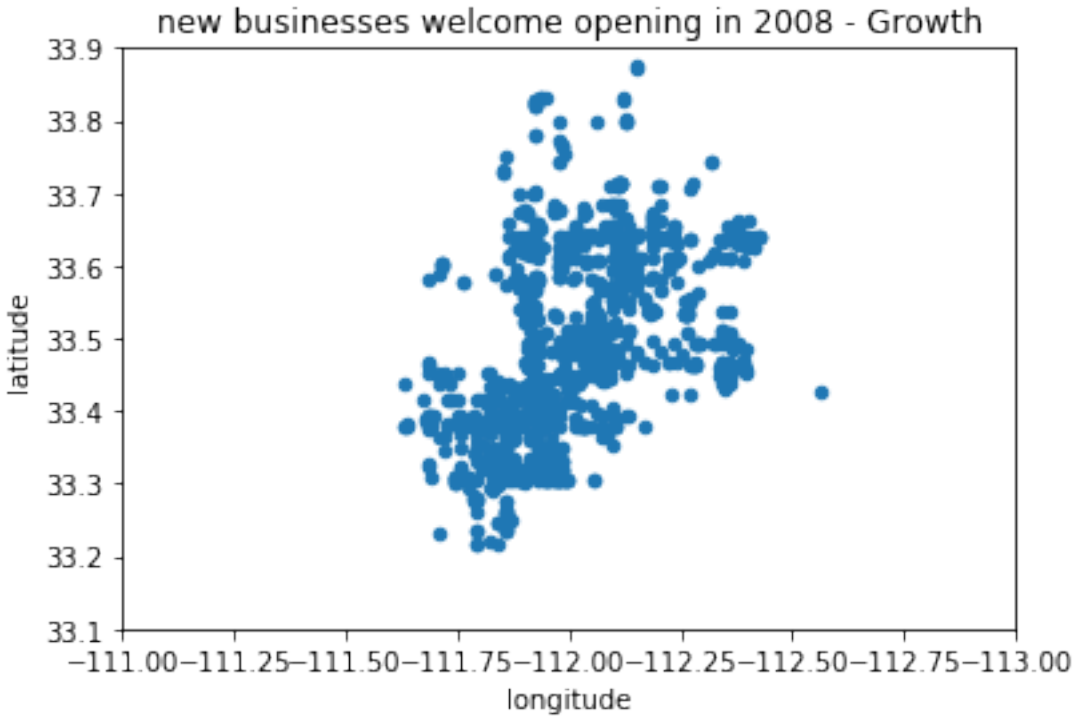
502


new businesses welcome opening in 2006 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

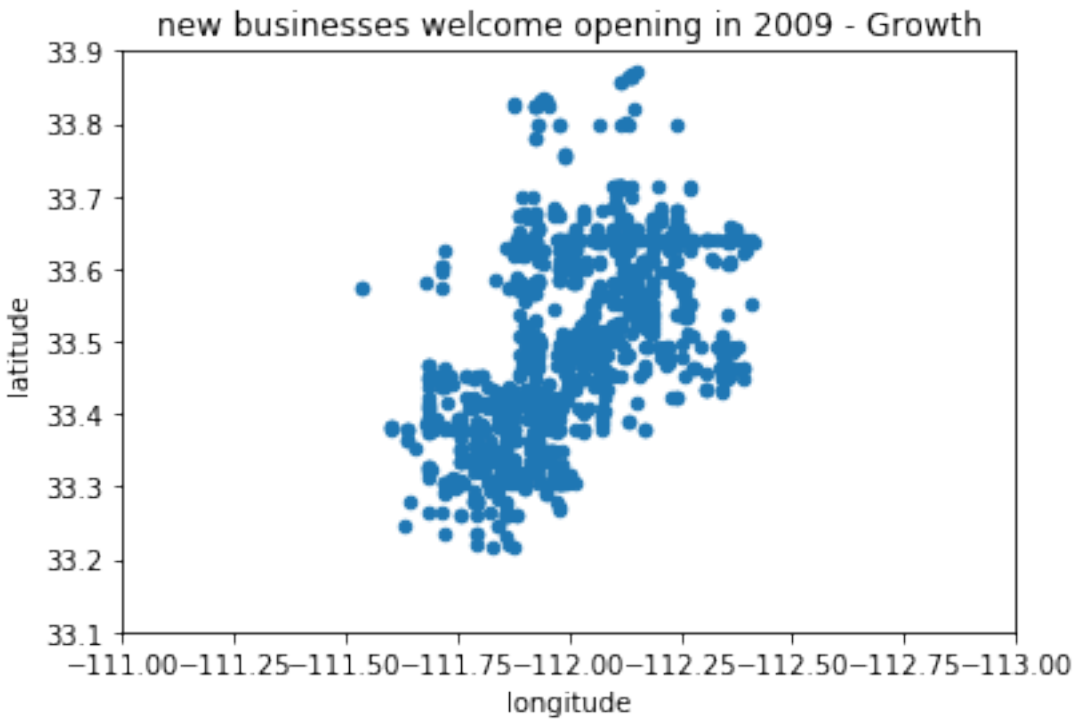
1210


new businesses welcome opening in 2007 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

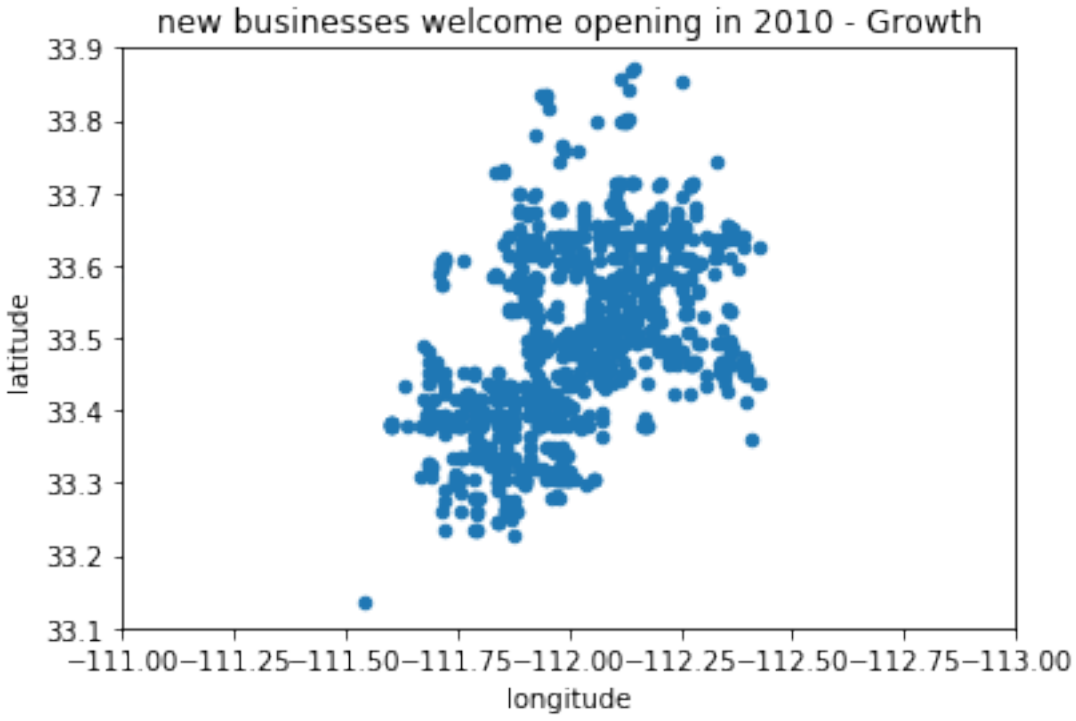
1361


new businesses welcome opening in 2008 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

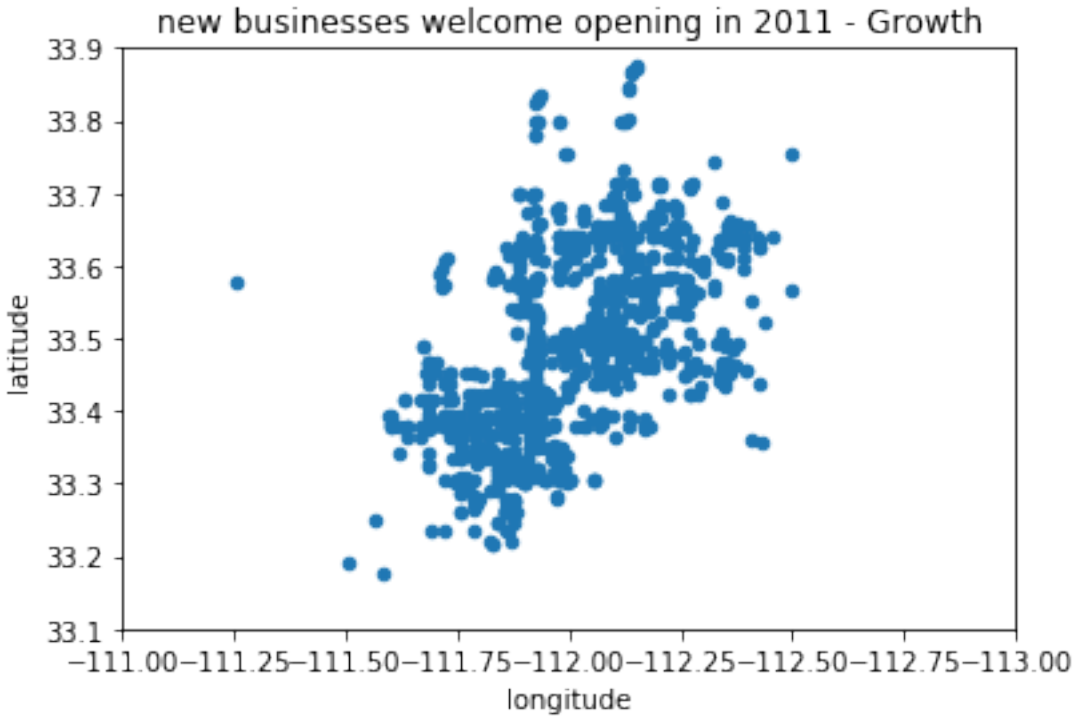
1220


new businesses welcome opening in 2009 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

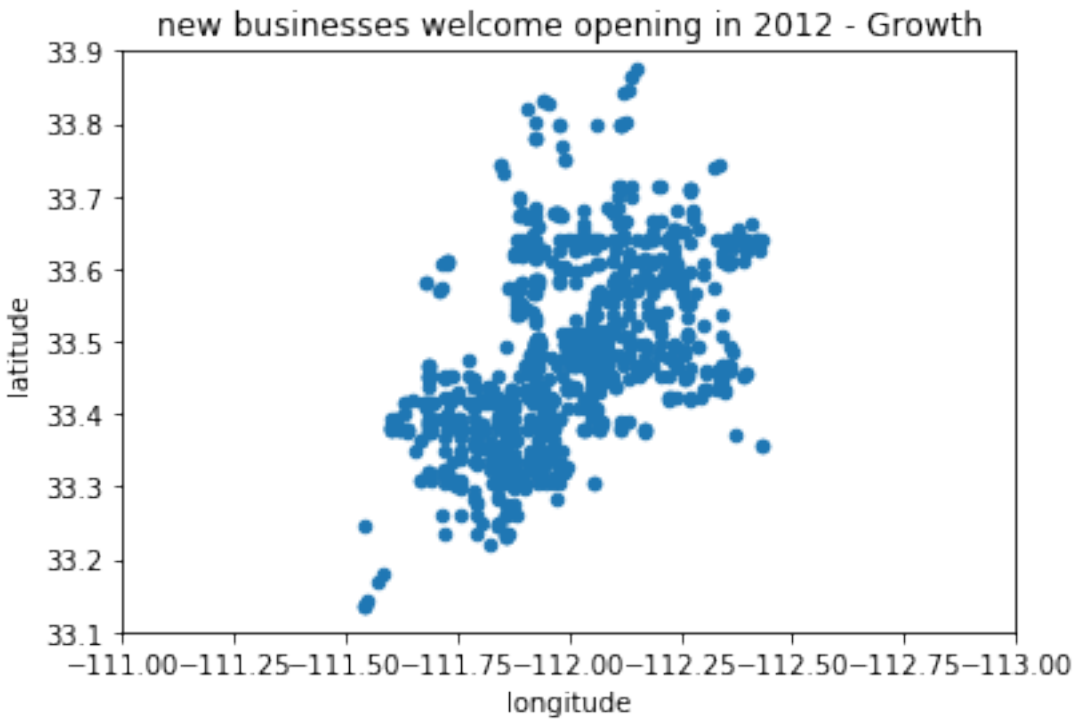
1178


new businesses welcome opening in 2010 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

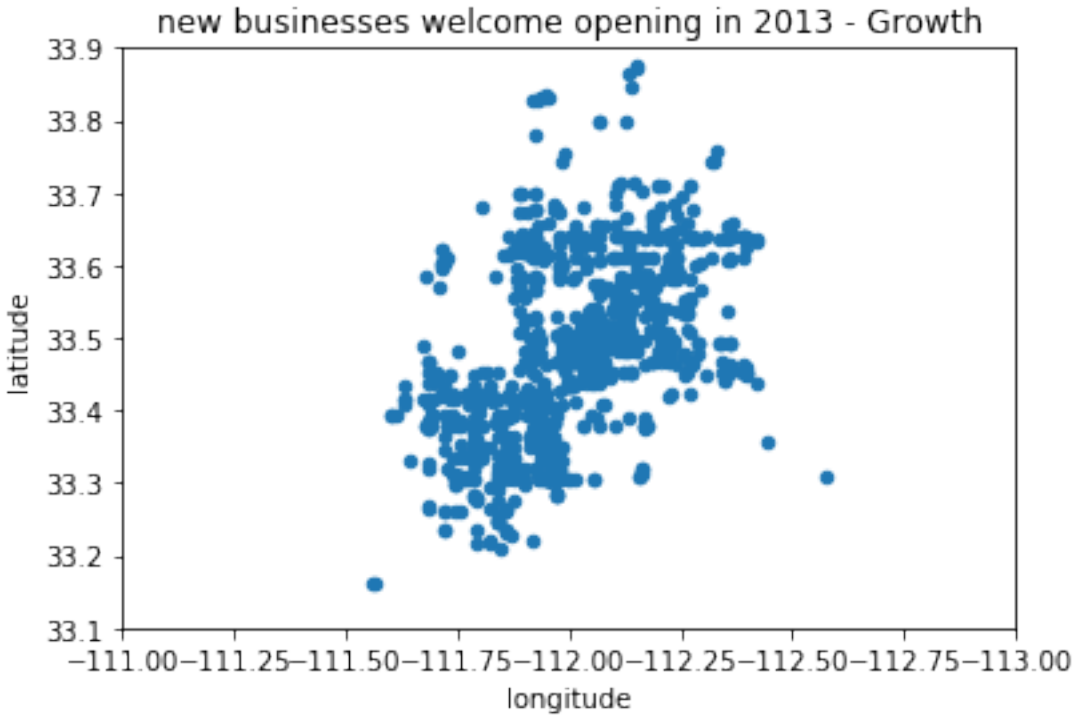
1159


new businesses welcome opening in 2011 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

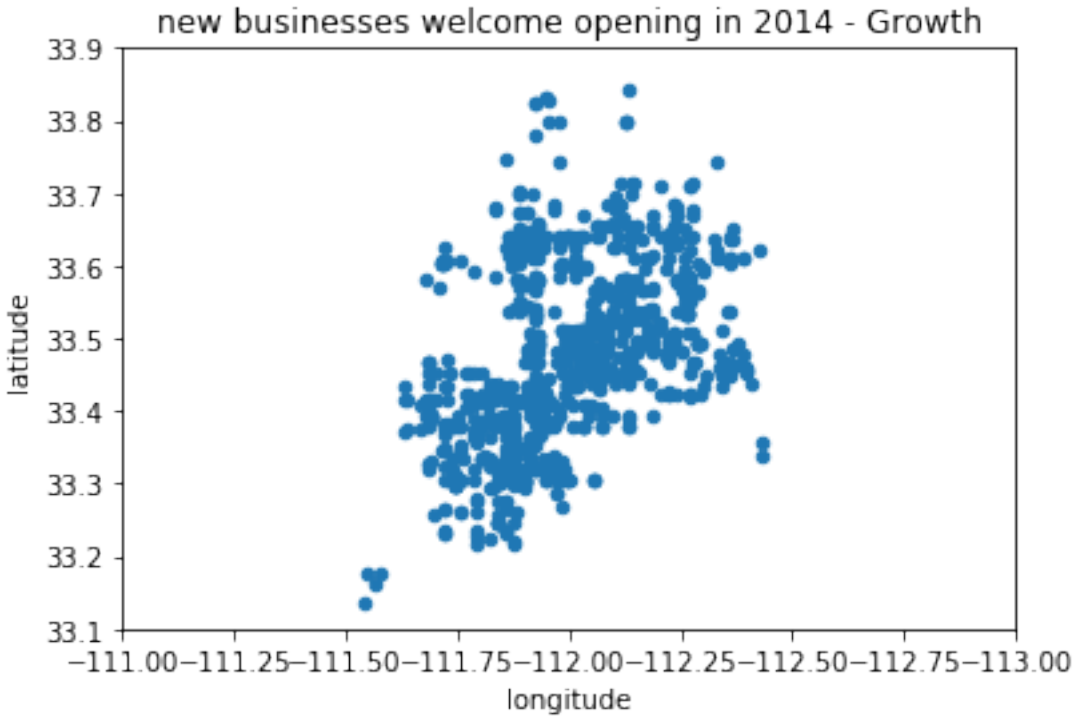
1079


new businesses welcome opening in 2012 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

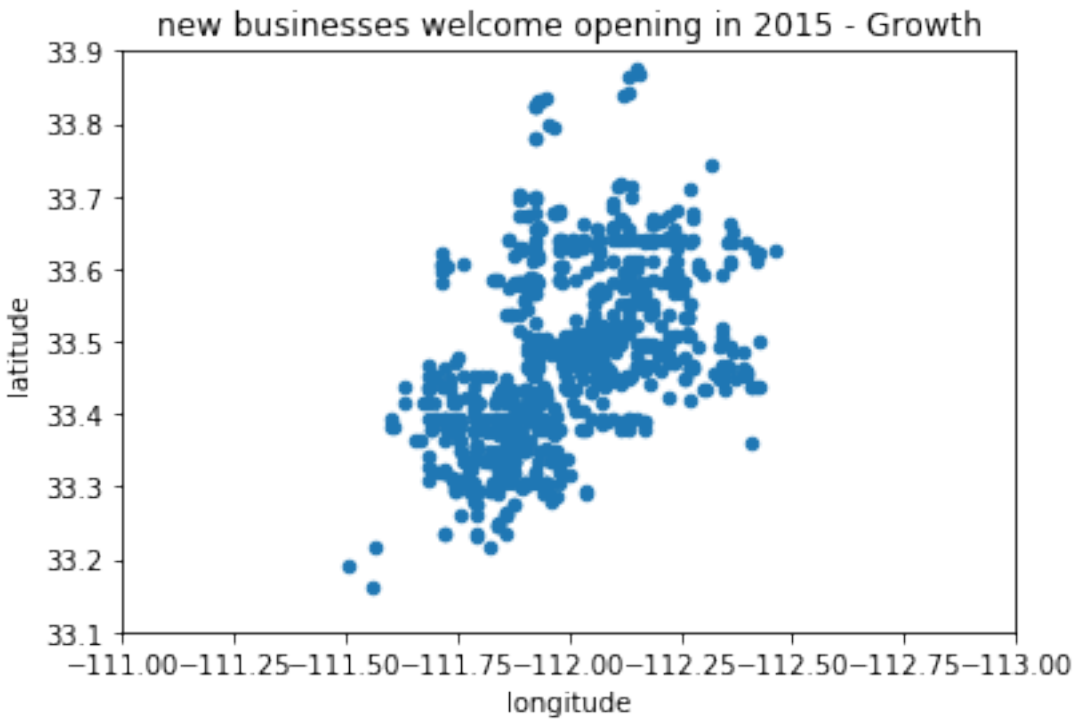
1044


new businesses welcome opening in 2013 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

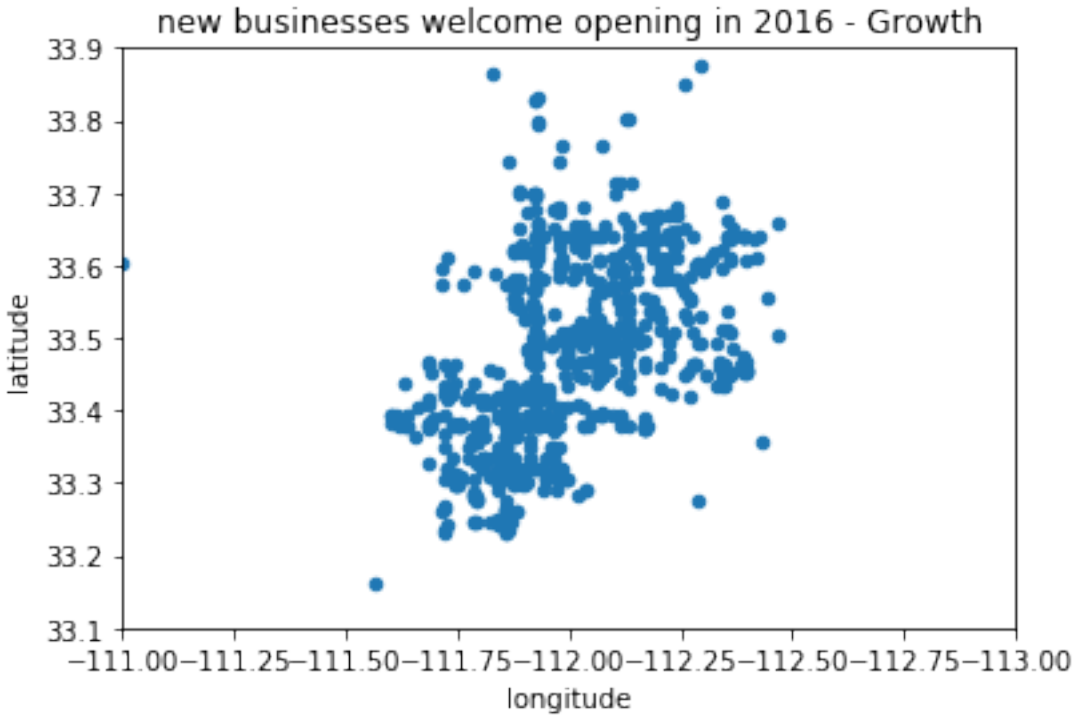
994


new businesses welcome opening in 2014 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

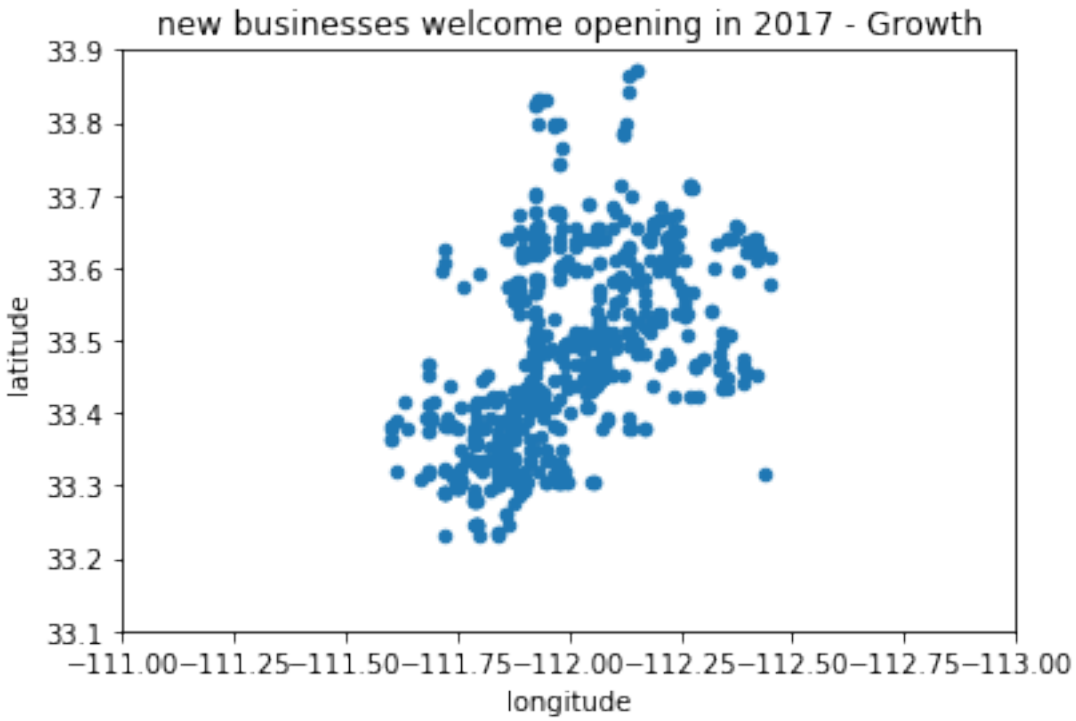
936


new businesses welcome opening in 2015 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

887

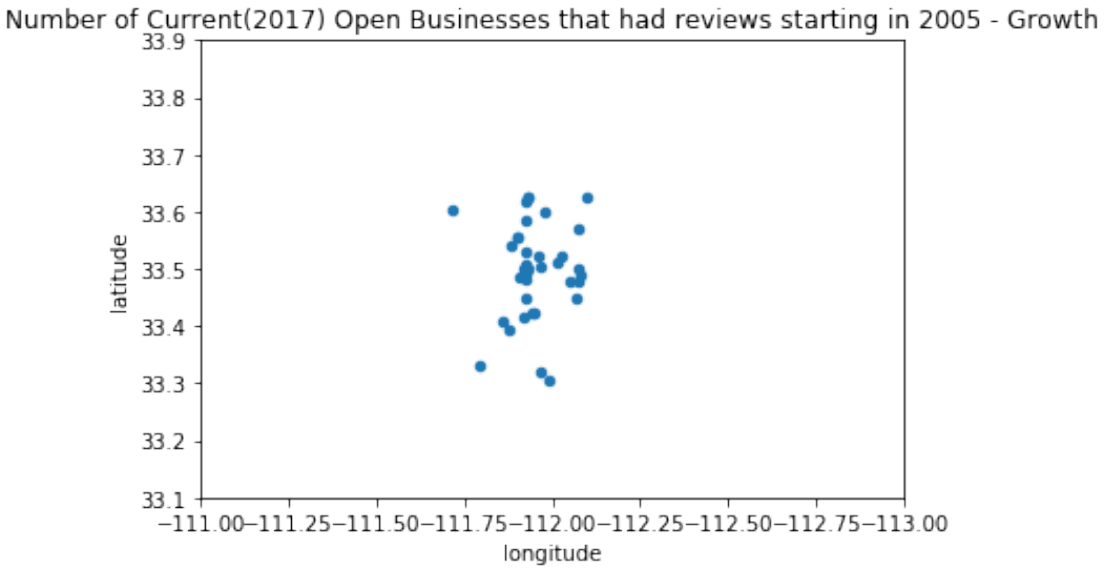

new businesses welcome opening in 2016 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

716

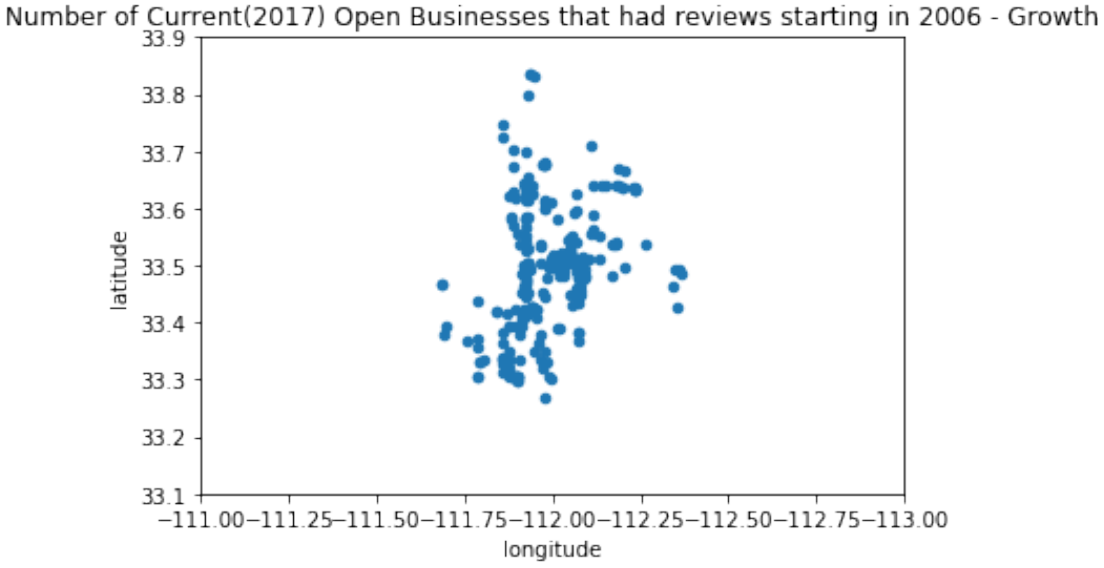


[36]:
```
for year in range(2005, 2018,1):
    dta = openazbusiness[openazbusiness['minyear'] == year]
    print(dta.shape[0])
    dta.plot(
    kind="scatter", x="longitude", y="latitude")
    plt.title('Number of Current(2017) Open Businesses that had reviews starting␣
 ↪in {0} - Growth'.format(year))
    plt.ylim(33.1, 33.9)
    plt.xlim(-111,
        -113)
    plt.show()
```

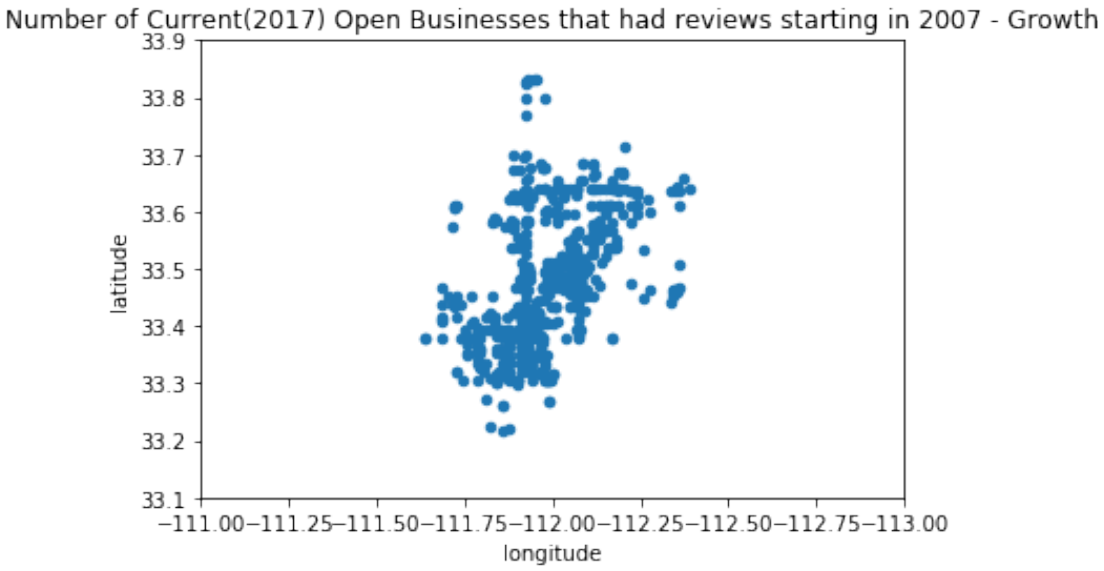
39


Number of Current(2017) Open Businesses that had reviews starting in 2005 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

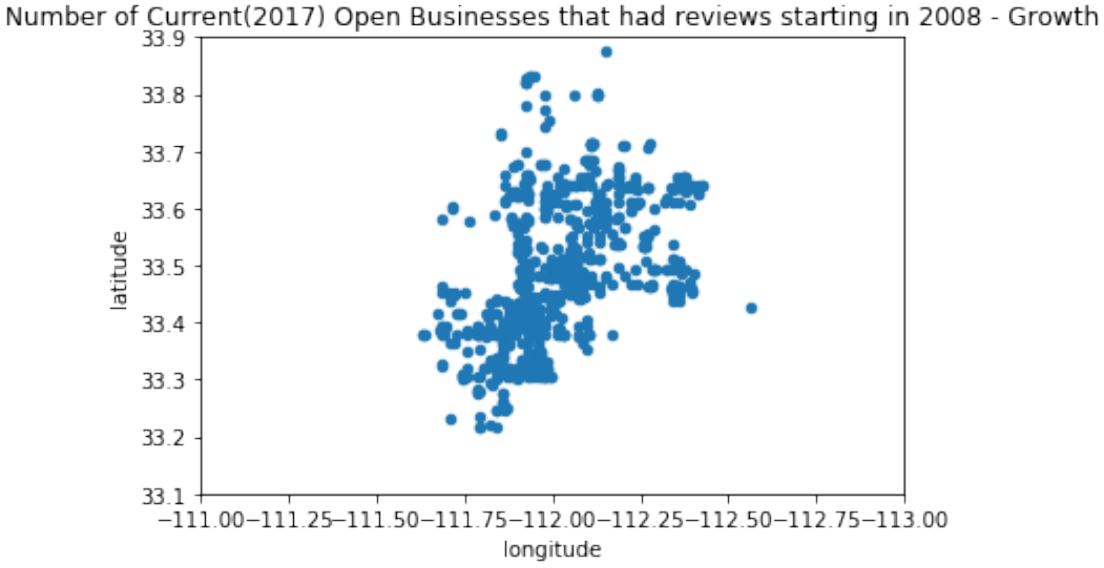
311


Number of Current(2017) Open Businesses that had reviews starting in 2006 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

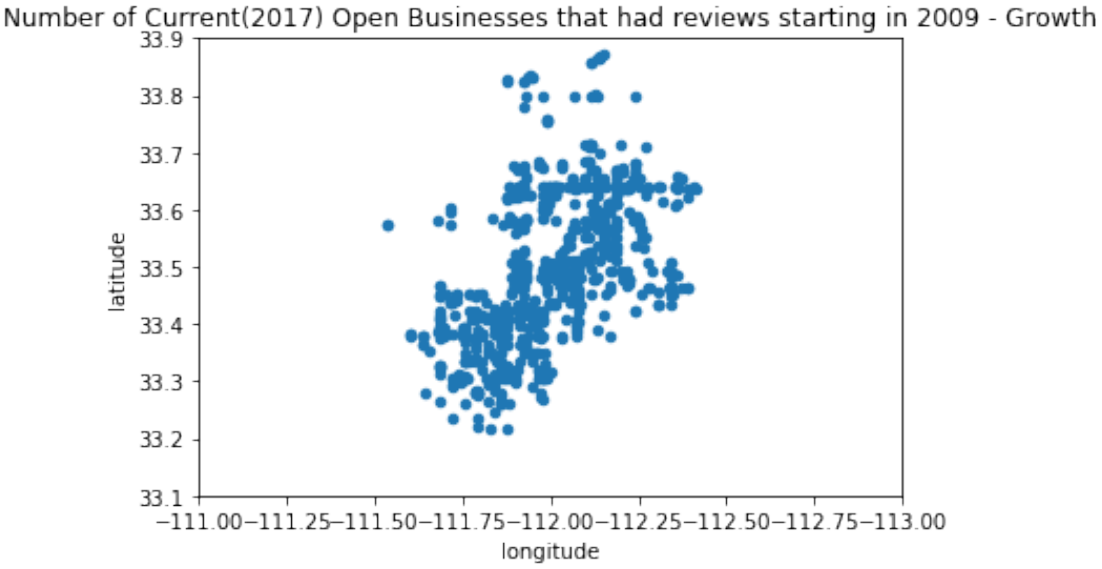
690


Number of Current(2017) Open Businesses that had reviews starting in 2007 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

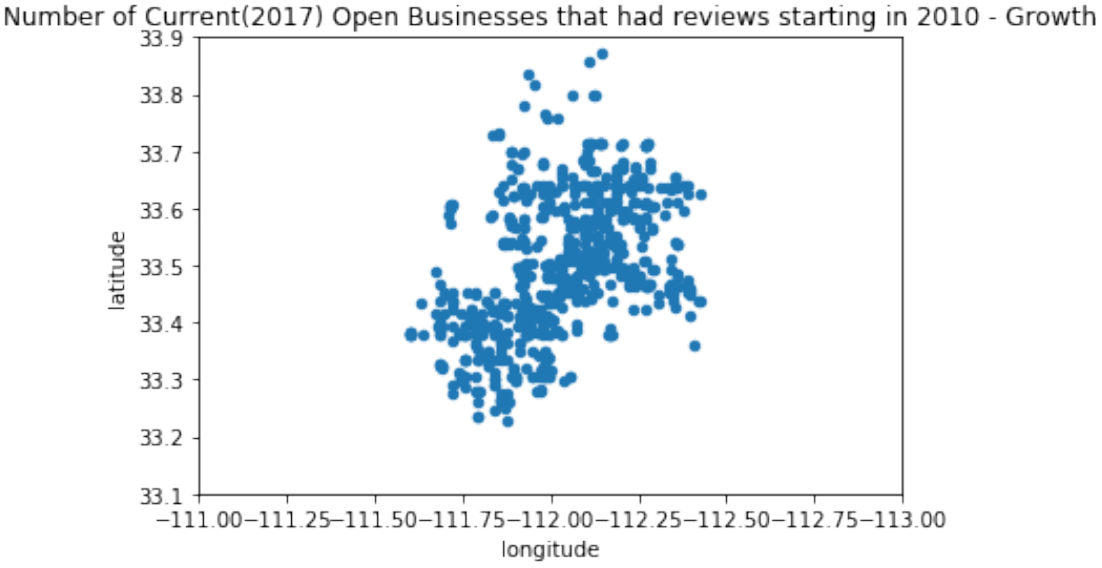
799


Number of Current(2017) Open Businesses that had reviews starting in 2008 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

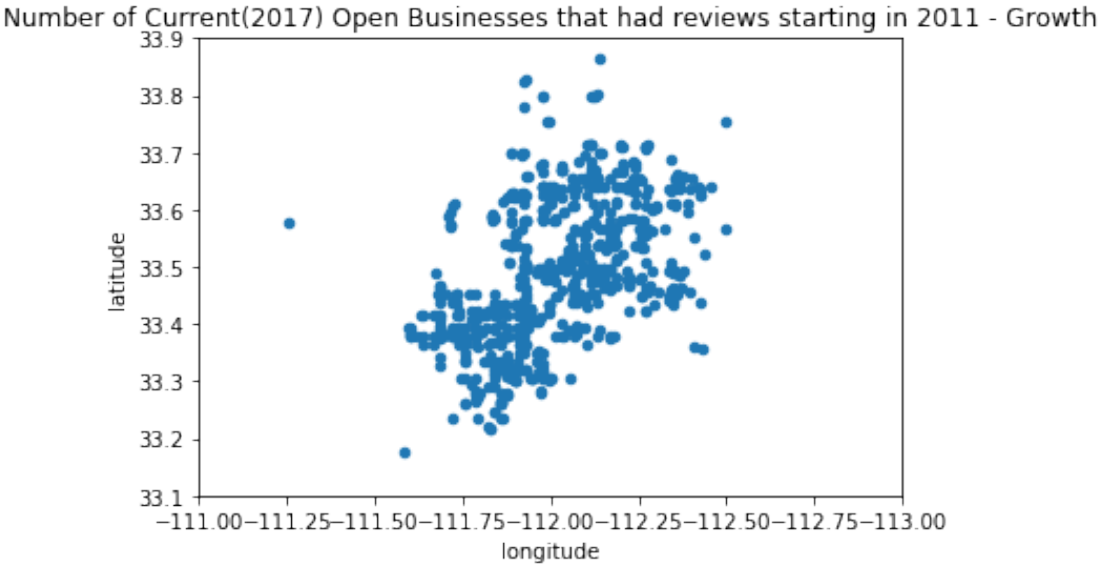
765


Number of Current(2017) Open Businesses that had reviews starting in 2009 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

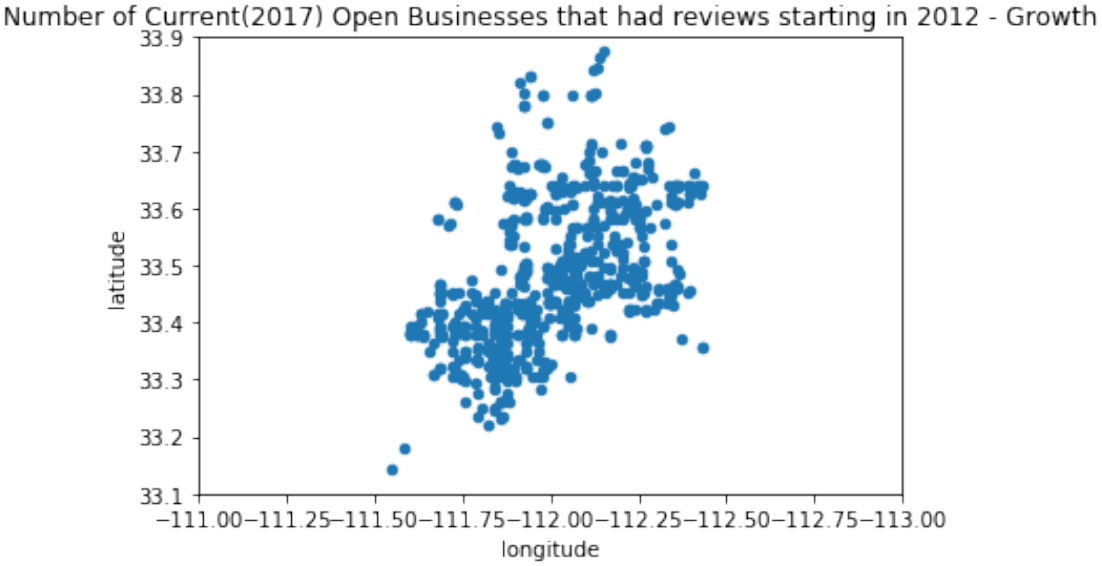
730


Number of Current(2017) Open Businesses that had reviews starting in 2010 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

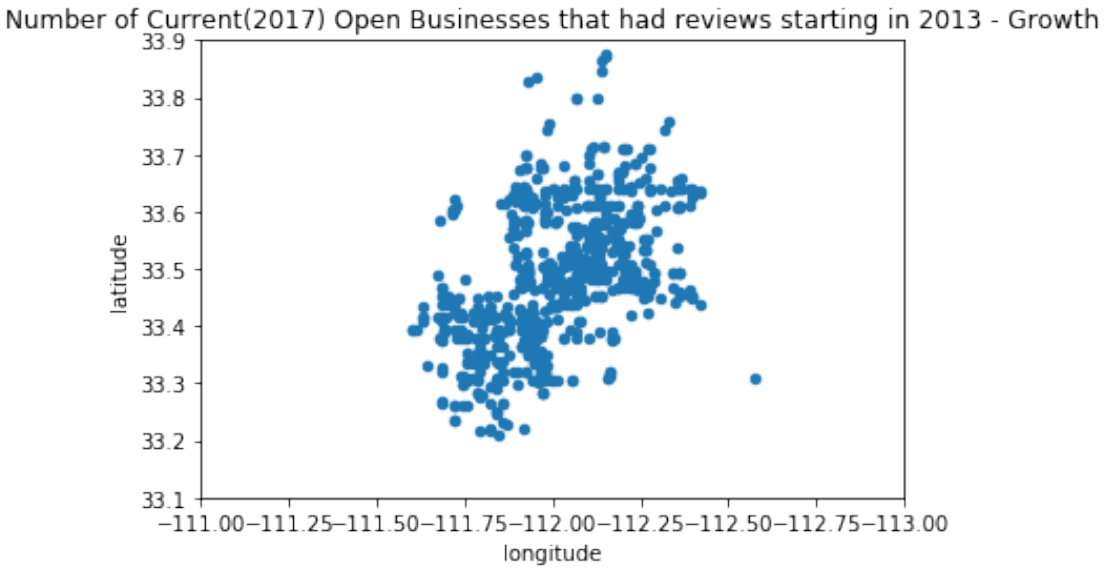
774


Number of Current(2017) Open Businesses that had reviews starting in 2011 - Growth
latitude
longitude
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00

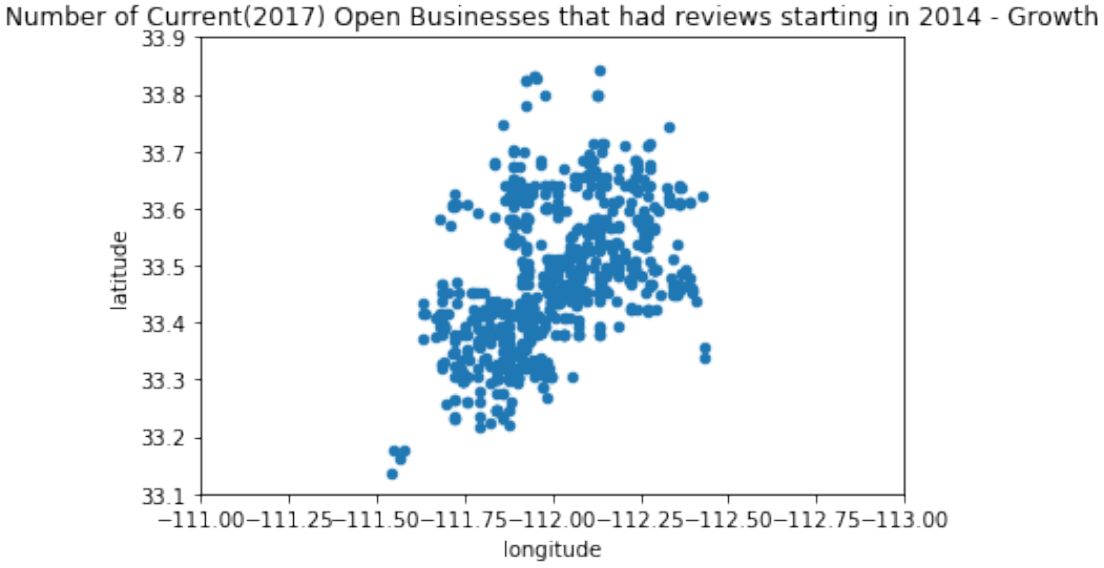
741


Number of Current(2017) Open Businesses that had reviews starting in 2012 - Growth
latitude
longitude
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00

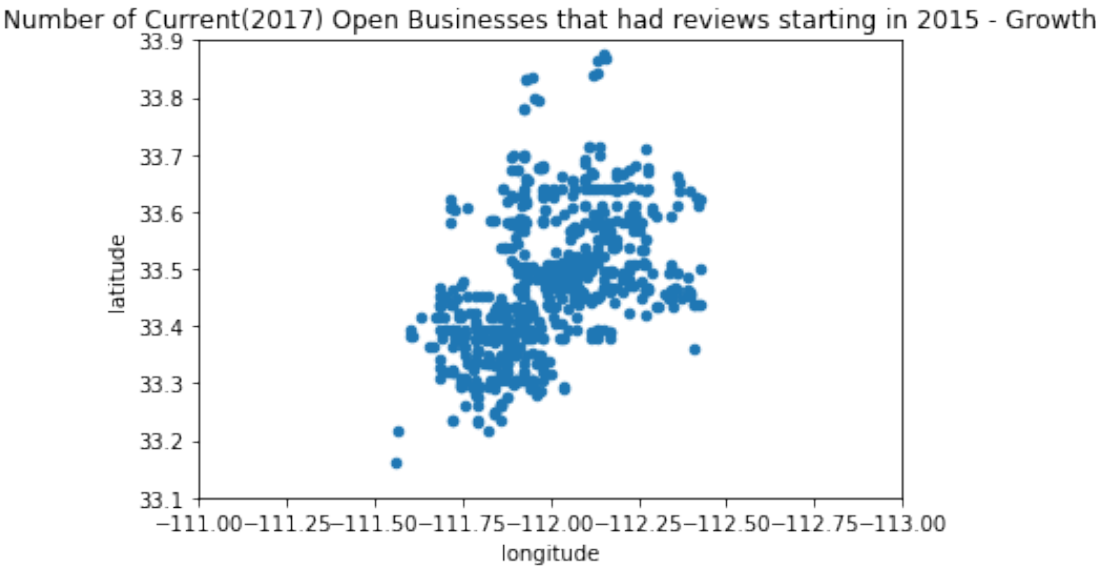
764


Number of Current(2017) Open Businesses that had reviews starting in 2013 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

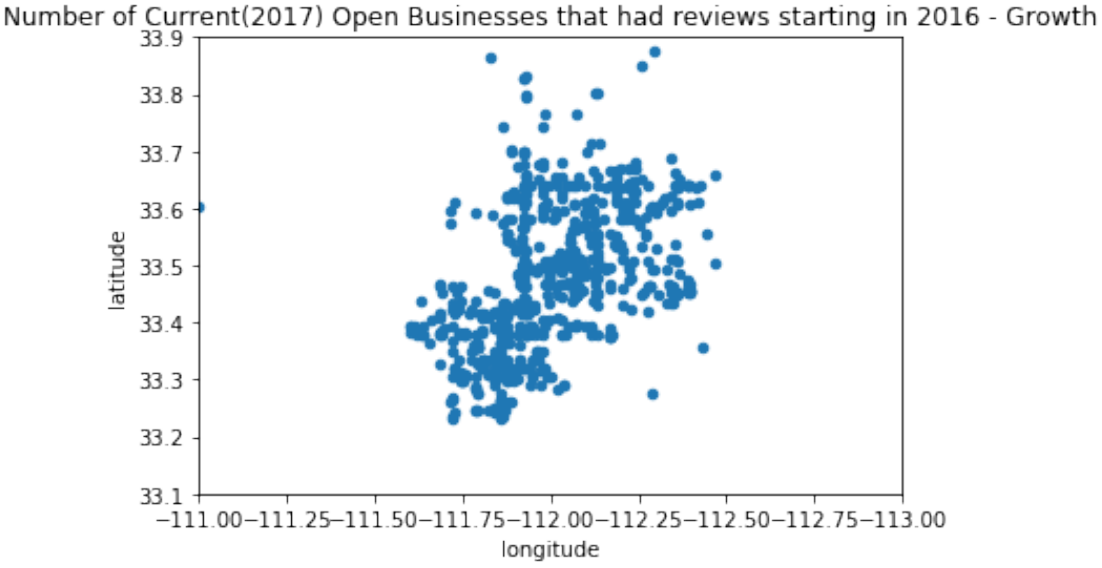
762


Number of Current(2017) Open Businesses that had reviews starting in 2014 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00−111.25−111.50−111.75−112.00−112.25−112.50−112.75−113.00
longitude

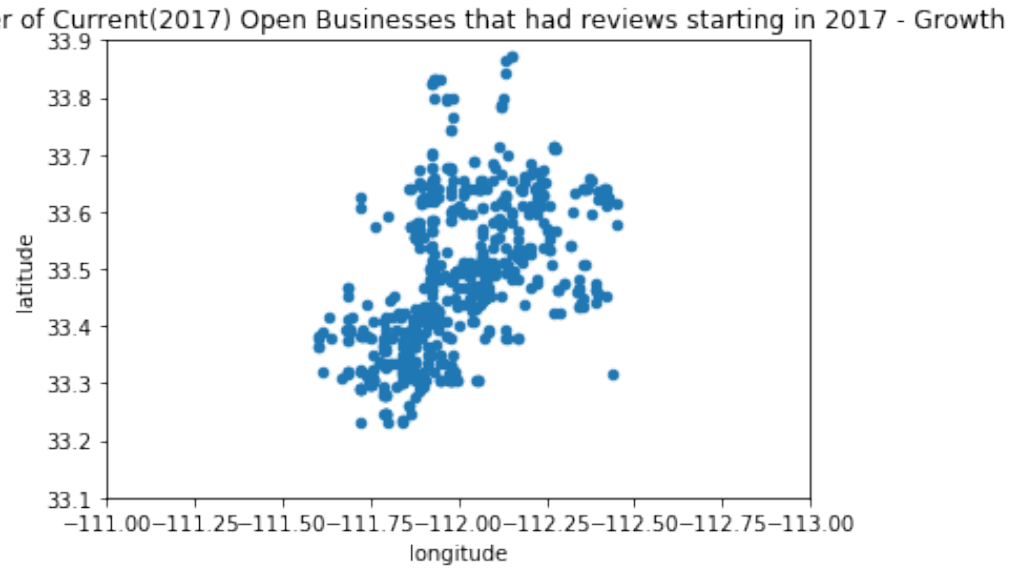
776


Number of Current(2017) Open Businesses that had reviews starting in 2015 - Growth
latitude
longitude
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00

780

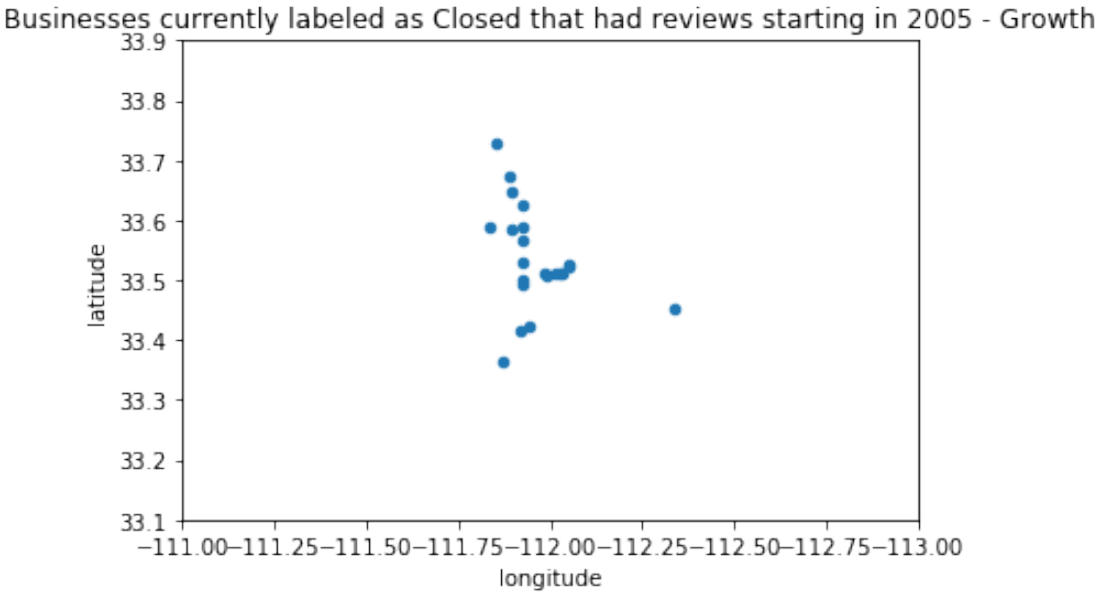

Number of Current(2017) Open Businesses that had reviews starting in 2016 - Growth
latitude
longitude
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00

691

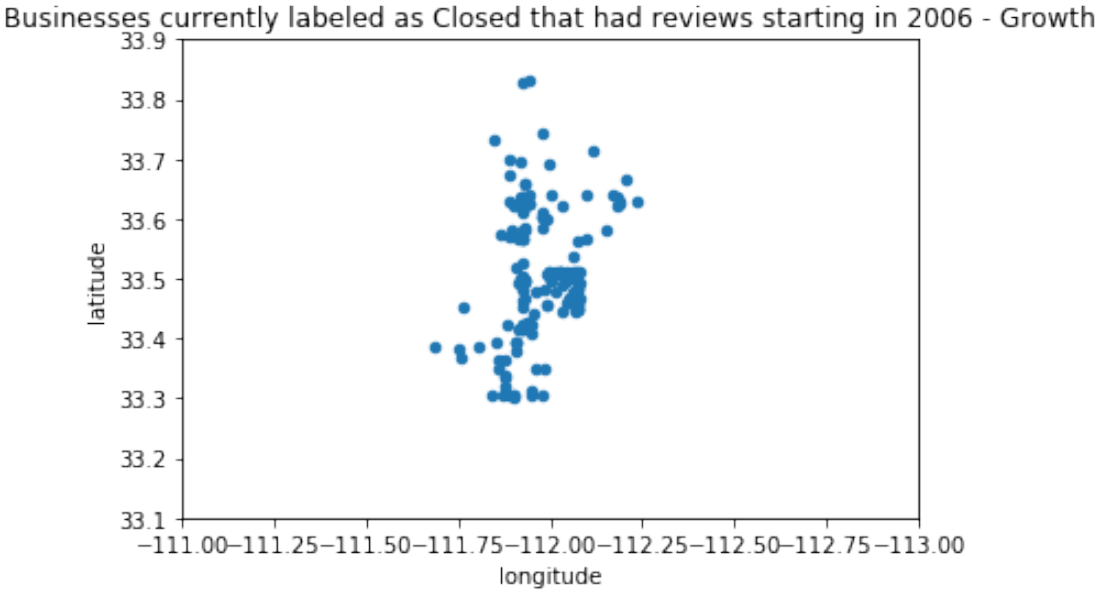


[37]:
```
for year in range(2005, 2018,1):
    dta = closedazbusiness[closedazbusiness['minyear'] == year]
    print(dta.shape[0])
    dta.plot(
    kind="scatter", x="longitude", y="latitude")
    plt.title('Businesses currently labeled as Closed that had reviews starting␣
 ↪in {0} - Growth'.format(year))
    plt.ylim(33.1, 33.9)
    plt.xlim(-111,
        -113)
    plt.show()
```

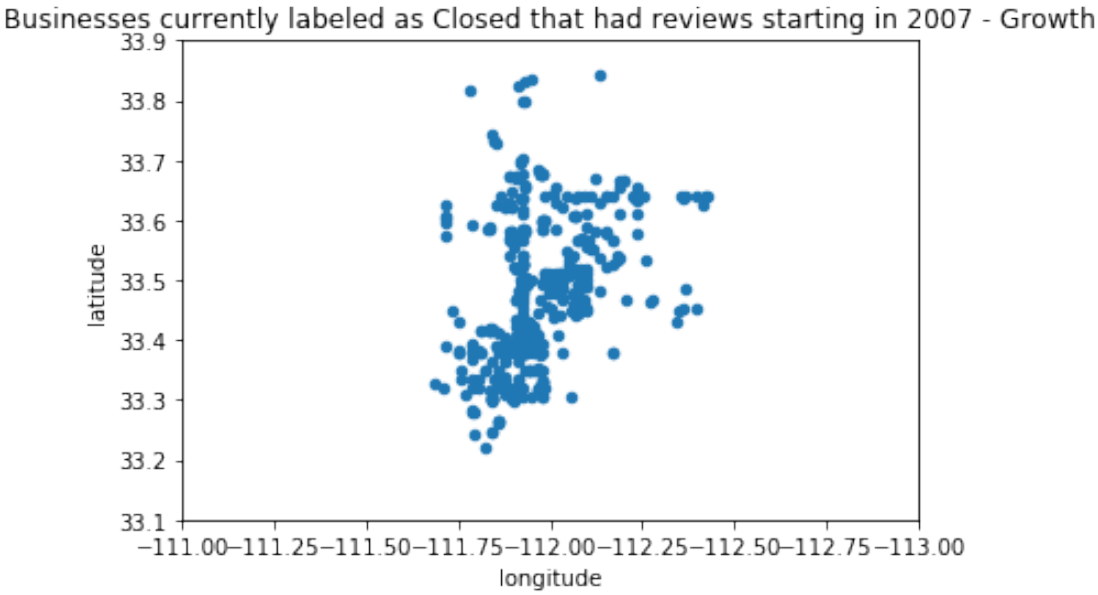
22


Businesses currently labeled as Closed that had reviews starting in 2005 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

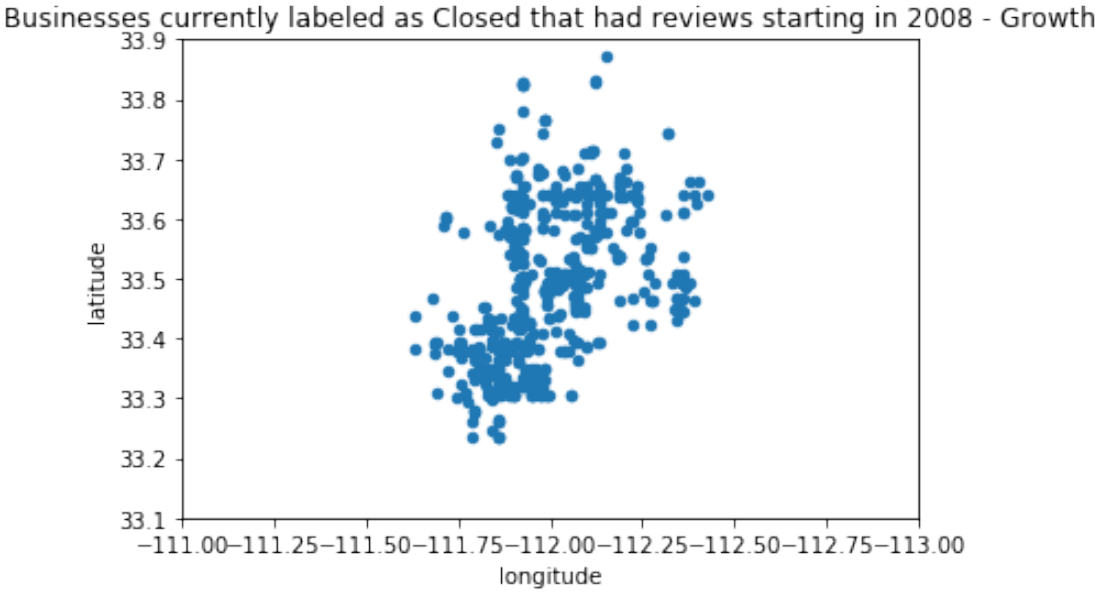
191


Businesses currently labeled as Closed that had reviews starting in 2006 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

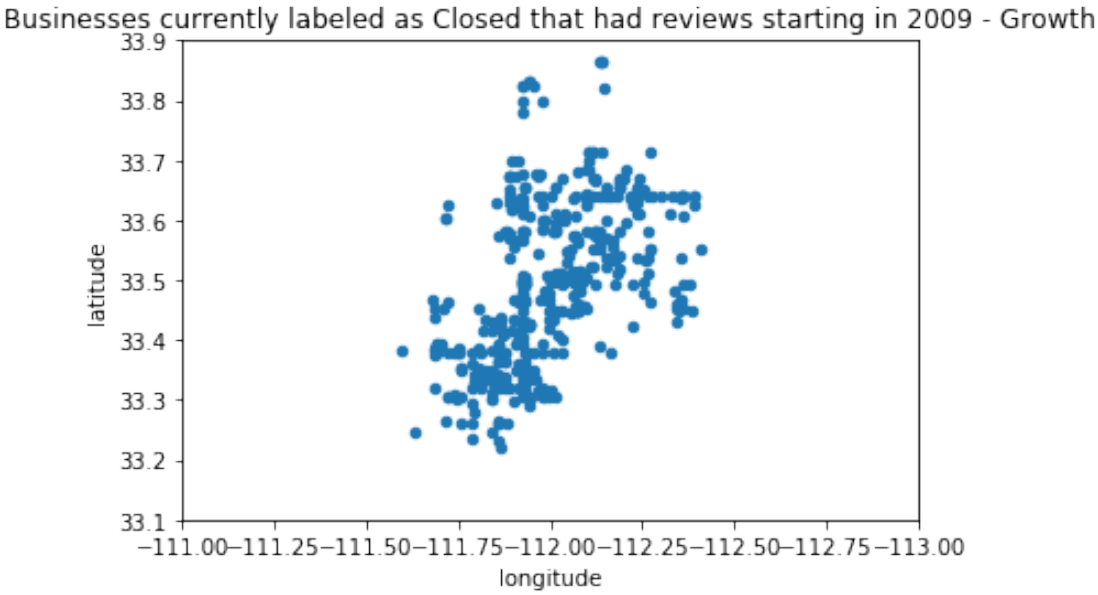
520


Businesses currently labeled as Closed that had reviews starting in 2007 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

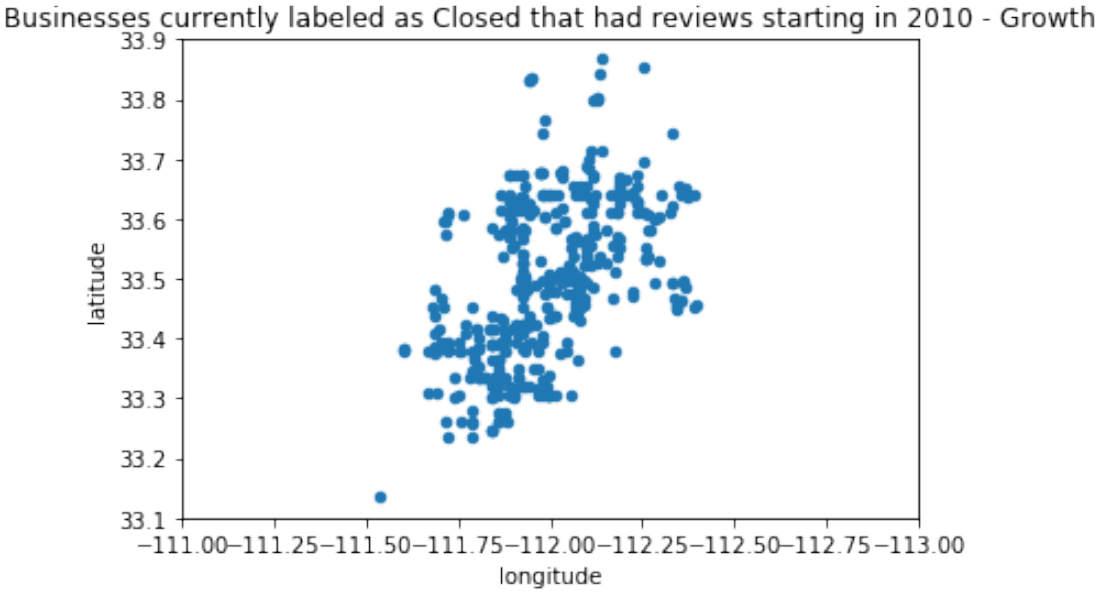
562


Businesses currently labeled as Closed that had reviews starting in 2008 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

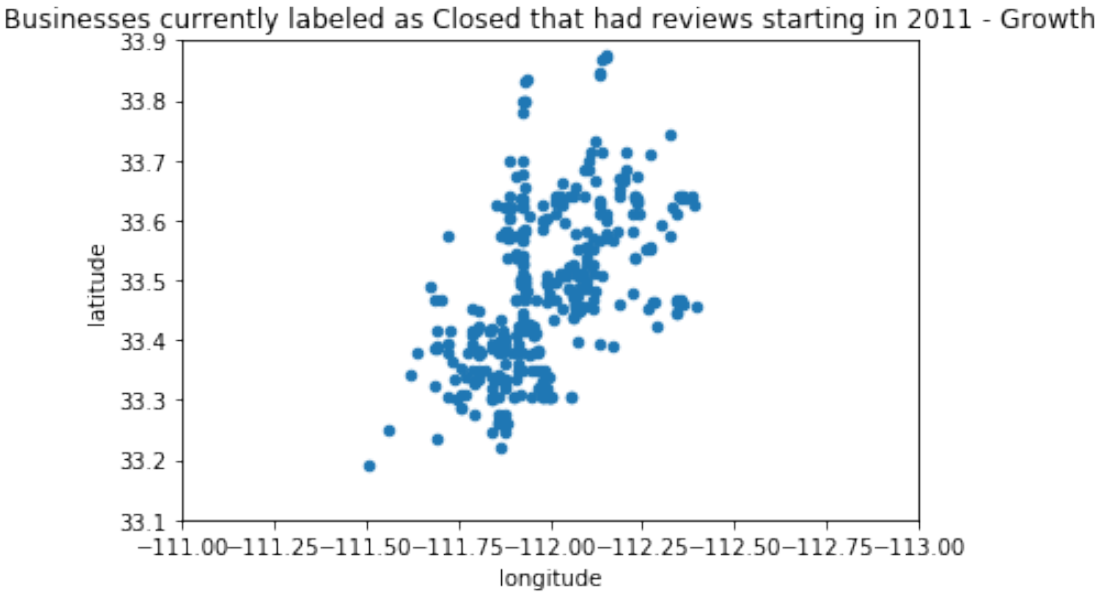
455


Businesses currently labeled as Closed that had reviews starting in 2009 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

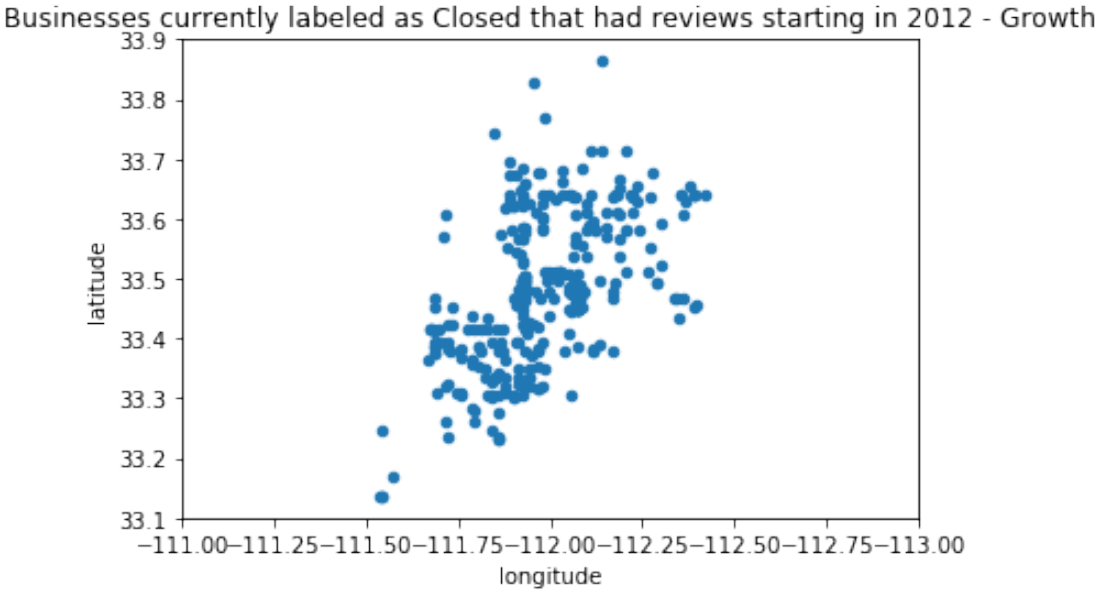
448


Businesses currently labeled as Closed that had reviews starting in 2010 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

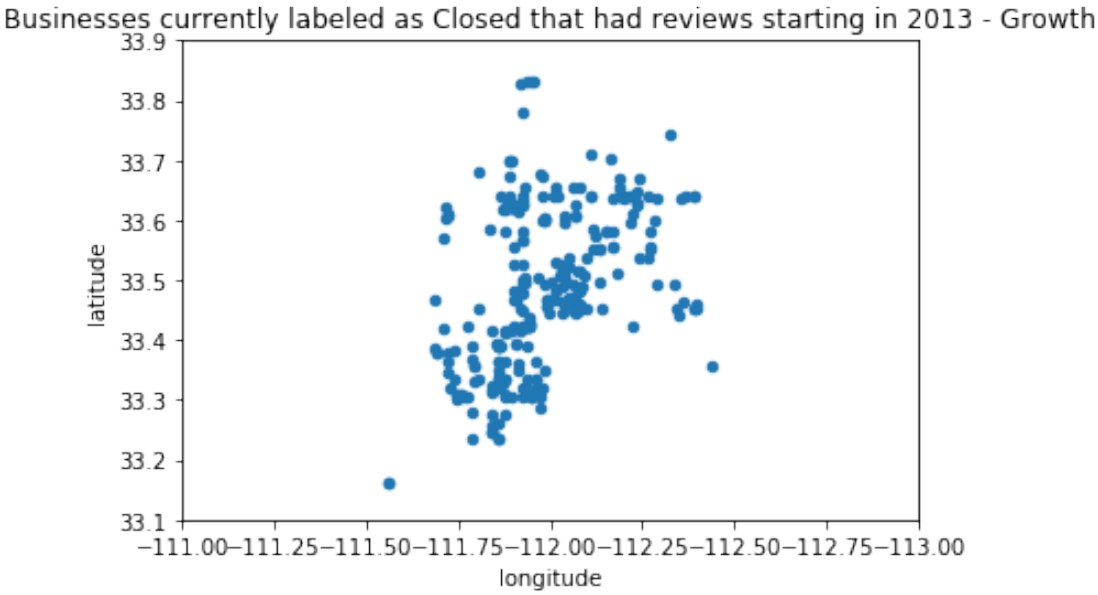
385


Businesses currently labeled as Closed that had reviews starting in 2011 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

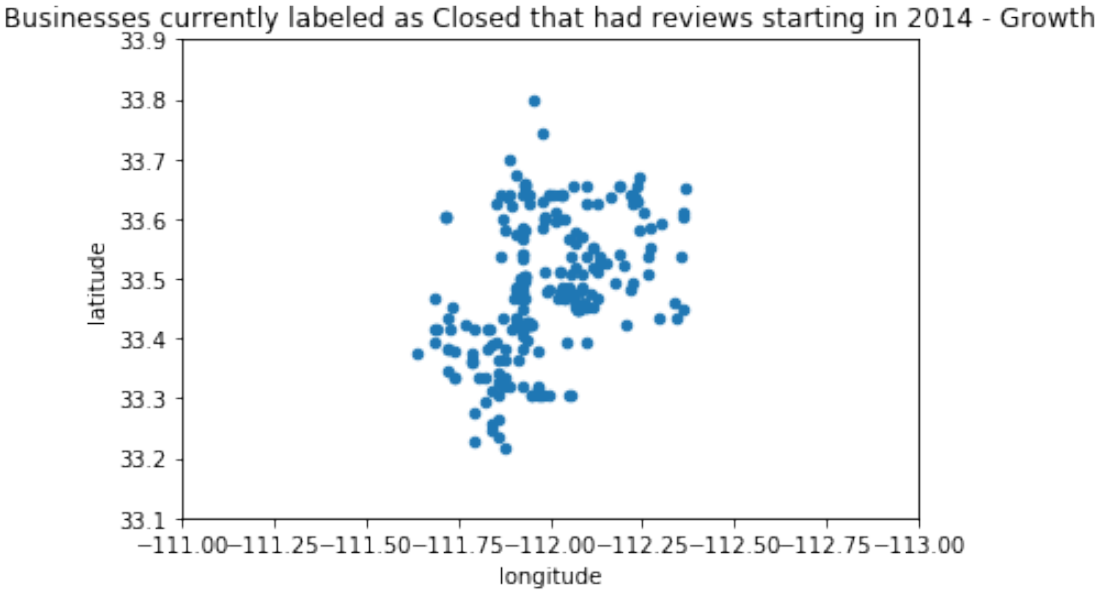
338


Businesses currently labeled as Closed that had reviews starting in 2012 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

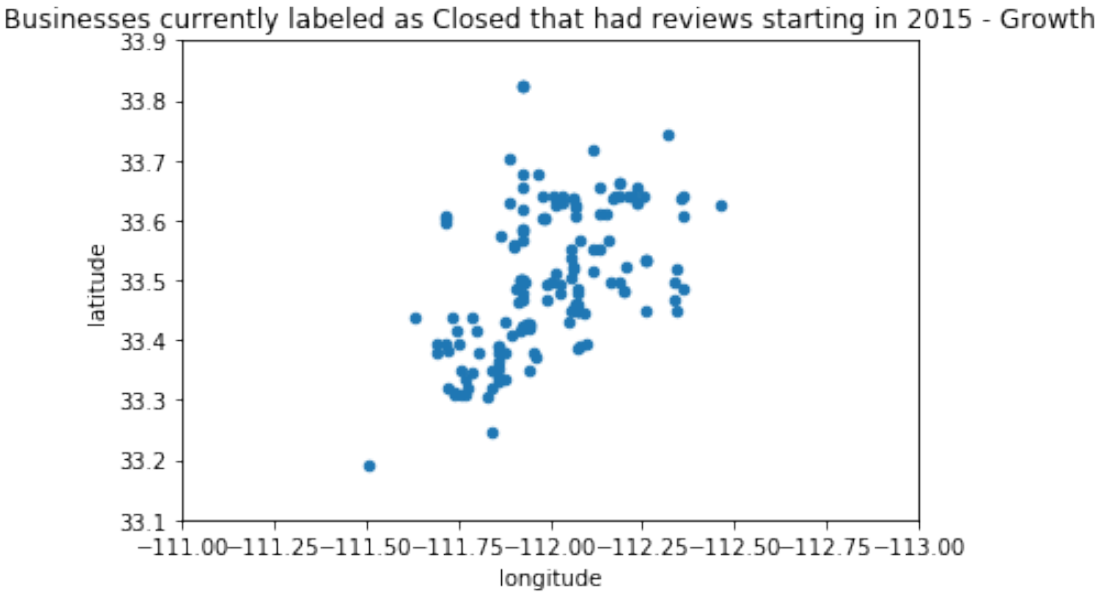
280


Businesses currently labeled as Closed that had reviews starting in 2013 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

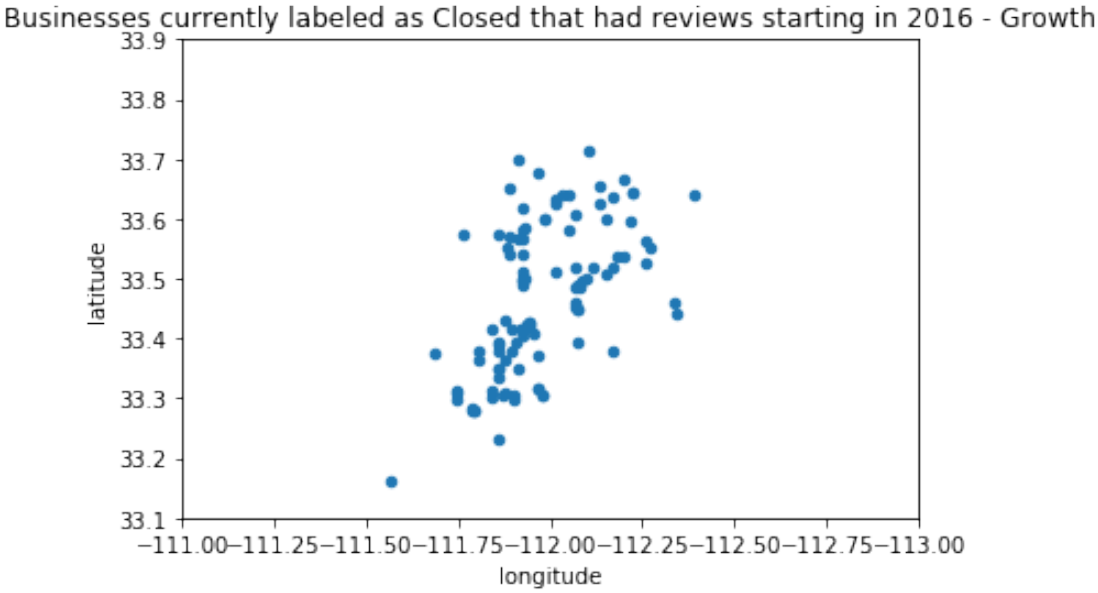
232


Businesses currently labeled as Closed that had reviews starting in 2014 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

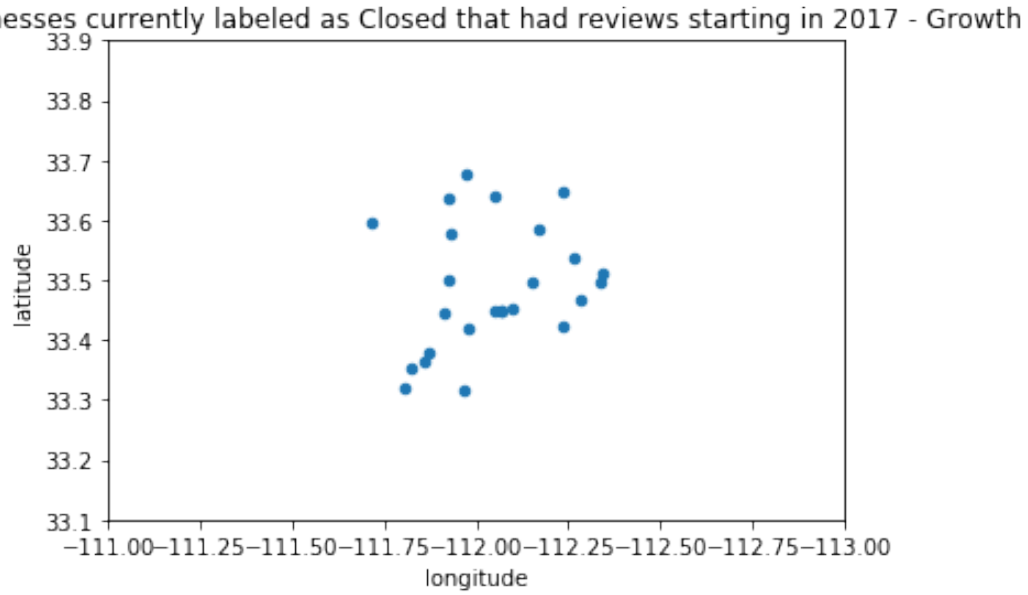
160


Businesses currently labeled as Closed that had reviews starting in 2015 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

107


Businesses currently labeled as Closed that had reviews starting in 2016 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−111.00
−111.25
−111.50
−111.75
−112.00
−112.25
−112.50
−112.75
−113.00
longitude

25



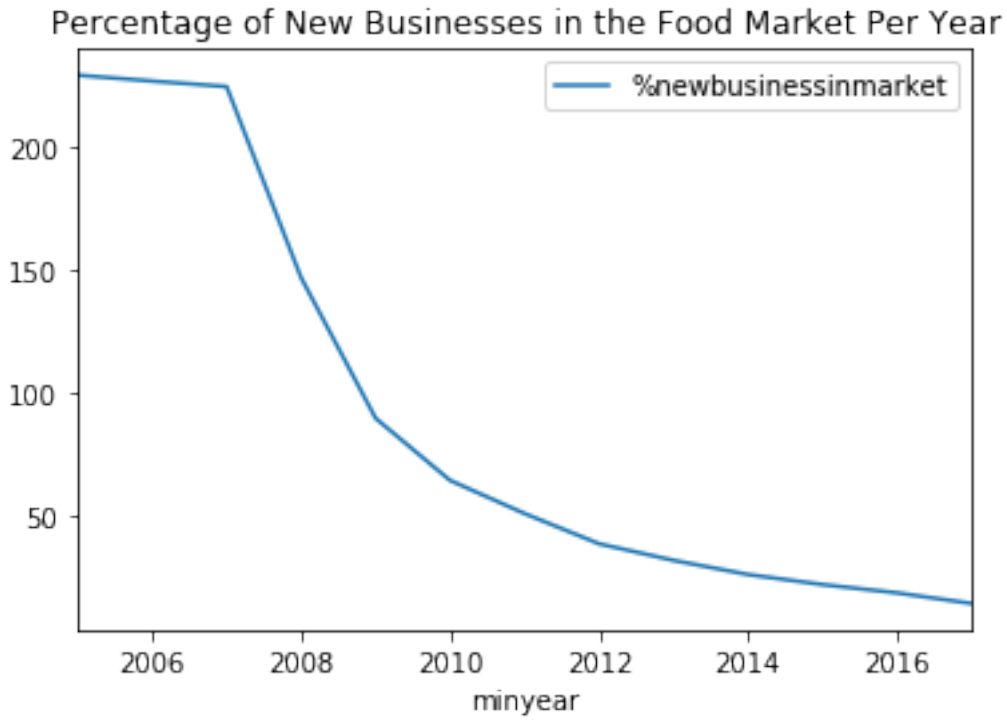
New Restaurant Opening

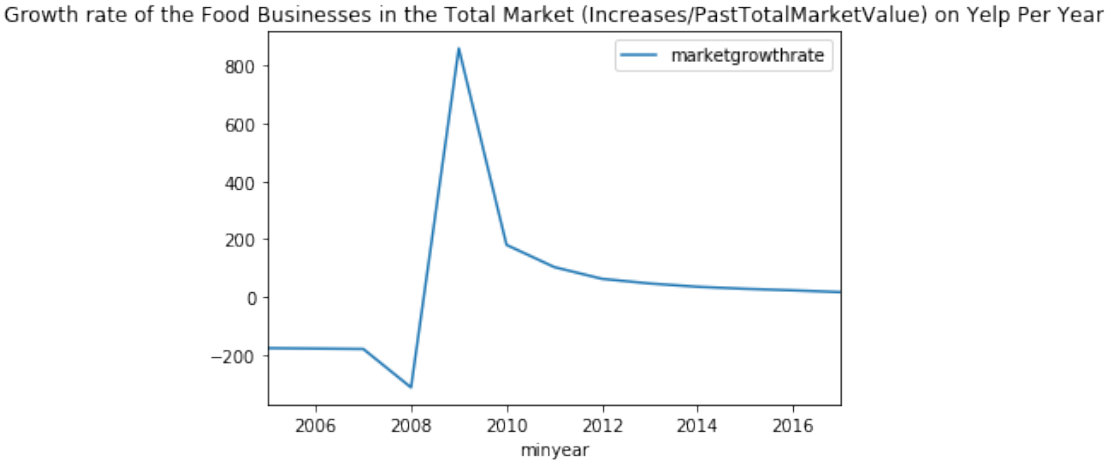
[38]:
```
minyearcount = arizonafoodbusiness.groupby(
    ['is_open','minyear']).count().reset_index().rename(
    columns = {'business_id': 'count of new businesses'})[['is_open', 'minyear',␣
 ↪'count of new businesses']]

#ex: 2015 cumsum
#cumsum(count) for all years in 2015 and before until 2005

def cumsum(year):
    pos = sum(
        minyearcount[minyearcount['minyear'] <= year][minyearcount['is_open'] ==␣
 ↪1]['count of new businesses'])
    neg = sum(
        minyearcount[minyearcount['minyear'] <= year][minyearcount['is_open'] ==␣
 ↪0]['count of new businesses'])
    return pos - neg

cumsumyear = dict(zip(range(2005, 2018,1),[cumsum(x) for x in range(2005,␣
 ↪2018,1)]))
newbusiness = minyearcount[minyearcount['is_open'] == 1]
newbusiness['totalbusinessmarket'] = newbusiness['minyear'].map(cumsumyear)
newbusiness['marketgrowthrate'] = newbusiness['count of new businesses']/
 ↪(newbusiness['totalbusinessmarket']-newbusiness['count of new businesses'])␣
 ↪*100
newbusiness['%newbusinessinmarket'] = newbusiness['count of new businesses']/
 ↪newbusiness['totalbusinessmarket'] *100
```

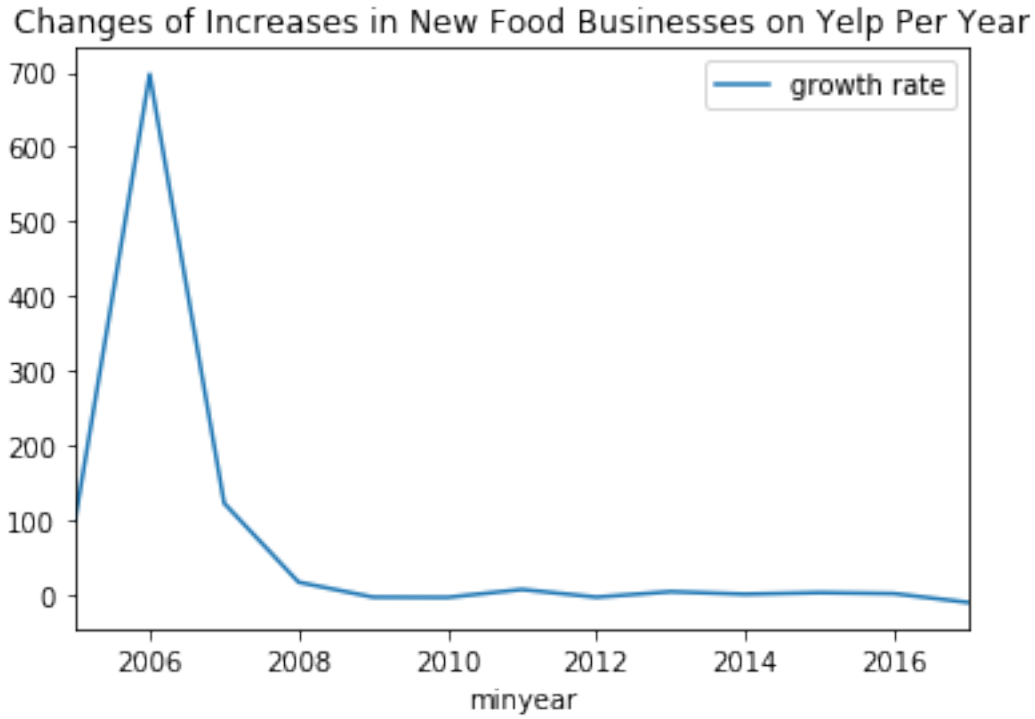
```
def newbusinessgr(year):
    data = sum(newbusiness[newbusiness['minyear'] == year]['count of new␣
 ↪businesses'])
    data2 = sum(newbusiness[newbusiness['minyear'] == year-1]['count of new␣
 ↪businesses'])
    return ((data - data2)/data2) * 100

newbusiness['growth rate'] = newbusiness['minyear'].apply(lambda x:␣
 ↪newbusinessgr(x) if x != 2005 else 100)

newbusiness.plot(x = 'minyear', y = '%newbusinessinmarket')
plt.title('Percentage of New Businesses in the Food Market Per Year')
plt.show()

newbusiness.plot(x = 'minyear', y = 'marketgrowthrate')
plt.title('Growth rate of the Food Businesses in the Total Market (Increases/
 ↪PastTotalMarketValue) on Yelp Per Year')
plt.show()


newbusiness.plot(x = 'minyear', y = 'growth rate')
plt.title('Changes of Increases in New Food Businesses on Yelp Per Year')
plt.show()


print('totalbusinessmarket = open businesses minus closed businesses from before␣
 ↪and in that year')
```

```
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:10: UserWarning:
Boolean Series key will be reindexed to match DataFrame index.
  # Remove the CWD from sys.path while we load stuff.
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:12: UserWarning:
Boolean Series key will be reindexed to match DataFrame index.
  if sys.path[0] == '':
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:18:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:19:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
/anaconda/lib/python3.6/site-packages/ipykernel_launcher.py:26:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/indexing.html#indexing-view-versus-copy
```







`totalbusinessmarket = open businesses minus closed businesses from before and in that year`

The food business market in 2017 grew by 13.67%. The food business market is growing exponentially. It's around 55% in 2017. The entering growth rate for new businesses were initially very high in the 2005-2007 when more businesses were enrolling than they did in the year before. The