SQL FOR DATA SCIENCE CAPSTONE PROJECT

DONE BY: KAI TAN POK HSUAN

DATE: 08/1/2020

CLIENT

- The Client that I have chosen to serve in this project is a Restaurant Startup who is interested to set up a restaurant in Las Vegas.
- Based on Yelp Dataset, I have been tasked to advise the management of this Restaurant Startup where is the preferred location that they should set up in. Another question that may need to be answer is what is the operating hours they should follow in order to maximize business revenue which eventually leads to business success.

HYPOTHESES

- There are actually a few variables that I am analysing. They are namely, “Business Address”, “stars(review rating)”, “review_count”, “Number of checkin by customers” and “hours(Operating hours)”
- My initial hypothesis is that there would be relationship between “Business Address” and “stars(review rating) and review_count”.

APPROACH

- My approach is to load to Yelp Dataset into Jupyter Notebook first.
- Subsequently, I will do explorative data analysis.
- After I have understand the data better, I would go into deeper analysis.

PROBLEMS FACED

- I have faced a problem when I am loading the dataset into Jupyter Notebook. Based on the Yelp dataset, there are actually separated into a few json files. They are namely “User”, “Review”, ”Business” and “Checkin” and more..
- As each json file is very big. I have run into memory problem when loading into Jupyter Notebook. Therefore, I am left with no choice but to subsample the data into smaller set and load them in.
- After Loading in the data, I have found out the based on the tasks I am required to do. “User” and “Review” seems to be irrelevant. Therefore, I have discontinued working on them, and thus primarily focus on “Business” and “Checkin” dataset.

SOLUTION USED TO OVERCOME MEMORY ERROR

```
~\Anaconda3\lib\encodings\cp1252.py in decode(self, input, final)
     21 class IncrementalDecoder(codecs.IncrementalDecoder):
     22     def decode(self, input, final=False):
---> 23         return codecs.charmap_decode(input,self.errors,decoding_table)[0]
     24
     25 class StreamWriter(Codec,codecs.StreamWriter):

MemoryError:
```

In [6]:
```
"""
    Due to Memory constraint, we will read in only 100,000 records from the dataset
    Calculation: 1000 x 100(chunksize) = 100,000
"""

appended_data = []

for gm_chunk in pd.read_json("C:\\Users\\kai\\Desktop\\user.json",chunksize=100, lines=True):

    appended_data.append(gm_chunk)
    if len(appended_data) >= 1000:
        break
```

In [7]:
```
type(appended_data[0])
```

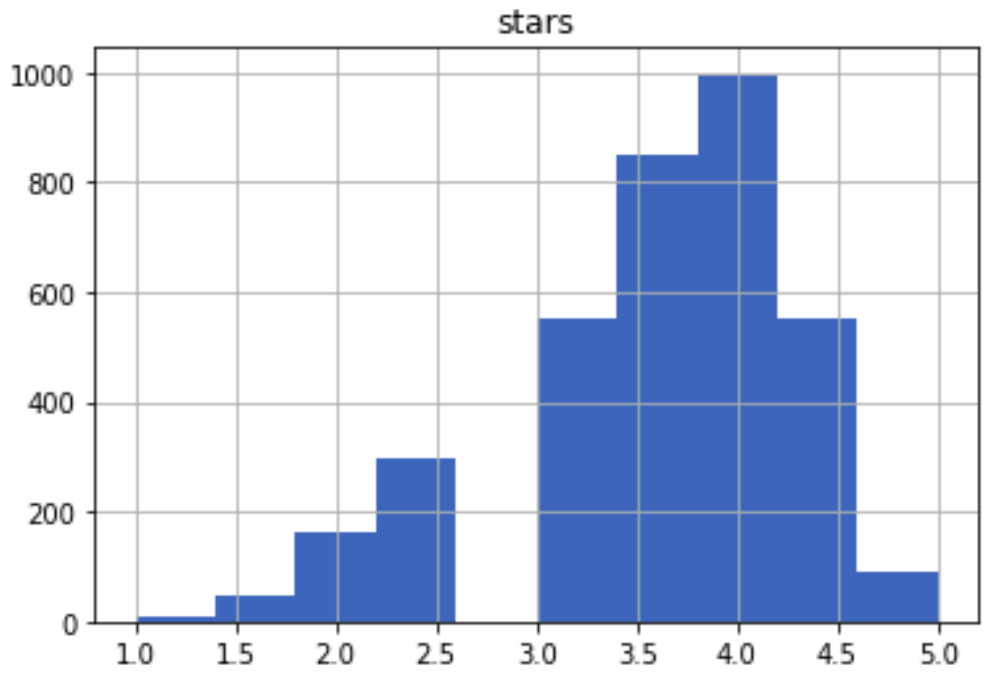
Out[7]: pandas.core.frame.DataFrame

In [8]:
```
type(appended_data)
```

DISTRIBUTION OF STARS (USER RATING)

```
In [225]: business_checkin_merged_LasVegas_Restaurant.hist('stars')
```

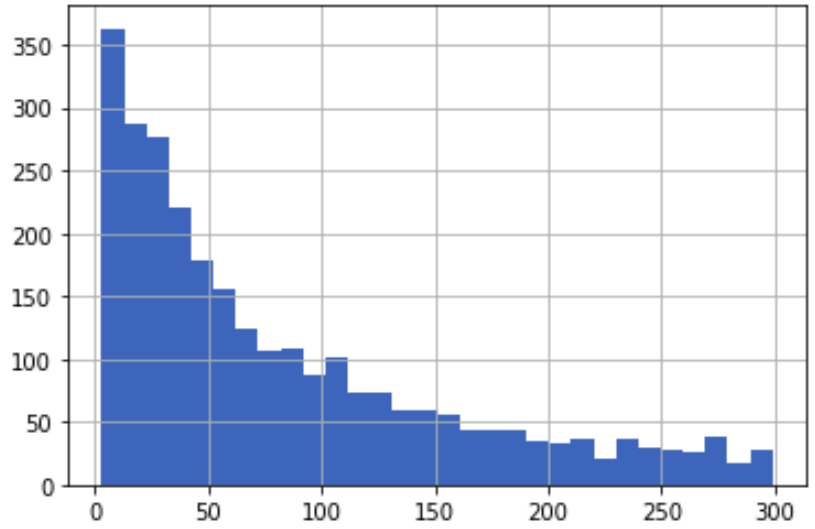
```
Out[225]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001BC77A079C8>]],
          dtype=object)
```



DISTRIBUTION OF REVIEW COUNT

In [228]:
```
# We have removed some outlier so that it will be easier to visualize

business_checkin_merged_LasVegas_Restaurant.loc[business_checkin_merged_LasVegas_Restaurant['review_count'] < 300 ]['review_count
```

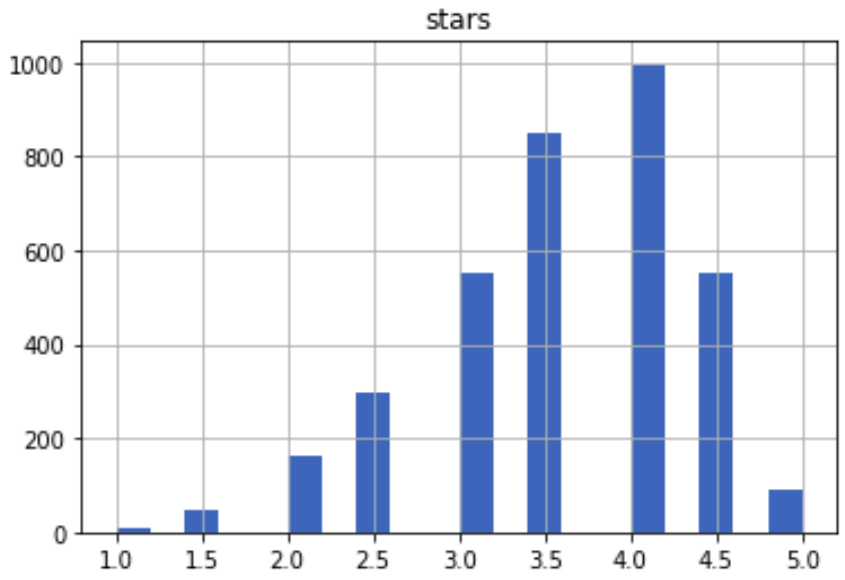
Out[228]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc77a9b948>



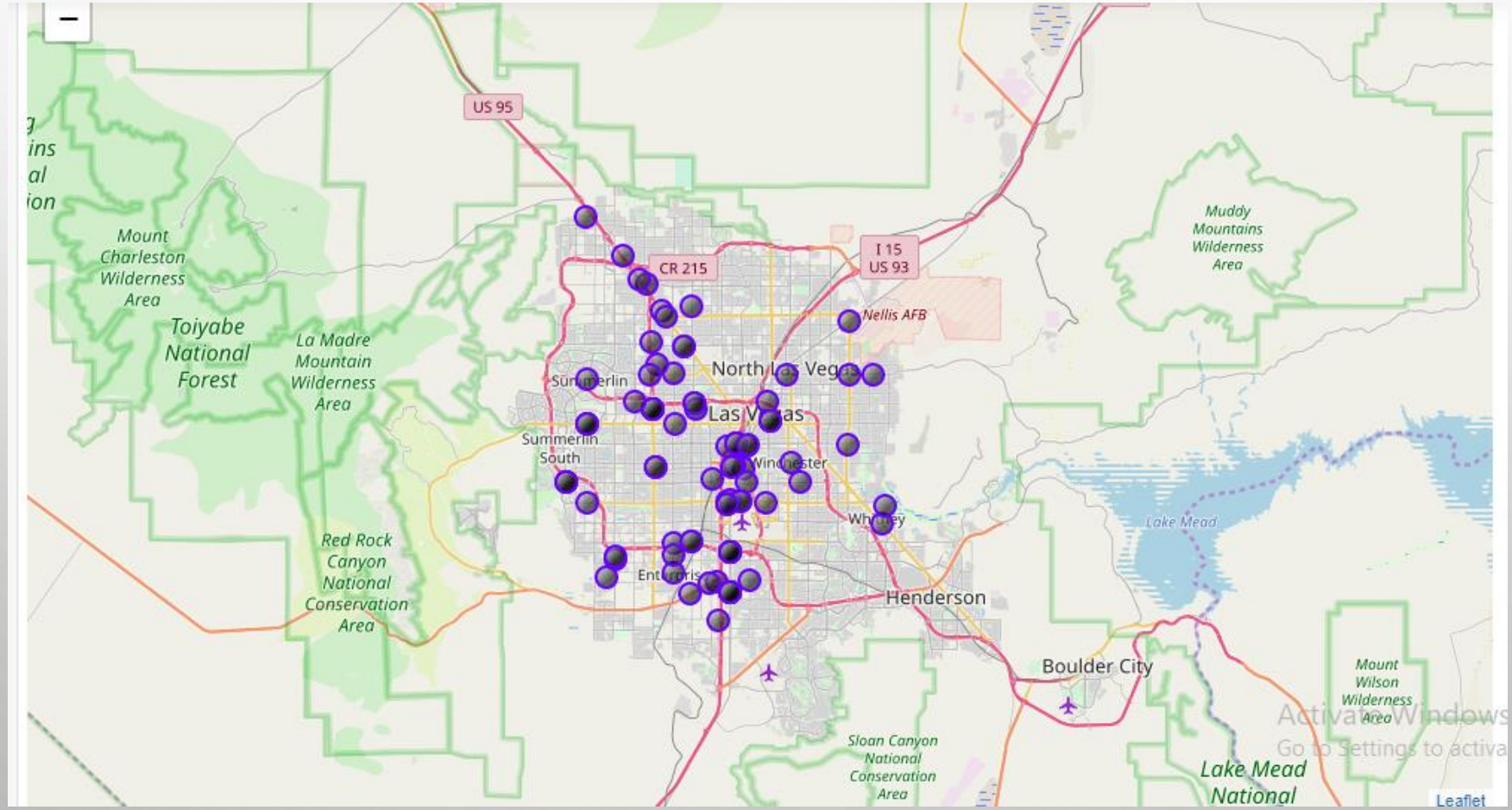
ADDITIONAL PLOT

```
In [229]: business_checkin_merged_LasVegas_Restaurant.hist('stars', bins=20)
```

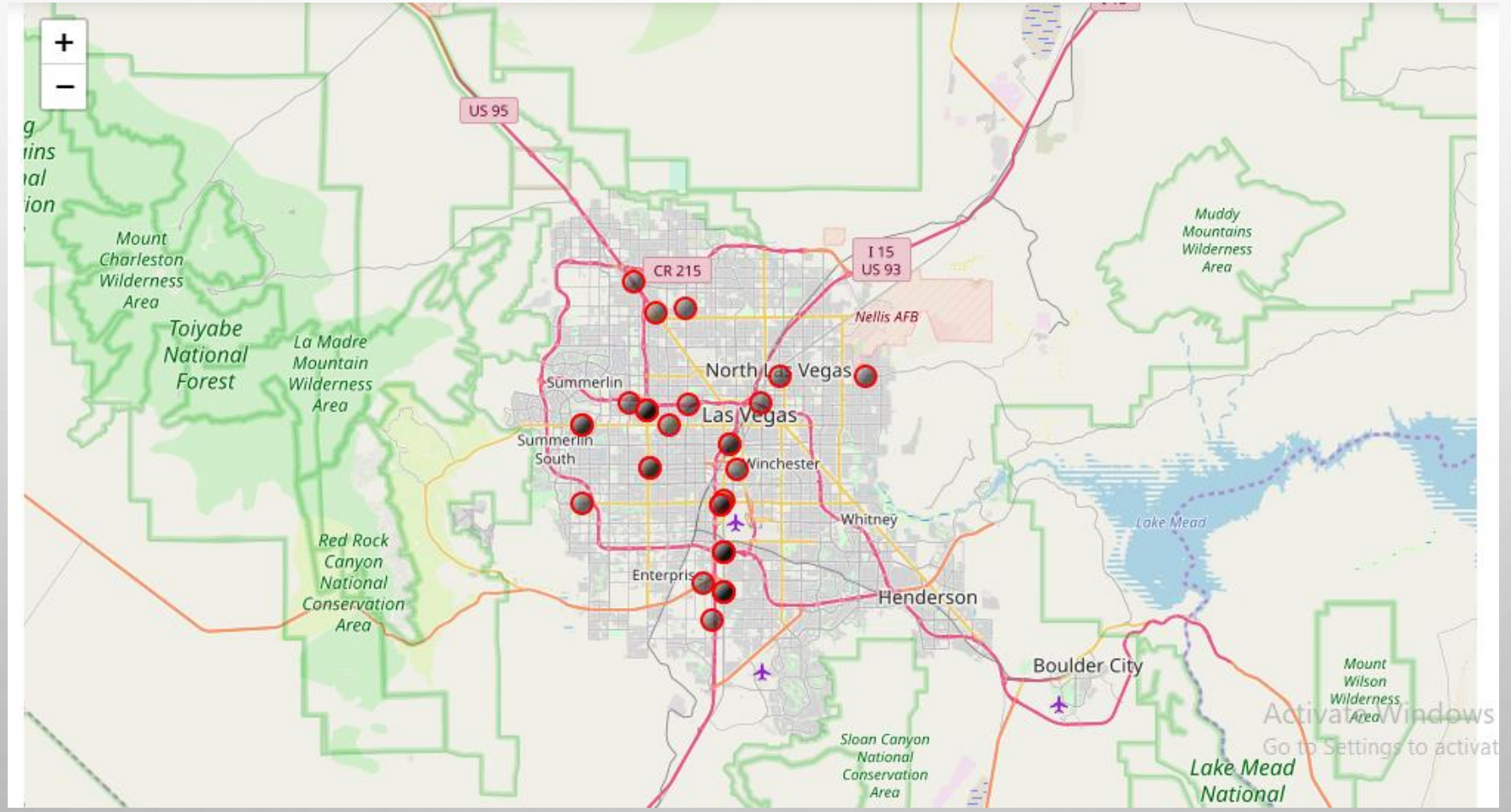
```
Out[229]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001BC77B4F588>]],
          dtype=object)
```



124 RESTAURANT IN LAS VEGAS



TOP 50 BASED ON HIGHEST STARS(USER RATING)



CONVERT “DATE OF CHECKIN” COLUMN TO “NO OF CHECKIN” COLUMN

In [291]:
```
tesst = list(filtered_data['date'])
# for i in tesst:
#     print(i)
i=0
my_list=[]
while i < 124:
    my_list.append(len(tesst[i].split(", ")))
    i += 1

# tesst[123]

# for index, value in filtered_data['date'].items():
#     print(index)
#     print("Index : {}, Value : {}".format(index, value))
```

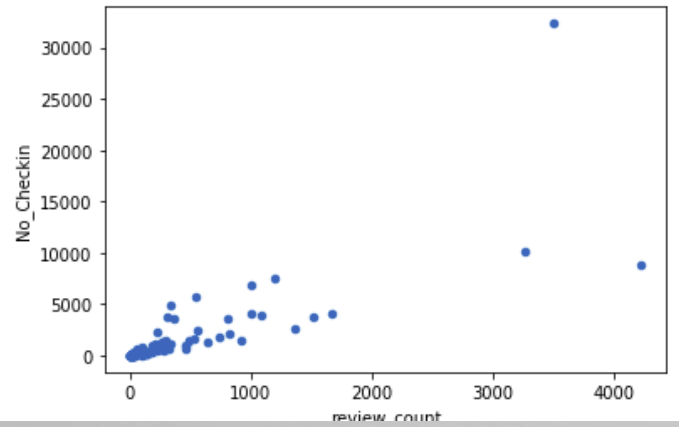
In [297]:
```
qqq = np.array(my_list)
qqq = pd.Series(qqq)
filtered_data['No_Checkin'] = qqq
filtered_data
```

HIGH CORRELATION BETWEEN "REVIEW_COUNT" AND "NO_CHECKIN"

	lat	long	stars	review_count	No_Checkin
lat	1.0000	0.8027	0.0870	-0.0657	-0.0331
long	0.8027	1.0000	0.1642	-0.0871	-0.0713
stars	0.0870	0.1642	1.0000	0.1650	0.1740
review_count	-0.0657	-0.0871	0.1650	1.0000	0.7946
No_Checkin	-0.0331	-0.0713	0.1740	0.7946	1.0000

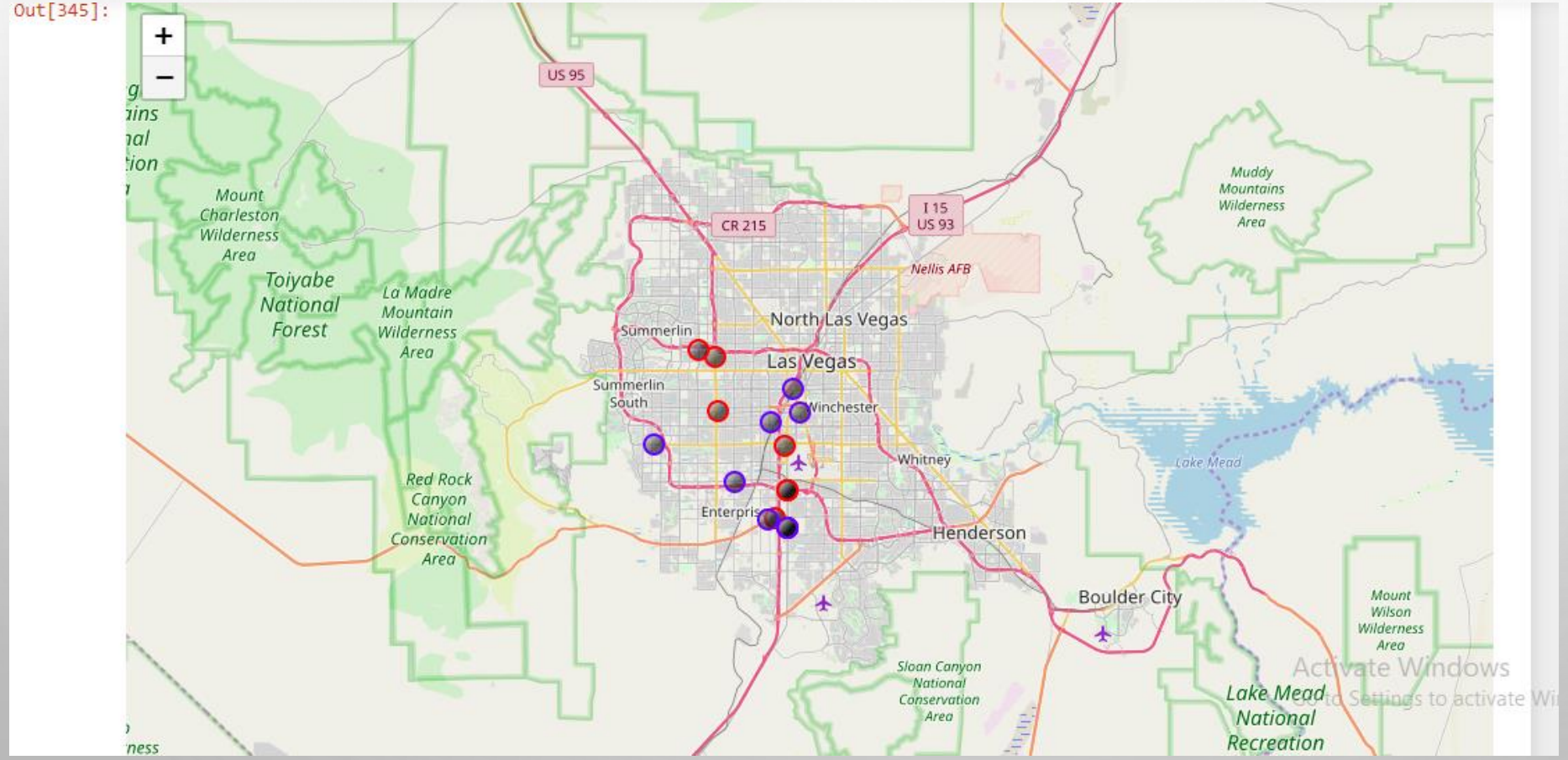
In [301]:
```
filtered_data[['review_count','No_Checkin']].plot(kind="scatter", x='review_count',y="No_Checkin")

# There seems to be a high correlation between "review_count" and "No_Checkin"
# It seems like most of the customers leave a review when they are check-in to the business
```

Out[301]: <matplotlib.axes._subplots.AxesSubplot at 0x1bc80001c88>



TOP 20 BUSINESSES BASED ON “NO_CHECKIN” DESC

Out[345]:



RED -> Top 10
BLUE -> Next 10

DISCUSS INSIGHTS DISCOVERED

- After completing the analysis, i have discovered that my initial hypothesis is wrong. as there don't seems to be much relation between business address and "user rating and review_count".
- Nevertheless, i have managed to uncover some insignt. first, there is high correlation between "review_count" and "no_checkin". this might mean that customer are more likely to give a review when they have checkin to the business.
- Another insight i have uncovered is that based on the on "top 20 businesses based on "no_checkin" in desc order. the businesses that has high number of checkin seems to cluster in a certain region. as more checkin imply more traffic for the business, i feel that the more checkin a business has, the better it is.
- Therefore based on the analysis, if possible, I would advise the new start-up management to locate within that cluster.

RECOMMENDATIONS AND ACTIONS

- I have felt that I needed more data in order to make a better decision. As my Jupyter Notebook faced memory constraints error when load in too much record. It would be better if I could harness Apache Spark or Hadoop system to solve the Big Data problem that I am facing.
- In addition, due to limiting CPU core and processing power, it would be better if I can host it in cloud so as to capitalize on the additional clusters to solve the Big data problem.
- In addition, if further data like rental prices of different region is given, I would be able to make a better decision as I can compare between the rental price and the traffic that a particular region enjoyed. This is so that we can access whether it is feasible to set up restaurant in a particular region based on several factors.