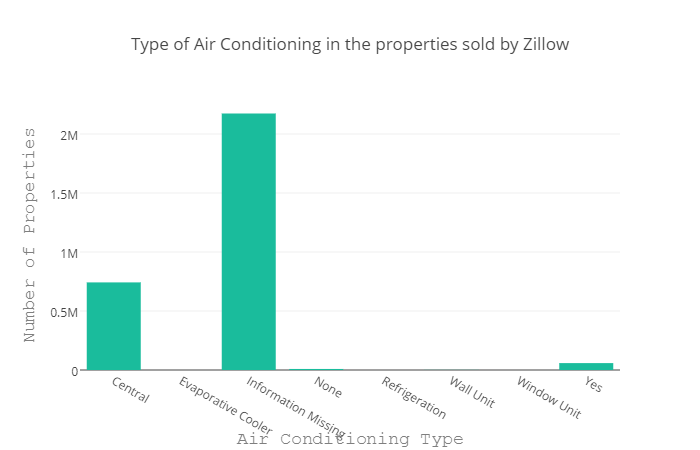
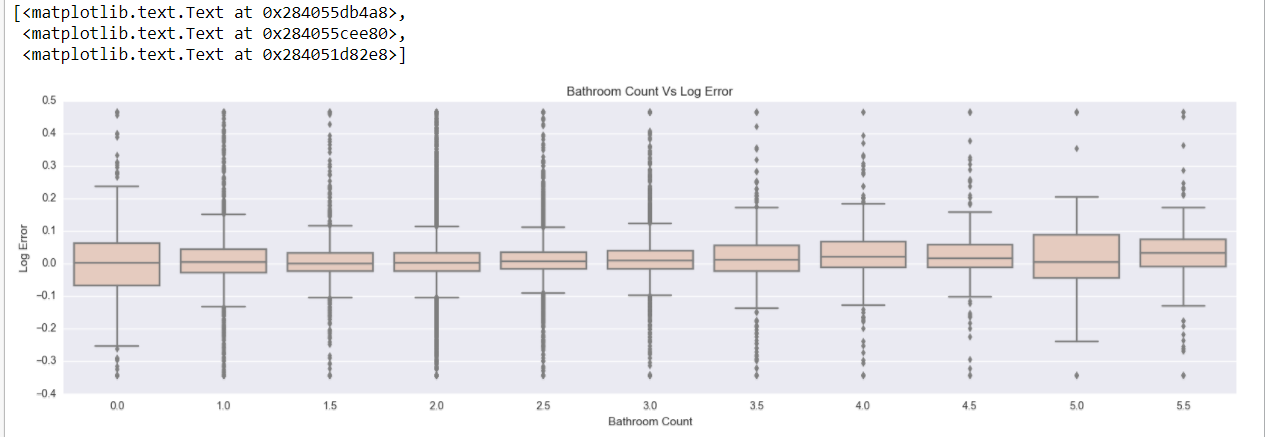
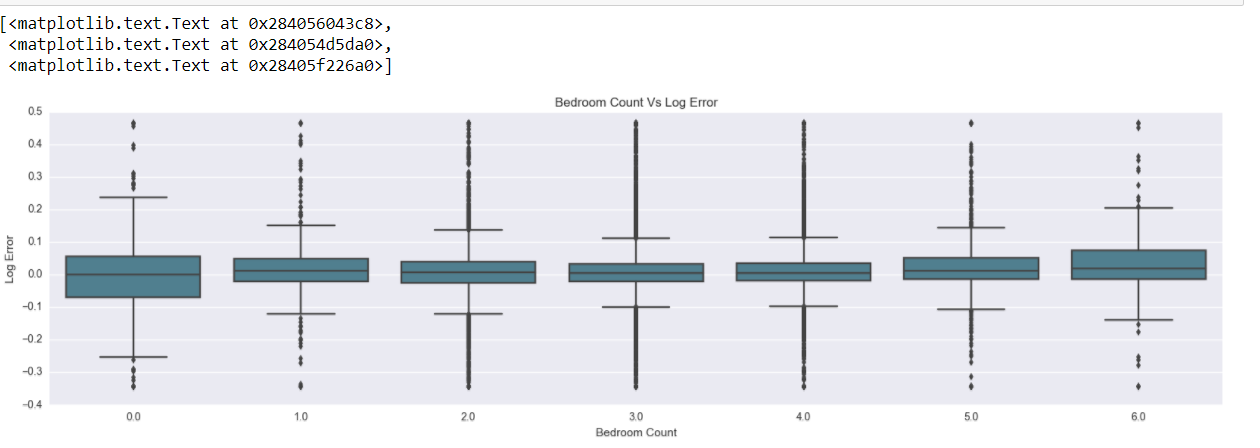
**Report Assignment 2**

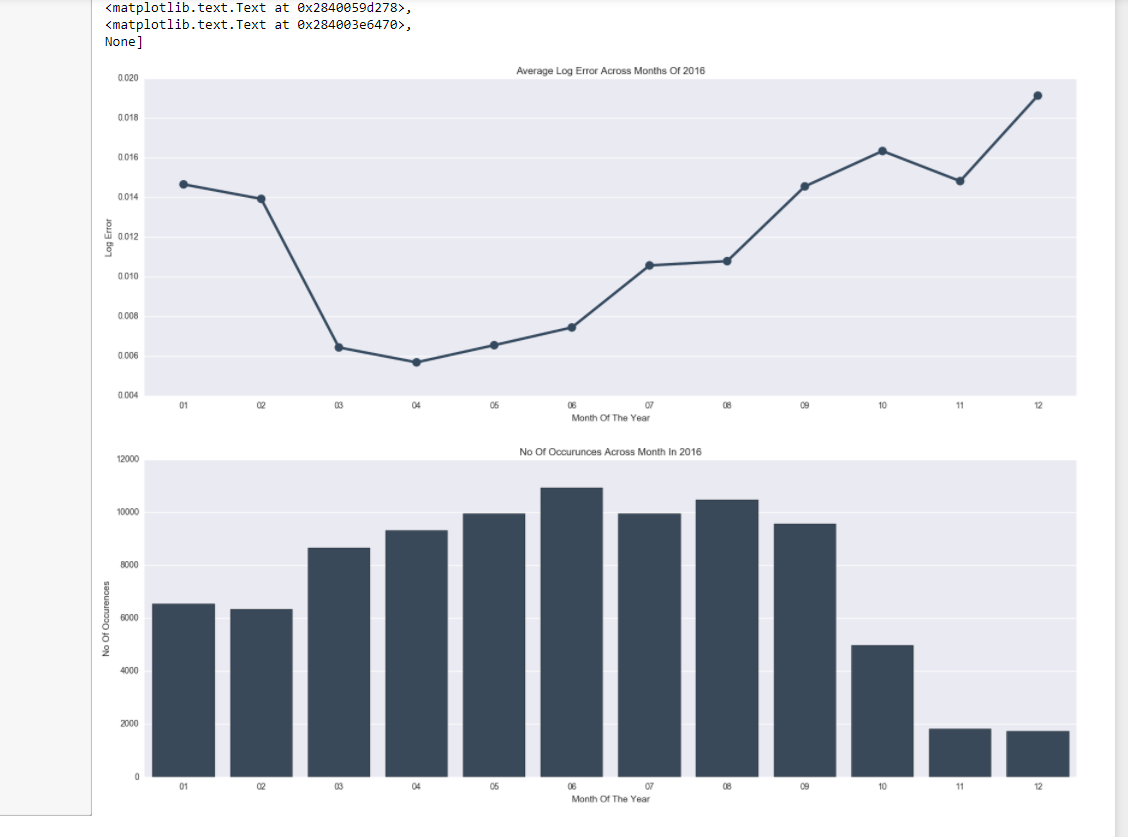
**Step 1:**

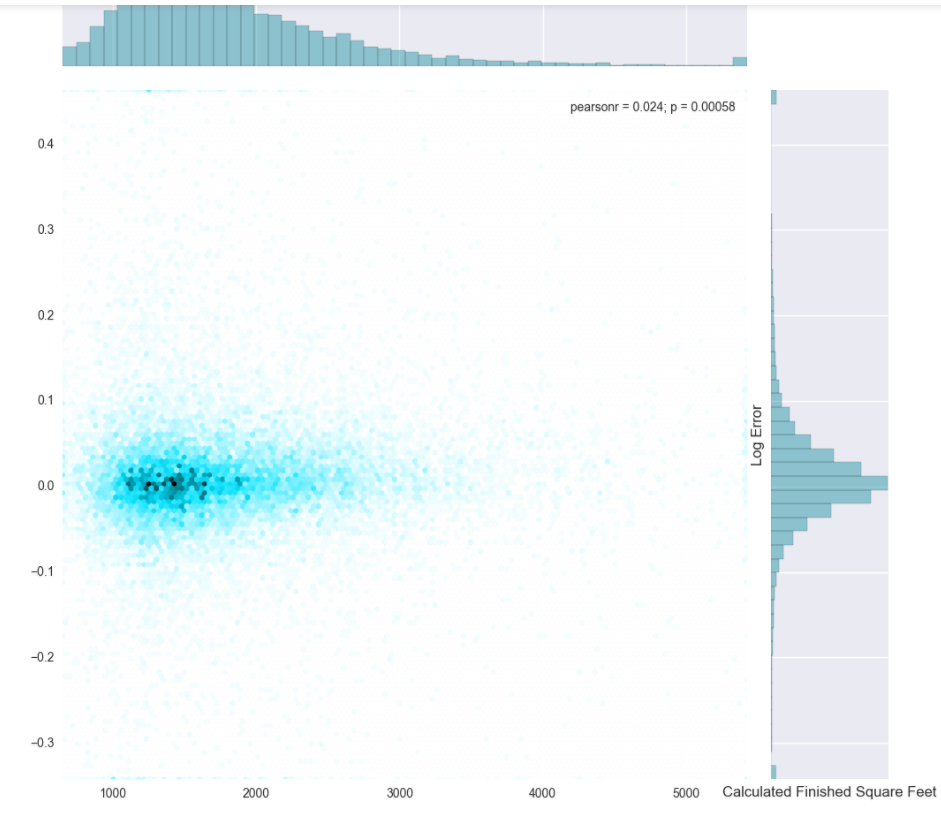
* As specified from the assignment document downloaded the data set from kaggle.
* The files contains properties file, Train dataset, sample\_submission, and data dictonary.
* After downloading the dataset the files are stored on local machine.
* Performed EDA on the dataset
* The results and conclusions are Witten in python notebook RAWDATAEDA
* The graphs of the Eda can be seen below:

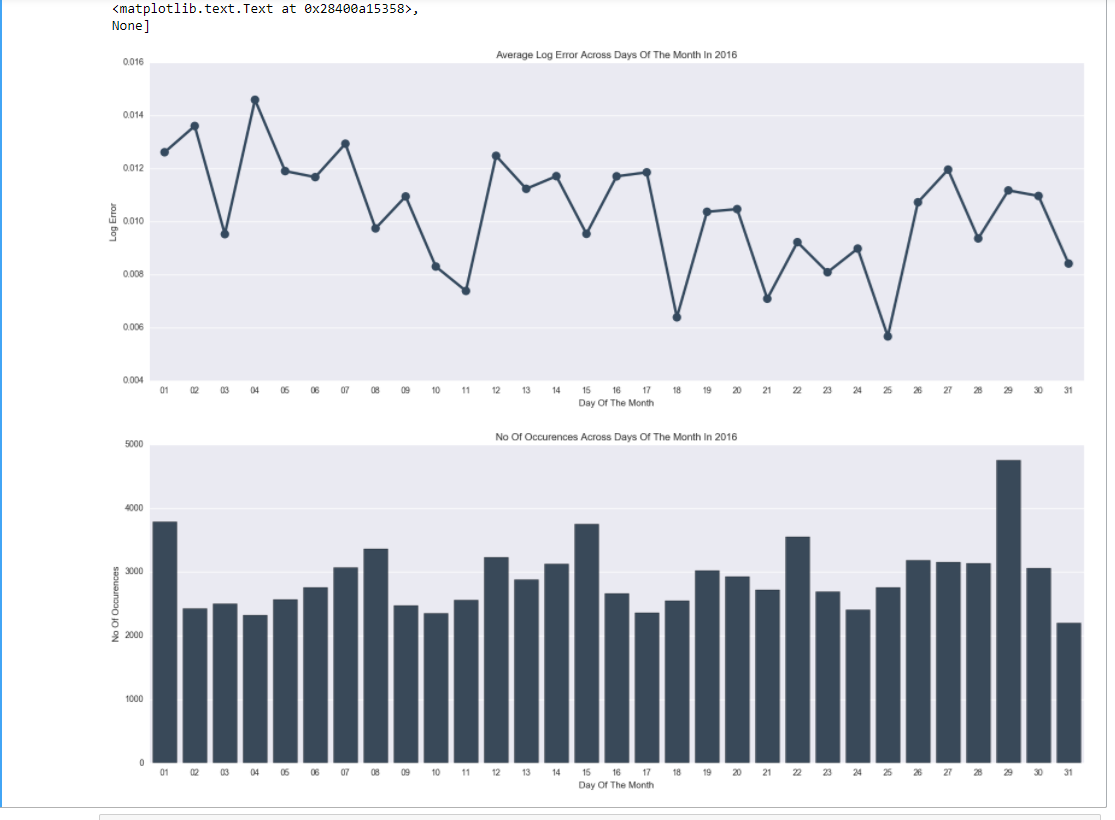


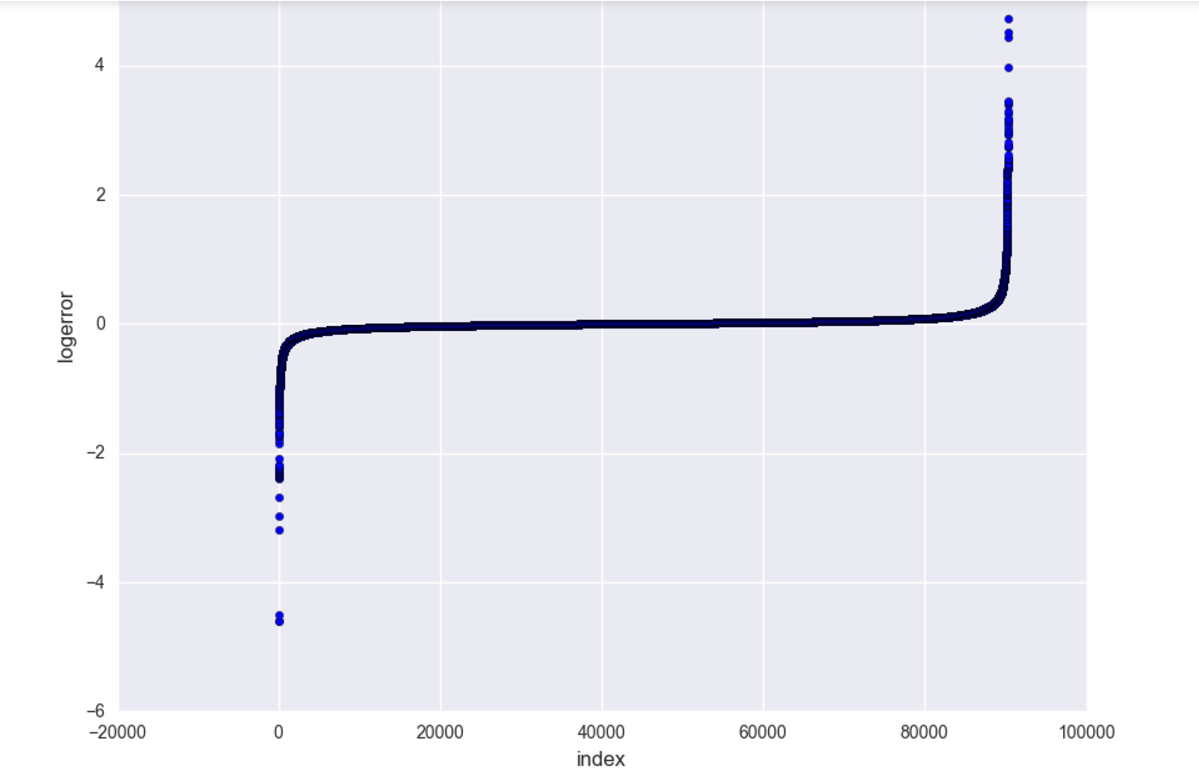


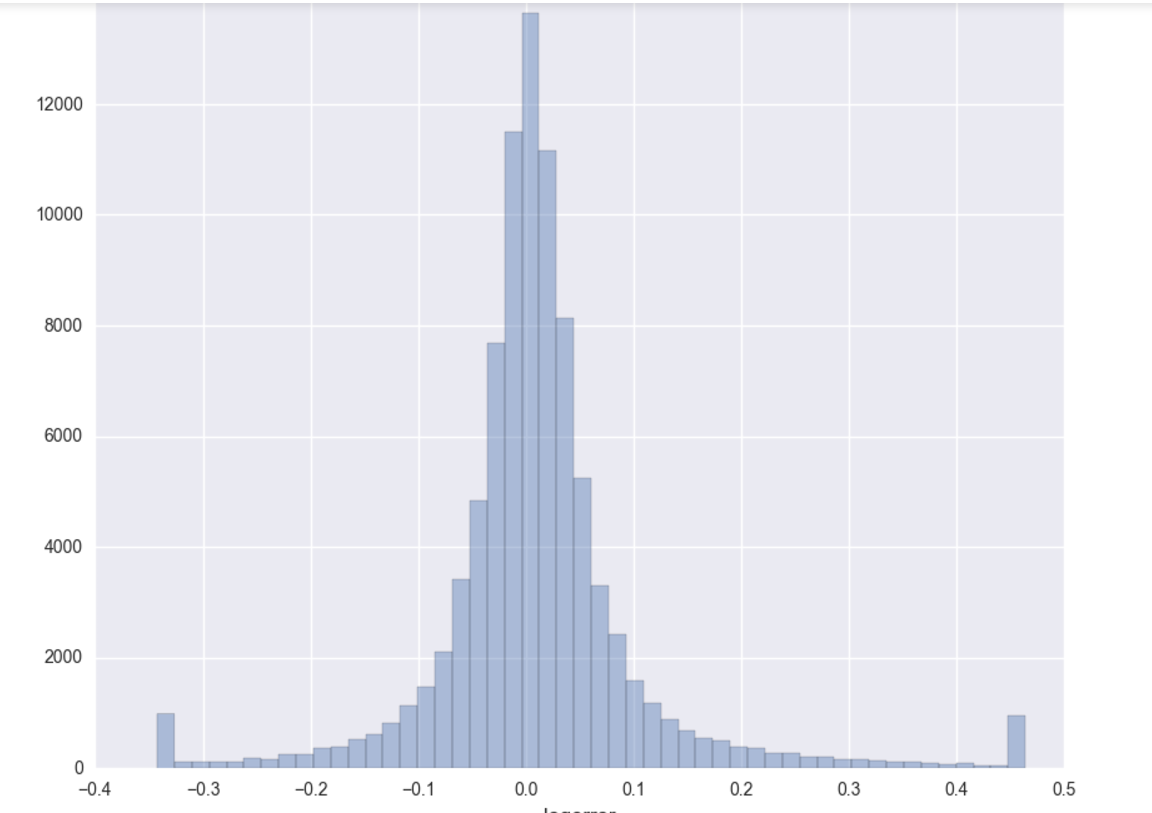


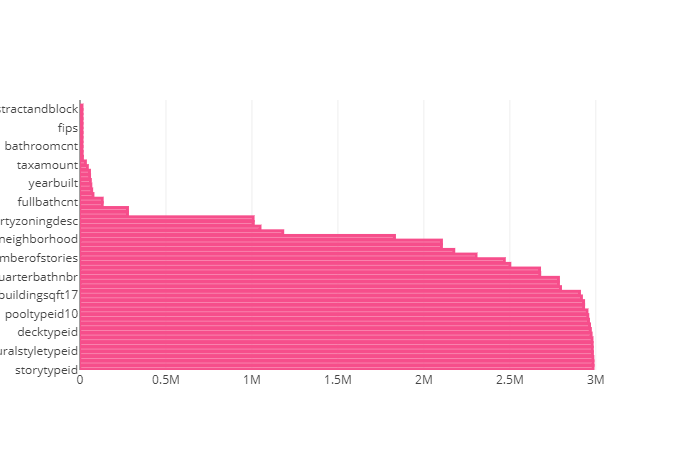


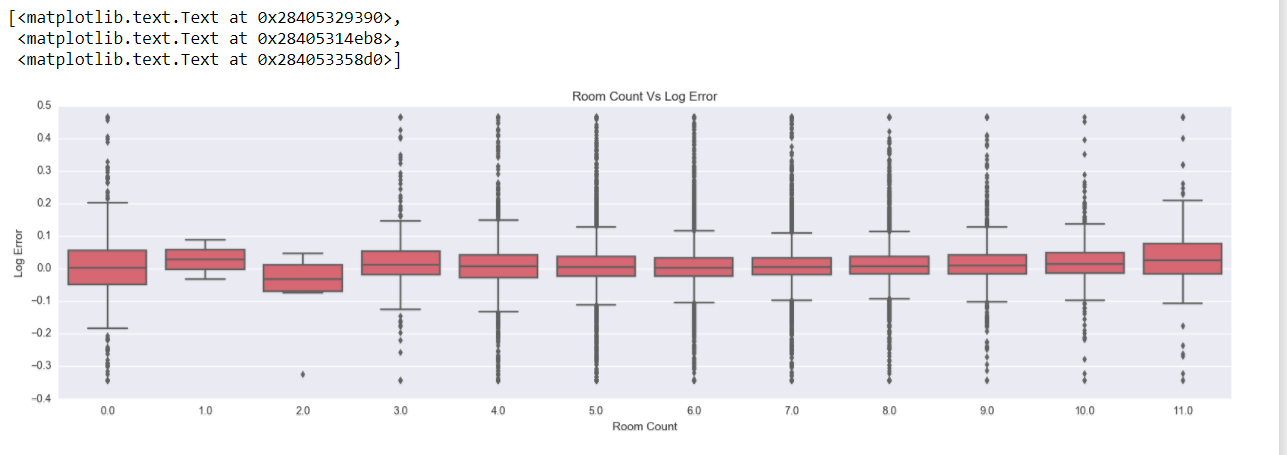


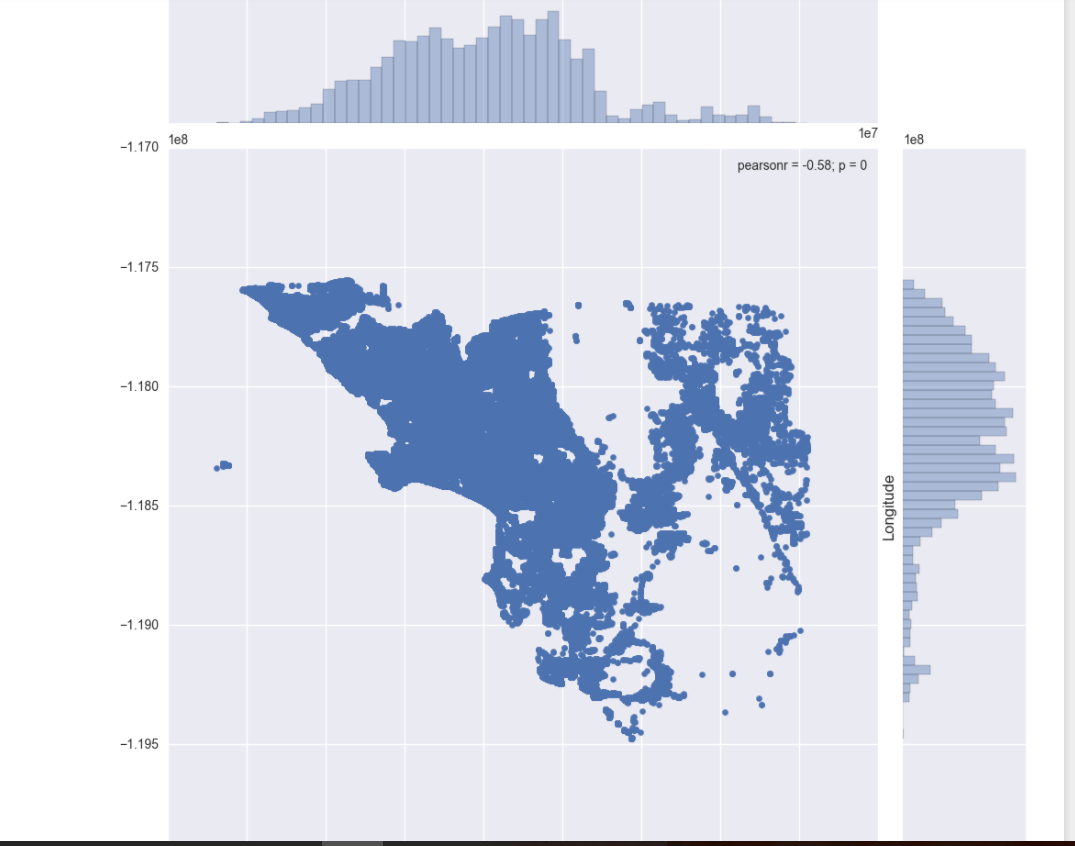


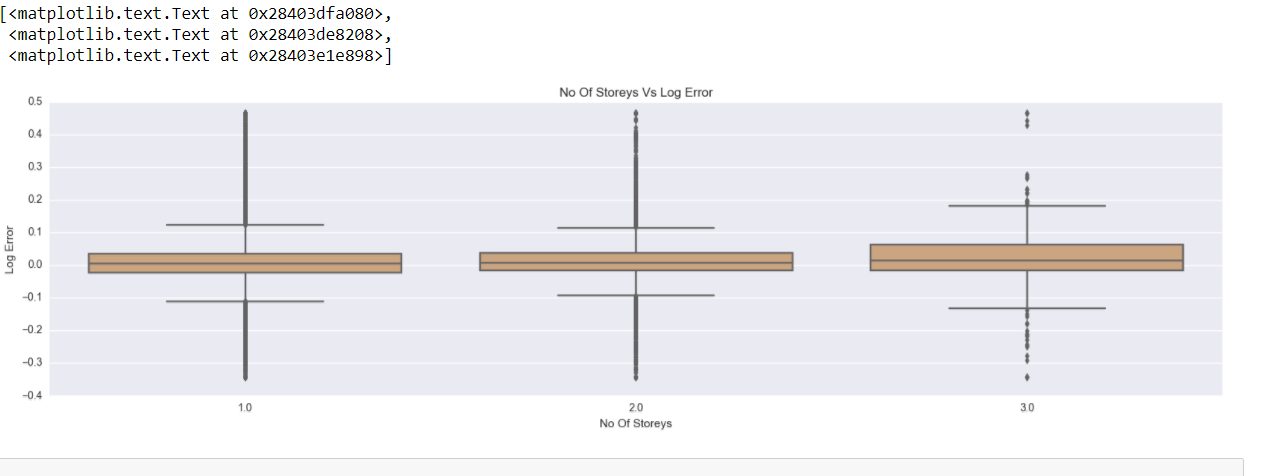


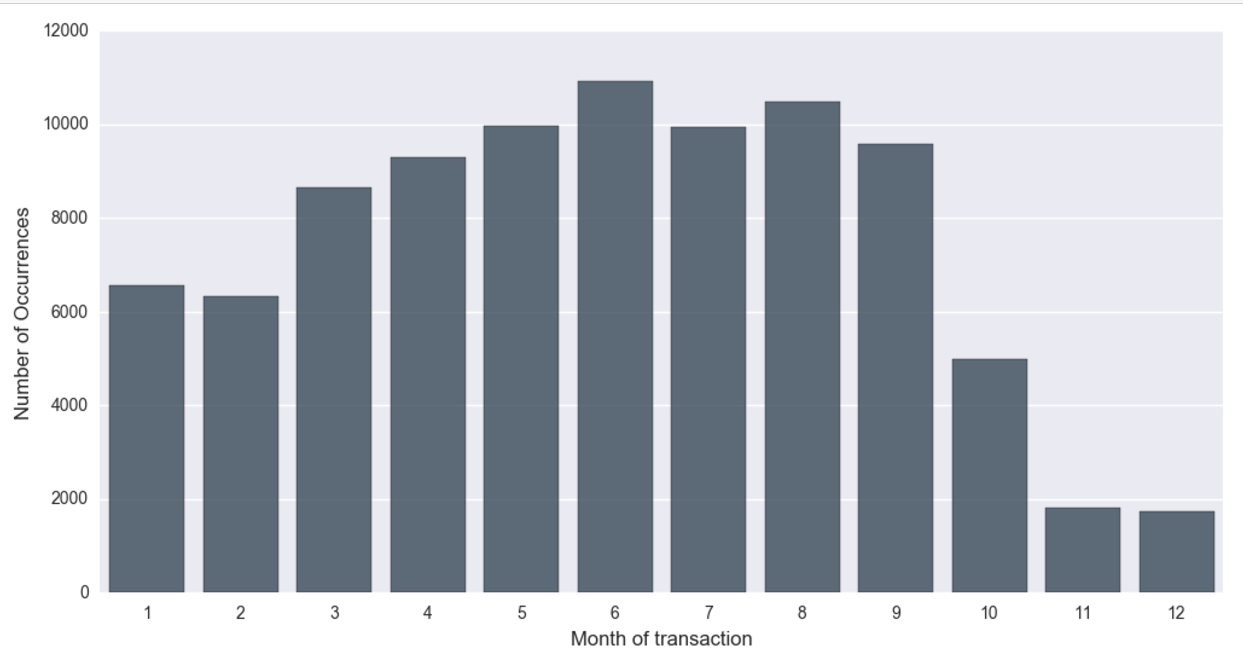












**Step 2:**

**Create a DBaas (Database as a service)**

* AS provided by the document created the account on Microsoft azure .
* Using that account created the DB on SQL server Online db.
* Using SQL cmd we created the table on online db. Named Zillow
* As in previous step we cleaned the file we have downloaded
* Cleaning details of the files are written in python notebook.
* Using the bulk loading operation from SQL CMD we pushed the clean file on SQL online database.
* So this way we created database on could.

**Step 3:**

* Create a Rest API to serve the data:
* We created a flask application which connects to azure database and retrieves the results online.
* In this application we connected to database using “pyodbc” drivers in python.
* By specifying the routs and query in code we retrieved the results.

**Step 4:**

* We developed a function in db which will give you nearest 10 places .
* As an input we need latitude and longitude and we paas those to our function and that function will retrieve the result for us.
* This flask application will fetch the result from cloud.
* This is the definition of the function we wrote:

function [dbo].[distance](@latitude float, @longitude float)

returns TABLE

as

return

(

select z.parcelid, z.latitude, z.longitude,3956\*2\*ASIN( SQRT(

POWER(SIN((@latitude-abs(z.latitude)) \* pi()/180/2),

2) + COS(@latitude \* pi()/180) \* COS(abs(z.latitude) \*

pi()/180) \* POWER(SIN((@longitude - z.longitude) \*

pi()/180/2), 2) )) as distance

from ZillowData as z

--group by dist.latitude, dist.longitude, dist.distance

--having dist.distance < 10

)

And when we pass the values we will get result in this format.

