Yelp Restaurant Demand and Supply

August 30, 2019

1 RESTAURANT DEMAND AND SUPPLY

BIG QUESTIONS: WHY DID THE RESTAURANT CLOSE DOWN? WHAT DOES THE RESTAURANT MARKET IN THE AREA LOOK LIKE?

1. Section **??**
2. Section **??**
3. Section **??**

[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
#can't install new tensorflow cuz of macos so use an older version
# import tensorflow as tf
# from tensorflow import keras
import geopandas
import descartes
import requests
import urllib
from shapely.geometry import Point
# from arcgis.gis import GIS
```

[ ]:
```
%%time
# 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')
checkin = pd.read_csv('yelp_checkin.csv')
user = pd.read_csv('yelp_user.csv')
review = pd.read_csv('yelp_review.csv')

# 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

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()

# 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()

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

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']

arizonafoodbusiness['maincuisine'] = arizonafoodbusiness['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);

# 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)

# 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()

#business id to name
businessidtoname = dict(zip(arizonafoodbusiness['business_id'],␣
 ↪arizonafoodbusiness['name']))
#business name to lon
businessidtolong = dict(zip(arizonafoodbusiness['business_id'],␣
 ↪arizonafoodbusiness['longitude']))
#business name to lat
businessidtolat = dict(zip(arizonafoodbusiness['business_id'],␣
 ↪arizonafoodbusiness['latitude']))

# add business name to AZ food reviews
arizonafoodreviews['business_name'] = arizonafoodreviews['business_id'].
 ↪map(businessidtoname);
```

```
#add lowername to the AZ food businesses
arizonafoodbusiness['lowername'] = arizonafoodbusiness['name'].apply(lambda x: x.
 ↪replace("'", "").lower());

#add set of lon and lat to AZ food reviews
arizonafoodreviews['lon'] = arizonafoodreviews['business_id'].
 ↪map(businessidtolong);
arizonafoodreviews['lat'] = arizonafoodreviews['business_id'].
 ↪map(businessidtolat)


# users profile from Arizona
arizonauserprofile = user['user_id'].isin(userarizona)
# elite users on Yelp
eliteusers = user[user['elite'] != 'None']
#getting the start year for yelping since
user['startyear'] = user['yelping_since'].apply(lambda x: x[0:4])

#remove outlier - gZGsReG0VeX4uKViHTB9EQ in Arizona
arizonafoodbusiness = arizonafoodbusiness[arizonafoodbusiness['business_id'] !=␣
 ↪'gZGsReG0VeX4uKViHTB9EQ']

# checkin values of the arizona food business
arizonafoodbusiness['sumcheckin'] = arizonafoodbusiness['business_id'].
 ↪map(sumfoodcheckin)

closedazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 0]
openazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 1]


#labeling review positive, negative, or neutral
def positiveornegative(star):
    if star > 3:
        return 'positive'
    elif star < 3:
        return 'negative'
    else:
        return 'neutral'
arizonafoodreviews['label'] = arizonafoodreviews['stars'].
 ↪map(positiveornegative);

import gc
gc.collect()

#the code goes 70x faster when it's cached right here than to be called␣
 ↪repeatedly
```

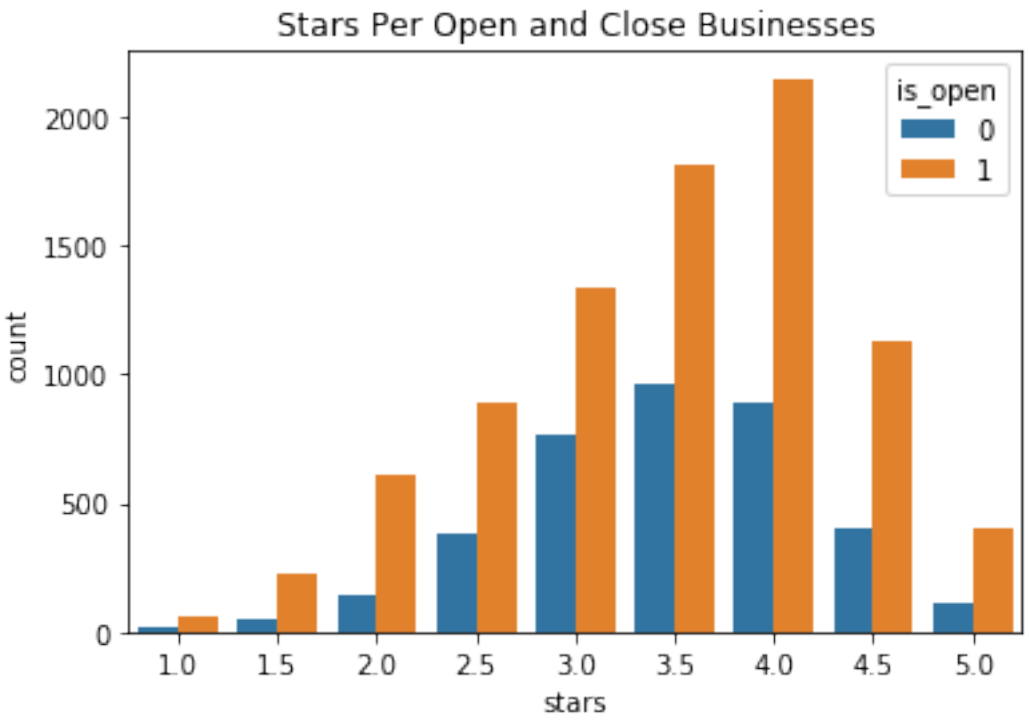
```
stopw = stopwords.words('english')
# remove punctuations, lowercase,
#remove numbers r'\d+
#remove stop words, and split string into a list of words
arizonafoodreviews['clean text'] = arizonafoodreviews['text'].apply(
    lambda word: re.sub(r'\d+', '', re.sub(r'[^\w\s]', '', word.replace(
    '\n', ' '))))).apply(
    lambda word: [w for w in word.lower().split(' ') if w not in stopw and w !=␣
 ↪''']);

positiveazfoodreviews = arizonafoodreviews[arizonafoodreviews['label'] ==␣
 ↪'positive']
negativeazfoodreviews = arizonafoodreviews[arizonafoodreviews['label'] ==␣
 ↪'negative']
```

[4]:
```
#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)
```

[5]:
```
#how many years of reviews on yelp
arizonafoodbusiness['diffyear']=arizonafoodbusiness['maxyear']-arizonafoodbusiness['minyear']
```

2 Closure of Restaurants

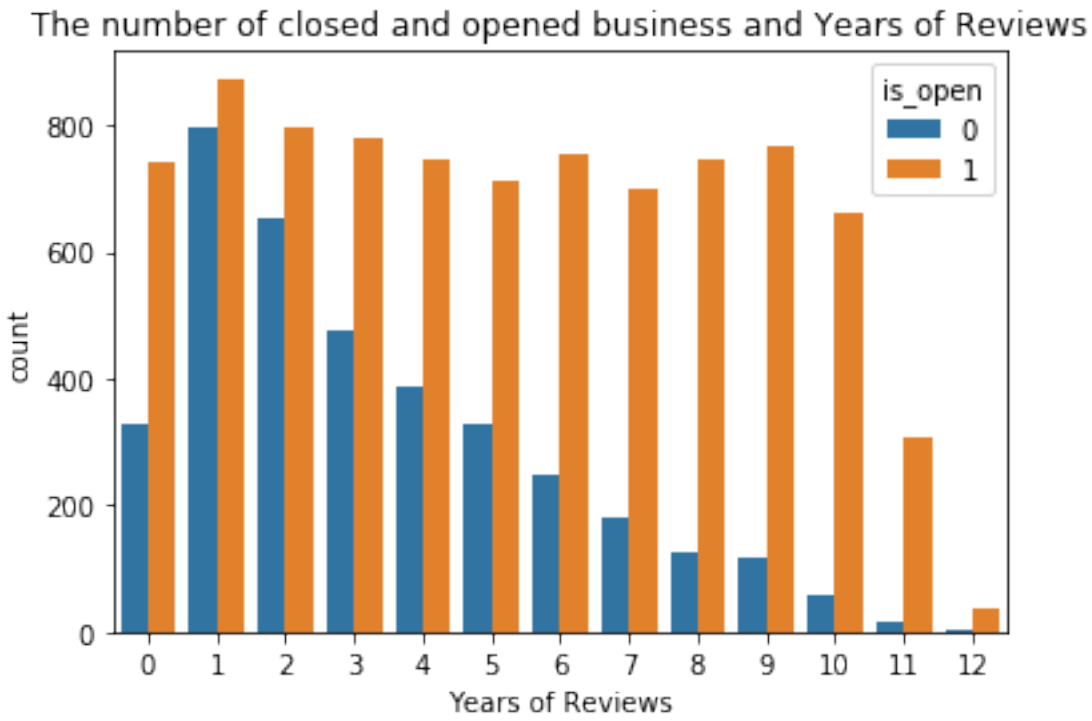
[5]:
```
#average star for closed businesses and average star for opened businesses
azstars = arizonafoodbusiness.groupby(['is_open', 'stars']).count().
 ↪reset_index()[['is_open', 'stars', 'business_id']].rename(
columns = {'business_id': 'count'})
sns.barplot(x = 'stars', y = 'count', hue = 'is_open', data = azstars)
plt.title("Stars Per Open and Close Businesses")
sns.color_palette("PuBuGn_d")
plt.show()
```



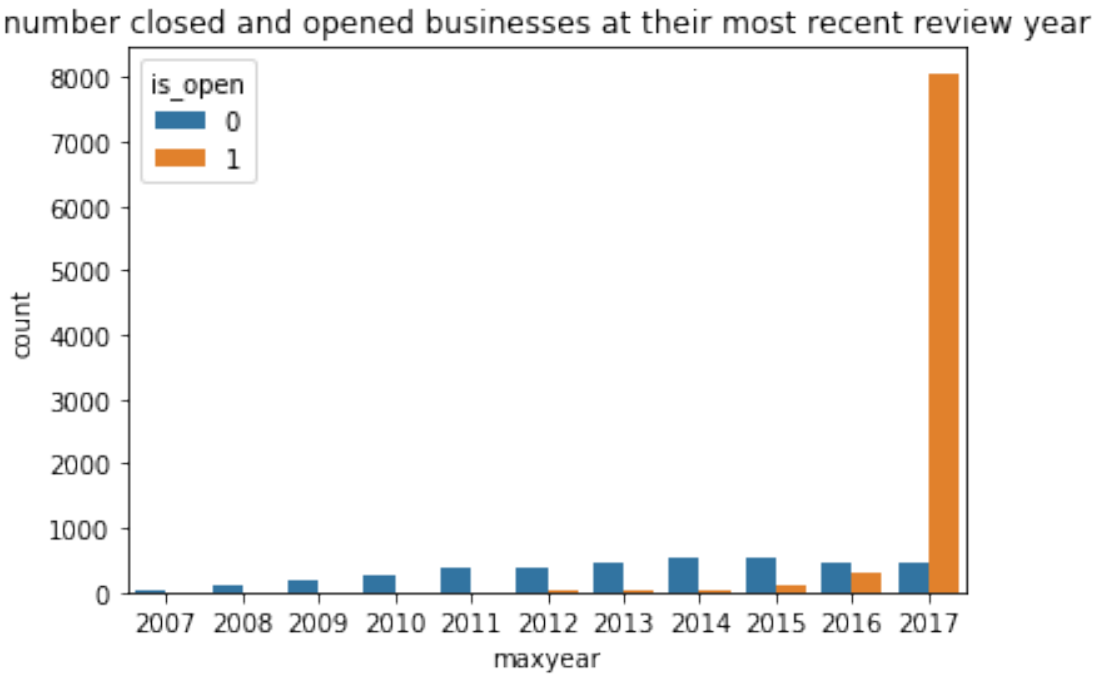
[6]:
```
arizonafoodbusiness.groupby('is_open').agg(
    {'review_count': 'mean'})
```

[6]:
```
          review_count
is_open
0            37.802148
1            96.557875
```

[7]:
```
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()
```

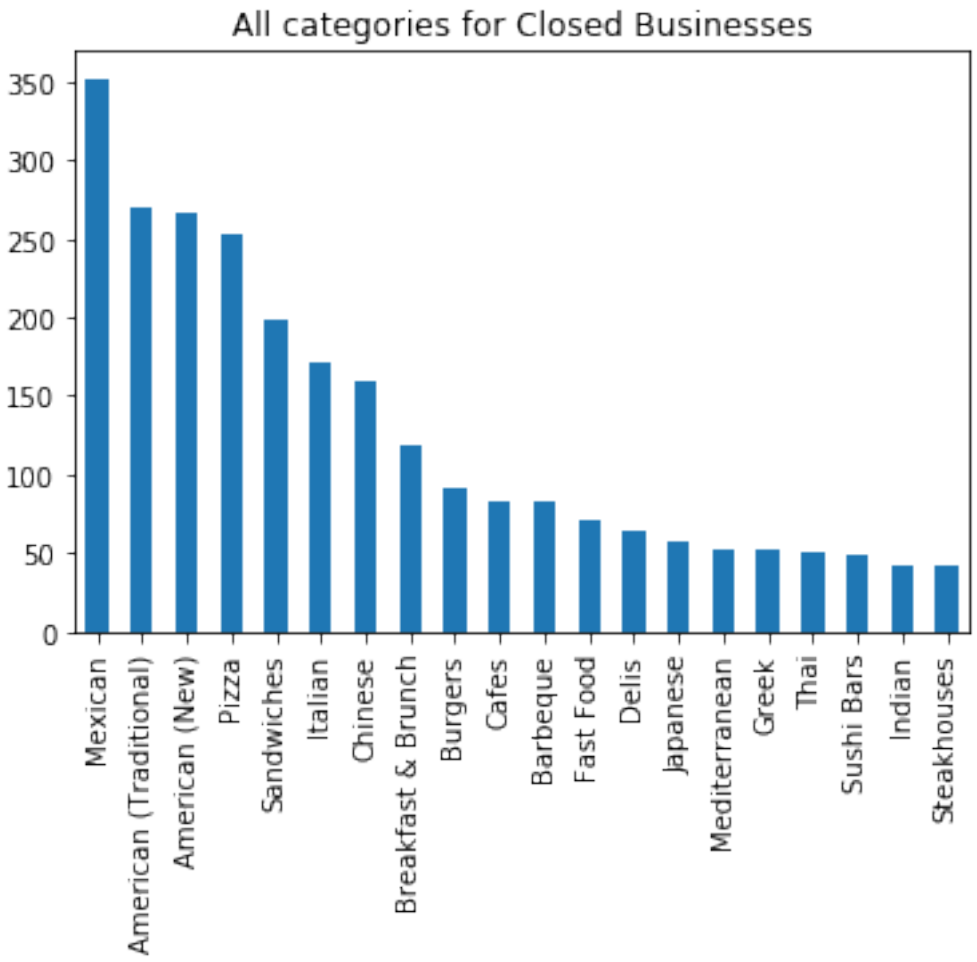


[8]:
```
maxyearo = arizonafoodbusiness.groupby(
    ['is_open', 'maxyear']).agg(
    {'business_id': 'count'}).reset_index().rename(
    columns = {'business_id': 'count'})
sns.barplot(x = 'maxyear', y = 'count', hue = 'is_open', data = maxyearo)
plt.title("number closed and opened businesses at their most recent review year")
plt.show()
```



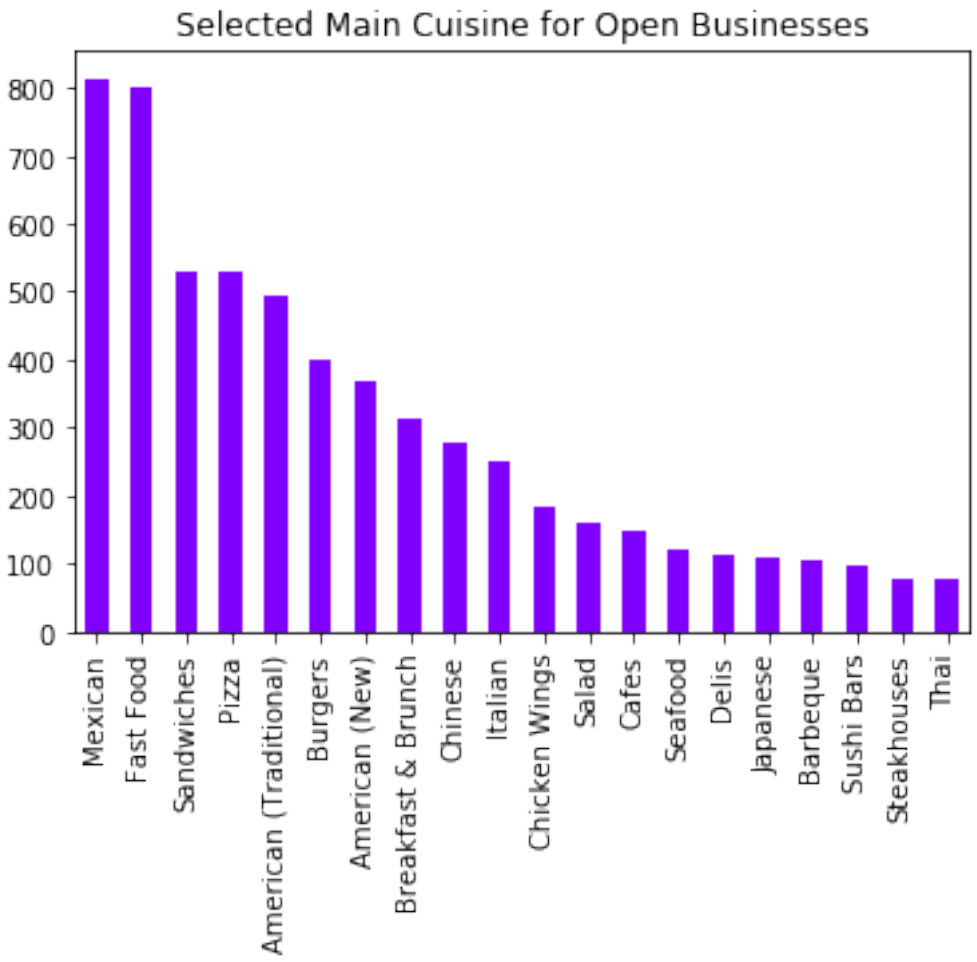
[9]:
```
closedazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 0]
closedazbusiness['maincuisine'] = closedazbusiness['categorieslist'].apply(
    lambda x: [a for a in x if a in restaurantgroups]).apply(
    lambda h: h[0] if len(h) != 0 else np.nan)
closedazbusiness['maincuisine'].value_counts()[0:20].plot.bar()
plt.title('All categories for Closed Businesses')
plt.show()
```

```
//anaconda3/lib/python3.7/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.
```



[12]:
```
openazbusiness = arizonafoodbusiness[arizonafoodbusiness['is_open'] == 1]
openazbusiness['maincuisine'] = openazbusiness['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)
openazbusiness['maincuisine'].value_counts()[0:20].plot.bar(colormap = 'rainbow')
plt.title('Selected Main Cuisine for Open Businesses')
plt.show()
```

```
//anaconda3/lib/python3.7/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.

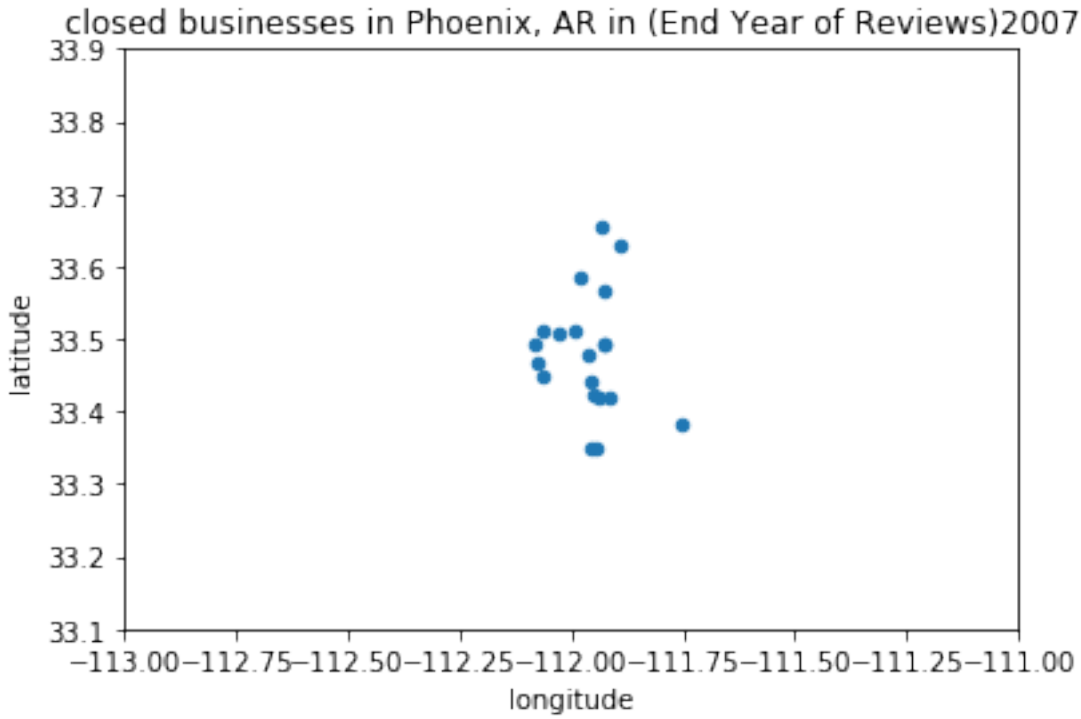
The number of closed businesses are higher when they have fewer years of reviews on yelp! This could either mean that they close during their startup or businesses close for other reasons besides startup especially the ones with more than 7 years on Yelp Reviews.

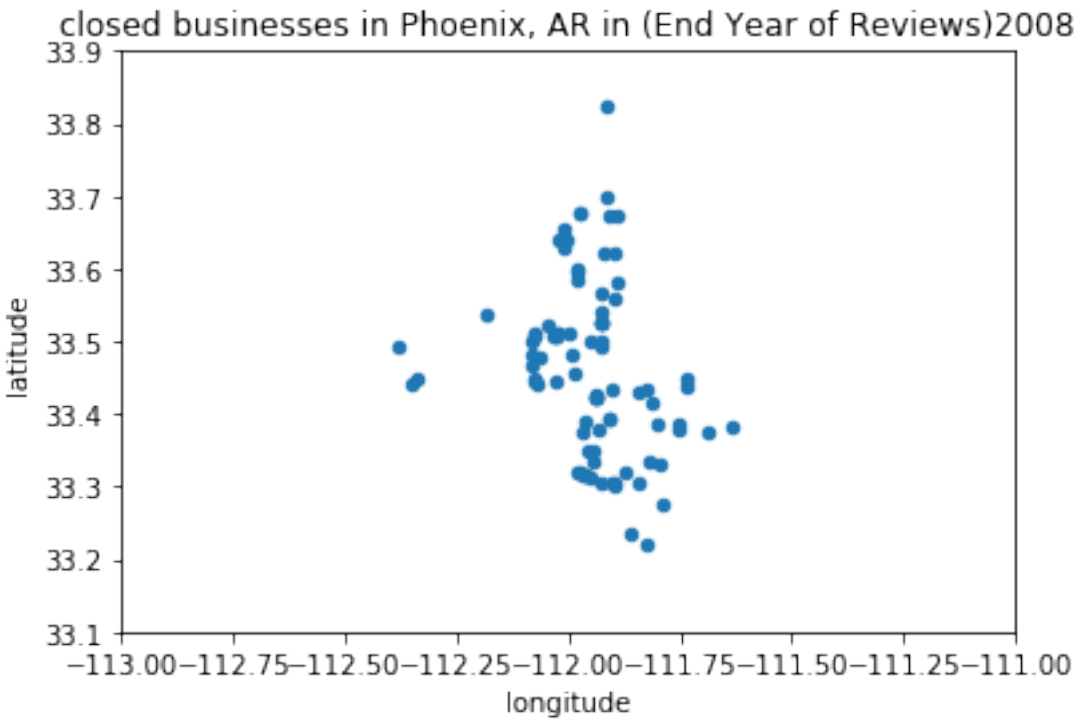
Fast food restaurants tend to stay open because they are part of a chain franchise so they got more funding and capital.

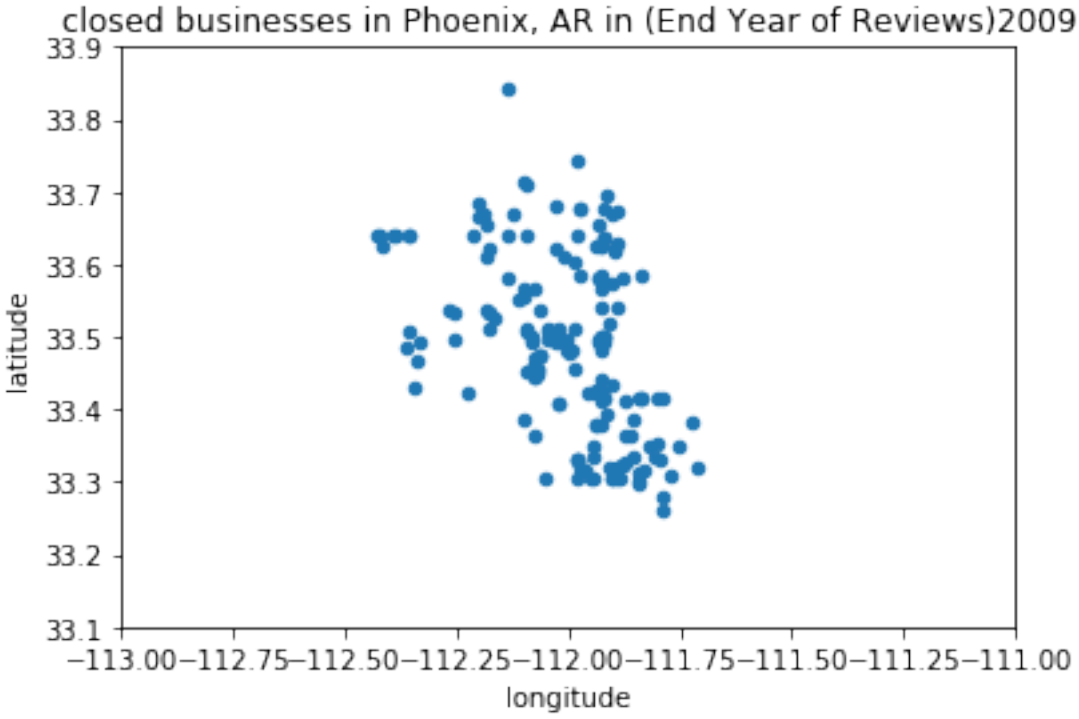
About 41 percent of Phoenix residents are of Hispanic descent(2018 BizJournal)

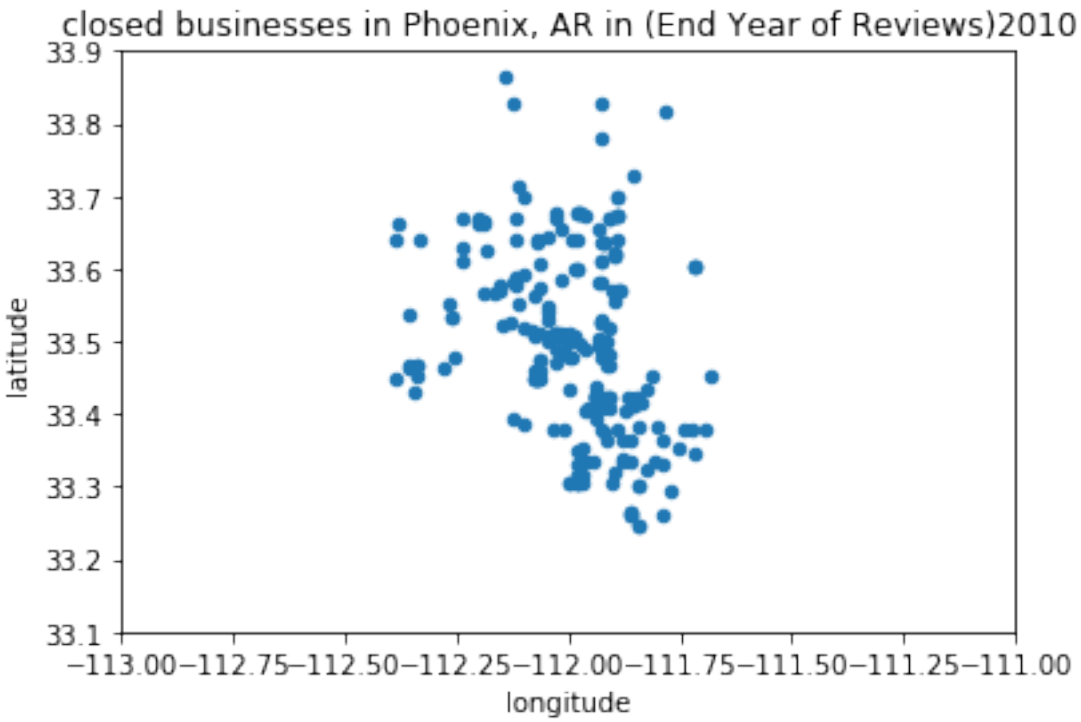
[13]:
```
closedazbusiness['maxyear'].unique()
for year in range(2007, 2018,1):
    dta = closedazbusiness[closedazbusiness['maxyear'] == year]
    dta.plot(
    kind="scatter", x="longitude", y="latitude")
```

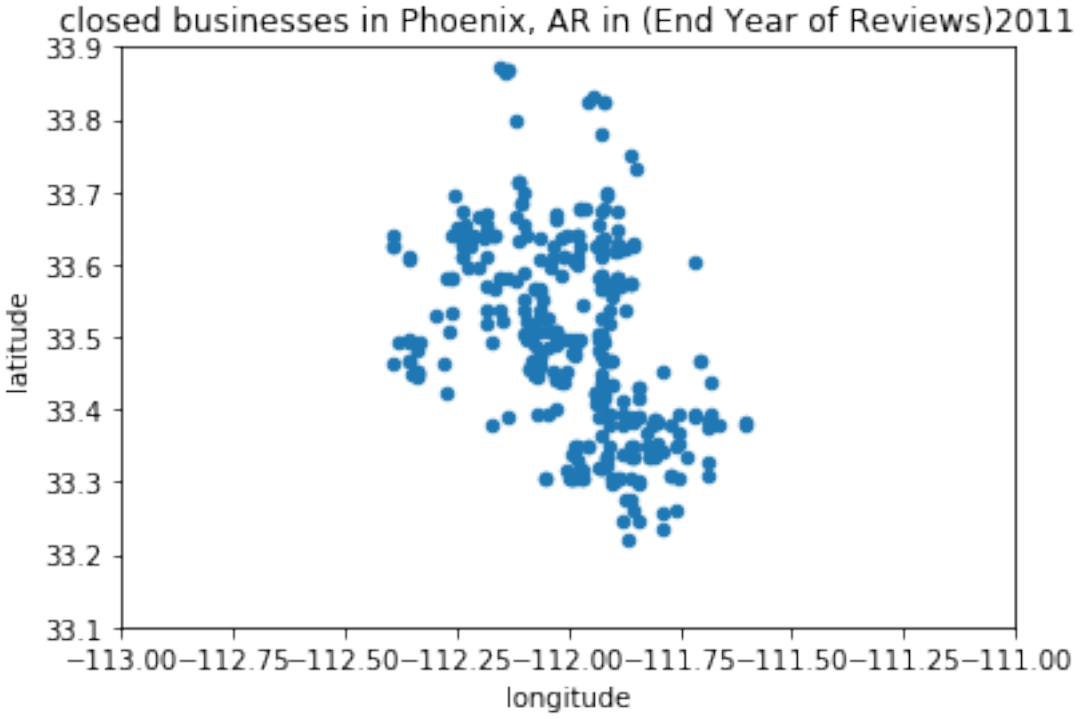
```
   plt.title('closed businesses in Phoenix, AR in (End Year of Reviews){0}'.
↪format(year))
   plt.ylim(33.1, 33.9)
   plt.xlim(-113,
       -111)
   plt.show()
```

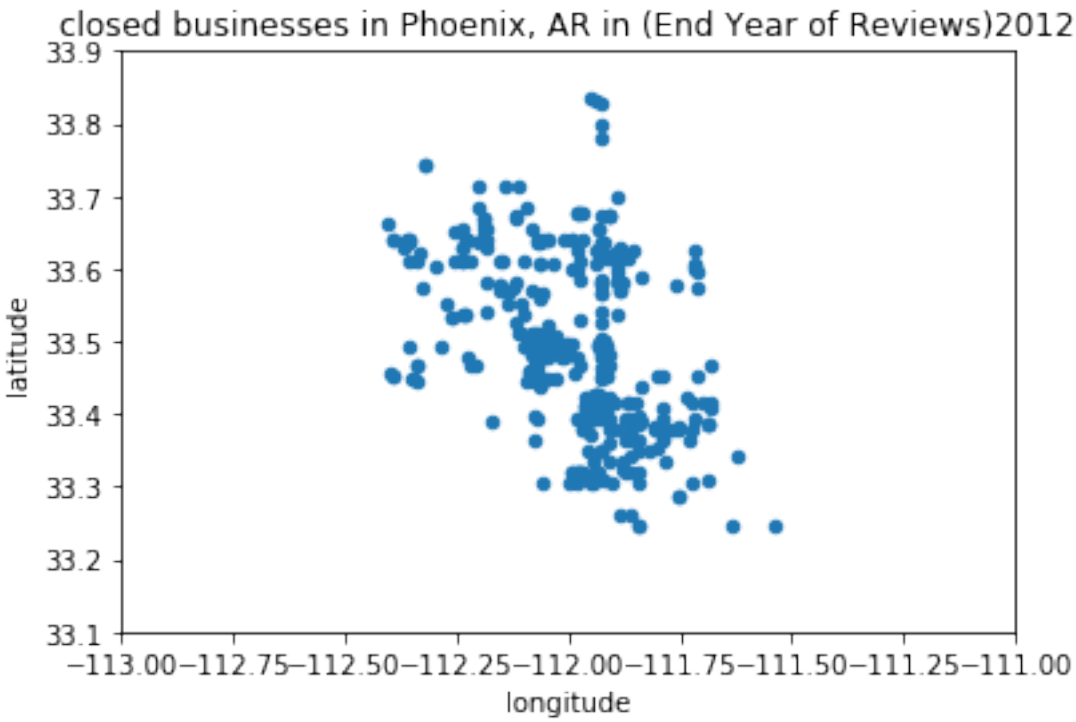


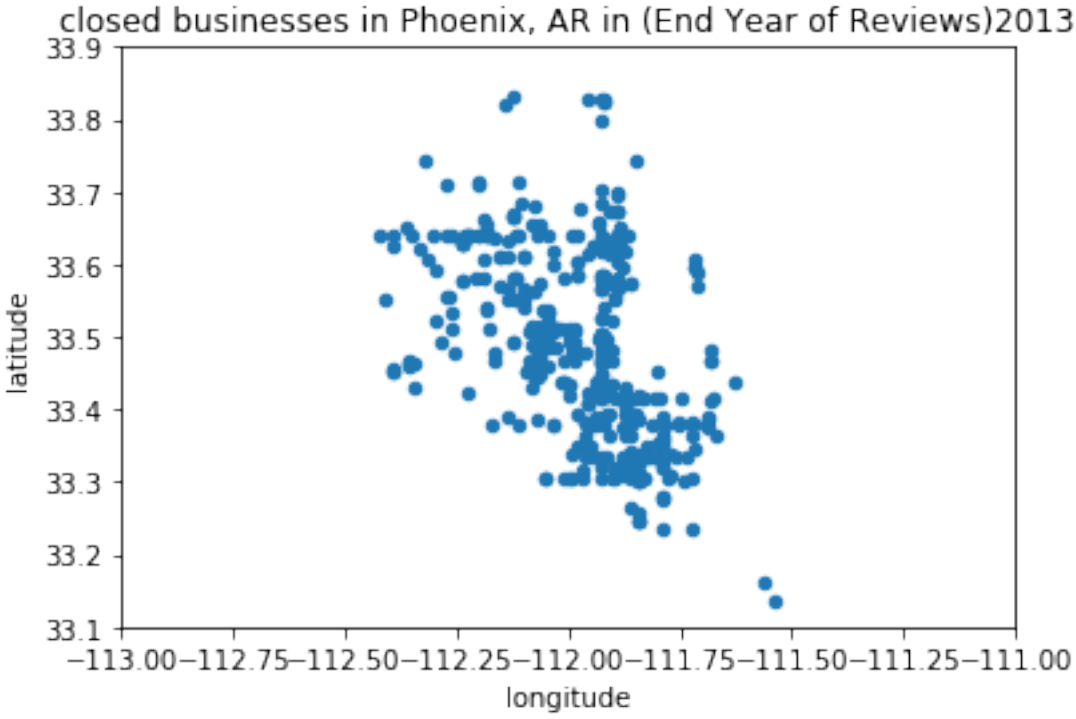

closed businesses in Phoenix, AR in (End Year of Reviews)2008
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

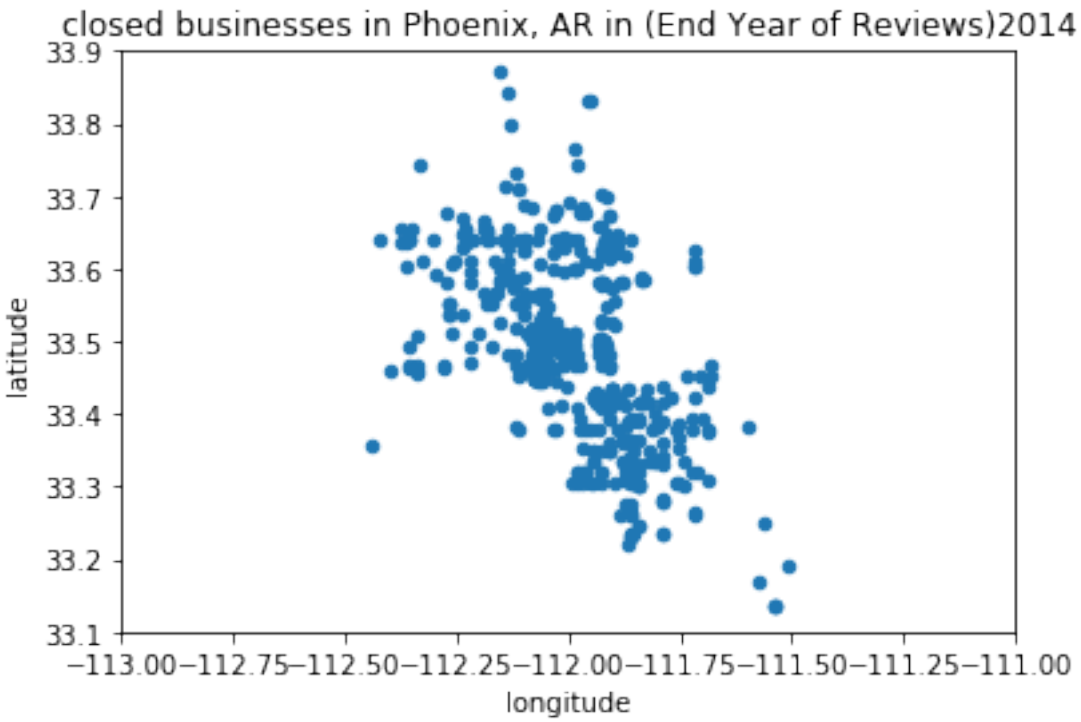

closed businesses in Phoenix, AR in (End Year of Reviews)2009
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

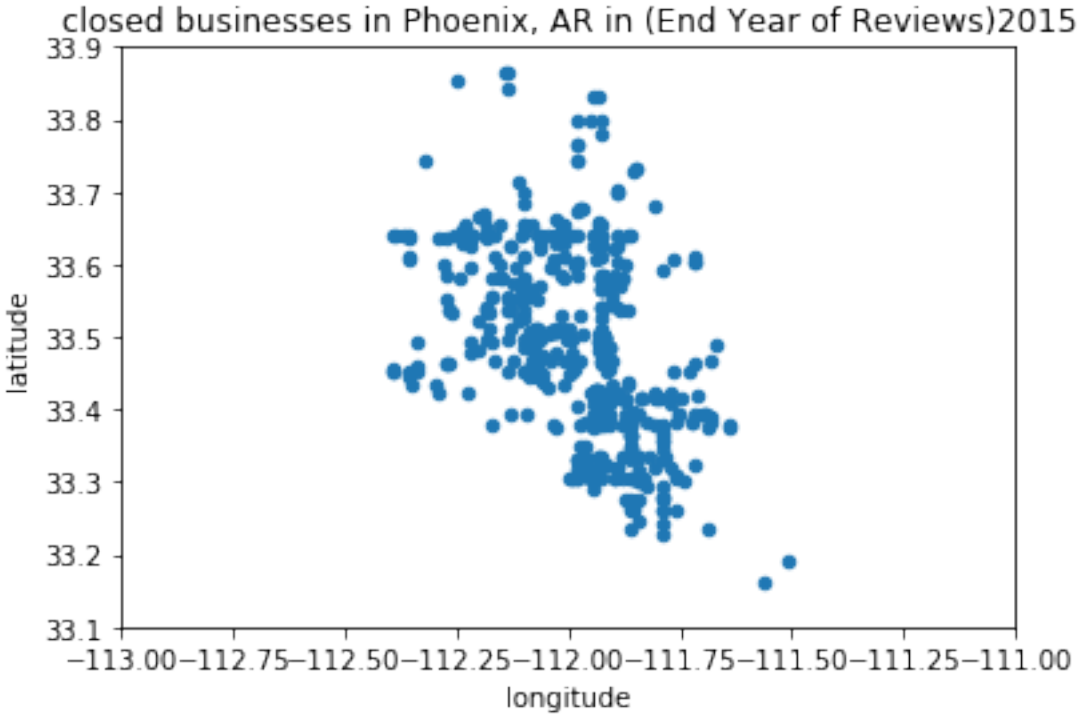

closed businesses in Phoenix, AR in (End Year of Reviews)2010
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

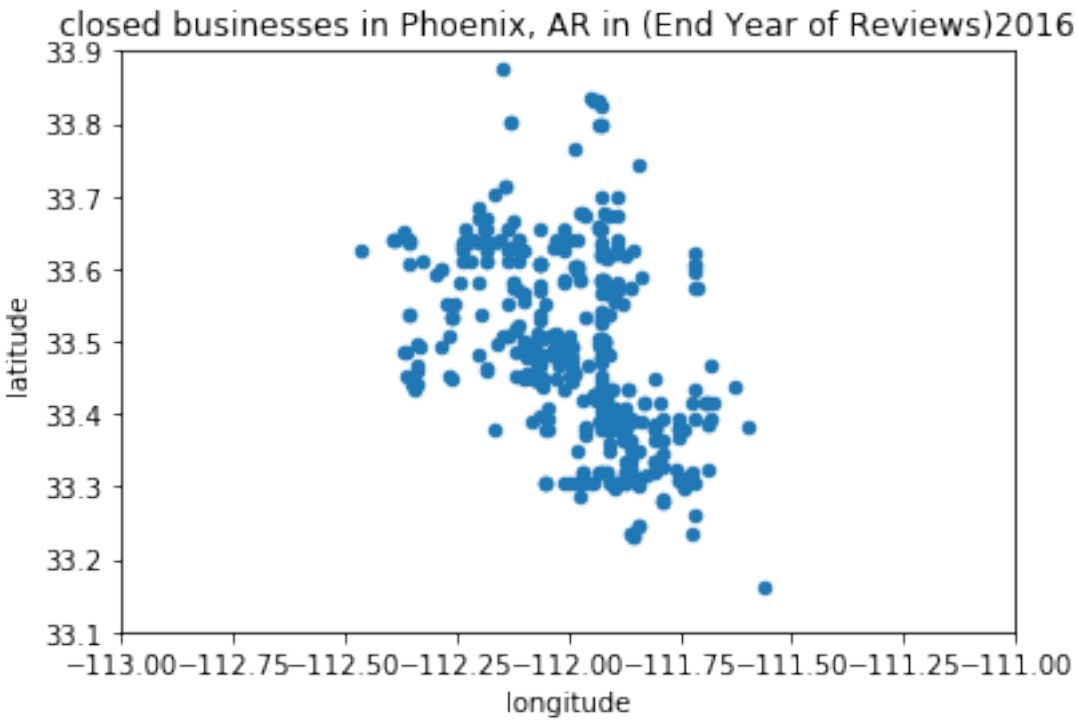

closed businesses in Phoenix, AR in (End Year of Reviews)2011
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

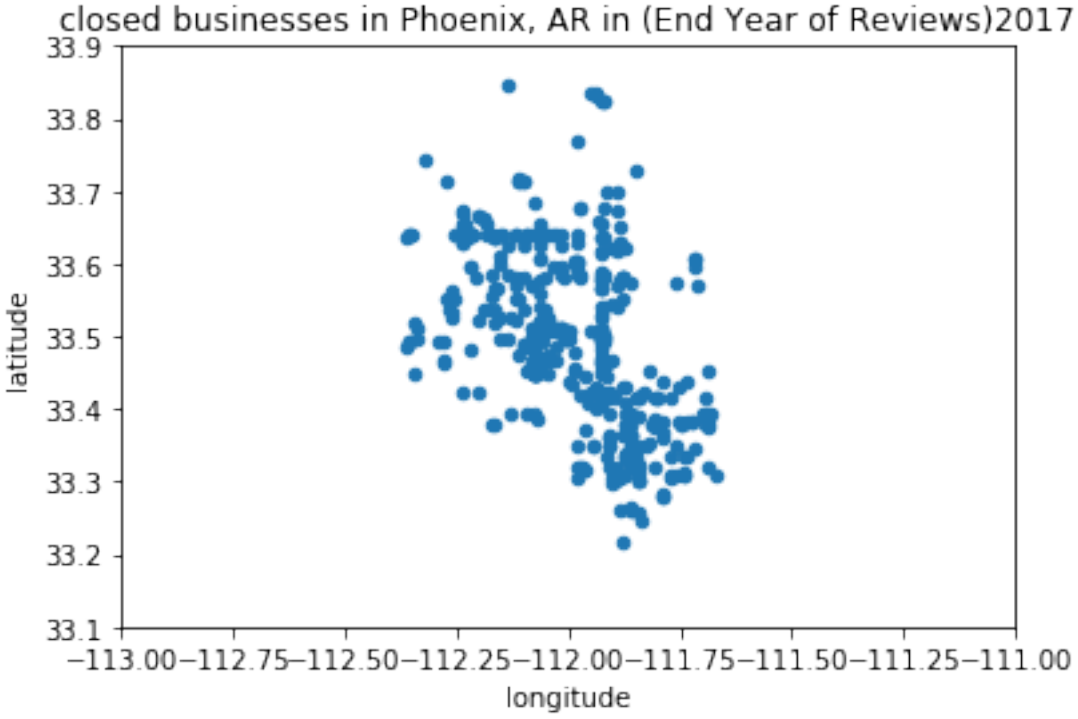

closed businesses in Phoenix, AR in (End Year of Reviews)2012
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

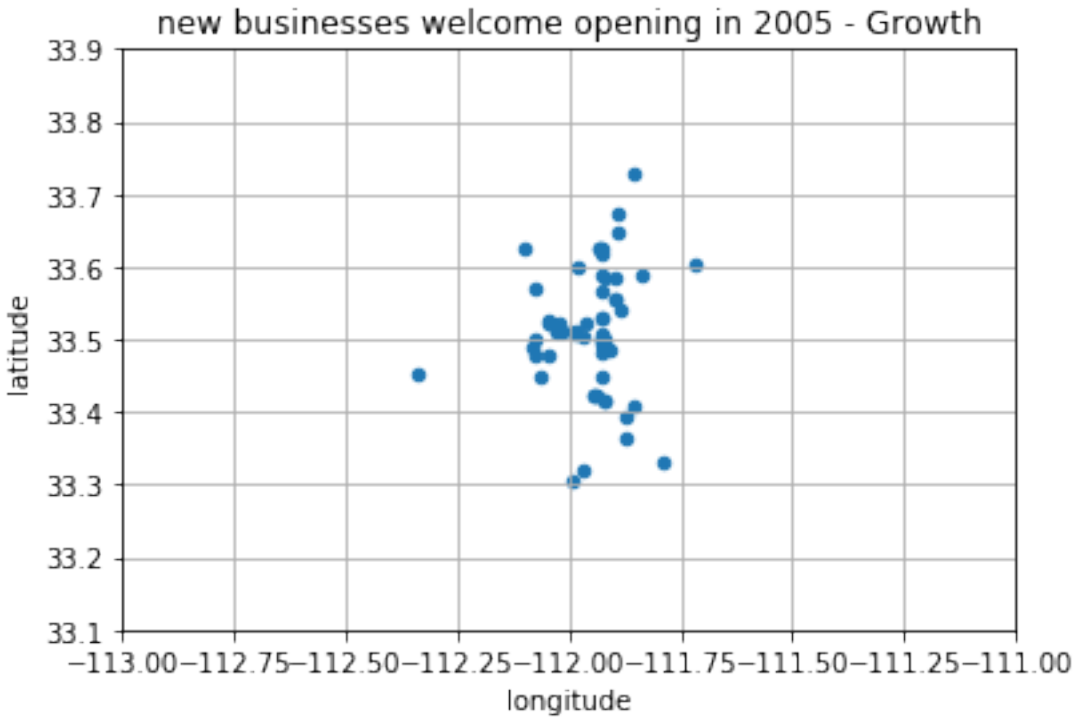

closed businesses in Phoenix, AR in (End Year of Reviews)2015
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude


closed businesses in Phoenix, AR in (End Year of Reviews)2016
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

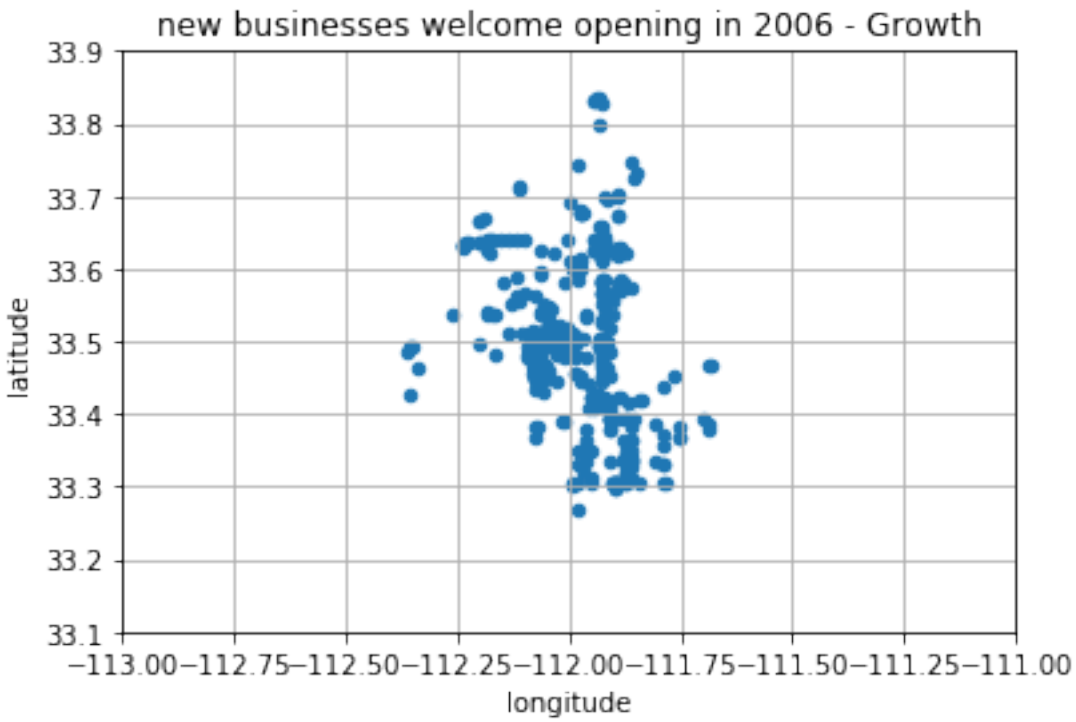

closed businesses in Phoenix, AR in (End Year of Reviews)2017
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

[14]:
```
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(-113,
        -111)
    plt.grid()
    plt.show()
```

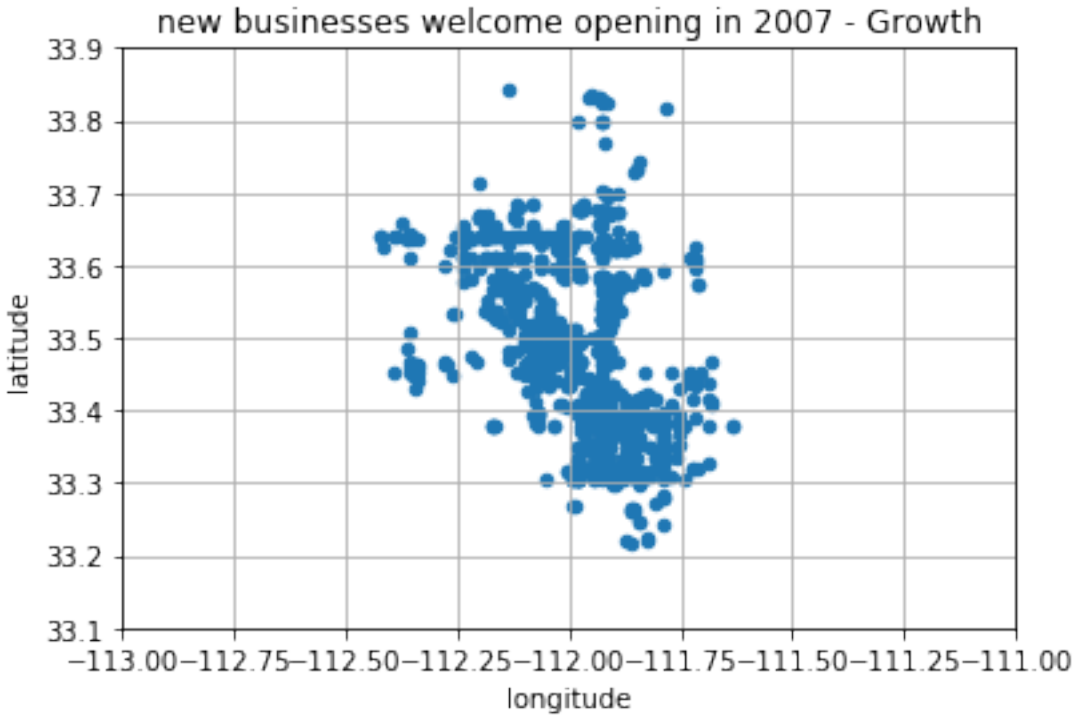
61



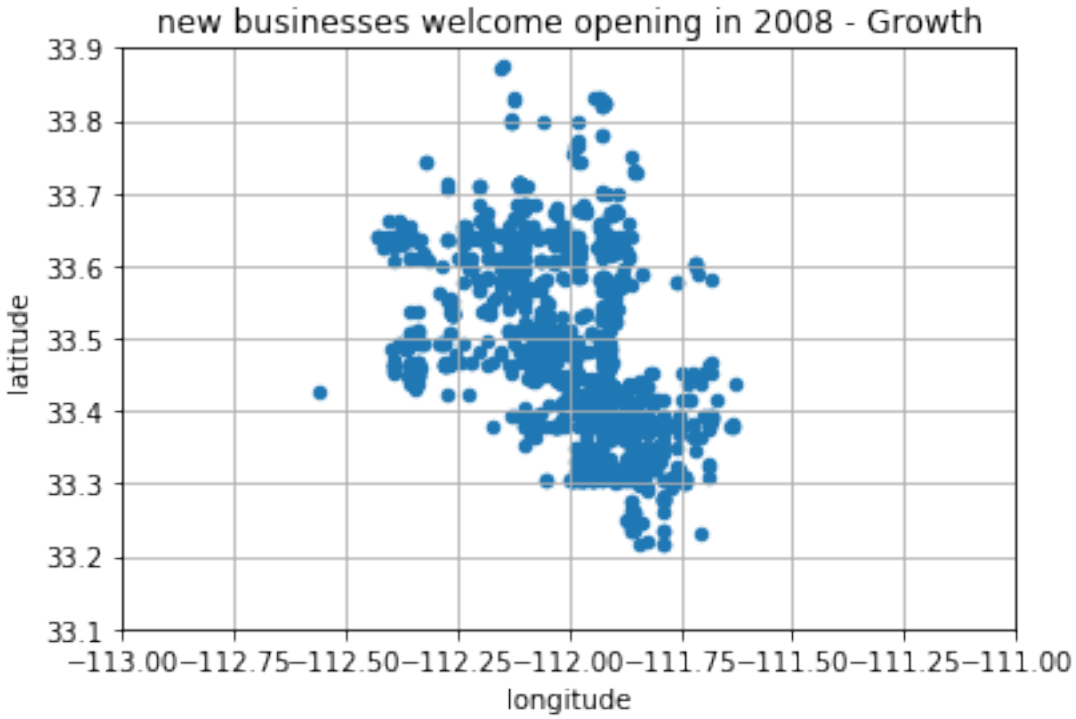
502


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

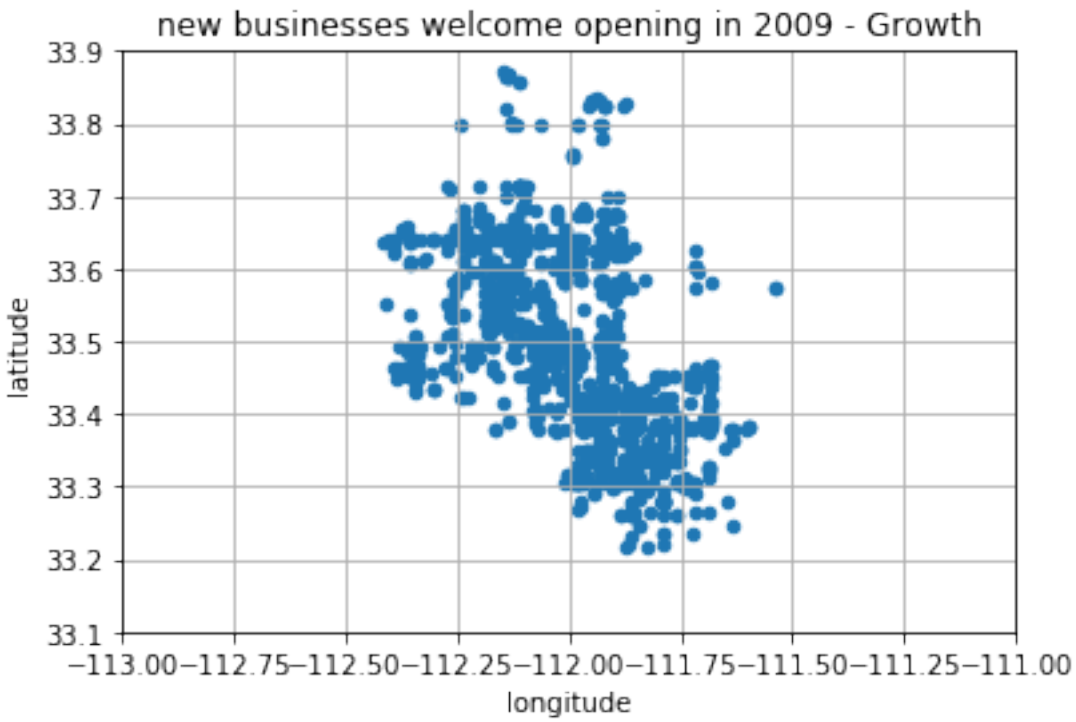
1210


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

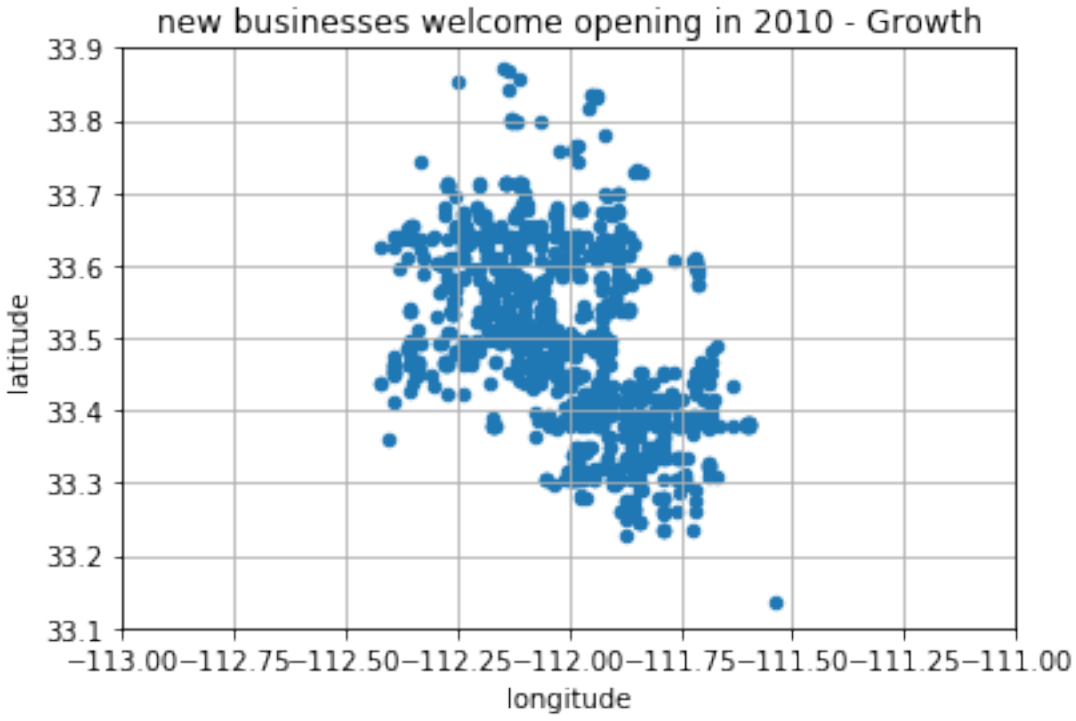
1361


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

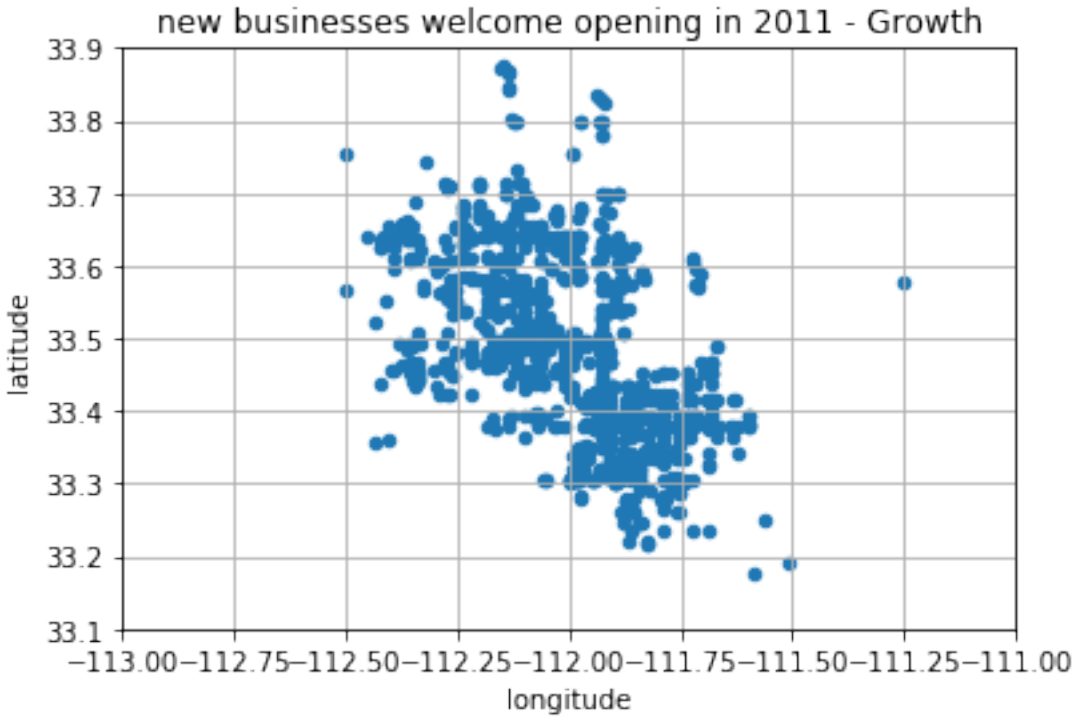
1220


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

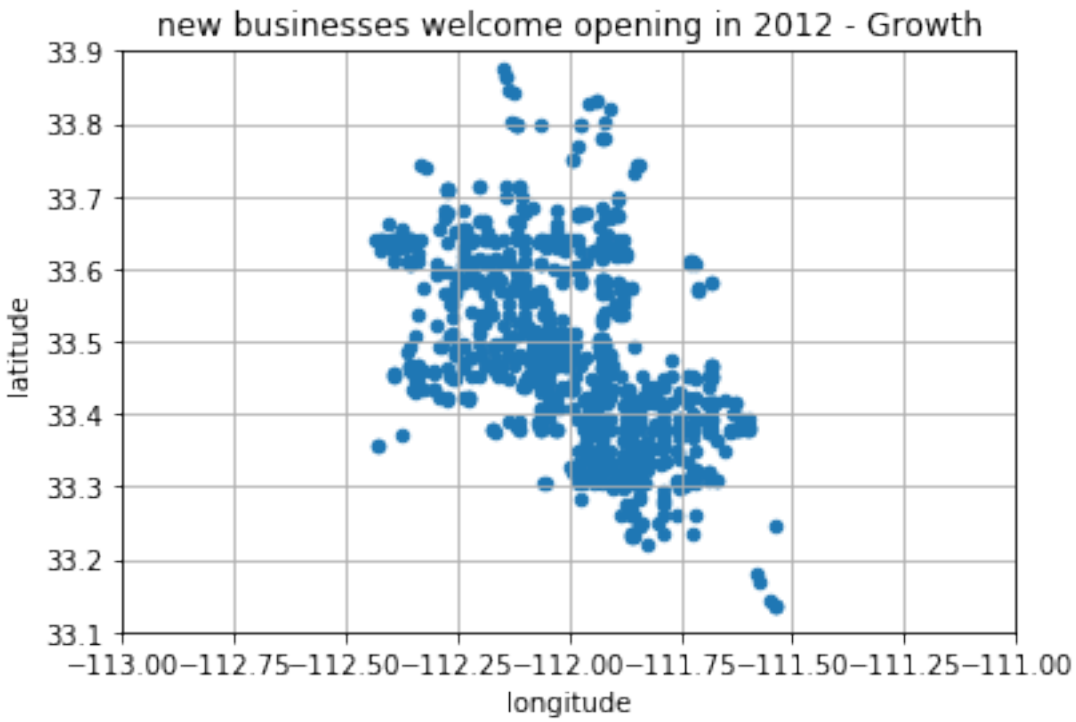
1178


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

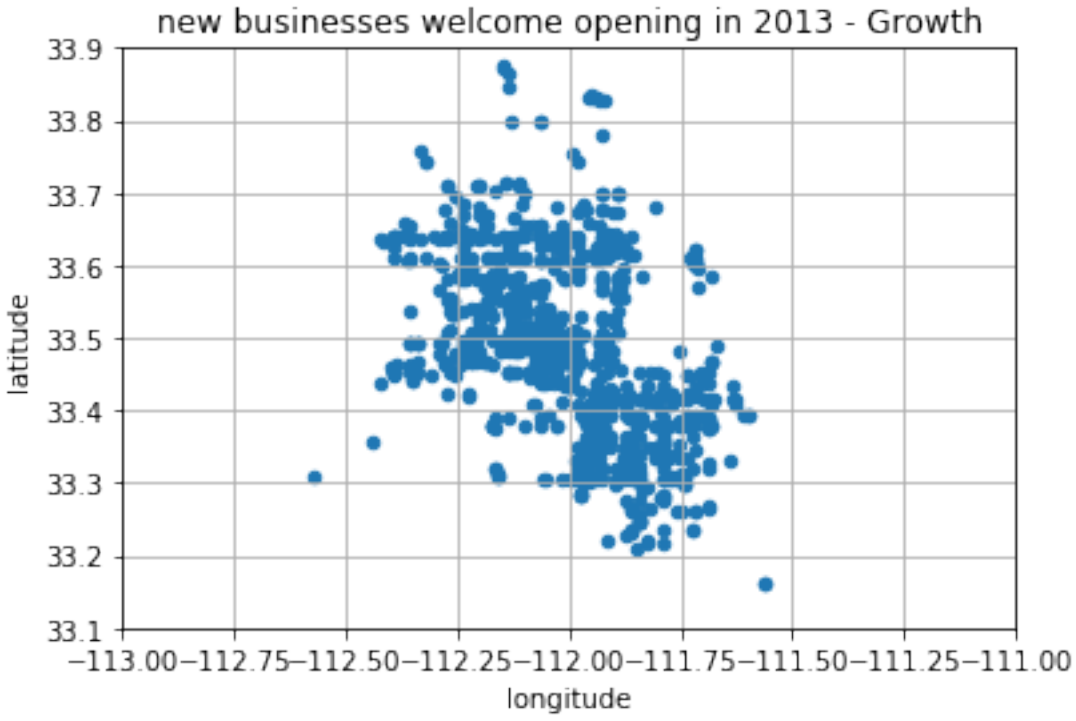
1159


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

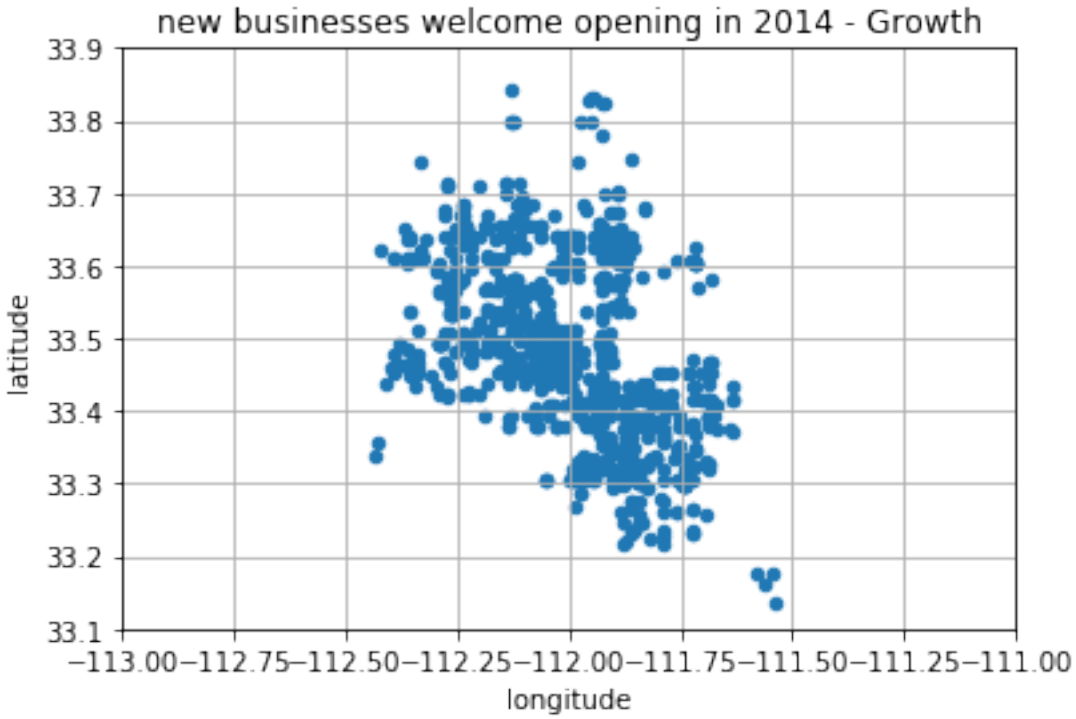
1079


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

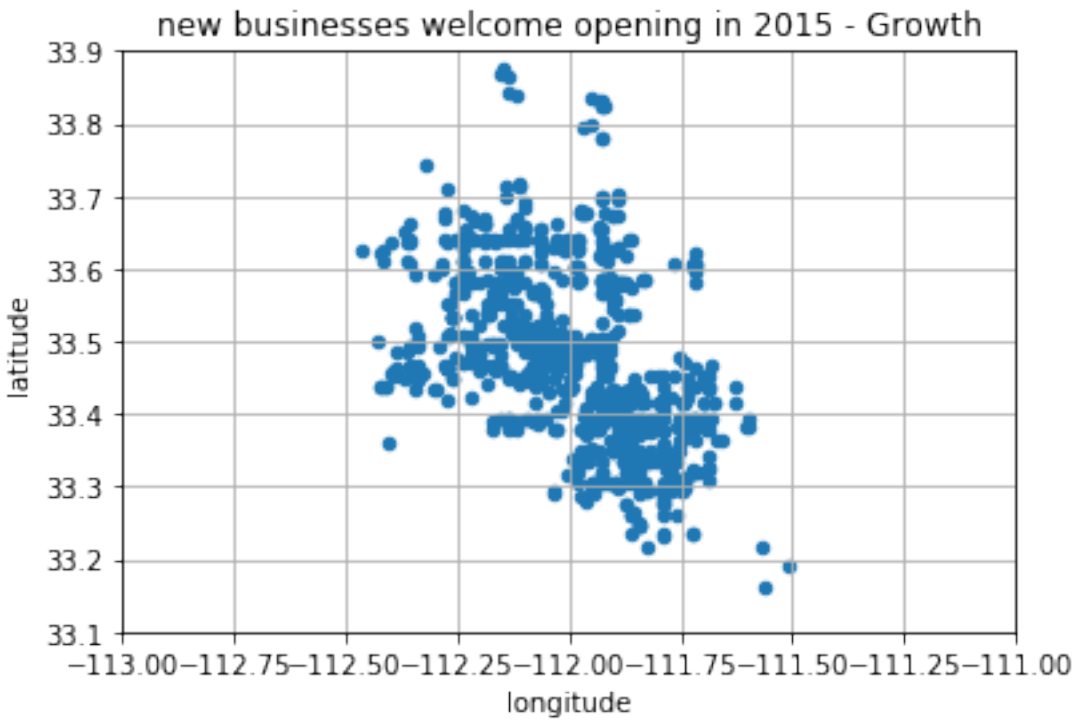
1044


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

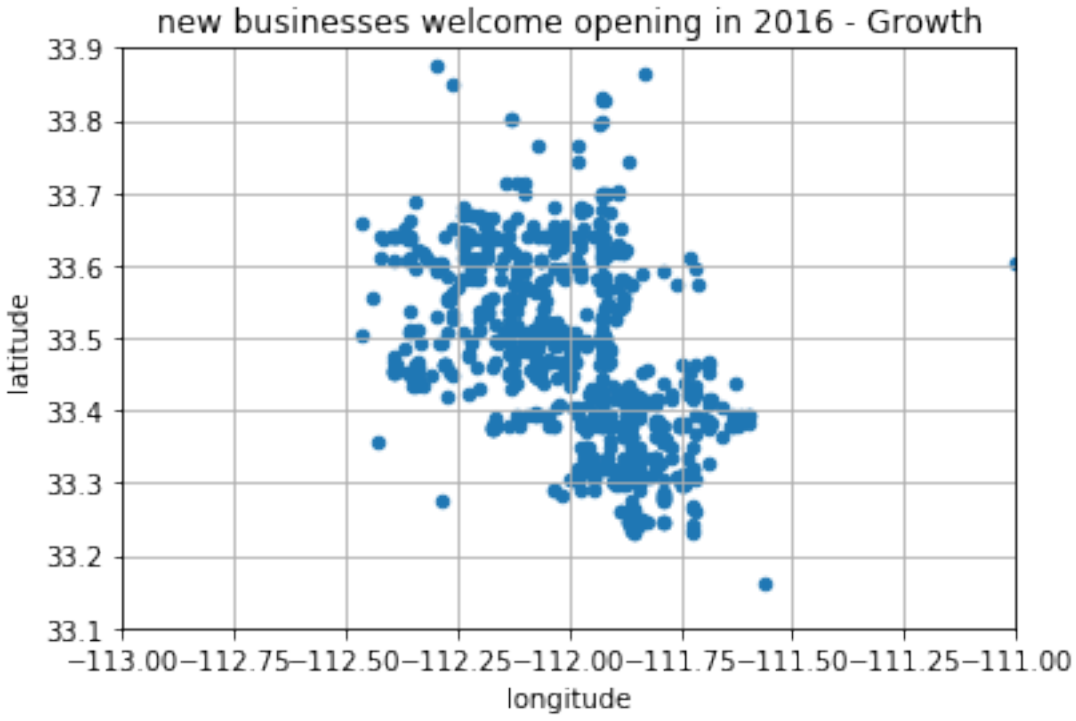
994


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

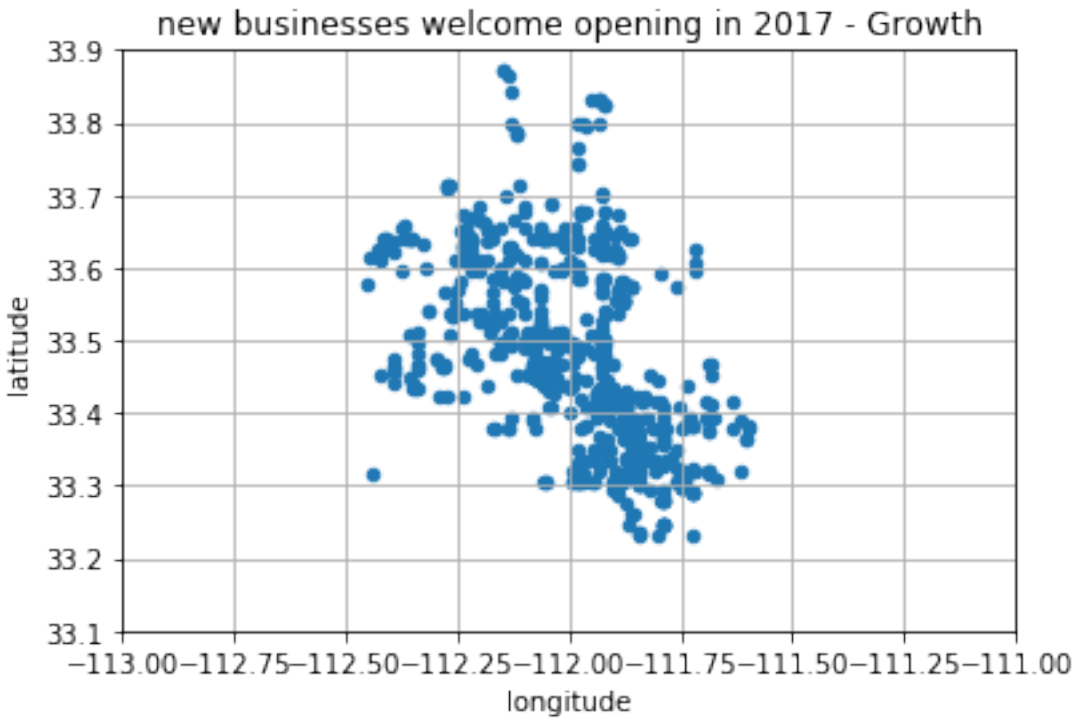
936


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

887

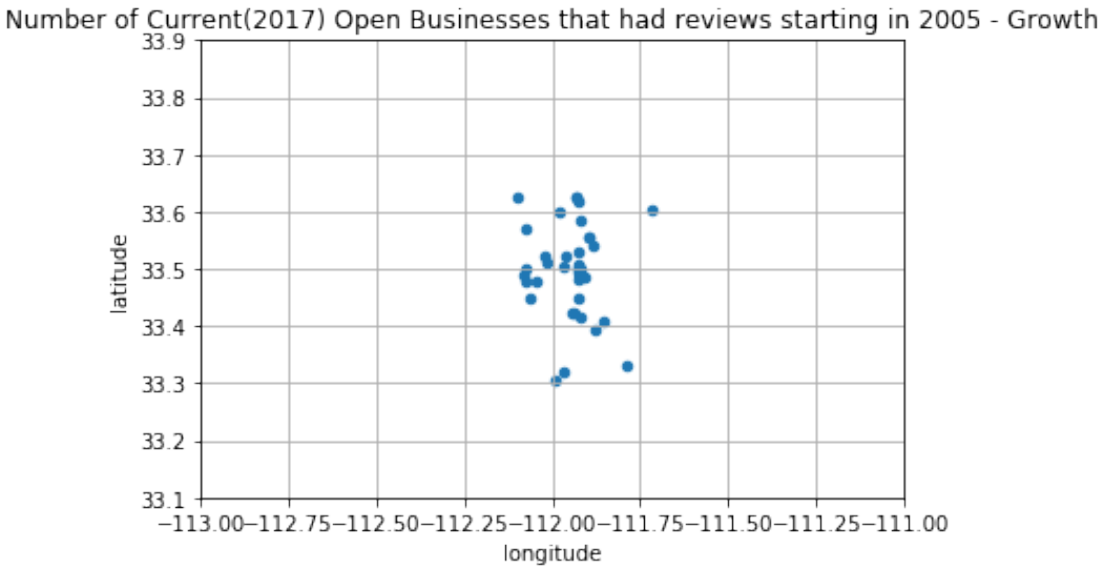

new businesses welcome opening in 2016 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

716

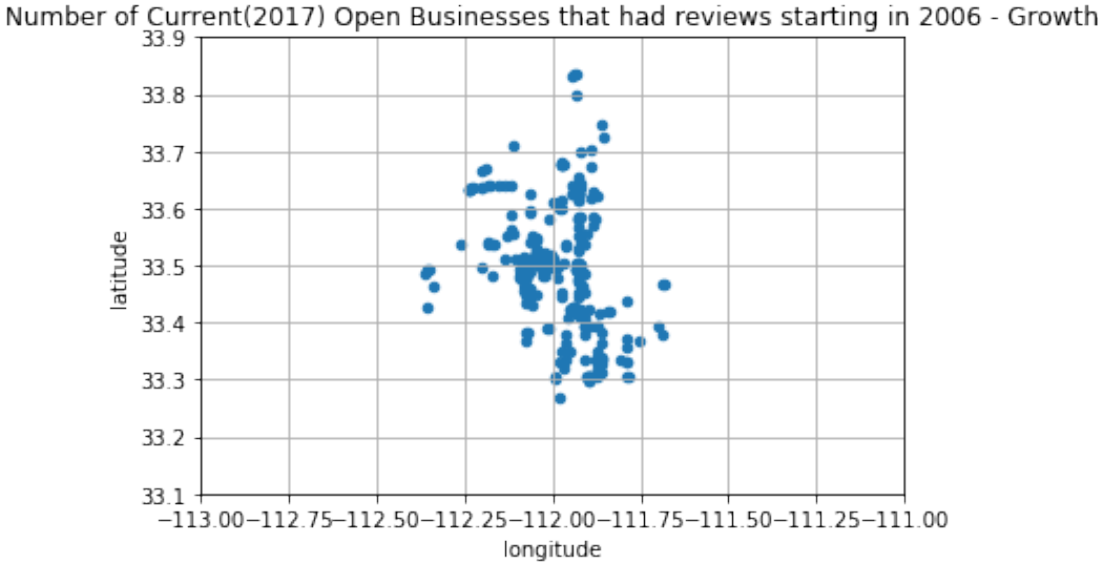


[15]:
```
for year in range(2005, 2018,1):
    dta = openazbusiness[openazbusiness['minyear'] == year]
    print(dta.shape[0])
    dta.plot(
    kind="scatter", x="longitude", y="latitude")
    plt.title('Number of Current(2017) Open Businesses that had reviews starting␣
 ↪in {0} - Growth'.format(year))
    plt.ylim(33.1, 33.9)
    plt.xlim(-113,
        -111)
    plt.grid()
    plt.show()
```

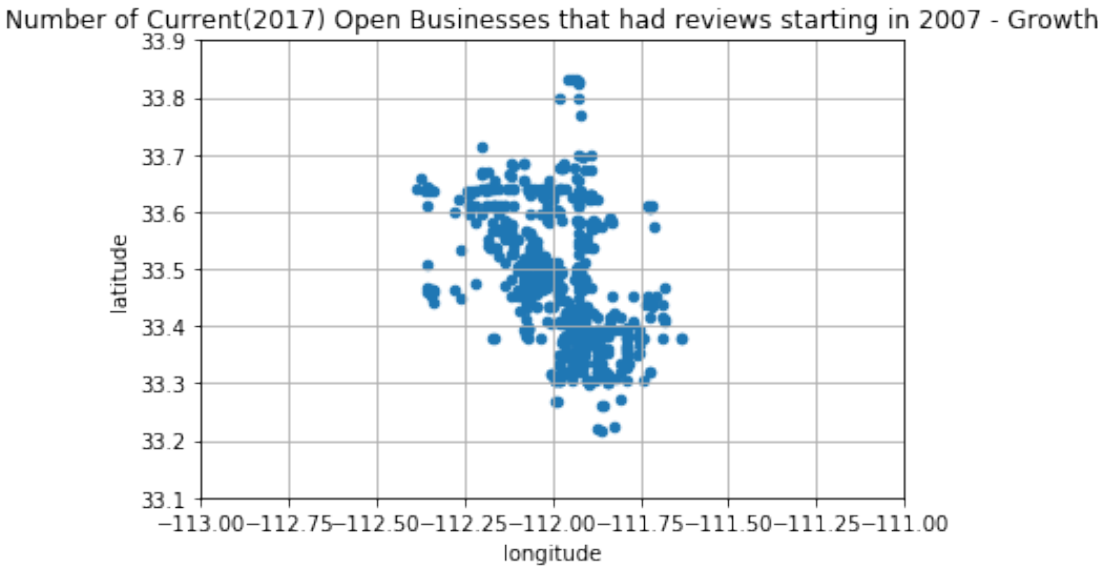
39


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

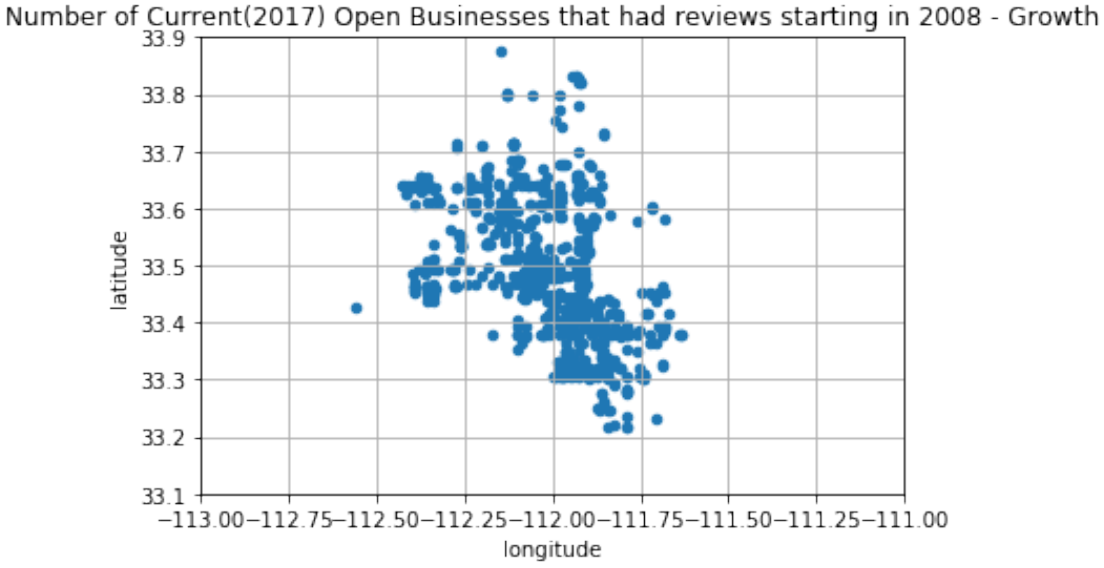
311


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

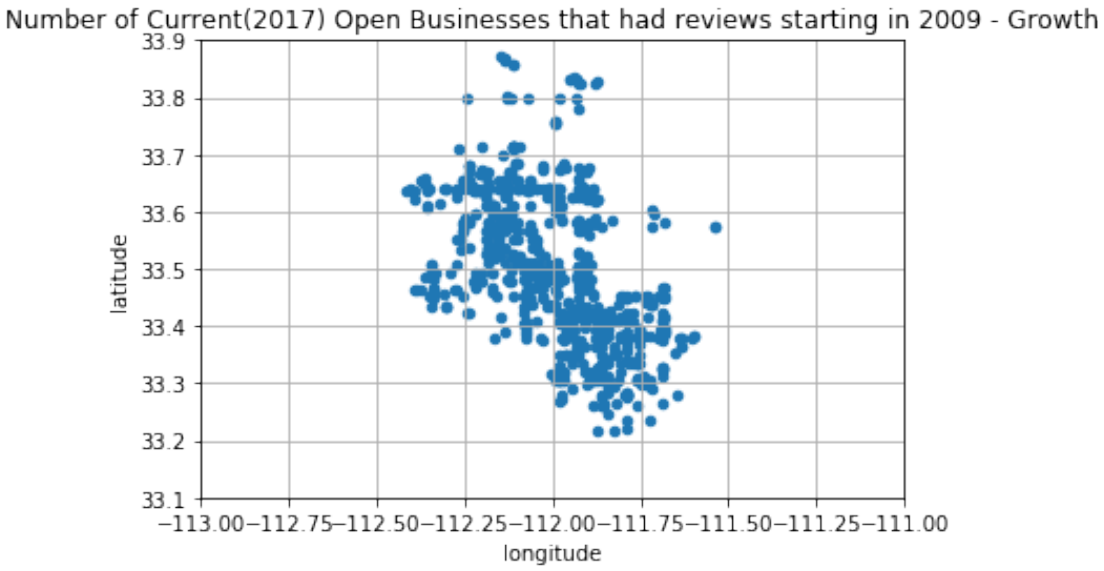
690


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
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

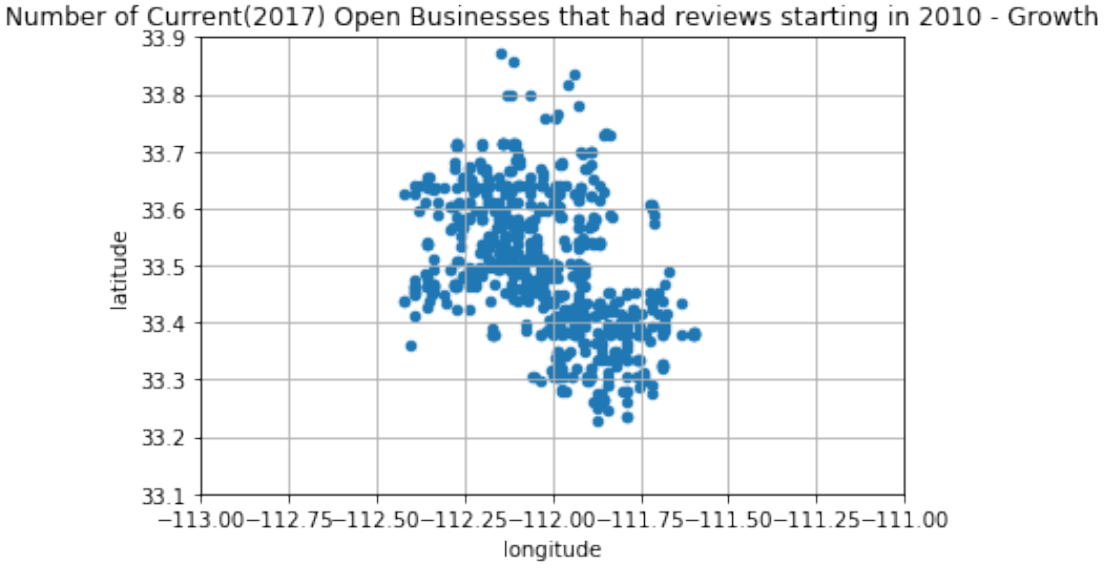
799


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
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

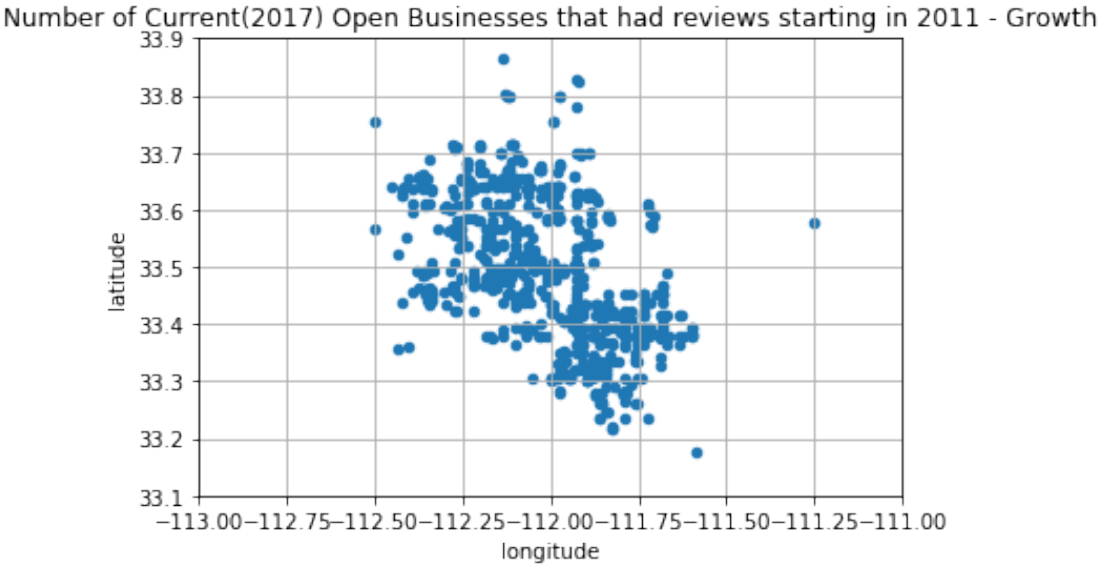
765


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
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

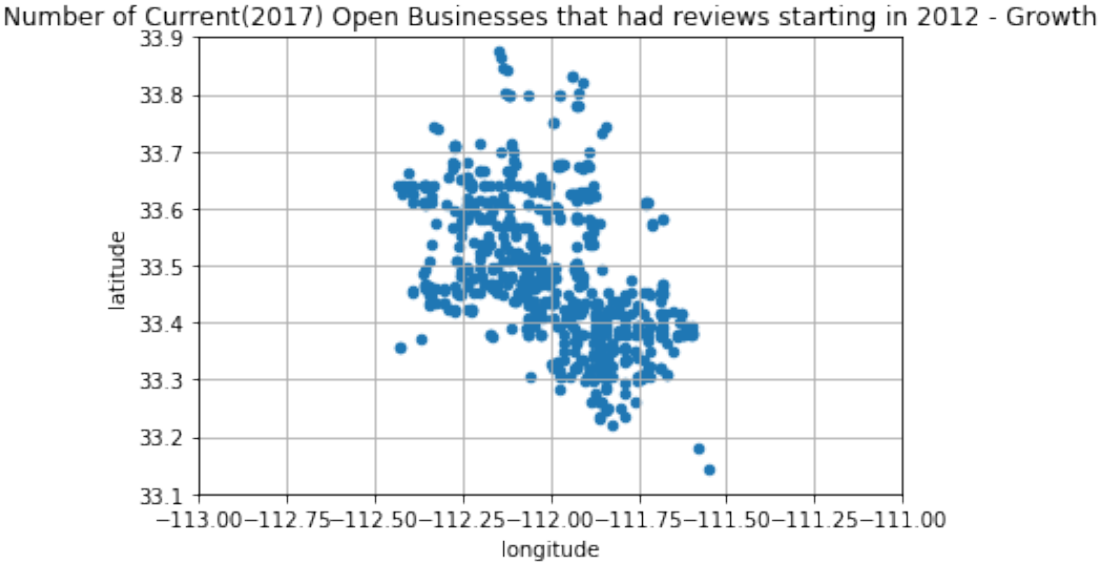
730


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
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

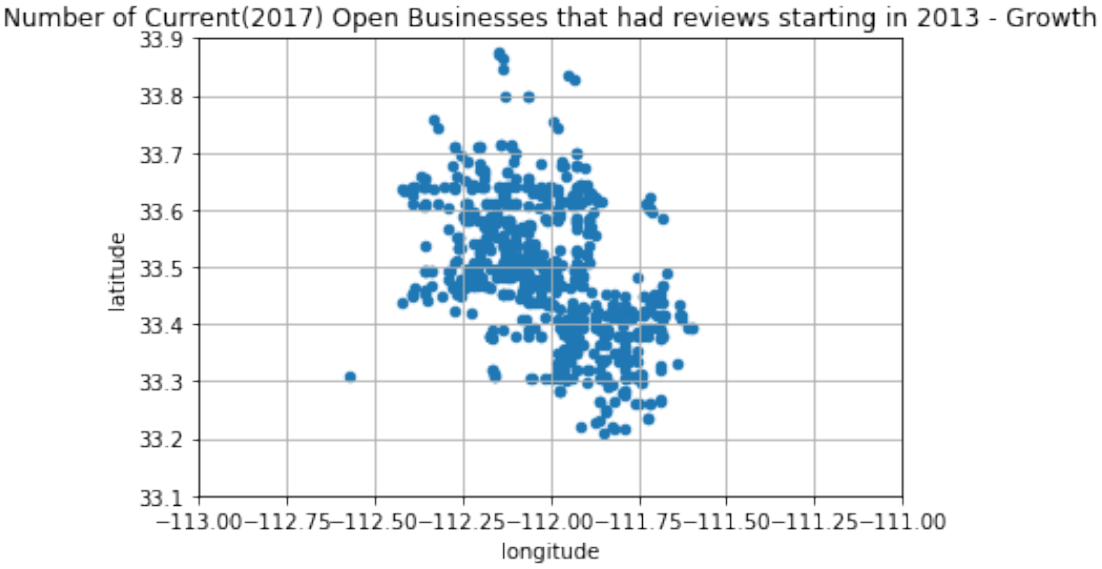
774


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00

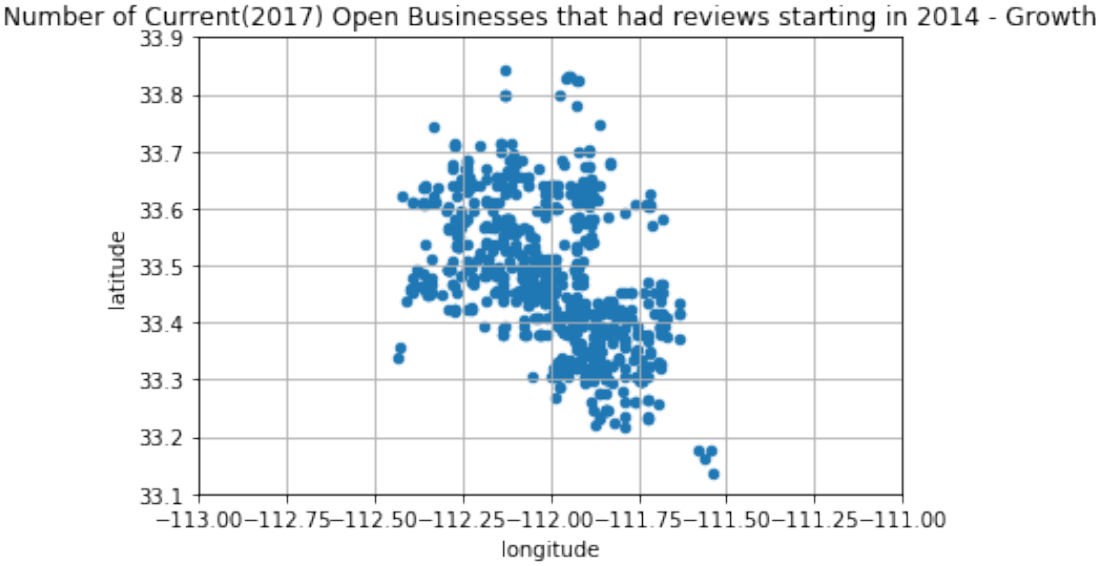
741


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00

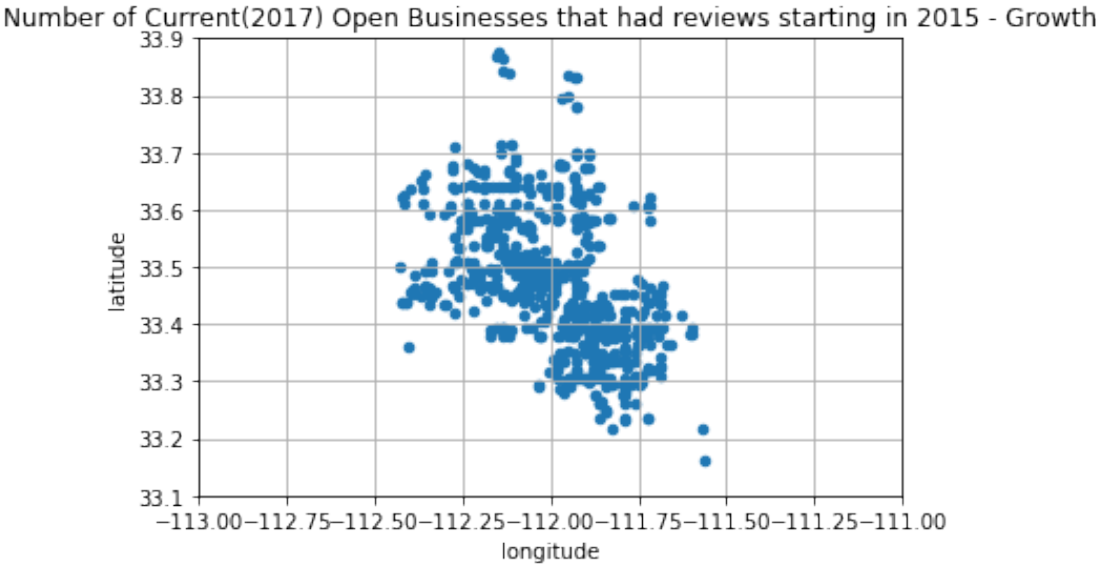
764


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

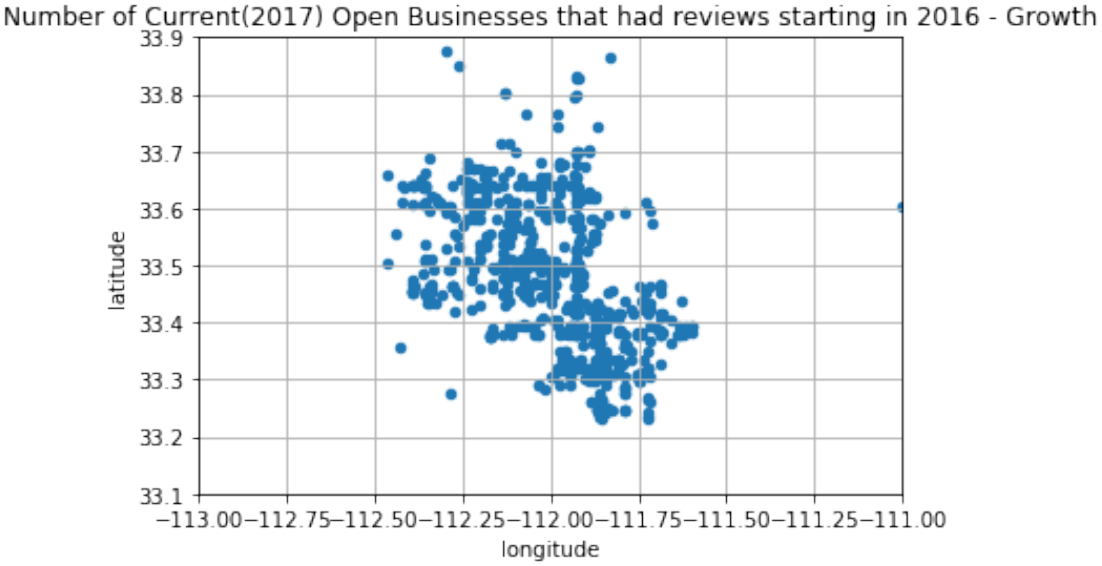
762


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

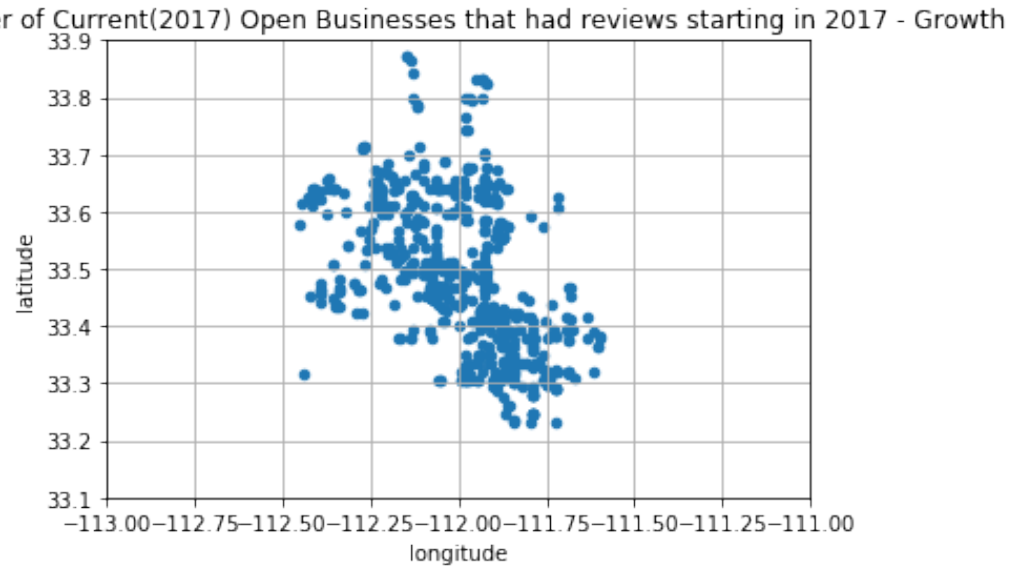
776


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00

780

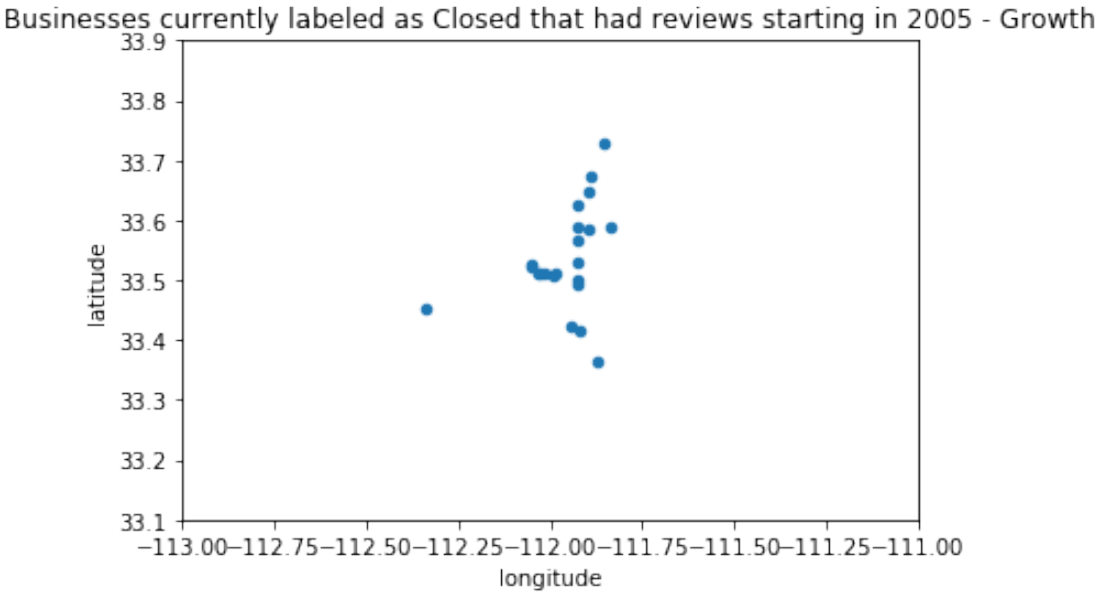

Number of Current(2017) Open Businesses that had reviews starting in 2016 - Growth
latitude
longitude
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00

691

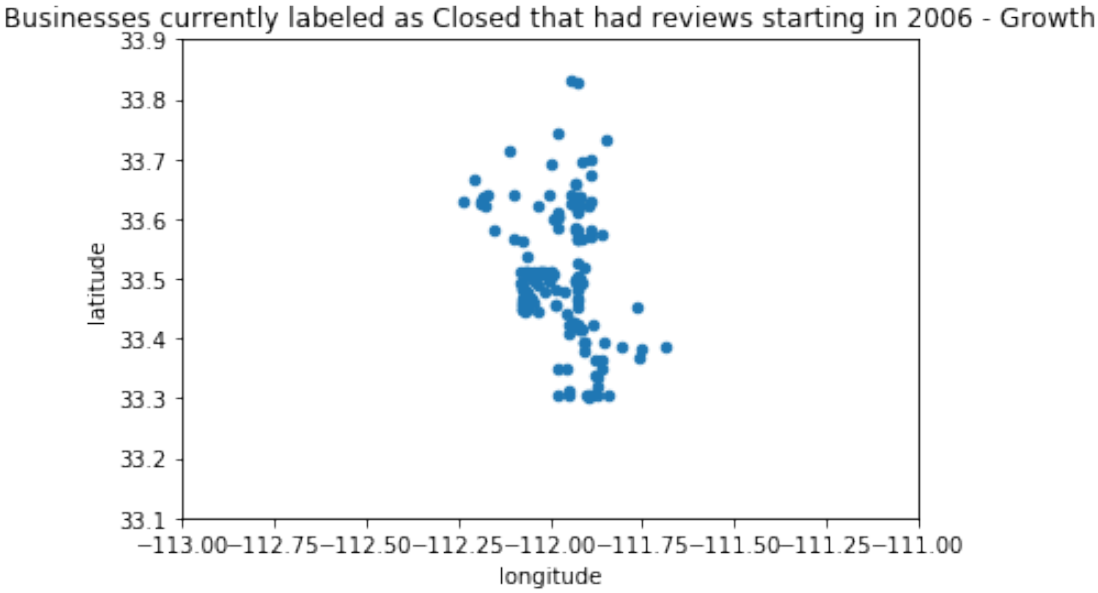


[16]:
```
for year in range(2005, 2018,1):
    dta = closedazbusiness[closedazbusiness['minyear'] == year]
    print(dta.shape[0])
    dta.plot(
    kind="scatter", x="longitude", y="latitude")
    plt.title('Businesses currently labeled as Closed that had reviews starting␣
 ↪in {0} - Growth'.format(year))
    plt.ylim(33.1, 33.9)
    plt.xlim(-113,
        -111)
    plt.show()
```

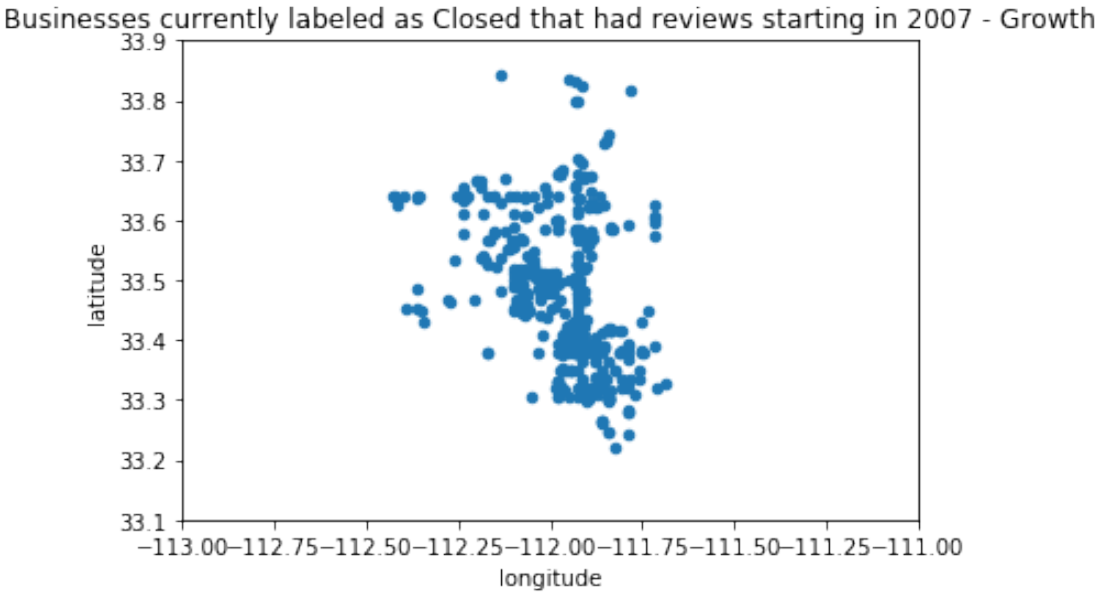
22


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

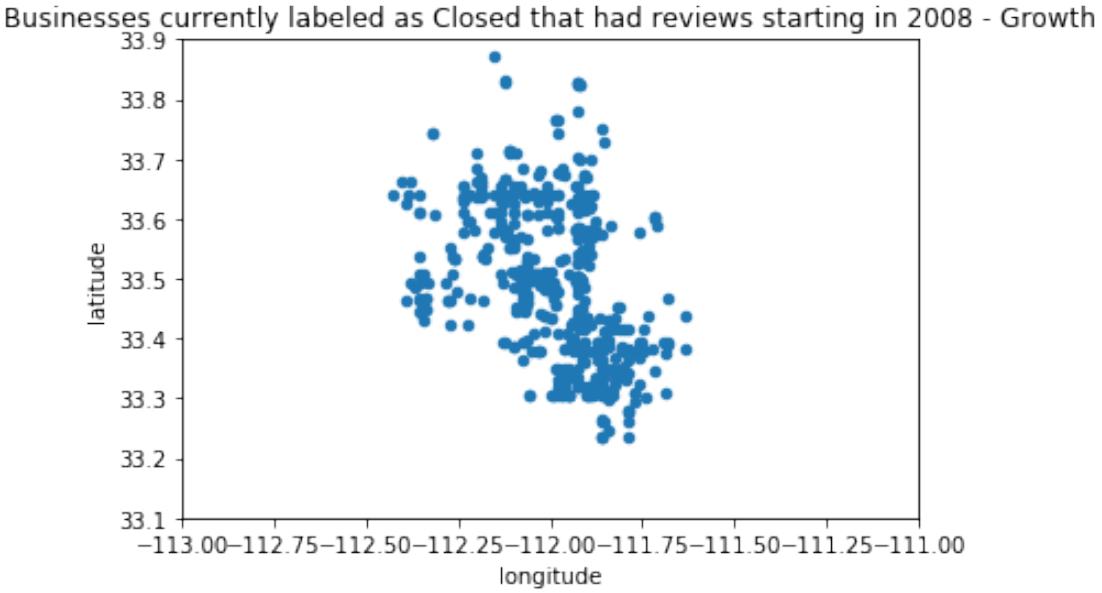
191


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

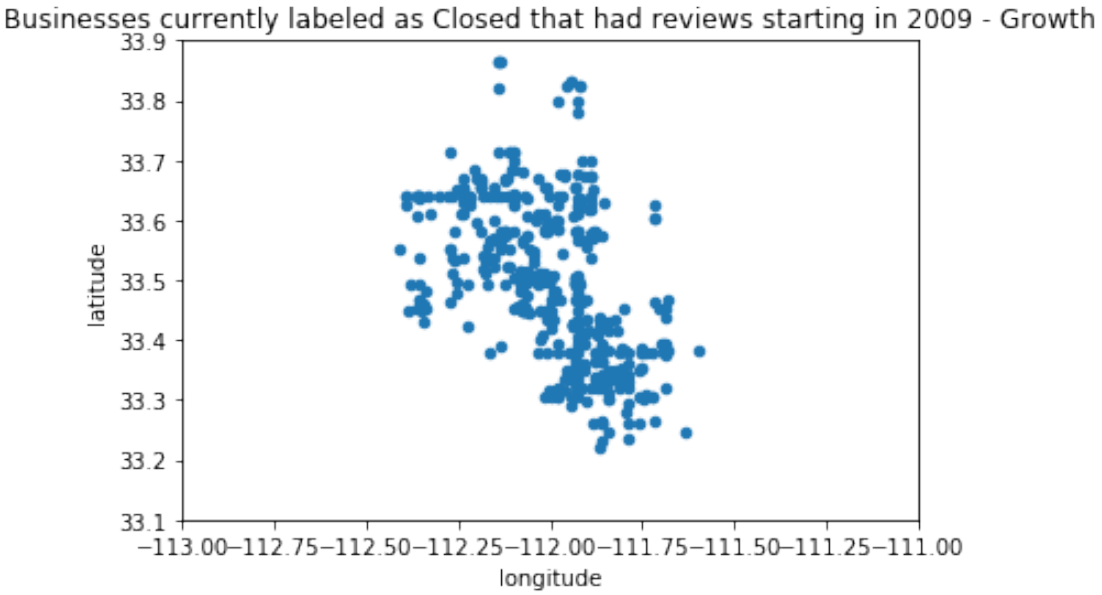
520


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

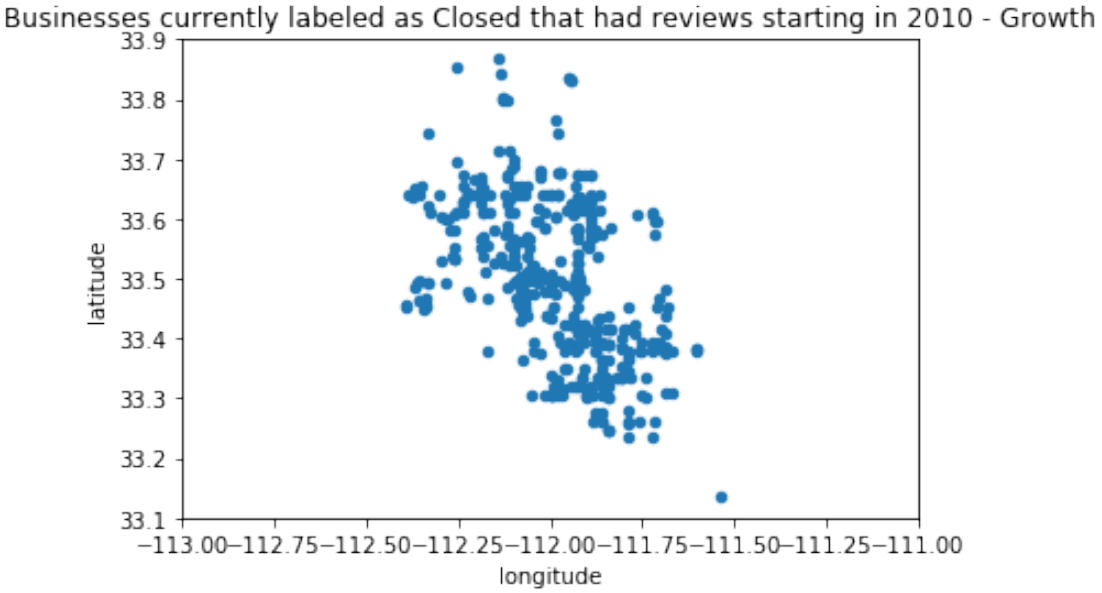
562


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

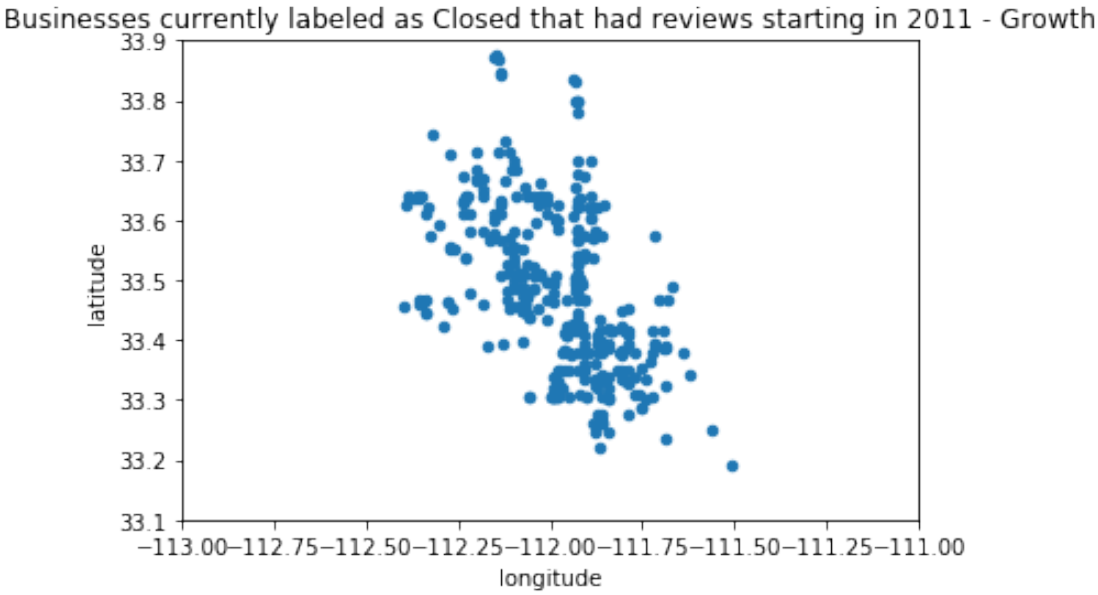
455


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
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

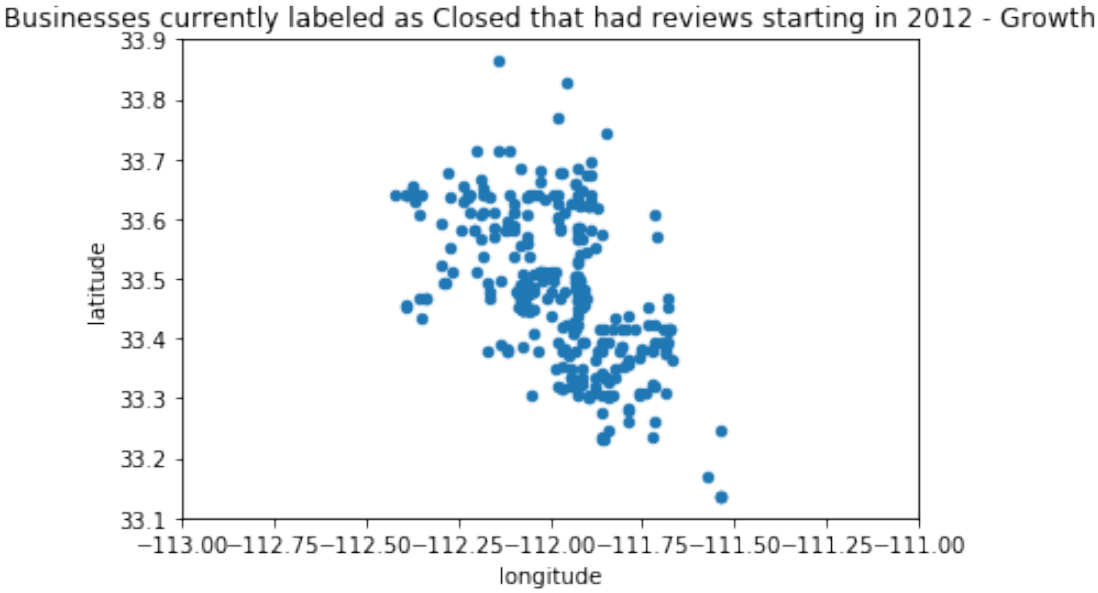
448


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
−113.00−112.75−112.50−112.25−112.00−111.75−111.50−111.25−111.00
longitude

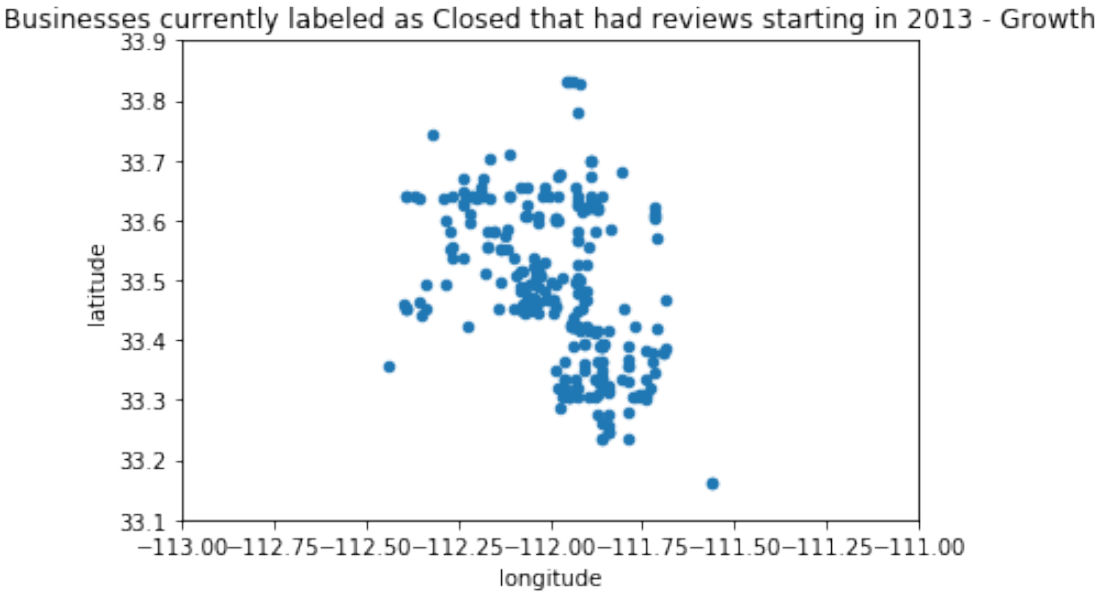
385


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

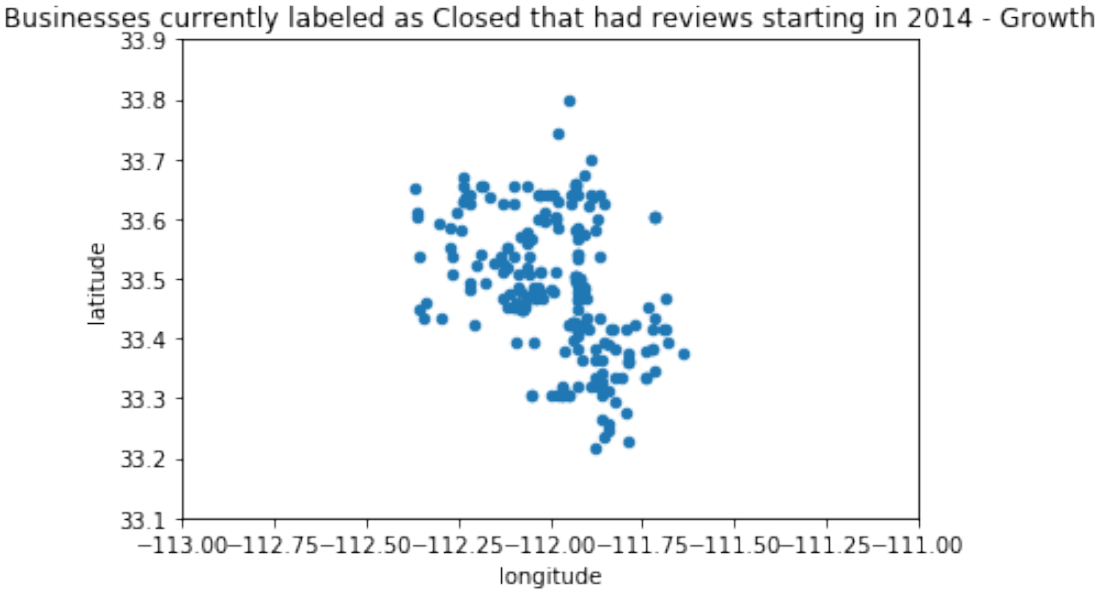
338


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

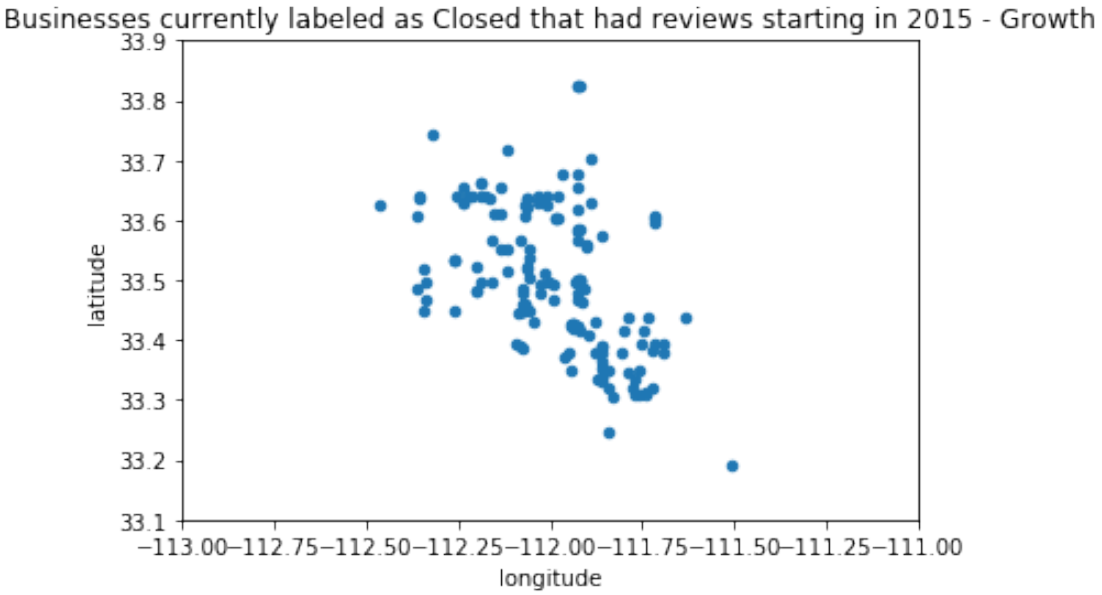
280


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

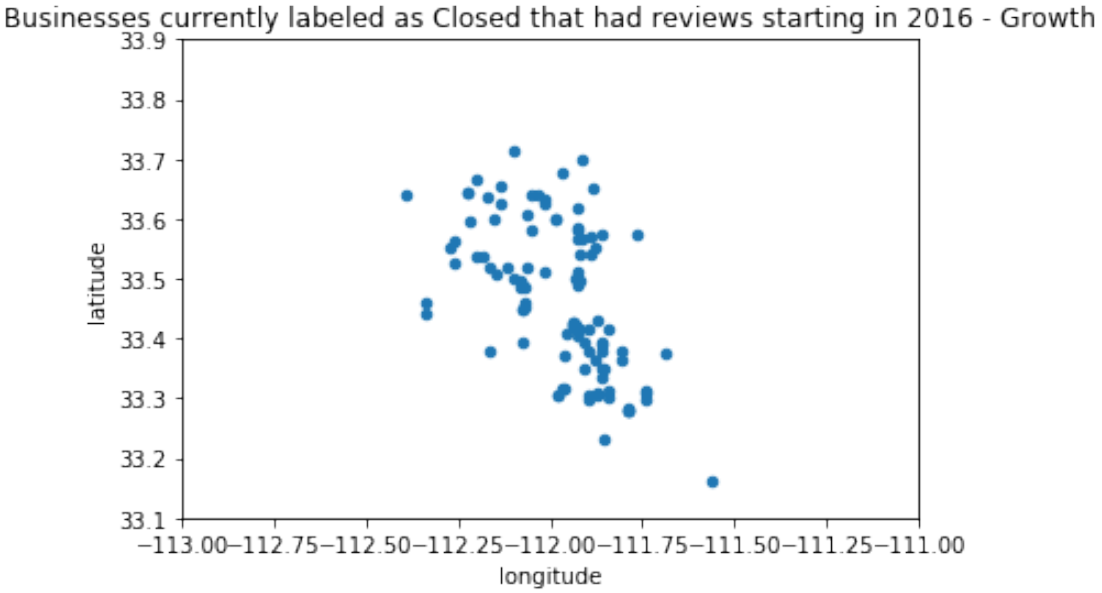
232


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

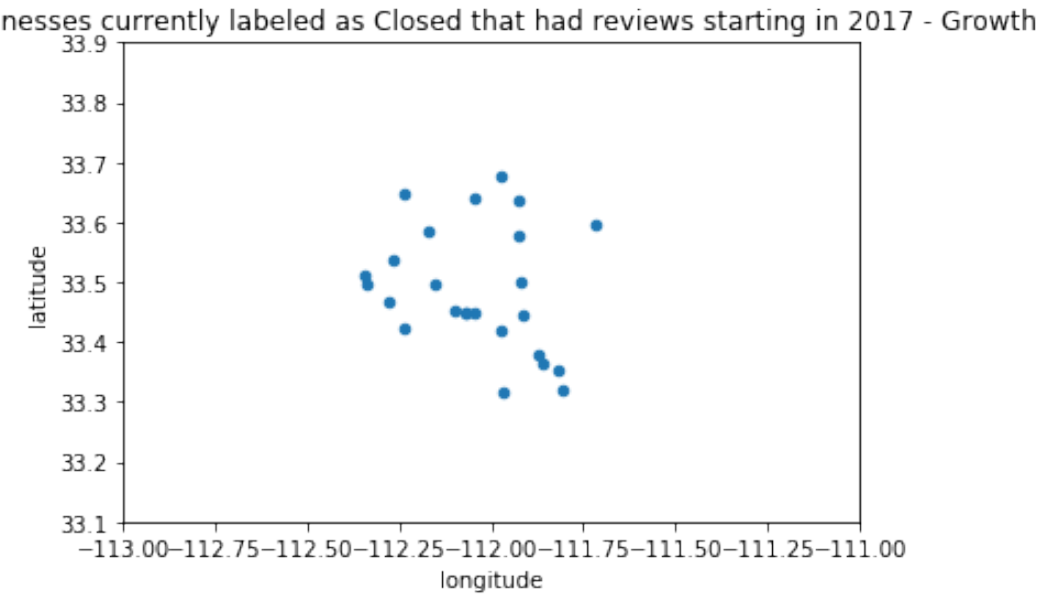
160


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
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

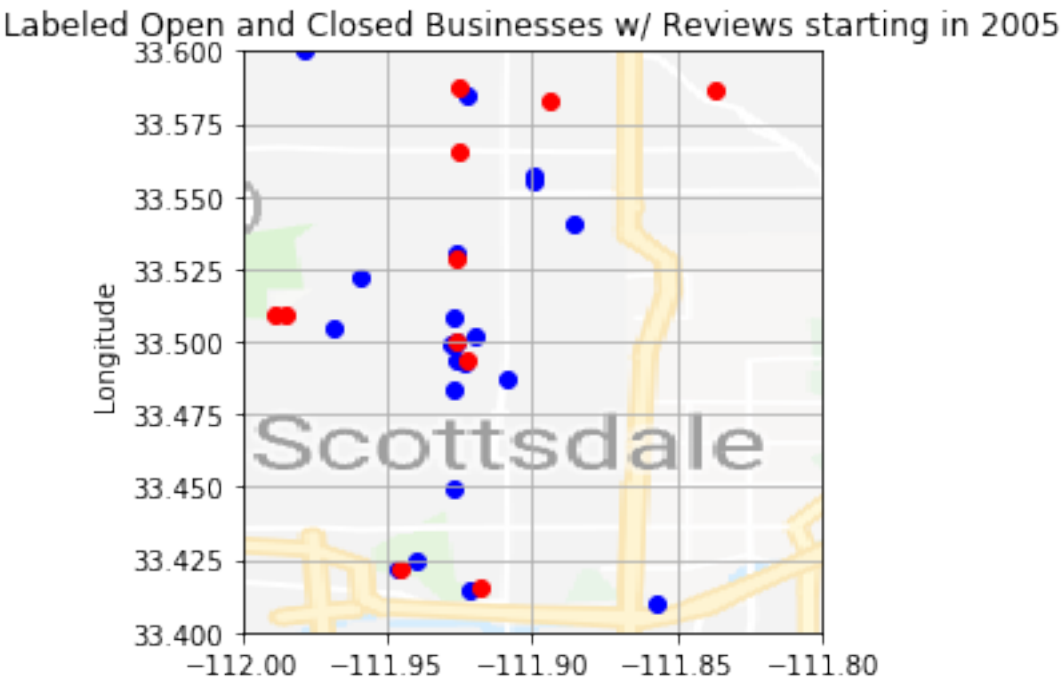
107

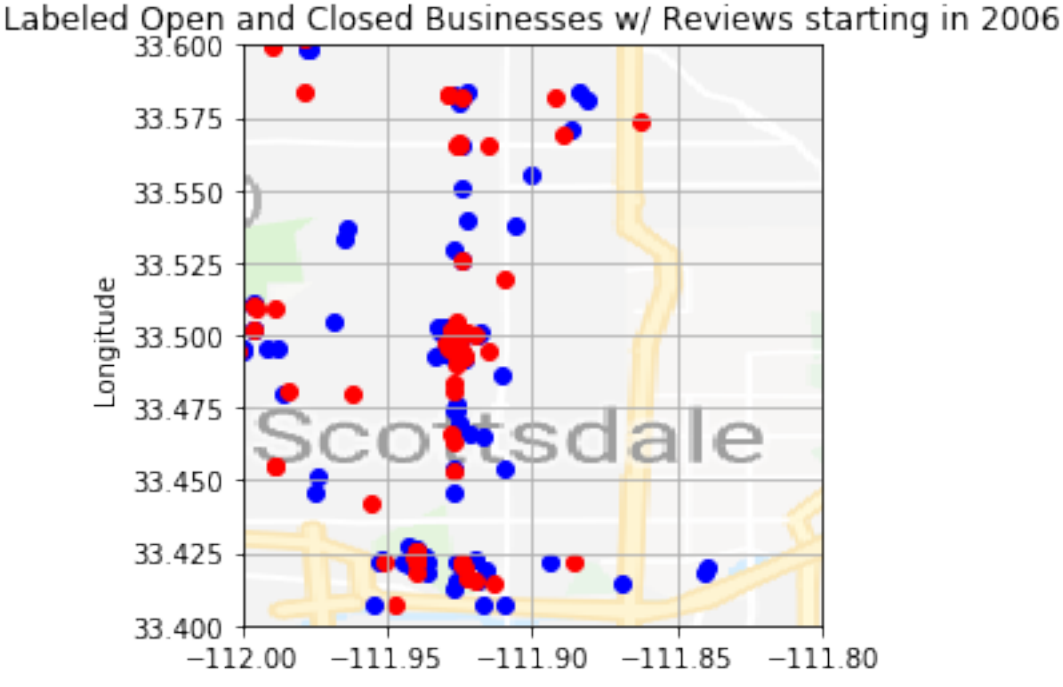

Businesses currently labeled as Closed that had reviews starting in 2016 - Growth
33.9
33.8
33.7
33.6
33.5
33.4
33.3
33.2
33.1
latitude
−113.00
−112.75
−112.50
−112.25
−112.00
−111.75
−111.50
−111.25
−111.00
longitude

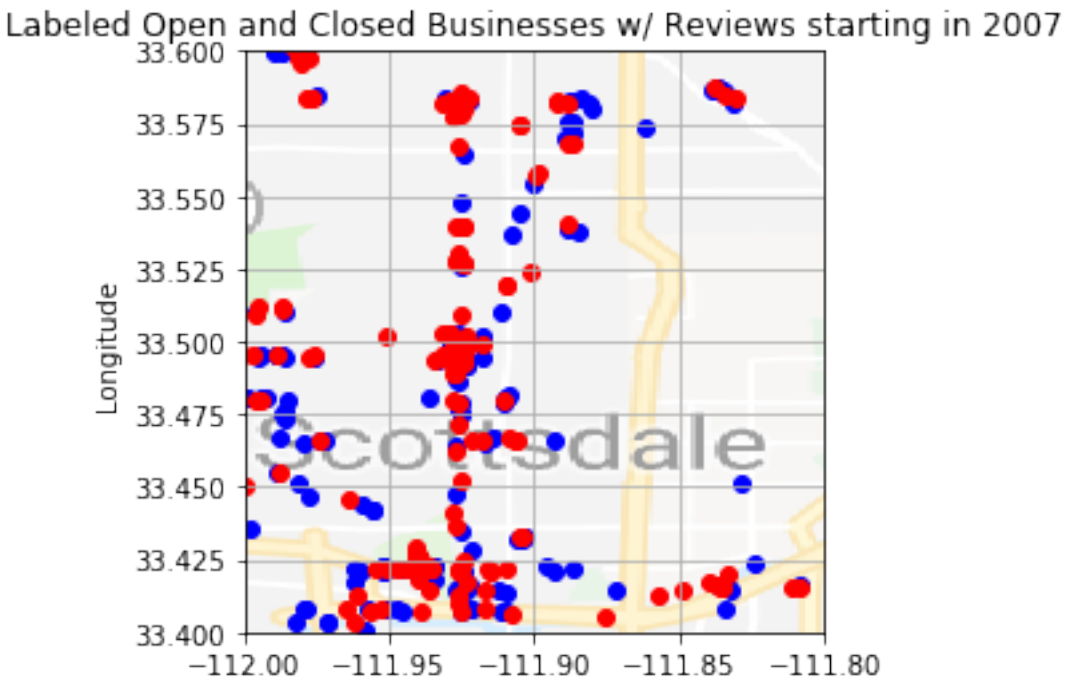
25

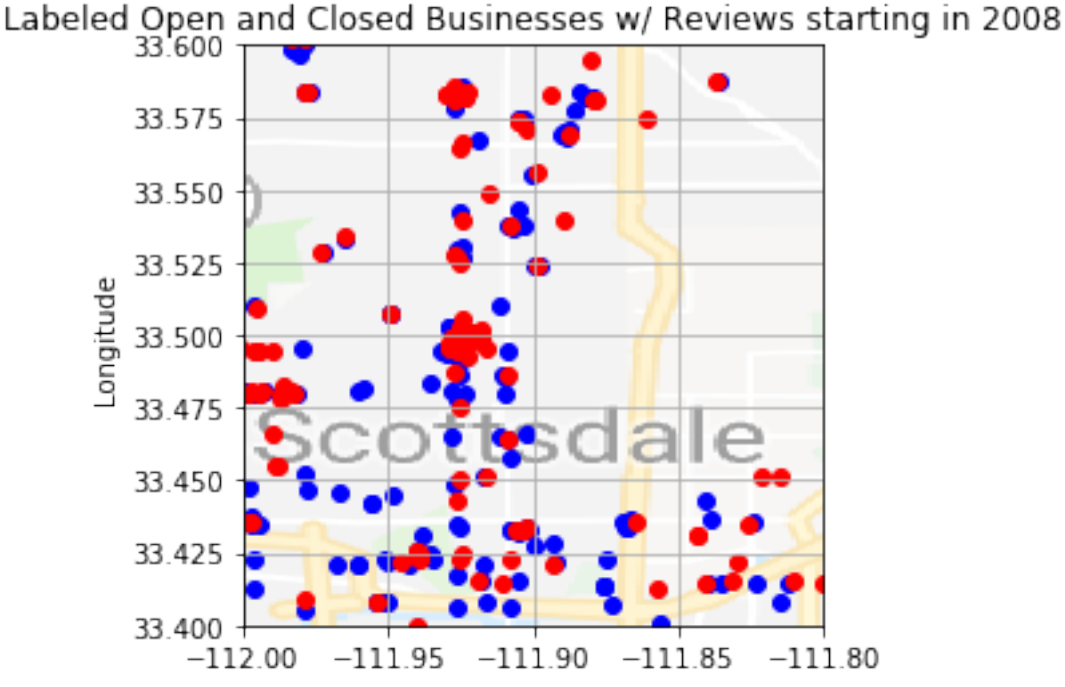


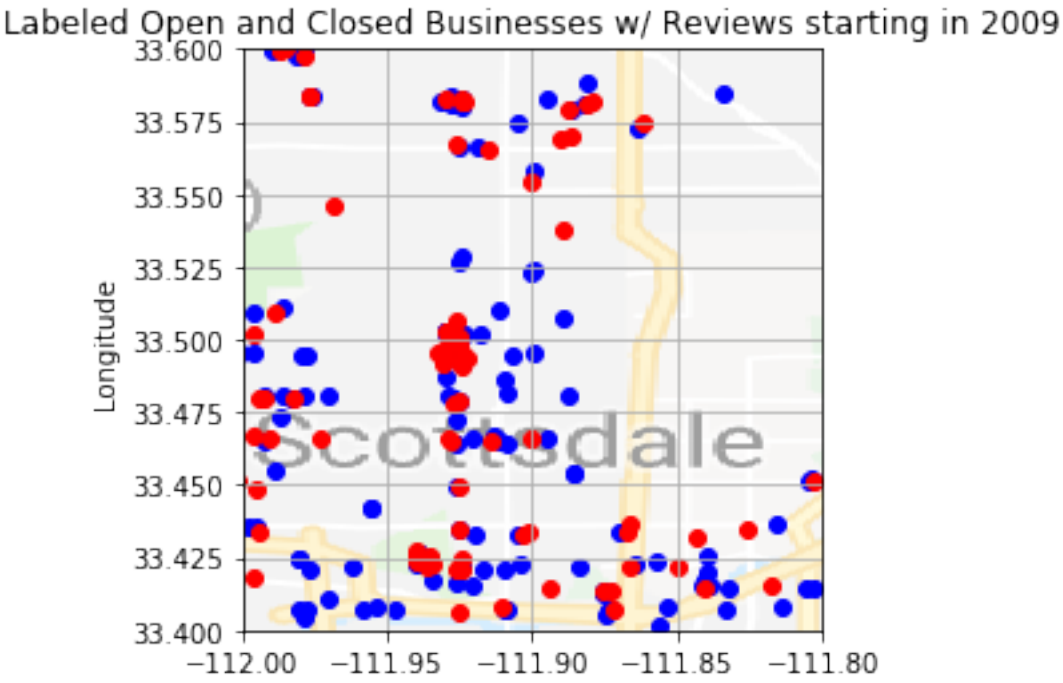
[17]:
```
import matplotlib.image as mpimg
phoenix_img=mpimg.imread('phoenix.png')
for year in range(2005, 2018,1):
    dta = openazbusiness[openazbusiness['minyear'] == year]
    cta = closedazbusiness[closedazbusiness['minyear'] == year]
    plt.imshow(phoenix_img, extent=[-112.6, -111.4, 33.1, 33.9], alpha=0.5)
    plt.scatter(dta['longitude'],dta['latitude'],color = 'blue', label = 'open')
    plt.scatter(cta['longitude'], cta['latitude'], color = 'red', label =␣
 ↪'closed')
    plt.title("Labeled Open and Closed Businesses w/ Reviews starting in {0}".
 ↪format(year))
    plt.ylim(33.4, 33.6)
    plt.ylabel("Latitude")
    plt.xlim(-112.0,
        -111.8)
    plt.ylabel("Longitude")
    plt.legend
    plt.grid()
    plt.show()
```

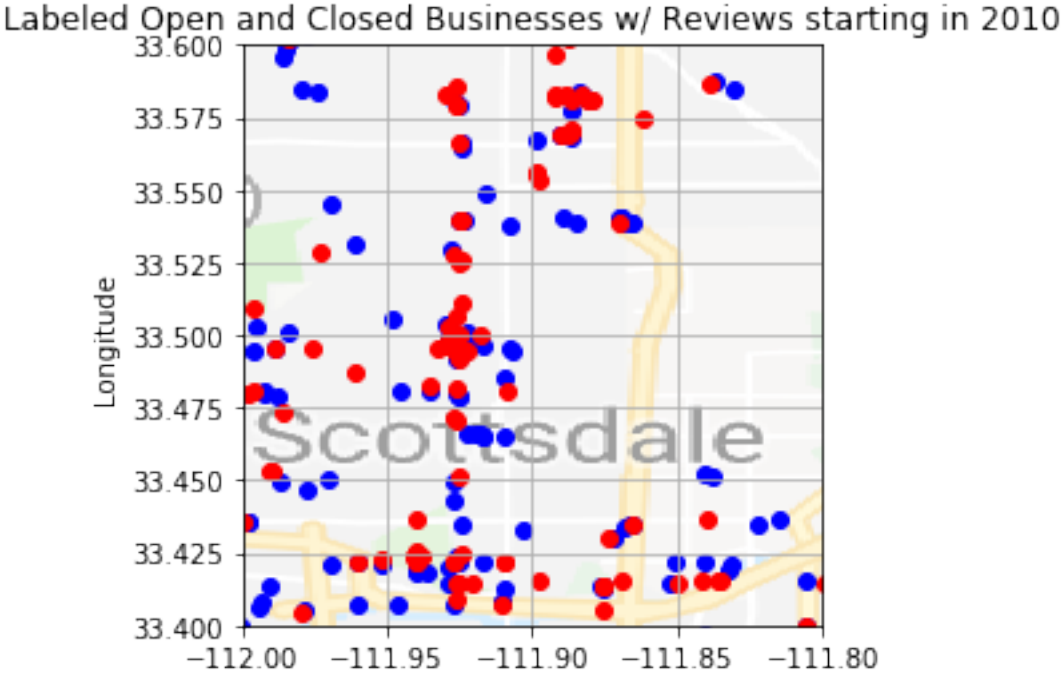

Labeled Open and Closed Businesses w/ Reviews starting in 2005
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale

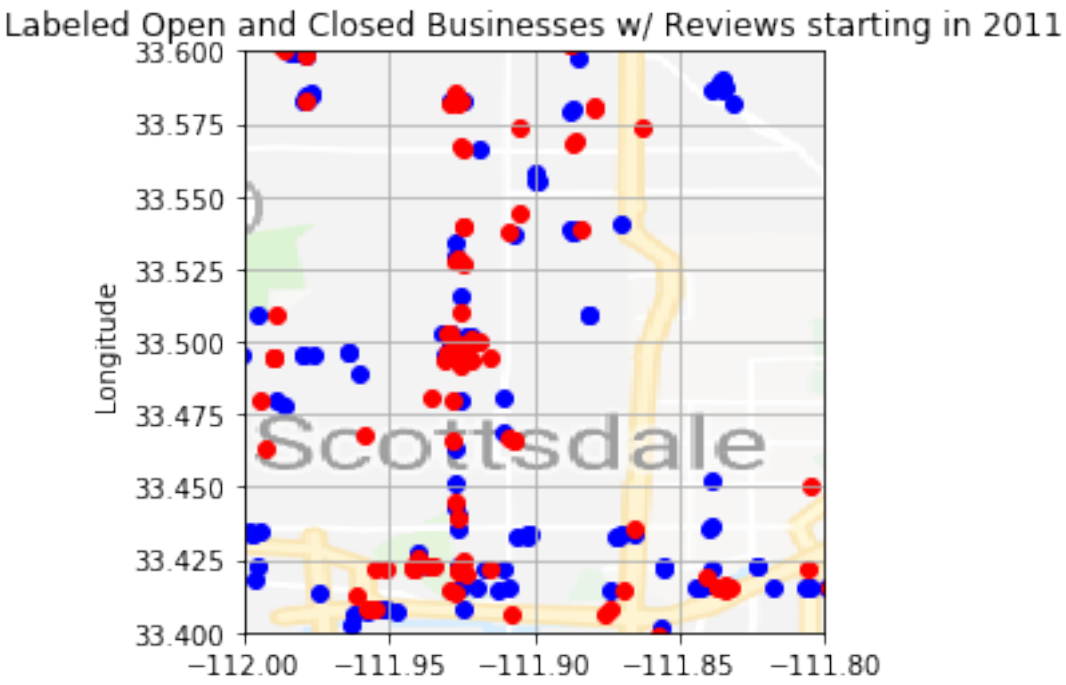

Labeled Open and Closed Businesses w/ Reviews starting in 2006
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale

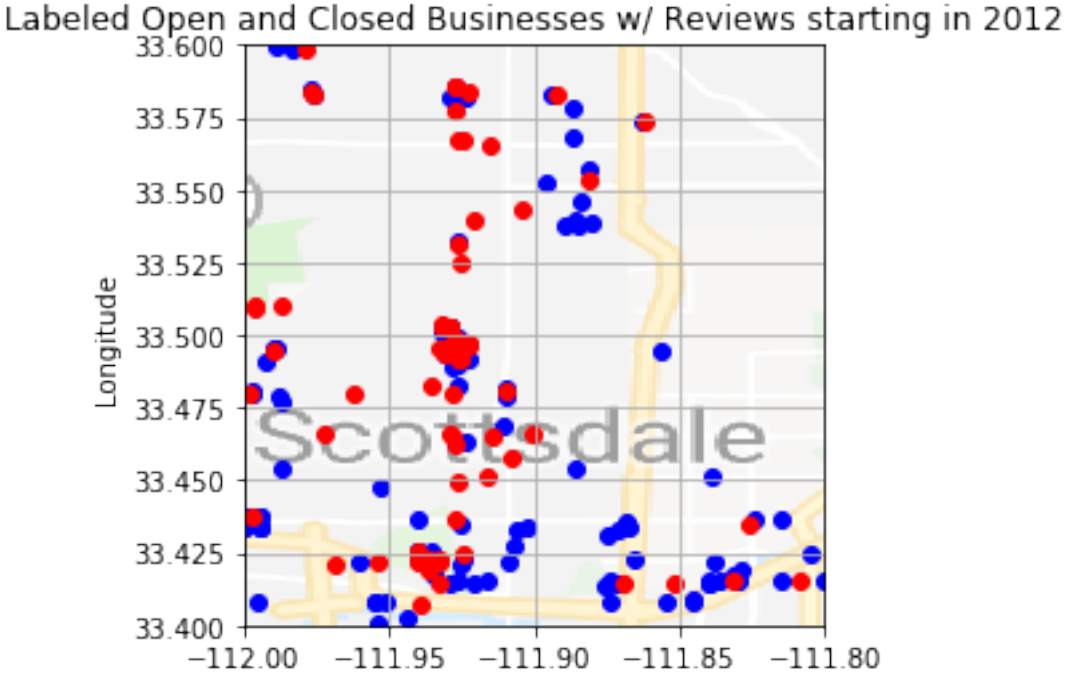

Labeled Open and Closed Businesses w/ Reviews starting in 2007
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale

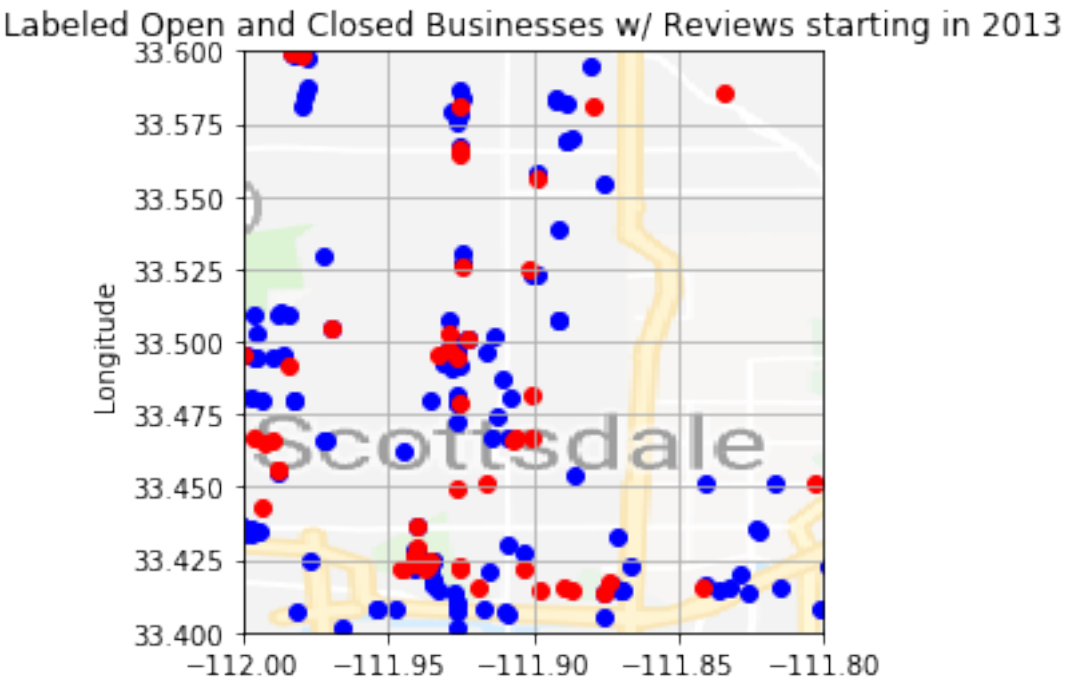

Labeled Open and Closed Businesses w/ Reviews starting in 2008
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale

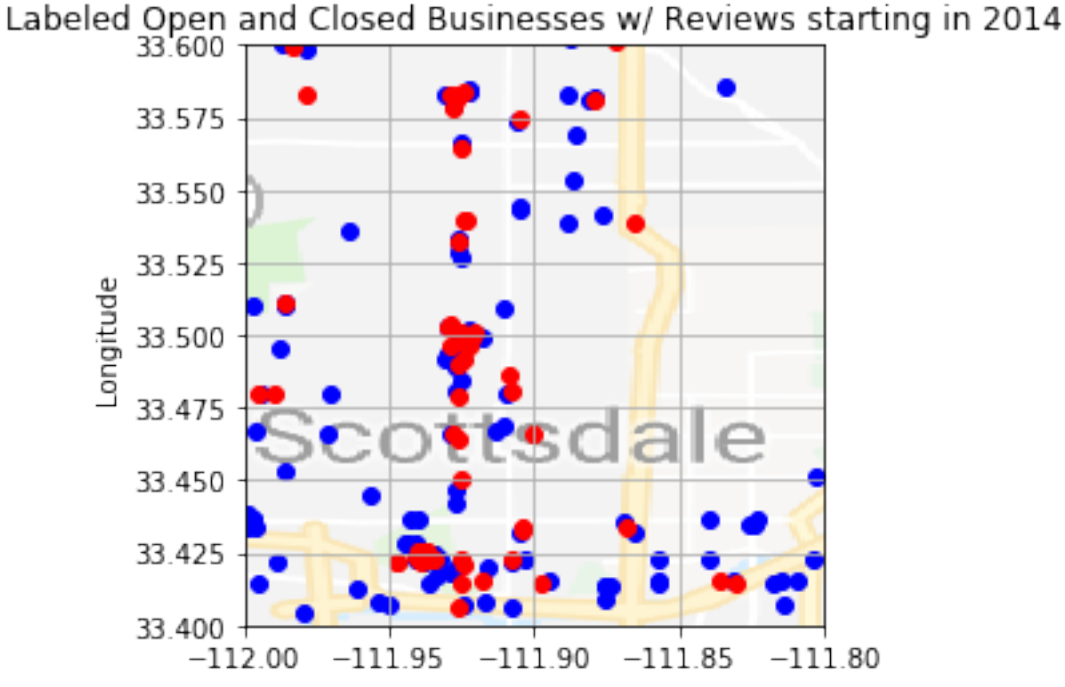

Labeled Open and Closed Businesses w/ Reviews starting in 2009
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale

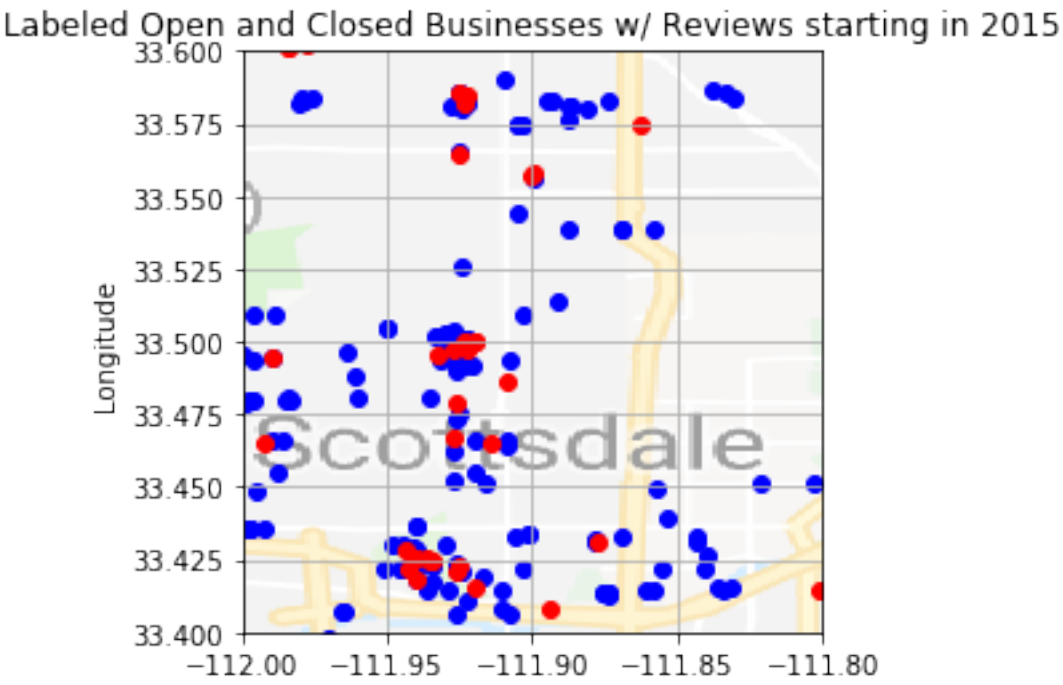

Labeled Open and Closed Businesses w/ Reviews starting in 2010
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale

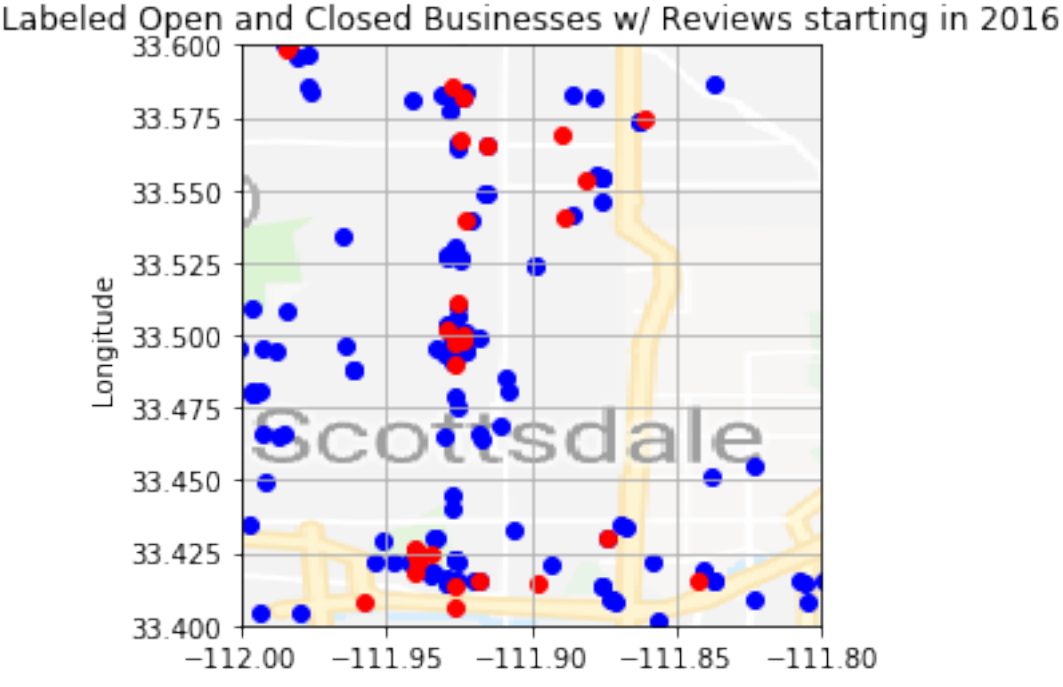

Labeled Open and Closed Businesses w/ Reviews starting in 2011
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale

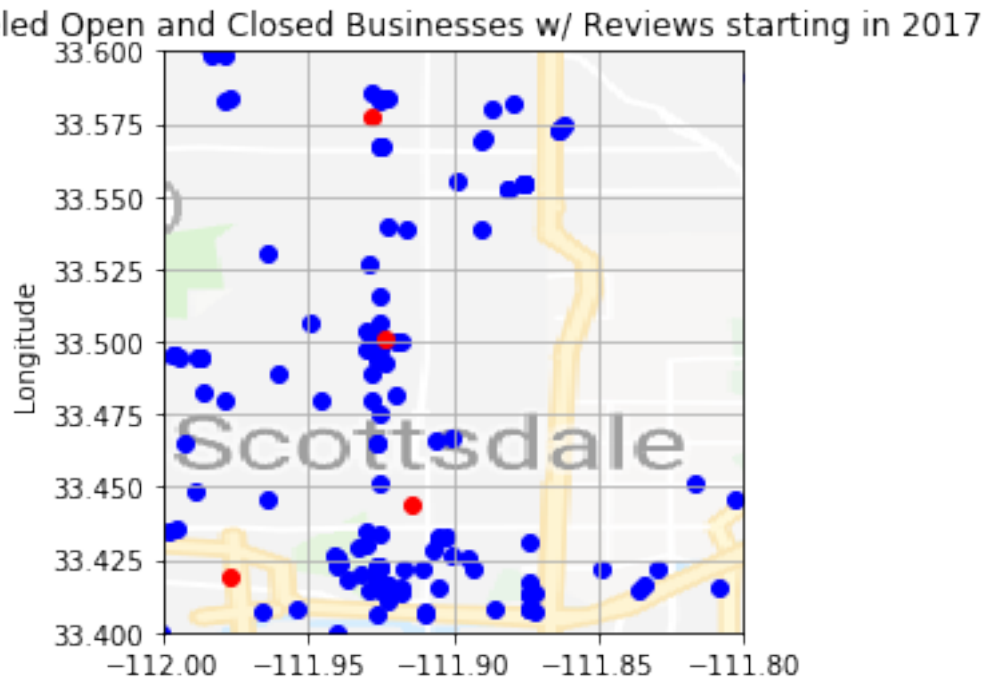

Labeled Open and Closed Businesses w/ Reviews starting in 2012
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale


Labeled Open and Closed Businesses w/ Reviews starting in 2013
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale


Labeled Open and Closed Businesses w/ Reviews starting in 2014
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale


Labeled Open and Closed Businesses w/ Reviews starting in 2015
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale


Labeled Open and Closed Businesses w/ Reviews starting in 2016
Longitude
33.600
33.575
33.550
33.525
33.500
33.475
33.450
33.425
33.400
−112.00
−111.95
−111.90
−111.85
−111.80
Scottsdale



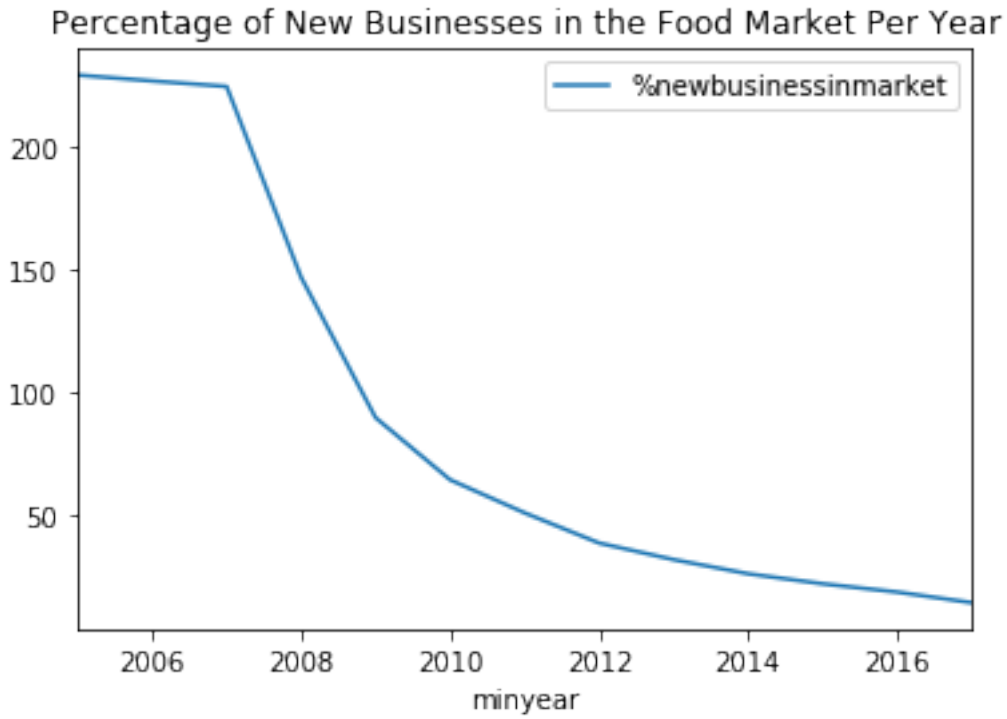
3 Market Growth and New Businesses Enrolled in Yelp

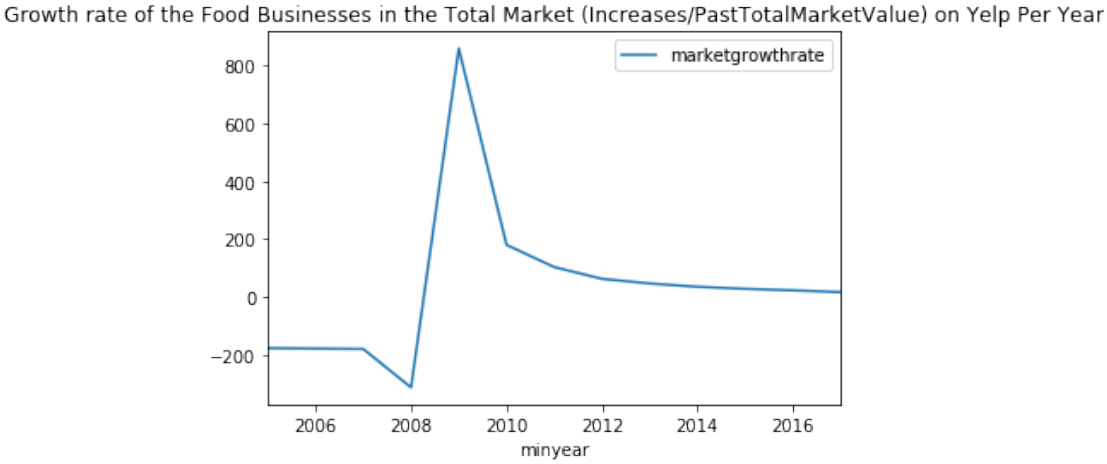
[7]:
```
minyearcount = arizonafoodbusiness.groupby(
    ['is_open','minyear']).count().reset_index().rename(
    columns = {'business_id': 'count of new businesses'})[['is_open', 'minyear',␣
 ↪'count of new businesses']]

#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)
```

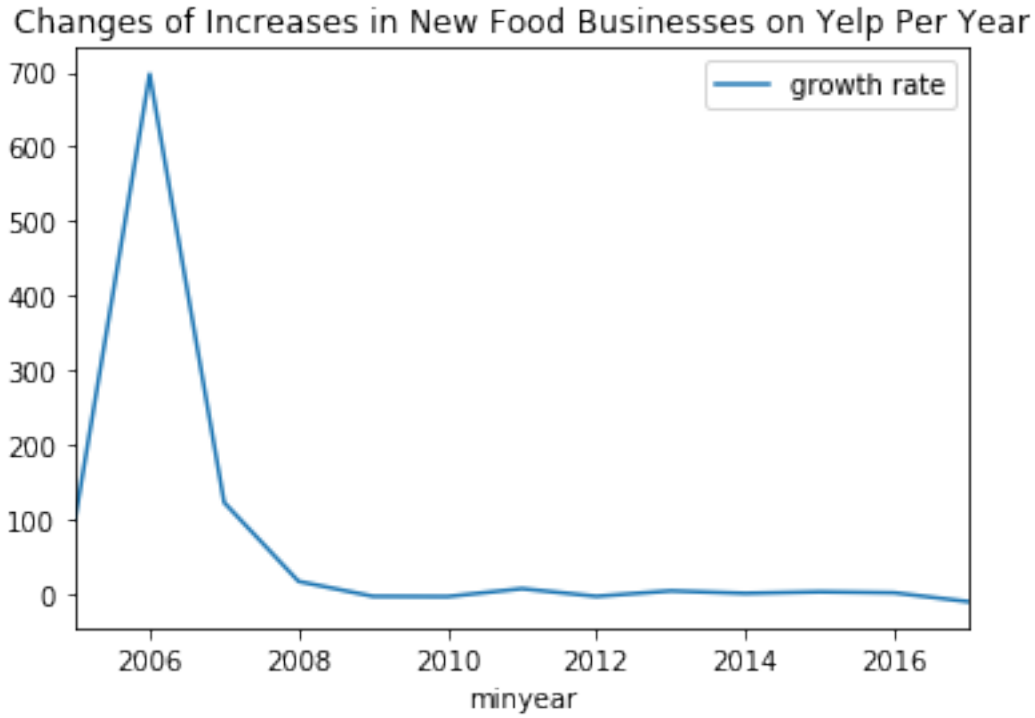
```
newbusiness['marketgrowthrate'] = newbusiness['count of new businesses']/
 ↪(newbusiness['totalbusinessmarket']-newbusiness['count of new businesses'])␣
 ↪*100
newbusiness['%newbusinessinmarket'] = newbusiness['count of new businesses']/
 ↪newbusiness['totalbusinessmarket'] *100

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')
newbusiness
```

```
//anaconda3/lib/python3.7/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.
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:12: UserWarning:
Boolean Series key will be reindexed to match DataFrame index.
  if sys.path[0] == '':
//anaconda3/lib/python3.7/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
//anaconda3/lib/python3.7/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
//anaconda3/lib/python3.7/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
//anaconda3/lib/python3.7/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

[7]:

	is_open	minyear	count of new businesses	totalbusinessmarket	\
13	1	2005	39	17	
14	1	2006	311	137	

15	1	2007	690	307
16	1	2008	799	544
17	1	2009	765	854
18	1	2010	730	1136
19	1	2011	774	1525
20	1	2012	741	1928
21	1	2013	764	2412
22	1	2014	762	2942
23	1	2015	776	3558
24	1	2016	780	4231
25	1	2017	691	4897

	marketgrowthrate	%newbusinessinmarket	growth rate
13	-177.272727	229.411765	100.000000
14	-178.735632	227.007299	697.435897
15	-180.156658	224.755700	121.864952
16	-313.333333	146.875000	15.797101
17	859.550562	89.578454	-4.255319
18	179.802956	64.260563	-4.575163
19	103.062583	50.754098	6.027397
20	62.426285	38.433610	-4.263566
21	46.359223	31.674959	3.103914
22	34.954128	25.900748	-0.261780
23	27.893602	21.810006	1.837270
24	22.602144	18.435358	0.515464
25	16.428911	14.110680	-11.410256

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 growth rate has declined and the number of new businesses per year on Yelp has been relatively constant around 700-800.

4 Findings

1. The growth rate of the restaurants creating a Yelp Page in Arizona is still positive, but has been constant in the recent years. There was a huge increase in new business pages before 2010. Yelp started in 2004 locally in San Francisco, CA. Yelp IPO in 2012. Thus, Yelp only grew in Arizona beginning 2007-2010.

2. Mexican and American are the two most popular cuisine, and this data is supported by 49% of residents in Phoenix, Arizona are Hispanics.

3. About 750 new businesses are gaining a review page star into Yelp each year.

[ ]:

[ ]:

[ ]: