In [1]:
```
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Data Exploration and Cleaning

1. How many columns are in the dataset? How many tracts' records are there?

In [2]:
```
# Load the Boston housing dataset
data = pd.read_csv('Project_dataset.csv')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2194 entries, 0 to 2193
Data columns (total 17 columns):
GEOID10           2194 non-null int64
County_Code       2194 non-null int64
Latitude          2194 non-null float64
Longitude         2194 non-null float64
Total_Pop         2194 non-null int64
Mean_Comm_Time    2194 non-null float64
Frac_College      2194 non-null float64
Med_HHincome      2194 non-null int64
Pop_Density       2194 non-null float64
Frac_Living       2194 non-null float64
Jobs_Total        2194 non-null int64
Frac_Nonwhite     2194 non-null float64
Underground       2194 non-null int64
Fire_Station      2194 non-null int64
Hazardous         2194 non-null int64
Gas_Station       2194 non-null int64
Economic_Mob      2157 non-null float64
dtypes: float64(8), int64(9)
memory usage: 291.5 KB
```

Is there any missing data? Which Column has the missing values?

In [3]:
```
data.isna().sum()
```

Out[3]:
```
GEOID10            0
County_Code        0
Latitude           0
Longitude          0
Total_Pop          0
Mean_Comm_Time     0
Frac_College       0
Med_HHincome       0
Pop_Density        0
Frac_Living        0
Jobs_Total         0
Frac_Nonwhite      0
Underground        0
Fire_Station       0
Hazardous          0
Gas_Station        0
Economic_Mob      37
dtype: int64
```

If there is missing data in a column, replace the missing values with the median of that column

In [4]:
```
median = data["Economic_Mob"].median()
data["Economic_Mob"].fillna(median, inplace=True)
```

In [5]:
```
data.isna().sum()
```

Out[5]:
```
GEOID10            0
County_Code        0
Latitude           0
Longitude          0
Total_Pop          0
Mean_Comm_Time     0
Frac_College       0
Med_HHincome       0
Pop_Density        0
Frac_Living        0
Jobs_Total         0
Frac_Nonwhite      0
Underground        0
Fire_Station       0
Hazardous          0
Gas_Station        0
Economic_Mob       0
dtype: int64
```

Querying a Dataset

What is the maximum and minimum Mean Commute Time?

In [6]:
```
data.describe()
```

Out[6]:

	GEOID10	County_Code	Latitude	Longitude	Total_Pop	Mean_Comm_Time
count	2.194000e+03	2194.000000	2194.000000	2194.000000	2194.000000	2194.000000
mean	3.710278e+10	102.517776	-79.692828	35.530926	4346.163628	24.979792
std	5.445647e+07	54.442906	1.611163	0.537127	1877.156472	5.332447
min	3.700102e+10	1.000000	-84.238718	33.879335	0.000000	0.000000
25%	3.706300e+10	63.000000	-80.825087	35.203012	3070.500000	21.683192
50%	3.710104e+10	101.000000	-79.801645	35.578522	4183.500000	24.720707
75%	3.714700e+10	147.000000	-78.617324	35.948509	5453.500000	28.136009
max	3.719996e+10	199.000000	-75.538012	36.542920	14249.000000	44.162933

What is the maximum value for fraction of residents living in poverty in a census tract? Which census tract has the maximum fraction of residents living in poverty?

In [7]:
```
data["Frac_Living"].max()
```

Out[7]: 0.962756

In [8]:
```
data["Frac_Living"].idxmax()
```

Out[8]: 267

How many census tracts have more than 10 hazardous waste sites?

In [9]:
```
data_h = data[data["Hazardous"]>10]
data_h["Hazardous"].count()
```

Out[9]: 11

How many census tracts do not have any fire stations?

In [10]:
```
data_fs = data[data["Fire_Station"]== 0]
data_fs["Fire_Station"].count()
```

Out[10]: 909

How many census tracts have more than the average number of underground tanks?

In [11]:
```
ug_mean = data["Underground"].mean()
ug_mean
```

Out[11]: 12.740656335460347

In [12]:
```
data_ug = data[data["Underground"]> ug_mean]
data_ug["Underground"].count()
```

Out[12]: 790

What is the mean population density of the census tracts with 10 or more Gas Stations vs. without any Gas Stations?

In [13]:
```
data_g = data[data["Gas_Station"]>10]
data_g["Pop_Density"].mean()
```

Out[13]: 194.0187777857143

In [14]:
```
data_ng = data[data["Gas_Station"]==0]
data_ng["Pop_Density"].mean()
```

Out[14]: 599.080856563077

What is the average of median house hold income for the tracts where 80% or more people are non-white vs. 20% or less people are non-white

In [15]:
```
data_nw = data[data["Frac_Nonwhite"] >= 0.80]
data_nw["Med_HHincome"].mean()
```

Out[15]: 27653.789156626506

In [16]:
```
data_nw = data[data["Frac_Nonwhite"] <= 0.20]
data_nw["Med_HHincome"].mean()
```

Out[16]: 58994.48574969021

Which census tract has the highest and lowest economic mobility? What are the fractions of college educated population there?

In [17]:
```
id = data["Economic_Mob"].idxmax()
print("The tract with highest economic mobility is ", id)
print("The fraction of college educated is ", data.iloc[id].Frac_Colle
```

```
The tract with highest economic mobility is  2187
The fraction of college educated is  0.8461540000000001
```

In [18]:
```
id = data["Economic_Mob"].idxmin()
print("The tract with lowest economic mobility is ", id)
print("The fraction of college educated is ", data.iloc[id].Frac_Colle
```

```
The tract with lowest economic mobility is  260
The fraction of college educated is  0.098814
```

What are the averages of median household income for each county? Sort them in acsending order

In [19]:
```
data.groupby("County_Code").agg({"Med_HHincome": np.mean}).sort_values
```

Out[19]:

County_Code	Med_HHincome
95	12580.333333
173	26881.400000
165	31015.714286
15	31282.500000
17	31289.666667
...	...
25	60398.675676
119	63527.763948
135	65841.178571
179	75391.365854
183	76731.406417

100 rows × 1 columns

Group the data by number of Hazardous sites. What are the average population densities?

In [20]:
```
data.groupby("Hazardous").agg({"Pop_Density": np.mean})
```

Out[20]:

Hazardous	Pop_Density
0	379.344966
1	435.409185
2	479.881593
3	507.012521
4	416.918211
5	476.443236
6	354.246688
7	275.390129
8	379.037356
9	370.629912
10	213.402645
11	38.192469
12	231.929924
13	510.643563
15	361.406100
19	90.967071
26	104.579510

Finding Correlations

What is the correlation between median household income and fraction of residents with a college degree?

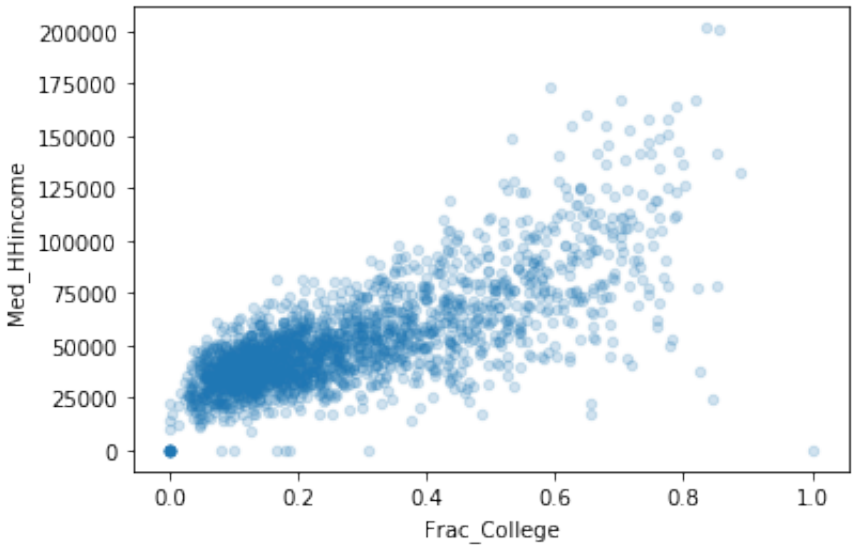
In [21]:
```
corr_matrix = data.corr()
corr_matrix["Med_HHincome"].sort_values()
```

Out[21]:
```
Frac_Living      -0.558110
Frac_Nonwhite    -0.380819
Gas_Station      -0.259324
Underground      -0.142778
Hazardous        -0.133935
Fire_Station     -0.098382
Pop_Density      -0.076429
Latitude         -0.017423
Longitude         0.002775
Total_Pop         0.124628
Jobs_Total        0.134132
GEOID10           0.153532
County_Code       0.155524
Mean_Comm_Time    0.251729
Economic_Mob      0.557117
Frac_College      0.749061
Med_HHincome      1.000000
Name: Med_HHincome, dtype: float64
```

Plot the correlation between median household income and fraction of residents with a college degree.

In [22]:
```
from pandas.plotting import scatter_matrix
data.plot(kind= "scatter", x="Frac_College", y="Med_HHincome", alpha =
```

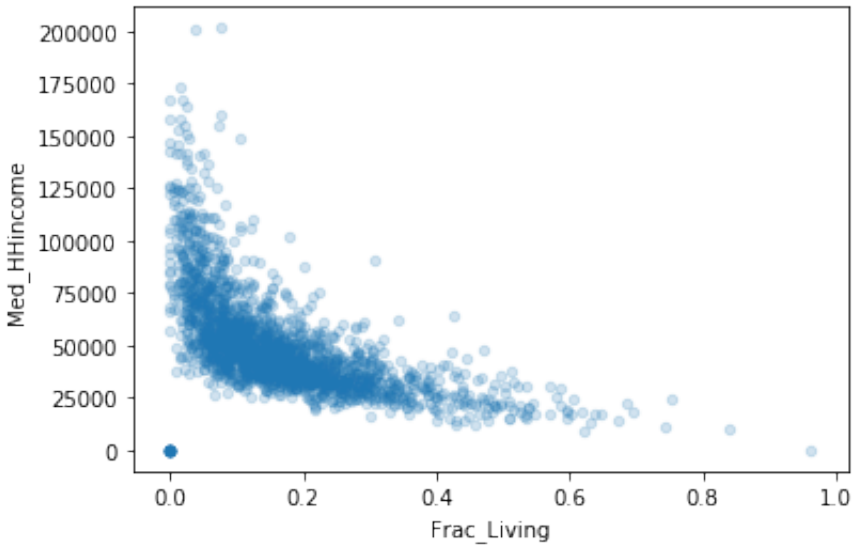
Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb772bd8a10>
```



Plot the correlation between median household income and fraction of residents living with poverty.

In [23]:
```
data.plot(kind= "scatter", x="Frac_Living", y="Med_HHincome", alpha =
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb773ccce50>
```



What is the correlation between Economic Mobility and fraction of residents living in poverty?

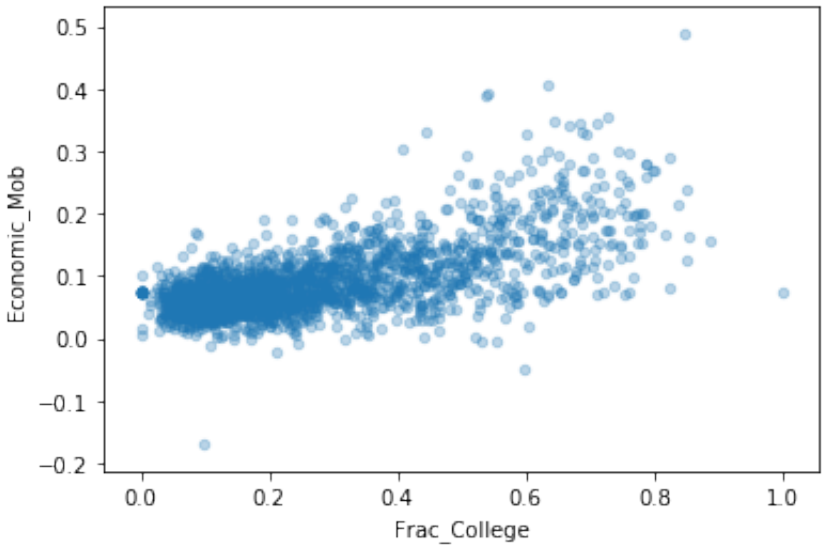
In [24]:
```
corr_matrix = data.corr()
corr_matrix["Economic_Mob"].sort_values()
```

Out[24]:
```
Frac_Living      -0.311195
Gas_Station      -0.209459
Fire_Station     -0.205983
Frac_Nonwhite    -0.141721
Underground      -0.058099
Hazardous        -0.031909
Longitude        -0.023764
Mean_Comm_Time    0.007836
Total_Pop         0.010618
Latitude          0.079433
GEOID10           0.170367
County_Code       0.171625
Pop_Density       0.208472
Jobs_Total        0.257989
Med_HHincome      0.557117
Frac_College      0.653986
Economic_Mob      1.000000
Name: Economic_Mob, dtype: float64
```

Plot the correlation between Economic Mobility and fraction of residents with a college degree.

In [25]:
```
data.plot(kind= "scatter", x="Frac_College", y="Economic_Mob", alpha =
```

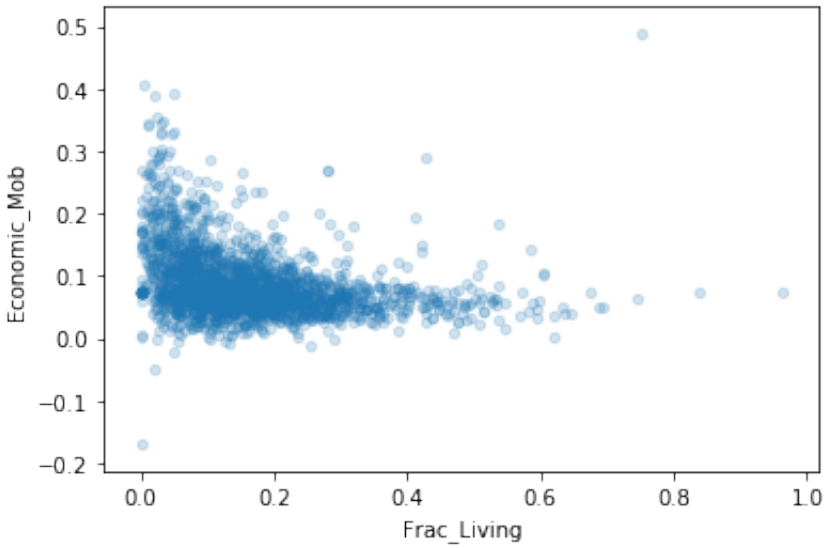
Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb7704f7c10>
```



Plot the correlation between Economic Mobility and fraction of residents living in poverty.

In [26]:
```
data.plot(kind= "scatter", x="Frac_Living", y="Economic_Mob", alpha =
```

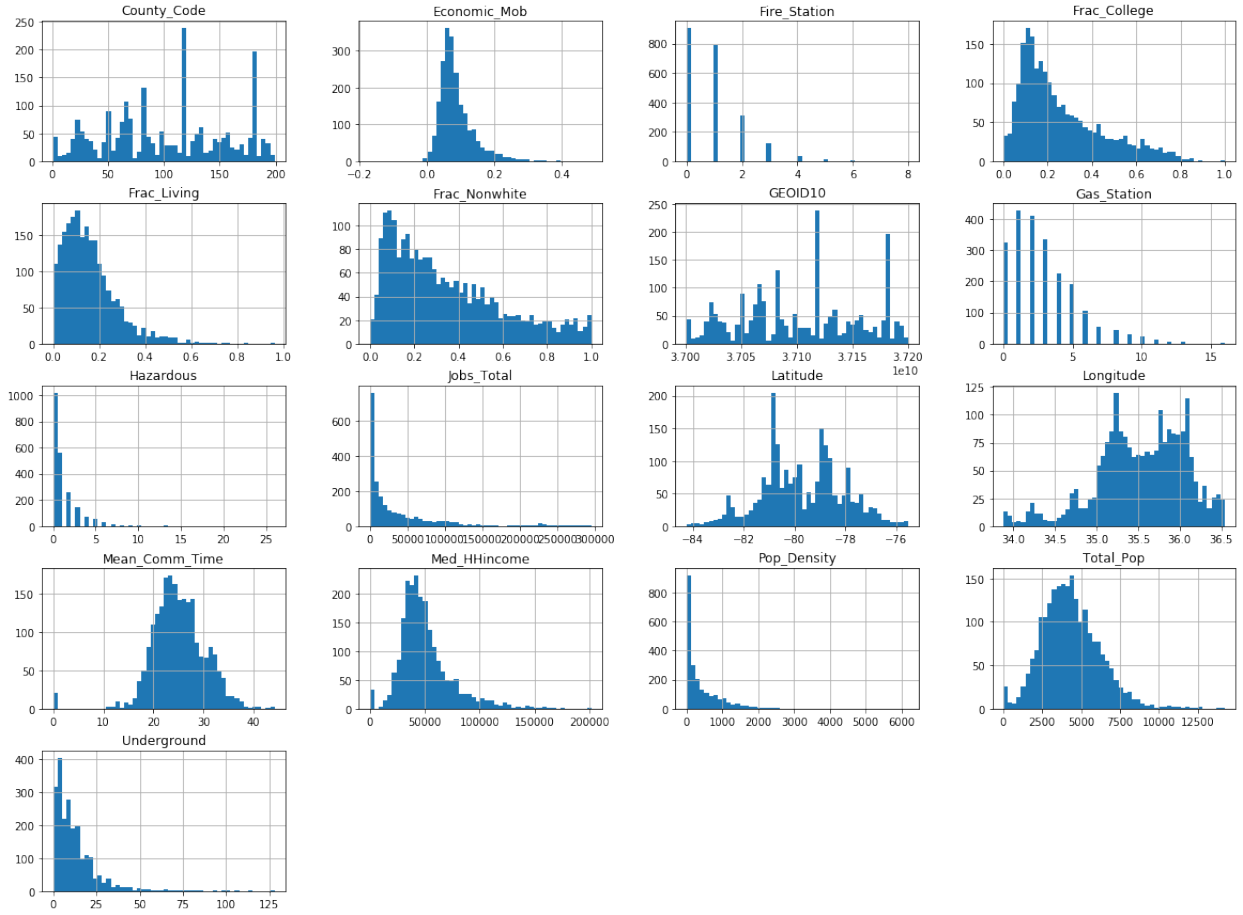
Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb772e96bd0>
```



Data Visualization

Show the histogram for each columns

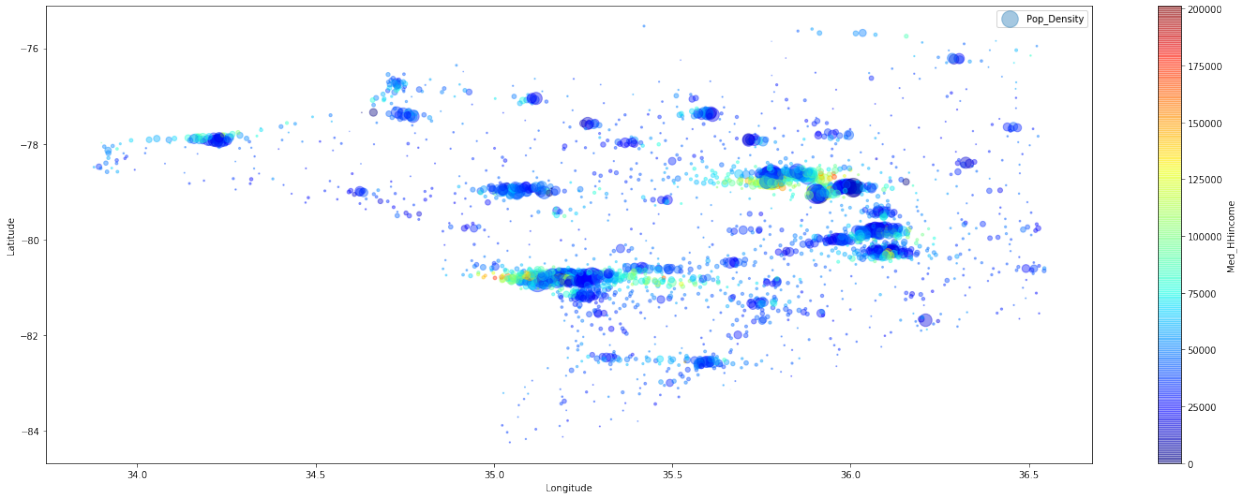
In [27]:
```
data.hist(bins=50, figsize=(20,15))
plt.show()
```



Use the color map to visualize population density and median household income for NC geo tracts.

In [28]:
```
data.plot(kind="scatter", x="Longitude", y="Latitude", alpha=0.4,
    s=data["Pop_Density"]/10, label="Pop_Density", figsize=(25,9),
    c="Med_HHincome", cmap=plt.get_cmap("jet"), colorbar=True,
    sharex=False)
plt.legend()
```

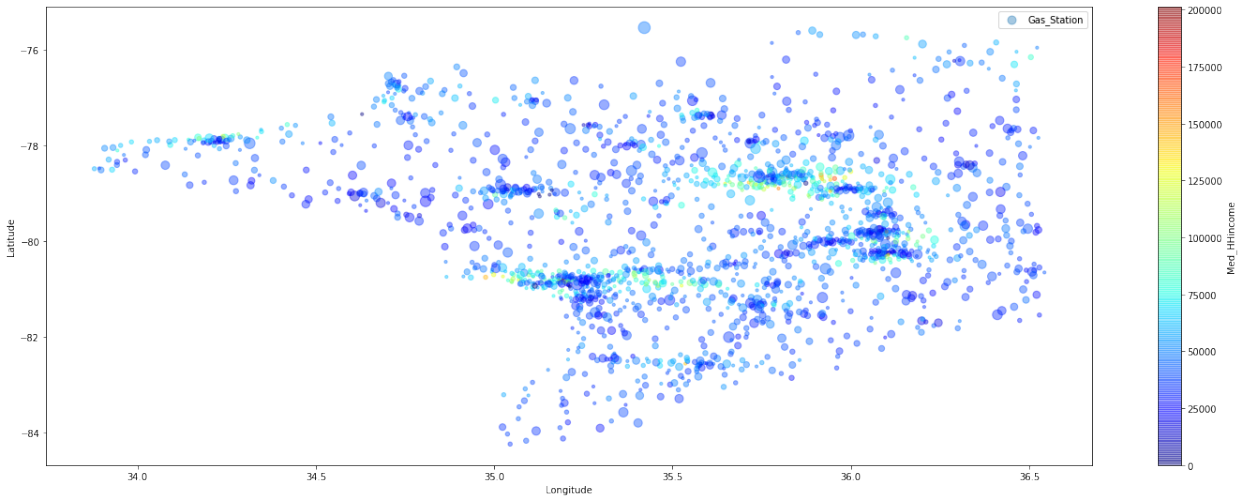
Out[28]: <matplotlib.legend.Legend at 0x7fb774523710>



Use the color map to visualize number of gas stations and median household income for NC geo tracts.

In [29]:
```
data.plot(kind="scatter", x="Longitude", y="Latitude", alpha=0.4,
    s=data["Gas_Station"]*10, label="Gas_Station", figsize=(25,9),
    c="Med_HHincome", cmap=plt.get_cmap("jet"), colorbar=True,
    sharex=False)
plt.legend()
```

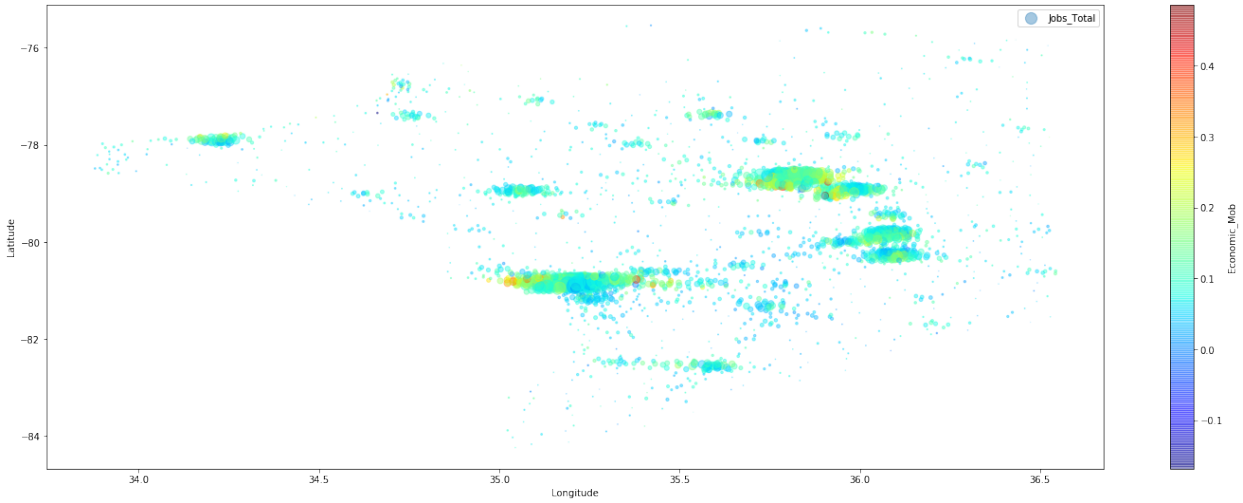
Out[29]: <matplotlib.legend.Legend at 0x7fb773f46750>



Use the color map to visualize number of total jobs and the value of Economic mibility for NC geo tracts.

In [30]:
```
data.plot(kind="scatter", x="Longitude", y="Latitude", alpha=0.4,
    s=data["Jobs_Total"]/1000, label="Jobs_Total", figsize=(25,9),
    c="Economic_Mob", cmap=plt.get_cmap("jet"), colorbar=True,
    sharex=False)
plt.legend()
```

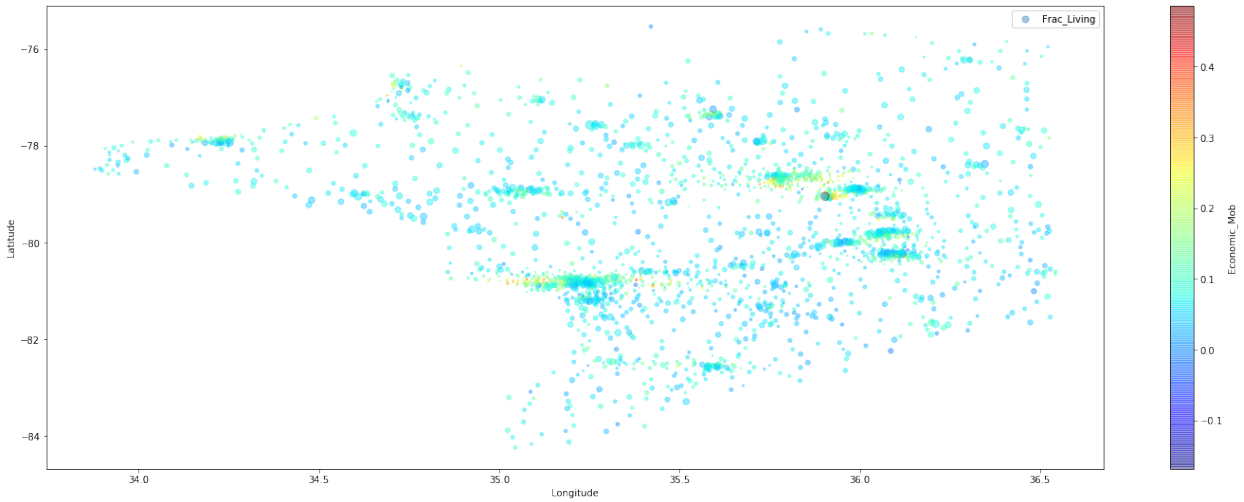
Out[30]:
```
<matplotlib.legend.Legend at 0x7fb775c51510>
```



Use the color map to visualize the fraction of people living with poverty and the value of Economic mibility for NC geo tracts.

In [31]:
```
data.plot(kind="scatter", x="Longitude", y="Latitude", alpha=0.4,
    s=data["Frac_Living"]*100, label="Frac_Living", figsize=(25,9),
    c="Economic_Mob", cmap=plt.get_cmap("jet"), colorbar=True,
    sharex=False)
plt.legend()
```

Out[31]:
```
<matplotlib.legend.Legend at 0x7fb7738b3250>
```



In [ ]: