In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

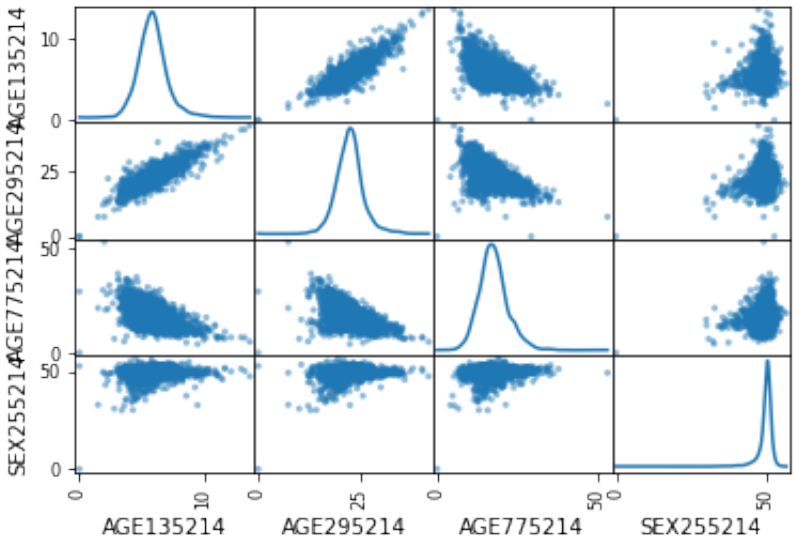
In [3]:
```
data = pd.read_csv("county_facts.csv")
primary_results = pd.read_csv("primary_results.csv")
```

In [4]:
```
data.loc[:, "PST045214":"POP060210"].describe()
```

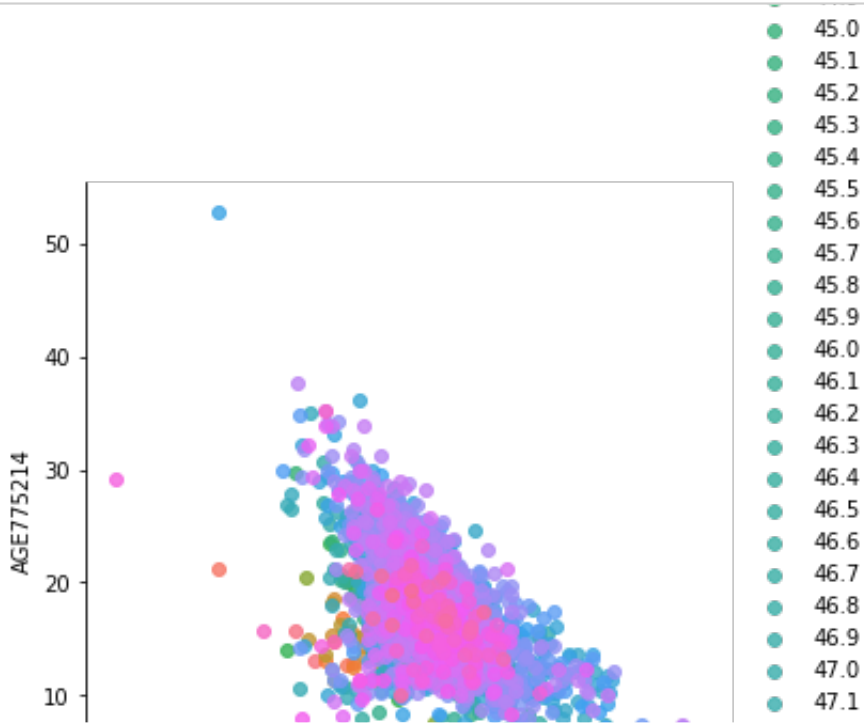
Out[4]:

| | PST045214 | PST040210 | PST120214 | POP010210 | AGE135214 | AGE295214 |
|---|---|---|---|---|---|---|
| **count** | 3.195000e+03 | 3.195000e+03 | 3195.000000 | 3.195000e+03 | 3195.000000 | 3195.000000 |
| **mean** | 2.993963e+05 | 2.899137e+05 | 0.508545 | 2.899019e+05 | 5.900782 | 22.545290 |
| **std** | 5.768468e+06 | 5.584181e+06 | 4.180066 | 5.583955e+06 | 1.211925 | 3.418896 |
| **min** | 0.000000e+00 | 0.000000e+00 | -17.000000 | 8.200000e+01 | 0.000000 | 0.000000 |
| **25%** | 1.113450e+04 | 1.129500e+04 | -1.900000 | 1.129500e+04 | 5.200000 | 20.500000 |
| **50%** | 2.655300e+04 | 2.644600e+04 | -0.100000 | 2.642400e+04 | 5.800000 | 22.500000 |
| **75%** | 7.247600e+04 | 7.150450e+04 | 2.300000 | 7.144800e+04 | 6.500000 | 24.200000 |
| **max** | 3.188571e+08 | 3.087581e+08 | 72.900000 | 3.087455e+08 | 13.700000 | 42.000000 |

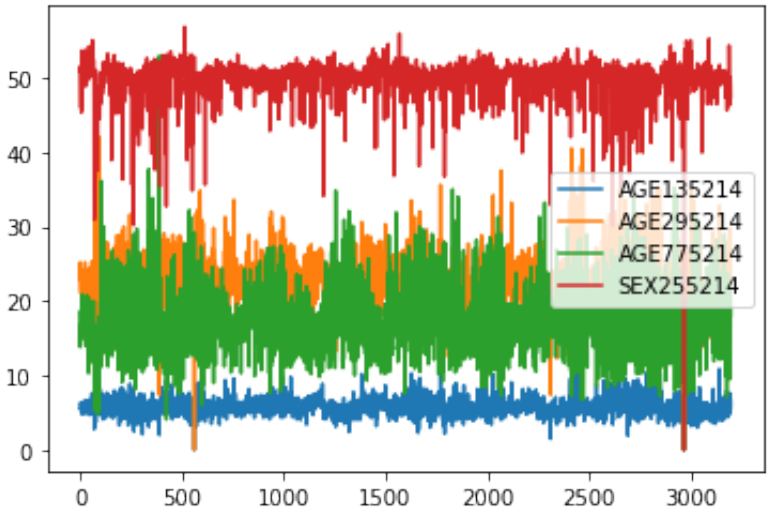
In [6]:
```
pd.plotting.scatter_matrix(data.loc[:, "AGE135214":"SEX255214"], di
plt.show()
```



In [9]:
```
sns.lmplot("AGE295214", "AGE775214", data, hue="SEX255214", fit_reg
```




In [10]:
```
ax = data.loc[:, "AGE135214":"SEX255214"].plot()
ax.legend(loc='center right', bbox_to_anchor=(1, 0.5));
```




In [11]:
```
y = primary_results.candidate

Trump = primary_results[y=='Donald Trump']
Hillary = primary_results[y=='Hillary Clinton']
Bernie = primary_results[y=='Bernie Sanders']
Trump = Trump.query('fraction_votes > 0')
Hillary = Hillary.query('fraction_votes > 0')
Bernie = Bernie.query('fraction_votes > 0')
```

In [12]:
```
data = data.replace(to_replace='County', value="",regex=True)
```

In [13]:
```
Trump.columns = ['state','state_abbreviation', 'area_name','fips','
Hillary.columns = ['state','state_abbreviation', 'area_name','fips'
Bernie.columns = ['state','state_abbreviation', 'area_name','fips',
```

In [14]:
```
Trump.result = pd.merge(data, Trump, on='area_name')
Hillary.result = pd.merge(data, Hillary, on='area_name')
Bernie.result = pd.merge(data, Bernie, on='area_name')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access (https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access)
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access (https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWarning: Pandas doesn't allow columns to be created via a new attribute name - see https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access (https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access)
  This is separate from the ipykernel package so we can avoid doing imports until
```

In [15]:
```
result = pd.concat([Trump.result, Hillary.result, Bernie.result], i
```

In [16]:
```
y = result.candidate
```

In [17]:
```
X = result.loc[:, "PST045214":"POP060210"]
```

In [18]:
```
def printMeanAndSdByGroup(variables, groupvariable):
    data_groupby = variables.groupby(groupvariable)
    print("## Means:")
    display(data_groupby.apply(np.mean))
    print("\n## Standard deviations:")
    display(data_groupby.apply(np.std))
    print("\n## Sample sizes:")
    display(pd.DataFrame(data_groupby.apply(len)))
```

In [19]:
```
printMeanAndSdByGroup(X, y)
```

```
## Means:
```

| candidate | PST045214 | PST040210 | PST120214 | POP010210 | AGE135214 | AGE295214 |
|---|---|---|---|---|---|---|
| **Bernie Sanders** | 6.382734e+06 | 6.097231e+06 | 4.238333 | 6.097058e+06 | 6.32000 | 23.030000 |
| **Donald Trump** | 6.359328e+06 | 6.075599e+06 | 4.212069 | 6.075421e+06 | 6.32069 | 23.041379 |
| **Hillary Clinton** | 6.382734e+06 | 6.097231e+06 | 4.238333 | 6.097058e+06 | 6.32000 | 23.030000 |

```
## Standard deviations:
```

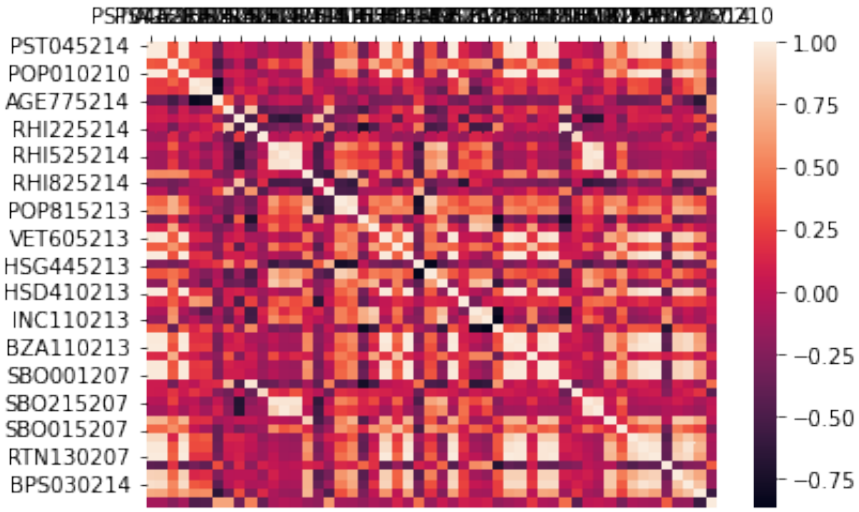
| candidate | PST045214 | PST040210 | PST120214 | POP010210 | AGE135214 | AGE295214 |
|---|---|---|---|---|---|---|
| **Bernie Sanders** | 5.144650e+06 | 4.844391e+06 | 1.68514 | 4.844192e+06 | 0.424185 | 1.435769 |
| **Donald Trump** | 5.231028e+06 | 4.925782e+06 | 1.70790 | 4.925579e+06 | 0.431421 | 1.458983 |
| **Hillary Clinton** | 5.144650e+06 | 4.844391e+06 | 1.68514 | 4.844192e+06 | 0.424185 | 1.435769 |

```
## Sample sizes:
```

| candidate | 0 |
|---|---|
| **Bernie Sanders** | 60 |
| **Donald Trump** | 58 |
| **Hillary Clinton** | 60 |

In [21]:
```
corrmat = X.corr()
```

In [22]:
```
sns.heatmap(corrmat, vmax=1., square=False).xaxis.tick_top()
```




In [ ]: