**MongoDB installation:**

[https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/#get-mongodb-community-edition (Links to an external site.)Links to an external site.](https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/#get-mongodb-community-edition)

Note: If you have version 3.2 installed, then upgrade to 3.4 first and then to 3.6. (I was working with MOngoDB 3.2 and wanted to upgrade to the latest version 3.6. But, an error was shown while running the mongod.exe saying all the data must be upgraded to version 3.4 first. So, installed version 3.4 first, cleaned the /data/db folder and then installed 3.6 again). The link to the installation guide is below:

**Loading and accessing our dataset into MongoDB:**

**Data source**: <https://www.kaggle.com/nsharan/h-1b-visa>

Command to load csv file into MongoDB:

"C:\Program Files\MongoDB\Server\3.6\bin\mongoimport" --db g5 --collection h1b --type csv --headerline --file "C:\Users\jaide\Downloads\4th Sem\INFO-I590 - SQL NoSQL\Project\h1b\_kaggle.csv"

database name (--db): g5  
collection (--collection): h1b  
file type (--type): csv  
file path (--file): path to the downloaded file.

To access the data, first switch to your desired database:

db.adminCommand( { listDatabases: 1 } ); // lists all databases

OR show dbs

use g5; //switches to database g5

db.getCollectionNames(); // lists all collections (tables)

OR show collections

db.h1b.findOne(); //fetches first row or doc from collection h1b

**Python with MongoDB:**

We are using the python library**PyMongo:**

Command to install PyMongo: pip install pymongo

Guide: [https://docs.mongodb.com/getting-started/python/client/ (Links to an external site.)Links to an external site.](https://docs.mongodb.com/getting-started/python/client/)

Acessing data using **pyMongo** and **pandas** from **MongoDB**:

from **pymongo** import **MongoClient**

client = MongoClient('mongodb://localhost:27017/') # connect to MongoDB localhost

db = client.g5 # g5 is our database  
coll = db.h1b #h1b is our collection in g5

**Cleaning** the data:

import **pandas** as pd

h1b\_data = pd.DataFrame(list(coll.find({}))) # load the entire data from collection h1b

h1b\_data.isnull().sum() # check for null values. We founf none in our dataset

h1b\_data['CASE\_STATUS'].unique() # displays all unique values of Case\_Status

h1b\_data[h1b\_data['CASE\_STATUS']=="NA"] # displays all rows with Case\_Status = "NA"

We found many rows with "NA" values which won't help our analysis, hence need to be deleted (cleaned).

h1b\_data = coll.delete\_many({"CASE\_STATUS": "NA"}) # Command to delete rows with NA values from mongoDB collection.

h1b\_data[h1b\_data['CASE\_STATUS']=="NA"] # should show no rows with NA values.

h1b\_data.PREVAILING\_WAGE.describe()

count 3.002373e+06

mean 1.469984e+05

std 5.287609e+06

min 0.000000e+00

25% 5.437100e+04

50% 6.502100e+04

75% 8.143200e+04

max 6.997607e+09

As seen PREVAILING\_WAGE mean is in terms of 100K, but the maximu value is very high, hence we can conclude there are lot of outliers. We will remove the outlier which are 3 standard deviation away from mean.

h1b\_data=h1b\_data[np.abs(h1b\_data.PREVAILING\_WAGE-h1b\_data.PREVAILING\_WAGE.mean())<=(3\*h1b\_data.PREVAILING\_WAGE.std())]

count 3.000677e+06

mean 7.082575e+04

std 5.615638e+04

min 0.000000e+00

25% 5.435040e+04

50% 6.500000e+04

75% 8.139040e+04

max 1.509460e+07

**Visualization:** We have done the following visualizations.

* Mean and median wage across all the states.
* Wage distribution for Different Years
* Mean and Median Wage for Different Years
* Number of petitions per year and the percentage change.
* State Wise number of petitions.
* Mean and median wage across top 5 states.

The WORKSITE in our dataset is in the format “city, state”. To visualize based on states in the map format (plotly.graph\_objs), we will need the state name abbreviations like IN for Indiana. First, we split the WORKSITE and extracted the state name. We created a dictionary to map state names to their abbreviation and then used it to visualize.

h1b\_data['State']=h1b\_data.WORKSITE.str.split(',', expand = True)[1]

This line of code creates a new column ‘State’ in our dataframe h1b\_data which contains the state name.

US\_STATE\_ABBREV = {

'ALABAMA': 'AL',

'ALASKA': 'AK',

'ARIZONA': 'AZ',

}

This is the dictionary to map state names to their abbreviation.

1. **Mean and median wage across all the states.**

The below code will create a US map with mean wage distribution across different states.

import plotly.plotly as py

import plotly.graph\_objs as go

from plotly.offline import download\_plotlyjs, init\_notebook\_mode, plot, iplot

init\_notebook\_mode(connected=True)

data = dict(type='choropleth',

colorscale = 'YIOrRd',

locations = mean\_wage['State'],

z = mean\_wage['PREVAILING\_WAGE'],

locationmode = 'USA-states',

text = mean\_wage['State'],

marker = dict(line = dict(color = 'rgb(255,255,255)',width = 2)),

colorbar = {'title':"Mean Wage Distribution"}

)

layout = dict(title = 'Mean Wage Distribution',

geo = dict(scope='usa',

showlakes = False,

lakecolor = 'rgb(85,173,240)')

)

choromap = go.Figure(data = [data],layout = layout)

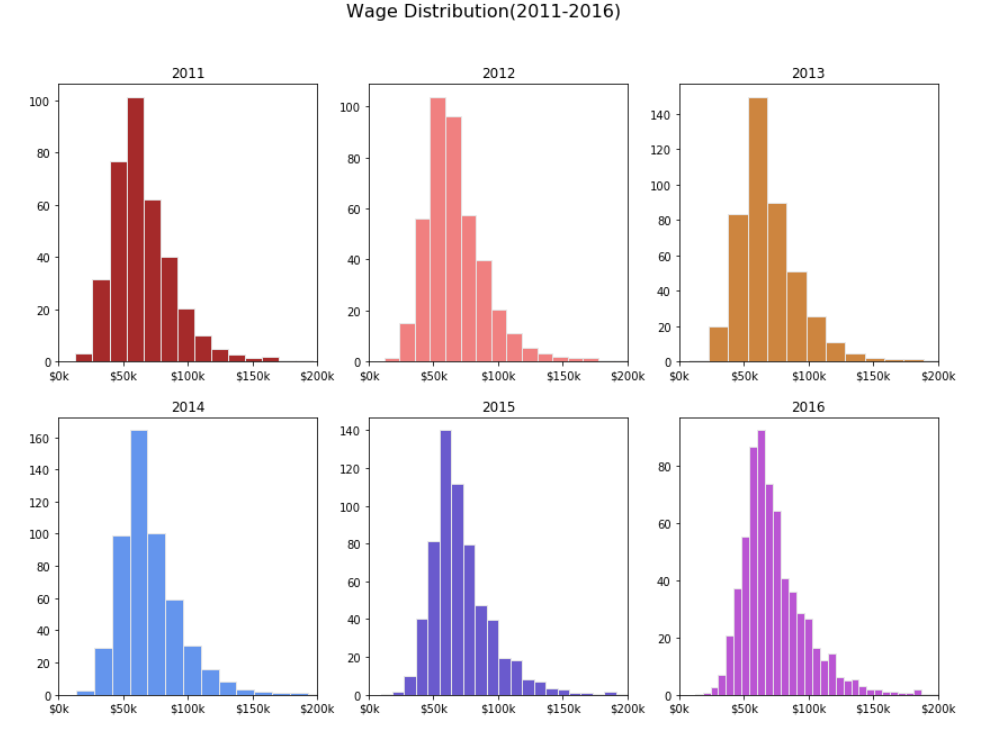
iplot(choromap)

Link to the visualization: <https://plot.ly/create/?fid=jaidevyd:23>

Similarly, we have visualized median wage distribution across different states. <https://plot.ly/create/?fid=jaidevyd:25>

1. **Wage distribution for Different Years**

Take the prevailing wage of each year separately and put it in bins of 1000 and by using plotly’s subplot function, plot the graph for each year. The wage 60k was the prominent wage from 2011 untill 2015. It raised to 70K in 2016.



1. **Mean and Median Wage for Different Years**

Group the prevailing wage by year and take the mean and median of it in different data frames. The data frame will have year and its mean wage (and median wage in another data frame). We plotted the graph using iplot.

Mean wage distribution: <https://plot.ly/create/?fid=jaidevyd:27>

Median wage distribution: <https://plot.ly/create/?fid=jaidevyd:29>

1. **Number of petitions per year and the percentage change.**

Group the data by year and and plot the graph using iplot.

<https://plot.ly/create/?fid=jaidevyd:31>

1. **State Wise Yearly number of petitions.**

Take all the data for a particular year and select the top 10 states with maximum petitions. Plot the graph using iplot.

2011 & 2012: <https://plot.ly/create/?fid=jaidevyd:37>

2013 & 2014: <https://plot.ly/create/?fid=jaidevyd:39>

2015 & 2016: <https://plot.ly/create/?fid=jaidevyd:41>

1. **Mean and median wage across top 5 states.**

From the above graph, we now know the top 10 states with maximum petitions. For top 5 states, we took the mean and median Prevailing wage for each yea and plotted the graph using iplot.

Mean wage across top 5 states: <https://plot.ly/create/?fid=jaidevyd:35>

Median wage across top 5 states: <https://plot.ly/create/?fid=jaidevyd:33>

1. **No. of petitions certified across years (Certified petitions).**

Take all the data for a particular year with status as ‘certified’ and select the top 10 states with maximum petitions. Plot the graph using iplot.

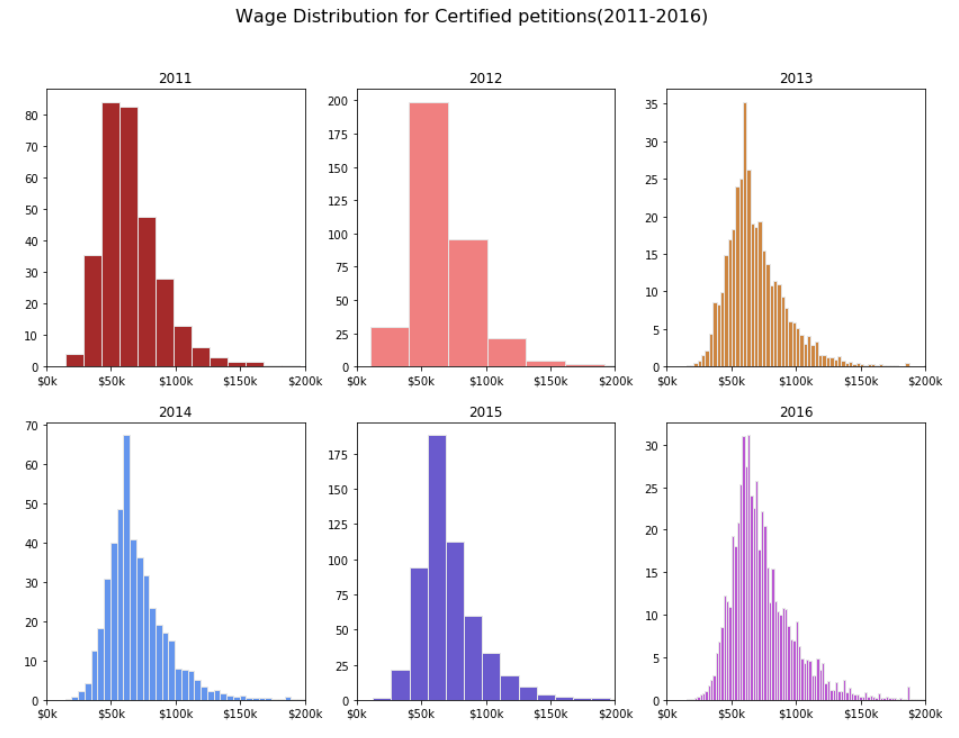
Certified petitions 2011 & 2012: <https://plot.ly/create/?fid=jaidevyd:43>

Certified petitions 2013 & 2014: <https://plot.ly/create/?fid=jaidevyd:45>

Certified petitions 2015 & 2016: <https://plot.ly/create/?fid=jaidevyd:47>

1. **Wage Distribution for Certified petitions across years.**

Take the prevailing wage of each year for the petitions that have status ‘certified’ separately and put it in bins of 300 and by using plotly’s subplot function, plot the graph for each year. The wage between 50k and 60k was the prominent wage from 2011 untill 2015 to get certified. It is close to 70K in 2016.



1. **Top 10 states with highest percentage of Certified petitions across years.**Take the count of total petitions of each state for every year (a). Take the count of certified petitions for every year(b).  Plot the graph by deviding these two entities (b/a) and multiply by 100.   
     
   State\_2011\_overall=h1b\_data[h1b\_data['YEAR']==2011]['State'].value\_counts()  
   State\_2011\_certified\_overall=h1b\_data[np.logical\_and(h1b\_data['YEAR']==2011,h1b\_data['CASE\_STATUS']=='CERTIFIED')]['State'].value\_counts()  
   petitions\_percentage\_2011\_overall=pd.Series(State\_2011\_certified\_overall[State\_2011\_certified\_overall.index.sort\_values()]/State\_2011\_overall[State\_2011\_overall.index.sort\_values()])  
   [Overall Certified percentage 2011 & 2012 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:5)  
   [Overall Certified percentage 2011 & 2012 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:7)  
   [Overall Certified percentage 2011 & 2012 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:9)
2. **Percentage of Certified petitions across years for top 10 states with most petitions.**We now know the top 10 states with most no. of petitions. Take the count of total petitions of these states for every year (a). Take the count of certified petitions for every year(b).  Plot the graph by deviding these two entities (b/a) and multiply by 100.   
     
   state\_data\_2011=h1b\_data[np.logical\_and(h1b\_data.State.isin(State\_2011.index),h1b\_data['YEAR']==2011)]  
   state\_petitions\_2011=state\_data\_2011['State'].value\_counts()  
   state\_2011\_certified=state\_data\_2011[state\_data\_2011['CASE\_STATUS']=='CERTIFIED']['State'].value\_counts()  
   petitions\_percentage\_2011=pd.Series(state\_2011\_certified[state\_2011\_certified.index.sort\_values()]/state\_petitions\_2011[state\_petitions\_2011.index.sort\_values()])[Percentage Certified(top 10 states with max petitions) 2011 & 2012 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jaidevyd:49)[Percentage Certified(top 10 states with max petitions) 2013 & 2014 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:1)  
   [Percentage Certified(top 10 states with max petitions) 2015 & 2016 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:3)

1. **Top 10 employers with most no. of petitions across years.**  
     
   Take all the data for a particular year and select the top 10 employers with maximum petitions. Plot the graph using iplot.  
     
   [2011 & 2012 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:11)  
   [2013 & 2014 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:13)  
   [2015 & 2016 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:15)
2. **Percentage of Certified petitions across years for top 10 employers with most petitions.**We now know the top 10 employers with most no. of petitions. Take the count of total petitions of these employers for every year (a). Take the count of certified petitions for every year(b).  Plot the graph by deviding these two entities (b/a) and multiply by 100.   
     
   Employer\_petitions\_2011=h1b\_data[h1b\_data['YEAR']==2011]['EMPLOYER\_NAME'].value\_counts()[:10]  
   employer\_data\_2011=h1b\_data[np.logical\_and(h1b\_data.EMPLOYER\_NAME.isin(Employer\_petitions\_2011.index),h1b\_data['YEAR']==2011)]  
   employer\_petitions\_2011=employer\_data\_2011['EMPLOYER\_NAME'].value\_counts()  
   employer\_2011\_certified=employer\_data\_2011[employer\_data\_2011['CASE\_STATUS']=='CERTIFIED']['EMPLOYER\_NAME'].value\_counts()  
   employer\_percentage\_2011=pd.Series(employer\_2011\_certified[employer\_2011\_certified.index.sort\_values()]/employer\_petitions\_2011[employer\_petitions\_2011.index.sort\_values()])  
     
   [Percentage Certified(top 10 employers with max petitions) 2011 & 2012 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:17)[Percentage Certified(top 10 employers with max petitions) 2013 & 2014 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:19)  
   [Percentage Certified(top 10 employers with max petitions) 2015 & 2016 (Links to an external site.)Links to an external site.](https://plot.ly/create/?fid=jyekkard07:21)