# Group2_FinalNotebook

October 6, 2021

[2]:
```
import requests
from pathlib import Path
import zipfile
import os, glob
import csv

import pandas as pd
import numpy as np
from numpy.linalg import inv
import statsmodels as sm

import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import seaborn as sns
sns.set_context("talk")
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Lasso, LassoCV, RidgeCV
from sklearn.model_selection import cross_val_score, train_test_split, KFold,␣
 ↪GridSearchCV
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn import ensemble

import statsmodels.api as sm
```

## 1 Predicting EV Adoption Using Census Block Data

Fall 2020

Team membes: Kunal Malhotra, Payton McSweeney, Ally Novales, Cameron Wright

Kunal: Helped with the idea generating / brainstorm process. Downloaded fleet database. Worked with Cameron on thee Project Background and Project Objective. Created county level EDA plots and wrote descriptions for them. Worked on the second prediction problem to predict adoption at the county level. Collaborated with Cameron to write interpretation and conclusions.

Payton: Helped come up with project ideas, downloaded half of the census data for features (with Cameron doing the other half). Grouped features and created proportions to create our final feature set, wrote the input data description section. Worked on prediction problem 3 with Ally (predicting commute time from our other data). Helped Kunal write descriptions of EDA process, did external validity testing with NorCal vs SoCal data.

Ally: Collaborated in ideation phase for model, researched data collection, and performed majority of Data Cleaning. Worked with all group members to have frequent meeting times over Zoom, checkpoints with action items, and organizing relevant files via GoogleDrive. Put together final notebook on Jupyter and fixed (multiple) bugs when inputting various parts of code.

Cameron: Contributed to the initial brainstorming process of choosing a topic. Downloaded half of the data used for our features (Payton did other half) from the Census Bureau. Wrote much of the Project Background and Project Objective sections, contributed in smaller part to the Interpretation and Conclusion and Abstract sections. Collaborated with Kunal in organizing and coding the first two prediction problems (I mostly worked on the first, Kunal on the second).

### 1.1 Abstract (5 points)

In recent years, we've seen growing awareness about the negative impact that internal combustion engine passenger vehicles can have on our planet. In response to this, we've seen a range of responses from electric vehicle (EV) manufacturers like Tesla that have created EVs that are indistinguishable from normal passenger vehicles, to states like California that have introduced legislation to ensure all new vehicles sold after 2035 are emission-free. Amidst these changes, the US has a long way to go in terms of achieving a meaningful level of fleet electrification. Most of the existing research on EV adoption focuses on state or county level adoption, consequently, we chose to predict EV adoption at a far greater granularity - the census block group level. We used census block group level variation in features such as race, income, education, and commute time to develop both parametric and non-parametric methods to determine the proportion of EVs registered in a census block group divided by the total number of passenger vehicles registered in the same region during the same year.

Through our exploration into this topic, we learned that demographic information (tracked in census data) can be used as fairly strong predictors of electric vehicle adoption. In addition, we came to see the value in creating both parametric and non-parametric models, especially in situations where we are focused on prediction rather than inference. A valuable takeaway for us is a newfound appreciation for how influential data provenance can be in the generation of data driven models.

### 1.2 Project Background (5 points)

Our motivation for embarking on this project came in part from our view that electric vehicles will be pivotal in the decarbonization of the transportation sector. According to the US EPA, the transportation sector (which includes all forms of transportation such as cars, trucks, plains, trains, and ships) accounted for 28 percent of all US greenhouse gas emissions in 2018, which was more than any other economic sector. Furthermore, passenger cars and light duty trucks contributed more than half of those emissions ("Sources of Greenhouse Gas Emissions").

Despite its purported merit, we recognize that electric vehicle adoption alone is not sufficient to ultimately reduce greenhouse gas emissions tied to transportation. The source of electricity for our electric vehicles will need to be renewably generated or else we will simply be transferring emissions from the transportation sector to the electricity generation sector (currently the second largest economic sector in terms of emissions). Given that transitions to renewables are already underway in the electricity generation sector, we feel confident that fleet electrification is an important step towards decarbonization.

California Governor, Gavin Newsom, shares our sentiment. In late September 2020, Newsom issued executive order N-79-20, requiring that all new passenger vehicles sold be emission-free by 2035. It is worth noting that in California, the transportation sector accounts for more than 50 percent of greenhouse gas emissions and successful execution of Newsom's order is projected to reduce total statewide emissions by 35 percent (State of California). Additionally, this order has implications that reach beyond the scope of climate change mitigation. California's transportation sector currently accounts for the vast majority of smog forming pollution and toxic diesel emissions. In a statement that addresses both climate change and human health concerns, the Governor proclaimed, "Californians shouldn't have to worry if our cars are giving our kids asthma. Our cars shouldn't make wildfires worse – and create more days filled with smoky air. Cars shouldn't melt glaciers or raise sea levels threatening our cherished beaches and coastlines" (State of California).

One might call into question whether executive order N-79-20 is realistic for California. After all, California's auto sales are currently far from 100 percent electric. Perhaps Newsom's 2035 goal is too lofty.

Another key piece of background for this research project is the changing nature of the EV market, particularly with regards to their price and range parity with gasoline vehicles. While cars like the Nissan Leaf and other non-mainstream vehicles introduced the idea of EVs to the market, the recent work of companies like Tesla, especially their Model 3 vehicle, have created a new breed of EVs that offer an identical driving experience and near price parity. Tesla's success, alongside changing global regulations, have led many other large car manufacturers such as Volkswagen, Volvo, and Ford (the EV Mustang will be coming out in 2021) to prioritize the manufacturing of EVs.

We think that use of economic incentives will be key in reaching California's goal. Understanding the current regional differences in demand for electric vehicles will be important in figuring out exactly how to quantify these incentives. In addition, these incentives may lie at both the producer and the consumer level. Fortunately, California provides data on electric vehicle adoption at the census block group level to help make these decisions. However, not all states collect this information at such a granular level, posing a barrier in effectively influencing demand for electric vehicles. A mechanism for predicting regional differences in electric vehicle adoption in areas where such information is not available could be instrumental in advising state governments on how to help the auto industry promote sales of electric vehicles. There are further applications of this research looking towards EV charging locations and how adoption rates may influence where charging stations are placed. This wasn't factored into our research, but would be an interesting follow up topic.

[**Sources**]

[1] California, State of. "Governor Newsom Announces California Will Phase Out Gasoline-Powered Cars & Drastically Reduce Demand for Fossil Fuel in California's Fight Against Climate Change." Office of Governor Gavin Newsom, CA.gov, 23 Sept. 2020, www.gov.ca.gov/2020/09/23/governor-newsom-announces-california-will-phase-out-gasoline-powered-cars-drastically-reduce-demand-for-fossil-fuel-in-californias-fight-against-climate-change/.

[2] "Sources of Greenhouse Gas Emissions." Greenhouse Gas Emissions, Environmental Protection

Agency, www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions.

### 1.3 Project Objective (5 points)

The purpose of this project is to train and evaluate different models to predict regional variation in electric vehicle adoption in areas that do not provide EV information, using census block level variation in features such as education, income, race, and commute time. Prediction power over regional variations in electric vehicle adoption will give states an easy, low effort way to effectively make decisions about where to grant subsidies for electric vehicle sales.

The main resource allocation problem that we aim to tackle is the fact that states typically have limited budgets, so we want to help them better understand census block level EV adoption, so they can efficiently target subsidies and other outreach methods to the areas that need them most. We see this as a very important problem, since EV adoption will be a key part of the global transition to a more sustainable future. In addition, as discussed above, EV manufacturers are at a key inflection point where they have drastically shifted attention from gas vehicles to producing equivalent EVs.

During our research, we didn't come across much work that looked at EV adoption at the census block level, though many have studied state-level adoption. I think that this adds novelty to the approach that we've taken and the research that we're doing. Much of the spatially focussed research has focussed on EV charging locations and their placement, rather than overall EV adoption.

### 1.4 Input Data Description (5 points)

We used the US Census Bureau to obtain data on a block group basis for all of the counties in California. From the Census data, we collected datasets on yearly income, average commute time, education level, and race per census block. We also collected data from the California Air Resources Board's Fleet Database, which contains data on the number of EV's as well as the total number of vehicles per census block group. The census data was collected via the yearly census done by the US Census Bureau, whereas the Fleet Database was generated from vehicle registration data from California Department of Motor Vehicles.

Structure: For each of our individual data sets (which we merged into one dataframe) the data is grouped by census block groups in each county. Each row in the data frame corresponds to a census block in California, and each column represents either an identifier (such as which county the census block belongs to) or a feature (race, income, etc). We downloaded all of these files as CSV files.

Granularity: Each row represents an observation from a specific block group and county from 2018. Scope: For our geographic scope, our data covers all of the census block groups in the state of California. The temporal scope of our data covers 2018, as this is the most recent (and relevant) data that we have access to.

Temporality: Our data set covers the 2018 year, however there is only one data point per census block per year since these are yearly measurements.

Faithfulness: Our Census data comes from the Census Bureau, which is a government group that collects self-reported data on the population. Given that the data is self reported and voluntarily recorded, our data may be potentially skewed. The fleet data contains data from the DMV's vehicle registrations, which are very reliable.

## 1.5 Data Cleaning (10 points)

For our final dataframe, we knew that we wanted to group it by Census Block Group Code and County. However, since we're working with 2 different data sources (US Census Bureau and EMFAC), we will need to do various data cleaning for each dataset until (and after) we perform the final merge. Since the Fleet Database (from EMFAC) is less comprehensive, let's start with that one first:

### 1.5.1 Fleet Database

**Problem:**

1. The data was downloaded by county, and we want EV and overall vehicle population for all counties in CA
2. EV data listed that the population of EVs was listed under the header "Vehicle Population"
3. No proportion calculated for proportion of EVs per census block group
4. Breifly looking at the data, it's clear that there are some NaN values as well as entries listed as "Unkown" or "Scrubbed" for certain block groups

Let's merge all the counties into its respective datasets (EV population and vehicle population) first.

```
[3]: # ADDRESSING PROBLEM 1: downloading each county's csv and merging them into␣
      ↪their respective datasets
     path = 'Fleet Database'

     #creating a path for python to read through each county's csv file
     #there is one character that is different for EV data vs all vehicle data
     all_files_EV = glob.glob(os.path.join(path,␣
      ↪'*P-GVWR-E-Agg-Agg-Agg-Agg-ByCensusBlockGroupCode.csv'))
     all_files_all = glob.glob(os.path.join(path,␣
      ↪'*P-GVWR-Agg-Agg-Agg-Agg-Agg-ByCensusBlockGroupCode.csv'))

     #concatenating each county's data into respective datasets
     df_from_each_file_EV = (pd.read_csv(f, sep=',') for f in all_files_EV)
     df_EV = pd.concat(df_from_each_file_EV, ignore_index=True)

     df_from_each_file_all = (pd.read_csv(f, sep=',') for f in all_files_all)
     df_all = pd.concat(df_from_each_file_all, ignore_index=True)
```

```
[4]: df_EV.head()
```

```
[4]:   Vehicle Category      GVWR Class Fuel Type County    MPO   Sub-Area  \
     0                P  Not Applicable  Electric   YUBA  SACOG  Yuba (SV)
     1                P  Not Applicable  Electric   YUBA  SACOG  Yuba (SV)
     2                P  Not Applicable  Electric   YUBA  SACOG  Yuba (SV)
     3                P  Not Applicable  Electric   YUBA  SACOG  Yuba (SV)
     4                P  Not Applicable  Electric   YUBA  SACOG  Yuba (SV)
```

```
  Census Block Group Code ZIP Code  EV Population
0             60570004022    95977              1
1             61150402002    95901              1
2             61150403021    95901              1
3             61150403031    95901              5
4             61150404003    95961              1
```

[5]:

```
df_all.head()
```

[5]:

```
  Vehicle Category      GVWR Class County    MPO   Sub-Area  \
0                P  Not Applicable   YUBA  SACOG  Yuba (SV)
1                P  Not Applicable   YUBA  SACOG  Yuba (SV)
2                P  Not Applicable   YUBA  SACOG  Yuba (SV)
3                P  Not Applicable   YUBA  SACOG  Yuba (SV)
4                P  Not Applicable   YUBA  SACOG  Yuba (SV)

  Census Block Group Code ZIP Code  Vehicle Population
0             60070033001    95901                   5
1             60070033002    95901                  19
2             60070033003    95901                   4
3             60570004022    95977                 186
4             60570004024    95977                  42
```

Since both datasets were collected in the same way, let's merge them together and clean them afterwards:

[6]:

```
# ADDRESSING PROBLEMS 2 AND 3 : keeping only relevant columns and calculating␣
 ↪EV Population per census block group
df_merged = df_EV.merge(df_all, how='outer')
census_block = df_merged.groupby(by=['Census Block Group Code', 'County'],␣
 ↪as_index=False).sum()
EV_pop = census_block['EV Population'].values
all_pop = census_block['Vehicle Population'].values
census_block['Proportion EV'] = EV_pop / all_pop
prop_ev = census_block

prop_ev.head()
```

```
<ipython-input-6-c21e03b9a9b7>:6: RuntimeWarning: divide by zero encountered in
true_divide
  census_block['Proportion EV'] = EV_pop / all_pop
```

[6]:

```
  Census Block Group Code   County  EV Population  Vehicle Population  \
0             60110001004   COLUSA            1.0                 0.0
1             60110003004   COLUSA            1.0                 0.0
2             60210105021   COLUSA            1.0                 0.0
3             60014001001  ALAMEDA          122.0              1506.0
4             60014002001  ALAMEDA           26.0               431.0
```

```
   Proportion EV
0            inf
1            inf
2            inf
3       0.081009
4       0.060325
```

[7]:

```
# ADDRESSING PROBLEM 4: checking for null values, filtering out "Unknown" and␣
 ↪"Scrubbed" entries

unknown = prop_ev.loc[prop_ev['Census Block Group Code'] == 'Unknown']
prop_ev = prop_ev.drop(unknown.index)

scrubbed = prop_ev.loc[prop_ev['Census Block Group Code'] == 'Scrubbed']
prop_ev = prop_ev.drop(scrubbed.index)

prop_ev.isnull().sum()
```

[7]:

```
Census Block Group Code    0
County                     0
EV Population              0
Vehicle Population         0
Proportion EV              0
dtype: int64
```

[8]:

```
#Checking if dtypes make sense
prop_ev.dtypes
```

[8]:

```
Census Block Group Code     object
County                      object
EV Population              float64
Vehicle Population         float64
Proportion EV              float64
dtype: object
```

This makes sense since Census Block Group Code is a categorical feature, EV/Vehicle Population should be whole numbers (integers), and Proportion EV should be a decimal.

[9]:

```
prop_ev
```

[9]:

```
      Census Block Group Code   County  EV Population  Vehicle Population  \
0                 60110001004   COLUSA            1.0                 0.0
1                 60110003004   COLUSA            1.0                 0.0
2                 60210105021   COLUSA            1.0                 0.0
3                 60014001001  ALAMEDA          122.0              1506.0
4                 60014002001  ALAMEDA           26.0               431.0
```

```
…                        …      …             …                  …
23154              61150411001    YUBA          0.0              74.0
23155              61150411002    YUBA          1.0              65.0
23156              61150411003    YUBA          0.0              66.0
23157              61150411004   BUTTE          0.0              18.0
23158              61150411004    YUBA          0.0              52.0

       Proportion EV
0                inf
1                inf
2                inf
3           0.081009
4           0.060325
…                …
23154       0.000000
23155       0.015385
23156       0.000000
23157       0.000000
23158       0.000000

[23159 rows x 5 columns]
```

## 1.6 ACS Data

**Note: Various datasets were downloaded per county based on these desired features (does this word make sense): Education, Income, Race, and Commute Time. The following data cleaning was performed on all sections, but the following explanation is in regards to the Education dataset**

#### Problems

1. Like the Fleet Database, the data was downloaded by county, and we want all counties to be in one dataframe
2. Wrong column names or unneeded columns, with unneeded data: columns that start with "B15003" and "Margin of Error!!"
3. The entries for "Census Block Group Code" and "County" in the Fleet Database don't match with those in the ACS data

### 1.6.1 Education

[10]:
```
# ADRESSING PROBLEM 1: download each csv, modeled after the way the fleet␣
 ↪database was extracted

def make_df(path):
    #path is a string
    all_files = glob.glob(os.path.join(path, '*.csv'))
    df_from_each_file = (pd.read_csv(f, sep=',') for f in all_files)
    df = pd.concat(df_from_each_file, ignore_index=True)
```

```python
    return df

education = make_df('Education')
education.head()
```

[10]:

```
                  GEO_ID                                               NAME  \
0                     id                               Geographic Area Name   
1  1500000US060570001024  Block Group 4, Census Tract 1.02, Nevada Count…   
2  1500000US060570012041  Block Group 1, Census Tract 12.04, Nevada Coun…   
3  1500000US060570001051  Block Group 1, Census Tract 1.05, Nevada Count…   
4  1500000US060570006002  Block Group 2, Census Tract 6, Nevada County, …   

       B15003_001E            B15003_001M  \
0  Estimate!!Total  Margin of Error!!Total   
1             1032                    162   
2              637                    221   
3              699                    168   
4              765                    214   

                               B15003_002E  \
0  Estimate!!Total!!No schooling completed   
1                                        0   
2                                        0   
3                                        0   
4                                        0   

                                      B15003_002M  \
0  Margin of Error!!Total!!No schooling completed   
1                                              12   
2                                              12   
3                                              12   
4                                              12   

                       B15003_003E                             B15003_003M  \
0  Estimate!!Total!!Nursery school  Margin of Error!!Total!!Nursery school   
1                                0                                      12   
2                                0                                      12   
3                                0                                      12   
4                                0                                      12   

                     B15003_004E                           B15003_004M  …  \
0  Estimate!!Total!!Kindergarten  Margin of Error!!Total!!Kindergarten  …   
1                              0                                    12  …   
2                              0                                    12  …   
3                              0                                    12  …   
4                              0                                    12  …   
```

```
                           B15003_021E  \
0  Estimate!!Total!!Associate's degree
1                                  111
2                                   82
3                                  115
4                                   86

                                  B15003_021M  \
0  Margin of Error!!Total!!Associate's degree
1                                          67
2                                          70
3                                          94
4                                          61

                          B15003_022E  \
0  Estimate!!Total!!Bachelor's degree
1                                 186
2                                 149
3                                 124
4                                  80

                                 B15003_022M  \
0  Margin of Error!!Total!!Bachelor's degree
1                                         80
2                                         76
3                                         80
4                                         65

                        B15003_023E                              B15003_023M  \
0  Estimate!!Total!!Master's degree  Margin of Error!!Total!!Master's degree
1                                61                                       41
2                               160                                      108
3                                60                                       43
4                                34                                       32

                                   B15003_024E  \
0  Estimate!!Total!!Professional school degree
1                                           16
2                                            8
3                                           32
4                                            4

                                         B15003_024M  \
0  Margin of Error!!Total!!Professional school de…
1                                                 25
2                                                 16
3                                                 33
```

```
4                                                 8

                         B15003_025E                                B15003_025M
0  Estimate!!Total!!Doctorate degree  Margin of Error!!Total!!Doctorate degree
1                                  0                                        12
2                                 18                                        19
3                                  7                                        12
4                                  5                                        12

[5 rows x 52 columns]
```

[11]:

```
# ADDRESSING PROBLEM 2: because of the way we concatenated each county's data,␣
 ↪there are duplicates
# of the original column names
def rename(df, new_names):
    old_names = df.columns

    for i in np.arange(len(new_names)):
        df = df.rename(columns = {old_names[i] : new_names[i]})

    return df


def fix_cols(df):
    duplicated = df.loc[df['GEO_ID'] == 'id']

    #actual column names
    new_names = duplicated.iloc[[0]].transpose()[0].values

    #don't run this twice
    df = df.drop(duplicated.index)

    #rename
    df = rename(df, new_names)

    return df

education = fix_cols(education)
```

[12]:

```
#ADDRESSING PROBLEM 2: Once adjusted, we also use regex to drop the columns the␣
 ↪columns that are labeled "Margin of Error!!",
# while storing them just in case. Then, we renamed the columns to its original␣
 ↪titles as cited in the ACS website

def moe(df, labels):
    moe = df.filter(regex='Margin of Error!!')
    df = df[df.columns.drop(list(moe))]
    df = rename(df, labels)
    return df, moe

#because of how `rename` is implemented, we need to start labels at id
labels_edu = ['id', 'Geographic Area Name', 'Total', 'No schooling completed',␣
 ↪'Nursery school', 'Kindergarten', '1st grade',
              '2nd grade', '3rd grade', '4th grade', '5th grade', '6th grade',␣
 ↪'7th grade', '8th grade', '9th grade',
              '10th grade', '11th grade', '12th grade, no diploma', 'Regular␣
 ↪high school diploma',
              'GED or alternative credential', 'Some college, less than 1␣
 ↪year', 'Some college, 1 or more years, no degree',
              "Associate's degree", "Bachelor's degree", "Master's degree",␣
 ↪"Professional school degree", 'Doctorate degree']
education, moe_edu = moe(education, labels_edu)
```

[13]:

```
#ADDRESSING PROBLEM 3: matching "Census Block Group Code" in the Fleet Database␣
 ↪and County Names (for convenience)

def cen_block_gc(df):
    id_values = df['id'].values
    cen_block_gc = [id_values[i].strip('1500000US0') for i in␣
 ↪range(len(id_values))]
    df['Census Block Group Code'] = cen_block_gc
    return df

def county_name(df):
    geo_area = df['Geographic Area Name'].values
    county_names = [geo_area[i].split(',')[2].strip().replace(' County', '').
 ↪upper() for i in range(len(geo_area))]
    df['County'] = county_names
    return df

education = cen_block_gc(education)
education = county_name(education)

education.head()
```

[13]:

```
                        id                                  Geographic Area Name  \
1  1500000US060570001024  Block Group 4, Census Tract 1.02, Nevada Count…
2  1500000US060570012041  Block Group 1, Census Tract 12.04, Nevada Coun…
3  1500000US060570001051  Block Group 1, Census Tract 1.05, Nevada Count…
4  1500000US060570006002  Block Group 2, Census Tract 6, Nevada County, …
5  1500000US060570005011  Block Group 1, Census Tract 5.01, Nevada Count…

  Total No schooling completed Nursery school Kindergarten 1st grade  \
1  1032                      0              0            0         0
2   637                      0              0            0         0
3   699                      0              0            0         0
4   765                      0              0            0         0
5   764                      0              0            0         0

  2nd grade 3rd grade 4th grade  … GED or alternative credential  \
1         0         0         0  …                            42
2         0         0         0  …                            27
3         0         0         0  …                             0
4         0         0         0  …                            14
5         0         0         0  …                             0

  Some college, less than 1 year Some college, 1 or more years, no degree  \
1                            150                                      221
2                              0                                      147
3                             54                                      186
4                            139                                      192
5                             31                                      175

  Associate's degree Bachelor's degree Master's degree  \
1                111               186              61
2                 82               149             160
3                115               124              60
4                 86                80              34
5                 47               151              55

  Professional school degree Doctorate degree Census Block Group Code  County
1                         16                0             60570001024  NEVADA
2                          8               18              6057001204  NEVADA
3                         32                7                    6057  NEVADA
4                          4                5             60570006002  NEVADA
5                         53               23                    6057  NEVADA

[5 rows x 29 columns]
```

[14]:

```
#checking for null values
education.isnull().values.any()
```

[14]:

```
False
```

[15]:

```
#double checking datatypes make sense
education.dtypes
```

[15]:

```
id                                          object
Geographic Area Name                        object
Total                                       object
```

```
No schooling completed                      object
Nursery school                              object
Kindergarten                                object
1st grade                                   object
2nd grade                                   object
3rd grade                                   object
4th grade                                   object
5th grade                                   object
6th grade                                   object
7th grade                                   object
8th grade                                   object
9th grade                                   object
10th grade                                  object
11th grade                                  object
12th grade, no diploma                      object
Regular high school diploma                 object
GED or alternative credential               object
Some college, less than 1 year              object
Some college, 1 or more years, no degree    object
Associate's degree                          object
Bachelor's degree                           object
Master's degree                             object
Professional school degree                  object
Doctorate degree                            object
Census Block Group Code                     object
County                                      object
dtype: object
```

Let's do the same for Income, Race, and Commute Time:

### 1.6.2 Income

[16]:

```
income = make_df('Income')

#fix columns
income = fix_cols(income)
income.head()

#original lablels, make sure to use \ to prevent making a formula via LaTexa
labels_i = ['id', 'Geographic Area Name', 'Total', 'Less than $10,000',␣
 ↪'\$10,000 to $14,999', '\$15,000 to $19,999',
           '\$20,000 to $24,999', '\$25,000 to $29,999', '\$30,000 to $34,999',␣
 ↪'\$35,000 to $39,999','\$40,000 to $44,999',
           '\$45,000 to $49,999', '\$50,000 to $59,999', '\$60,000 to $74,999',␣
 ↪'\$75,000 to $99,999', '\$100,000 to $124,999',
           '\$125,000 to $149,999', '\$150,000 to $199,999', '\$200,000 or␣
 ↪more']
```

```
income, moe_i = moe(income, labels_i)

income = cen_block_gc(income)
income = county_name(income)
income.head()
```

[16]:

```
                      id                               Geographic Area Name  \
1  1500000US060570001042  Block Group 2, Census Tract 1.04, Nevada Count…
2  1500000US060570001041  Block Group 1, Census Tract 1.04, Nevada Count…
3  1500000US060570001051  Block Group 1, Census Tract 1.05, Nevada Count…
4  1500000US060570004022  Block Group 2, Census Tract 4.02, Nevada Count…
5  1500000US060570001052  Block Group 2, Census Tract 1.05, Nevada Count…

  Total Less than $10,000 \$10,000 to $14,999 \$15,000 to $19,999  \
1   679                65                  33                  33
2   358                11                  28                  35
3   338                 7                  48                   8
4   811                62                   0                  30
5   332                41                   0                   0

  \$20,000 to $24,999 \$25,000 to $29,999 \$30,000 to $34,999  \
1                  37                  39                  90
2                  14                  26                  11
3                  21                   0                   8
4                  33                  55                  22
5                   0                   0                   9

  \$35,000 to $39,999  … \$45,000 to $49,999 \$50,000 to $59,999  \
1                  17  …                  38                  89
2                  34  …                  12                  47
3                   0  …                  11                  19
4                  14  …                   0                  41
5                  60  …                  17                  29

  \$60,000 to $74,999 \$75,000 to $99,999 \$100,000 to $124,999  \
1                  59                  45                    30
2                  25                  46                    12
3                  32                  82                    29
4                 208                  73                    17
5                   8                  34                    86

  \$125,000 to $149,999 \$150,000 to $199,999 \$200,000 or more  \
1                     8                    24                41
2                    23                     0                 4
3                    12                    34                19
4                    35                    62                62
```

```
5                    39                      9                  0

  Census Block Group Code  County
1             60570001042  NEVADA
2              6057000104  NEVADA
3                    6057  NEVADA
4             60570004022  NEVADA
5             60570001052  NEVADA

[5 rows x 21 columns]
```

[17]:

```
#Checking if any null values
income.isnull().values.any()
```

[17]:

```
False
```

[18]:

```
#Checking if all the types make sense
income.dtypes
```

[18]:

```
id                           object
Geographic Area Name         object
Total                        object
Less than $10,000            object
\$10,000 to $14,999          object
\$15,000 to $19,999          object
\$20,000 to $24,999          object
\$25,000 to $29,999          object
\$30,000 to $34,999          object
\$35,000 to $39,999          object
\$40,000 to $44,999          object
\$45,000 to $49,999          object
\$50,000 to $59,999          object
\$60,000 to $74,999          object
\$75,000 to $99,999          object
\$100,000 to $124,999        object
\$125,000 to $149,999        object
\$150,000 to $199,999        object
\$200,000 or more            object
Census Block Group Code      object
County                       object
dtype: object
```

### 1.6.3 Race

[19]:

```
race = make_df('Race')
#fix columns
race = fix_cols(race)

#last 1 labels are LC of "Two or More Races"
labels_r = ['id', 'Geographic Area Name', 'Total', 'White alone', 'Black or␣
 ↪African American alone',
            'American Indian and Alaska Native alone', 'Asian alone', 'Native␣
 ↪Hawaiian and Other Pacific Islander alone',
            'Some other race alone', 'Two or more races', 'Two races including␣
 ↪Some other race',
            'Two races excluding Some other race, and three or more races']
race, moe_r = moe(race, labels_r)

race = cen_block_gc(race)
race = county_name(race)
race.head()
```

[19]:

```
                       id                              Geographic Area Name  \
1  1500000US060014007004  Block Group 4, Census Tract 4007, Alameda Coun…
2  1500000US060014419242  Block Group 2, Census Tract 4419.24, Alameda C…
3  1500000US060014507461  Block Group 1, Census Tract 4507.46, Alameda C…
4  1500000US060014372004  Block Group 4, Census Tract 4372, Alameda Coun…
5  1500000US060014381004  Block Group 4, Census Tract 4381, Alameda Coun…

  Total White alone Black or African American alone  \
1  1172         672                             296
2  1444         262                               0
3  2594        1557                              11
4  2371         978                             161
5  1112         634                             147

  American Indian and Alaska Native alone Asian alone  \
1                                      65          74
2                                       0        1030
3                                       5         847
4                                      39         907
5                                       0         203

  Native Hawaiian and Other Pacific Islander alone Some other race alone  \
1                                                0                     8
2                                                0                    50
3                                                0                    35
4                                                0                    44
5                                                0                    46
```

```
  Two or more races Two races including Some other race  \
1                57                                   0
2               102                                  41
3               139                                   0
4               242                                  81
5                82                                   0

  Two races excluding Some other race, and three or more races  \
1                                                 57
2                                                 61
3                                                139
4                                                161
5                                                 82

  Census Block Group Code   County
1             60014007004  ALAMEDA
2             60014419242  ALAMEDA
3              6001450746  ALAMEDA
4             60014372004  ALAMEDA
5             60014381004  ALAMEDA
```

[20]:

```
#Checking if any null values
race.isnull().values.any()
```

[20]:

```
False
```

[21]:

```
#Checking if all the types make sense
race.dtypes
```

[21]:

```
id                                                              object
Geographic Area Name                                            object
Total                                                           object
White alone                                                     object
Black or African American alone                                 object
American Indian and Alaska Native alone                         object
Asian alone                                                     object
Native Hawaiian and Other Pacific Islander alone                object
Some other race alone                                           object
Two or more races                                               object
Two races including Some other race                             object
Two races excluding Some other race, and three or more races    object
Census Block Group Code                                         object
County                                                          object
dtype: object
```

### 1.6.4 Commute

[22]:

```
commute = make_df('Commute')
#fix columns
commute = fix_cols(commute)

labels_c = ['id', 'Geographic Area Name', 'Total', 'Less than 5 minutes', '5 to␣
 ↪9 minutes', '10 to 14 minutes',
           '15 to 19 minutes', '20 to 24 minutes', '25 to 29 minutes', '30 to␣
 ↪34 minutes', '35 to 39 minutes',
           '40 to 44 minutes', '45 to 59 minutes', '60 to 89 minutes', '90 or␣
 ↪more minutes']
commute, moe_c = moe(commute, labels_c)

commute = cen_block_gc(commute)
commute = county_name(commute)
commute.head()
```

[22]:

```
                      id                             Geographic Area Name  \
1  1500000US060930008003  Block Group 3, Census Tract 8, Siskiyou County…
2  1500000US060930008001  Block Group 1, Census Tract 8, Siskiyou County…
3  1500000US060930011003  Block Group 3, Census Tract 11, Siskiyou Count…
4  1500000US060930011001  Block Group 1, Census Tract 11, Siskiyou Count…
5  1500000US060930009001  Block Group 1, Census Tract 9, Siskiyou County…

  Total Less than 5 minutes 5 to 9 minutes 10 to 14 minutes 15 to 19 minutes  \
1   207                   9             36               41                9
2   754                  37            108              123              137
3   363                  22             71               30               99
4   105                  20             17               14               32
5   203                 102             23               60                9

  20 to 24 minutes 25 to 29 minutes 30 to 34 minutes 35 to 39 minutes  \
1               10                0               28                0
2                5               19              126               36
3               52               11               21               19
4                4                6                6                0
5                4                0                5                0

  40 to 44 minutes 45 to 59 minutes 60 to 89 minutes 90 or more minutes  \
1               13               38                0                 23
2                6              143               11                  3
3                3               15               20                  0
4                0                6                0                  0
5                0                0                0                  0

  Census Block Group Code    County
```

```
1             60930008003  SISKIYOU
2                60930008  SISKIYOU
3             60930011003  SISKIYOU
4                    6093  SISKIYOU
5                60930009  SISKIYOU
```

[23]:
```
#Checking if any null values
commute.isnull().values.any()
```

[23]:
```
False
```

[24]:
```
#Checking if all the types make sense
commute.dtypes
```

[24]:
```
id                           object
Geographic Area Name         object
Total                        object
Less than 5 minutes          object
5 to 9 minutes               object
10 to 14 minutes             object
15 to 19 minutes             object
20 to 24 minutes             object
25 to 29 minutes             object
30 to 34 minutes             object
35 to 39 minutes             object
40 to 44 minutes             object
45 to 59 minutes             object
60 to 89 minutes             object
90 or more minutes           object
Census Block Group Code      object
County                       object
dtype: object
```

### 1.6.5 Final Merged Dataset

[25]:
```
merge1 = education.merge(income, on=['Census Block Group Code', 'County', 'id',␣
 ↪'Geographic Area Name'], how='outer').rename(columns={'Total_x':'Total␣
 ↪Education', 'Total_y':'Total Income'})
merge2 = merge1.merge(race, on=['Census Block Group Code', 'County', 'id',␣
 ↪'Geographic Area Name'], how='outer').rename(columns={'Total':'Total Race'})
merge3 = merge2.merge(commute, on=['Census Block Group Code', 'County', 'id',␣
 ↪'Geographic Area Name'], how='outer').rename(columns={'Total':'Total␣
 ↪Commute'})
df = merge3.merge(prop_ev, on=['Census Block Group Code', 'County'],␣
 ↪how='outer')
```

```
#show all columns
pd.set_option('display.max_columns', None)
df.head()
```

[25]:

```
                      id                               Geographic Area Name  \
0  1500000US060570001024  Block Group 4, Census Tract 1.02, Nevada Count…
1  1500000US060570012041  Block Group 1, Census Tract 12.04, Nevada Coun…
2  1500000US060570001051  Block Group 1, Census Tract 1.05, Nevada Count…
3  1500000US060570005011  Block Group 1, Census Tract 5.01, Nevada Count…
4  1500000US060570005015  Block Group 5, Census Tract 5.01, Nevada Count…

  Total Education No schooling completed Nursery school Kindergarten  \
0            1032                      0              0            0
1             637                      0              0            0
2             699                      0              0            0
3             764                      0              0            0
4             407                      0              0            0

  1st grade 2nd grade 3rd grade 4th grade 5th grade 6th grade 7th grade  \
0         0         0         0         0         0         0         0
1         0         0         0         0         0         0         0
2         0         0         0         0         0         0         0
3         0         0         0         0         0         0         0
4         0         0         0         0         0         0         0

  8th grade 9th grade 10th grade 11th grade 12th grade, no diploma  \
0         0         0          0          0                      0
1         0         0          0          0                     34
2         0         0          0          0                      9
3         0         0          0          0                     19
4         0         0         11          0                     50

  Regular high school diploma GED or alternative credential  \
0                         245                            42
1                          12                            27
2                         112                             0
3                         210                             0
4                          84                            12

  Some college, less than 1 year Some college, 1 or more years, no degree  \
0                            150                                      221
1                              0                                      147
2                             54                                      186
3                             31                                      175
4                             32                                       33

  Associate's degree Bachelor's degree Master's degree  \
0                111               186             61
1                 82               149            160
2                115               124             60
3                 47               151             55
4                  0                93             73

  Professional school degree Doctorate degree Census Block Group Code  County  \
0                         16                0             60570001024  NEVADA
1                          8               18              6057001204  NEVADA
2                         32                7                    6057  NEVADA
3                         53               23                    6057  NEVADA
4                         19                0                    6057  NEVADA

  Total Income Less than $10,000 \$10,000 to $14,999 \$15,000 to $19,999  \
0          578                37                   0                  18
1          348                15                  34                   0
2          338                 7                  48                   8
3          438                18                  17                  18
4          273                32                   0                   0

  \$20,000 to $24,999 \$25,000 to $29,999 \$30,000 to $34,999  \
0                  13                   0                   0
1                   0                   0                   0
2                  21                   0                   8
3                   0                  51                  19
4                  13                  11                  20

  \$35,000 to $39,999 \$40,000 to $44,999 \$45,000 to $49,999  \
0                  30                  46                   0
1                   0                   8                   0
2                   0                   8                  11
3                  43                  36                  35
4                  43                  50                  28

  \$50,000 to $59,999 \$60,000 to $74,999 \$75,000 to $99,999  \
0                  32                  83                  98
1                  39                   0                  57
2                  19                  32                  82
3                  38                  65                   0
4                  21                   0                  19

  \$100,000 to $124,999 \$125,000 to $149,999 \$150,000 to $199,999  \
0                   113                    74                     0
1                    66                    14                    59
2                    29                    12                    34
3                    27                    71                     0
4                     0                    25                     0
```

```
  \$200,000 or more Total Race White alone Black or African American alone  \
0                34       1476        1152                               0
1                56        836         822                               0
2                19        846         822                               0
3                 0        941         752                               0
4                11        541         489                               0

  American Indian and Alaska Native alone Asian alone  \
0                                       0          36
1                                       0           0
2                                      24           0
3                                     172          17
4                                      52           0

  Native Hawaiian and Other Pacific Islander alone Some other race alone  \
0                                                0                     0
1                                                0                     0
2                                                0                     0
3                                                0                     0
4                                                0                     0

  Two or more races Two races including Some other race  \
0               288                                  41
1                14                                   0
2                 0                                   0
3                 0                                   0
4                 0                                   0

  Two races excluding Some other race, and three or more races Total Commute  \
0                                                          247           513
1                                                           14           486
2                                                            0           283
3                                                            0           371
4                                                            0           205

  Less than 5 minutes 5 to 9 minutes 10 to 14 minutes 15 to 19 minutes  \
0                  24              0               61              127
1                  35            156               67              101
2                   7              0               43              128
3                  31            108               28              182
4                  11             60               49               50

  20 to 24 minutes 25 to 29 minutes 30 to 34 minutes 35 to 39 minutes  \
0              149               14               56                0
1               14               21                0               55
2               41               13                0                0
```

```
3                0                0               22                0
4                0                0                0               11

  40 to 44 minutes 45 to 59 minutes 60 to 89 minutes 90 or more minutes  \
0               19                0               28                 35
1                0               23                0                 14
2                7               14               30                  0
3                0                0                0                  0
4               12                0               12                  0

   EV Population  Vehicle Population  Proportion EV
0            1.0               526.0       0.001901
1            NaN                 NaN            NaN
2            NaN                 NaN            NaN
3            NaN                 NaN            NaN
4            NaN                 NaN            NaN
```

That's odd, let's do some data cleaning to see why we have NaN values. I'm guessing that some of the county/census block group data don't match – let's rearrange the df so that we have those features first.

[26]:
```
#checking if we have null values
df.isnull().values.any()
```

[26]:
```
True
```

[27]:
```
#rearrange column order

col_names = list(df.columns.values)
del col_names[col_names.index('id'):col_names.index('id')+2]
del col_names[col_names.index('Census Block Group Code'):col_names.
 ↪index('Census Block Group Code')+2]

features = df[col_names]
groupedby = df[['Census Block Group Code', 'County', 'id', 'Geographic Area␣
 ↪Name']]
df = pd.concat([groupedby, features], axis=1)
```

[28]:
```
acs_null = df[['id']].isnull()
acs_null_index = acs_null.loc[acs_null['id'] == True].index
drop_acs_null = df.drop(acs_null_index)

fleet_null = drop_acs_null[['EV Population']].isnull()
fleet_null_index = fleet_null.loc[fleet_null['EV Population'] == True].index
drop_fleet_null = drop_acs_null.drop(fleet_null_index)

df = drop_fleet_null
```

```
df.isnull().values.any()
```

[28]: False

[29]:

```
#making sure all relevant observations are ints
num_cols = ['Total Education', 'No schooling completed', 'Nursery school',
       'Kindergarten', '1st grade', '2nd grade', '3rd grade', '4th grade',
       '5th grade', '6th grade', '7th grade', '8th grade', '9th grade',
       '10th grade', '11th grade', '12th grade, no diploma',
       'Regular high school diploma', 'GED or alternative credential',
       'Some college, less than 1 year',
       'Some college, 1 or more years, no degree', "Associate's degree",
       "Bachelor's degree", "Master's degree", 'Professional school degree',
       'Doctorate degree', 'Total Income', 'Less than $10,000',
       '\$10,000 to $14,999', '\$15,000 to $19,999', '\$20,000 to $24,999',
       '\$25,000 to $29,999', '\$30,000 to $34,999', '\$35,000 to $39,999',
       '\$40,000 to $44,999', '\$45,000 to $49,999', '\$50,000 to $59,999',
       '\$60,000 to $74,999', '\$75,000 to $99,999', '\$100,000 to $124,999',
       '\$125,000 to $149,999', '\$150,000 to $199,999', '\$200,000 or more',
       'Total Race', 'White alone', 'Black or African American alone',
       'American Indian and Alaska Native alone', 'Asian alone',
       'Native Hawaiian and Other Pacific Islander alone',
       'Some other race alone', 'Two or more races',
       'Two races including Some other race',
       'Two races excluding Some other race, and three or more races',
       'Total Commute', 'Less than 5 minutes', '5 to 9 minutes',
       '10 to 14 minutes', '15 to 19 minutes', '20 to 24 minutes',
       '25 to 29 minutes', '30 to 34 minutes', '35 to 39 minutes',
       '40 to 44 minutes', '45 to 59 minutes', '60 to 89 minutes',
       '90 or more minutes', 'EV Population', 'Vehicle Population',
       'Proportion EV']

for i in num_cols:
    df[i] = df[i].astype('float')
```

[30]:

```
#yay! no null values, let's see our final output
df = df.reset_index()
df
```

[30]:

```
       index Census Block Group Code  County                        id  \
0          0             60570001024  NEVADA  1500000US060570001024
1          5             60570006002  NEVADA  1500000US060570006002
2          6             60570005022  NEVADA  1500000US060570005022
3          7             60570004022  NEVADA  1500000US060570004022
4          8             60570001023  NEVADA  1500000US060570001023
…      …                         …       …                      …
14752  24165             60590992172  ORANGE  1500000US060590992172
```

```
14753  24167             60590762052  ORANGE  1500000US060590762052
14754  24168             60590636033  ORANGE  1500000US060590636033
14755  24171             60590631023  ORANGE  1500000US060590631023
14756  24172             60590637022  ORANGE  1500000US060590637022

                                    Geographic Area Name  Total Education  \
0      Block Group 4, Census Tract 1.02, Nevada Count…           1032.0
1      Block Group 2, Census Tract 6, Nevada County, …            765.0
2      Block Group 2, Census Tract 5.02, Nevada Count…            982.0
3      Block Group 2, Census Tract 4.02, Nevada Count…           1476.0
4      Block Group 3, Census Tract 1.02, Nevada Count…           1887.0
…                                                  …              …
14752  Block Group 2, Census Tract 992.17, Orange Cou…            859.0
14753  Block Group 2, Census Tract 762.05, Orange Cou…            750.0
14754  Block Group 3, Census Tract 636.03, Orange Cou…           1470.0
14755  Block Group 3, Census Tract 631.02, Orange Cou…            888.0
14756  Block Group 2, Census Tract 637.02, Orange Cou…            528.0

       No schooling completed  Nursery school  Kindergarten  1st grade  \
0                         0.0             0.0           0.0        0.0
1                         0.0             0.0           0.0        0.0
2                         0.0             0.0           0.0        0.0
3                         2.0             0.0           0.0        0.0
4                         0.0             0.0           0.0        0.0
…                       …             …           …        …
14752                     0.0             0.0           0.0        0.0
14753                     0.0             0.0           0.0        0.0
14754                     0.0             0.0           0.0        0.0
14755                    14.0             0.0           0.0        0.0
14756                     0.0             0.0           0.0        0.0

       2nd grade  3rd grade  4th grade  5th grade  6th grade  7th grade  \
0            0.0        0.0        0.0        0.0        0.0        0.0
1            0.0        0.0        0.0        0.0        0.0        0.0
2            0.0        0.0        0.0        0.0       58.0        0.0
3            0.0        0.0        0.0        0.0        0.0        0.0
4            0.0        0.0        0.0        0.0       43.0        0.0
…          …        …        …        …        …        …
14752        0.0        0.0        0.0        0.0        0.0        0.0
14753        0.0        0.0        0.0       11.0        8.0        0.0
14754        5.0        0.0        0.0        0.0        0.0        0.0
14755        0.0        0.0        0.0        0.0        0.0        0.0
14756        0.0        0.0        0.0        0.0        0.0        0.0

       8th grade  9th grade  10th grade  11th grade  12th grade, no diploma  \
0            0.0        0.0         0.0         0.0                     0.0
1            0.0       28.0         9.0         0.0                    11.0
```

```
2           52.0        0.0         0.0        15.0                     0.0
3            0.0        0.0         0.0         0.0                    16.0
4            0.0       20.0         0.0        51.0                    76.0
…          …        …         …         …                     …
14752        0.0        0.0         0.0         0.0                     4.0
14753        0.0       22.0         6.0         0.0                     0.0
14754        0.0        6.0         7.0        10.0                     0.0
14755        0.0        0.0         0.0         0.0                     0.0
14756        0.0        8.0         0.0         0.0                    18.0

       Regular high school diploma  GED or alternative credential  \
0                            245.0                           42.0
1                            163.0                           14.0
2                            258.0                           55.0
3                            241.0                           70.0
4                            240.0                           42.0
…                            …                            …
14752                         76.0                           20.0
14753                        140.0                           15.0
14754                        147.0                           23.0
14755                         58.0                           21.0
14756                         91.0                            0.0

       Some college, less than 1 year  \
0                               150.0
1                               139.0
2                               134.0
3                                62.0
4                               148.0
…                               …
14752                            40.0
14753                           110.0
14754                            51.0
14755                            19.0
14756                            54.0

       Some college, 1 or more years, no degree  Associate's degree  \
0                                         221.0               111.0
1                                         192.0                86.0
2                                         144.0               111.0
3                                         435.0               182.0
4                                         311.0               403.0
…                                         …                 …
14752                                     160.0                70.0
14753                                     107.0                99.0
14754                                     205.0                89.0
14755                                     220.0                86.0
```

```
14756                                      53.0                72.0

       Bachelor's degree  Master's degree  Professional school degree  \
0                  186.0             61.0                        16.0   
1                   80.0             34.0                         4.0   
2                   85.0             70.0                         0.0   
3                  343.0             62.0                        45.0   
4                  352.0            171.0                        30.0   
…                  …              …                         …
14752              333.0            111.0                        28.0   
14753              150.0             41.0                        14.0   
14754              653.0            156.0                        68.0   
14755              296.0            135.0                        21.0   
14756              112.0             92.0                        28.0   

       Doctorate degree  Total Income  Less than $10,000  \$10,000 to $14,999  \
0                   0.0         578.0               37.0                  0.0   
1                   5.0         475.0              101.0                 28.0   
2                   0.0         582.0               33.0                 14.0   
3                  18.0         811.0               62.0                  0.0   
4                   0.0         979.0               36.0                  0.0   
…                 …           …                …                  …
14752              17.0         453.0                4.0                  0.0   
14753              27.0         418.0               23.0                  8.0   
14754              50.0         957.0               72.0                 14.0   
14755              18.0         559.0                1.0                101.0   
14756               0.0         321.0                6.0                105.0   

       \$15,000 to $19,999  \$20,000 to $24,999  \$25,000 to $29,999  \
0                     18.0                 13.0                  0.0   
1                     14.0                101.0                 39.0   
2                     66.0                 91.0                 75.0   
3                     30.0                 33.0                 55.0   
4                     18.0                 47.0                 63.0   
…                    …                  …                  …
14752                 16.0                  8.0                 15.0   
14753                 15.0                 42.0                  7.0   
14754                 28.0                 52.0                 27.0   
14755                 24.0                 35.0                 18.0   
14756                 11.0                 54.0                  0.0   

       \$30,000 to $34,999  \$35,000 to $39,999  \$40,000 to $44,999  \
0                      0.0                 30.0                 46.0   
1                     14.0                 15.0                 13.0   
2                     15.0                 15.0                 53.0   
3                     22.0                 14.0                 97.0   
4                     14.0                 16.0                 47.0   
```

```
…                    …                  …                  …
14752                 12.0                  6.0                  0.0
14753                 26.0                 27.0                  0.0
14754                 49.0                 19.0                 61.0
14755                 13.0                  0.0                 21.0
14756                  0.0                 30.0                  0.0

       \$45,000 to $49,999  \$50,000 to $59,999  \$60,000 to $74,999  \
0                      0.0                 32.0                 83.0
1                      0.0                 59.0                 22.0
2                     63.0                 44.0                 64.0
3                      0.0                 41.0                208.0
4                    109.0                 48.0                 14.0
…                    …                  …                  …
14752                 25.0                 24.0                 23.0
14753                 13.0                 17.0                 11.0
14754                 12.0                104.0                 59.0
14755                  0.0                 53.0                 15.0
14756                  0.0                 30.0                 29.0

       \$75,000 to $99,999  \$100,000 to $124,999  \$125,000 to $149,999  \
0                     98.0                  113.0                   74.0
1                     44.0                   15.0                    0.0
2                     17.0                    0.0                   15.0
3                     73.0                   17.0                   35.0
4                    208.0                   78.0                   93.0
…                    …                    …                    …
14752                 90.0                   66.0                   19.0
14753                 45.0                   72.0                   31.0
14754                160.0                   97.0                   66.0
14755                 18.0                   68.0                   57.0
14756                 17.0                    0.0                    0.0

       \$150,000 to $199,999  \$200,000 or more  Total Race  White alone  \
0                        0.0               34.0      1476.0       1152.0
1                       10.0                0.0      1104.0        947.0
2                        0.0               17.0      1329.0       1155.0
3                       62.0               62.0      1844.0       1612.0
4                       97.0               91.0      2373.0       2289.0
…                      …                …         …          …
14752                   77.0               68.0      1245.0       1099.0
14753                   48.0               33.0      1069.0        835.0
14754                   58.0               79.0      1668.0       1464.0
14755                   27.0              108.0      1145.0       1016.0
14756                   21.0               18.0       605.0        464.0

       Black or African American alone  \
0                                  0.0
1                                 24.0
2                                  0.0
3                                210.0
4                                  0.0
…                                 …
14752                             25.0
14753                              0.0
14754                              0.0
14755                              0.0
14756                              0.0

       American Indian and Alaska Native alone  Asian alone  \
0                                          0.0         36.0
1                                         13.0         10.0
2                                          0.0         31.0
3                                          0.0          2.0
4                                          0.0          0.0
…                                        …          …
14752                                      0.0         76.0
14753                                     23.0        171.0
14754                                      7.0        130.0
14755                                      0.0         15.0
14756                                      0.0         55.0

       Native Hawaiian and Other Pacific Islander alone  \
0                                                   0.0
1                                                   0.0
2                                                   0.0
3                                                   0.0
4                                                   0.0
…                                                 …
14752                                               0.0
14753                                               0.0
14754                                               0.0
14755                                               0.0
14756                                               0.0

       Some other race alone  Two or more races  \
0                        0.0              288.0
1                       13.0               97.0
2                        0.0              143.0
3                        0.0               20.0
4                       25.0               59.0
…                      …                …
14752                   10.0               35.0
14753                   18.0               22.0
```

```
14754                    5.0               62.0
14755                  101.0               13.0
14756                   86.0                0.0

       Two races including Some other race  \
0                                     41.0
1                                      0.0
2                                      0.0
3                                      0.0
4                                      0.0
…                                    …
14752                                  5.0
14753                                 22.0
14754                                  0.0
14755                                  0.0
14756                                  0.0

       Two races excluding Some other race, and three or more races  \
0                                                 247.0
1                                                  97.0
2                                                 143.0
3                                                  20.0
4                                                  59.0
…                                                  …
14752                                              30.0
14753                                               0.0
14754                                              62.0
14755                                              13.0
14756                                               0.0

       Total Commute  Less than 5 minutes  5 to 9 minutes  10 to 14 minutes  \
0              513.0                 24.0             0.0              61.0
1              543.0                 11.0           292.0             149.0
2              344.0                 16.0           136.0              78.0
3              827.0                110.0            60.0              66.0
4              942.0                 47.0             0.0             206.0
…              …                   …             …               …
14752          569.0                  7.0            89.0              35.0
14753          469.0                  7.0            72.0              90.0
14754          885.0                  0.0            24.0             177.0
14755          648.0                 31.0            63.0             182.0
14756          296.0                  0.0            58.0              32.0

       15 to 19 minutes  20 to 24 minutes  25 to 29 minutes  30 to 34 minutes  \
0                 127.0             149.0              14.0              56.0
1                  35.0               0.0               0.0              12.0
2                  31.0               0.0              32.0              51.0
```

```
3                  94.0              91.0              73.0              48.0
4                 239.0             135.0              18.0              55.0
…                 …                 …                 …                 …
14752              96.0             125.0              26.0              51.0
14753              70.0              82.0              33.0              67.0
14754             168.0             151.0              40.0             191.0
14755              39.0              21.0              88.0              86.0
14756              28.0              90.0              20.0              46.0

       35 to 39 minutes  40 to 44 minutes  45 to 59 minutes  60 to 89 minutes  \
0                   0.0              19.0               0.0              28.0
1                   0.0               0.0               0.0              44.0
2                   0.0               0.0               0.0               0.0
3                  33.0              64.0             128.0               0.0
4                  50.0              25.0              39.0              44.0
…                 …                 …                 …                 …
14752              11.0              38.0              34.0              41.0
14753               9.0               7.0              17.0              15.0
14754              44.0               0.0              43.0              47.0
14755              10.0              13.0              77.0              38.0
14756               0.0              22.0               0.0               0.0

       90 or more minutes  EV Population  Vehicle Population  Proportion EV
0                    35.0            1.0               526.0       0.001901
1                     0.0            1.0               258.0       0.003876
2                     0.0            5.0               231.0       0.021645
3                    60.0            1.0               243.0       0.004115
4                    84.0            3.0               724.0       0.004144
…                    …             …                 …            …
14752                16.0            7.0               549.0       0.012750
14753                 0.0            7.0               618.0       0.011327
14754                 0.0            6.0               636.0       0.009434
14755                 0.0           11.0               556.0       0.019784
14756                 0.0            1.0               156.0       0.006410

[14757 rows x 73 columns]
```

## 1.7 Data Summary and Exploratory Data Analysis (10 points)

We ended up grouping our features into more interpretable categories. We did this because some of the categories, such as education, had way too many different categories. In education specifically, there were over 14 categories including Kindergarten, First Grade, Second Grade, etc all the way up into 12th grade, and then into the amount of college or professional school completed as well. The title of the ACS Data Set that we used was "Educational Attainment for the Population 25 Years and Older," so we ended up grouping the education categories into "More College Completed" which only included those data points for Master's degree, Professional school degree, and Doctorate degree, "College Completed" which included all of the categories from "More College Completed" plus Associates and Bachelors degrees, and finally we had “High School Completed,” which included the columns in “More College Completed” and “College Completed” as well as the high school degree related features.

After exploring the data, we decided to represent our features by proportions of people per county rather than total amount since each county has a different population and different numbers of representation in the data. Furthermore, we wanted to prevent skewing the data if one county had more people reported than the other.

Side note: in order to do this, some observations were dropped if that specific census block group didn’t have any data related to the ACS categories (Education, Race, Income, and Commute Time) and listed as 0 – otherwise we couldn’t calculate their representation in proportions).

### 1.7.1 Making Proportions

[31]:
```
#drop columns with no data
df.drop([238, 1595, 2357, 6116, 6179, 7387, 8717, 9057, 11561, 10029],␣
 ↪inplace=True)
```

[32]:
```
#grouping features by education
hs_completed = ['Regular high school diploma', 'GED or alternative credential',
       'Some college, less than 1 year',
       'Some college, 1 or more years, no degree', "Associate's degree",
       "Bachelor's degree", "Master's degree", 'Professional school degree',␣
 ↪'Doctorate degree']
college_completed = ["Associate's degree", "Bachelor's degree", "Master's␣
 ↪degree", 'Professional school degree',
       'Doctorate degree']
more_college_completed = ["Master's degree", 'Professional school degree',
       'Doctorate degree']

df['More College Completed'] = df["Master's degree"] + df['Professional school␣
 ↪degree'] + df['Doctorate degree']
df['College Completed'] = df['More College Completed'] + df["Bachelor's␣
 ↪degree"] + df["Associate's degree"]
df['High School Completed'] = df['College Completed'] + df['Some college, 1 or␣
 ↪more years, no degree'] + df['Some college, less than 1 year'] + df['GED or␣
 ↪alternative credential'] + df['Regular high school diploma']

education_columns = ['No schooling completed',
       'Nursery school', 'Kindergarten', '1st grade', '2nd grade', '3rd grade',
       '4th grade', '5th grade', '6th grade', '7th grade', '8th grade',
       '9th grade', '10th grade', '11th grade', '12th grade, no diploma',
       'Regular high school diploma', 'GED or alternative credential',
       'Some college, less than 1 year',
       'Some college, 1 or more years, no degree', "Associate's degree",
       "Bachelor's degree", "Master's degree", 'Professional school degree',
       'Doctorate degree']
```

```
df.drop(columns=education_columns, inplace=True)
```

[33]:

```
#grouping features by income
df['Less than $10k'] = df['Less than $10,000']
df['\$10,000 to $24,999'] = df['\$10,000 to $14,999'] + df['\$15,000 to␣
 ↪$19,999'] + df['\$20,000 to $24,999']
df['\$25,000 to $49,999'] = df['\$25,000 to $29,999'] + df['\$30,000 to␣
 ↪$34,999'] + df['\$35,000 to $39,999'] + df['\$40,000 to $44,999'] +␣
 ↪df['\$45,000 to $49,999']
df['\$50,000 to $99,999'] = df['\$50,000 to $59,999'] + df['\$60,000 to␣
 ↪$74,999'] + df['\$75,000 to $99,999']
df['\$100,000 to $199,999'] = df['\$100,000 to $124,999'] + df['\$125,000 to␣
 ↪$149,999'] + df['\$150,000 to $199,999']
df['$200,000 or more'] = df['\$200,000 or more']

income_columns = ['Less than $10,000',
       '\$10,000 to $14,999', '\$15,000 to $19,999', '\$20,000 to $24,999',
       '\$25,000 to $29,999', '\$30,000 to $34,999', '\$35,000 to $39,999',
       '\$40,000 to $44,999', '\$45,000 to $49,999', '\$50,000 to $59,999',
       '\$60,000 to $74,999', '\$75,000 to $99,999', '\$100,000 to $124,999',
       '\$125,000 to $149,999', '\$150,000 to $199,999', '\$200,000 or more']
df.drop(columns=income_columns, inplace=True)
```

[34]:

```
#grouping features commute time
df['Less than 10 Minutes'] = df['Less than 5 minutes'] + df['5 to 9 minutes']
df['10 to 19 Minutes'] = df['10 to 14 minutes'] + df['15 to 19 minutes']
df['20 to 29 Minutes'] = df['20 to 24 minutes'] + df['25 to 29 minutes']
df['30 to 44 Minutes'] = df['30 to 34 minutes'] + df['35 to 39 minutes'] +␣
 ↪df['40 to 44 minutes']
df['45 to 59 Minutes'] = df['45 to 59 minutes']
df['60 to 89 Minutes'] = df['60 to 89 minutes']
df['90 or more Minutes'] = df['90 or more minutes']

commute_columns = ['Less than 5 minutes', '5 to 9 minutes',
       '10 to 14 minutes', '15 to 19 minutes', '20 to 24 minutes',
       '25 to 29 minutes', '30 to 34 minutes', '35 to 39 minutes',
       '40 to 44 minutes', '45 to 59 minutes', '60 to 89 minutes',
       '90 or more minutes']
df.drop(columns=commute_columns, inplace=True)
```

[35]:

```
#grouping by race
df.drop(columns=['Two or more races', 'Two races including Some other race',␣
 ↪'Two races excluding Some other race, and three or more races'],␣
 ↪inplace=True)
```

[36]:
```
education_columns = ['More College Completed',
       'College Completed', 'High School Completed']
income_columns = ['Less than $10k',
       '\$10,000 to $24,999', '\$25,000 to $49,999', '\$50,000 to $99,999',␣
 ↪'\$100,000 to $199,999',
       '$200,000 or more']
race_columns = ['White alone', 'Black or African American alone',
       'American Indian and Alaska Native alone', 'Asian alone',
       'Native Hawaiian and Other Pacific Islander alone',
       'Some other race alone']
commute_columns = ['Less than 10 Minutes', '10 to 19 Minutes',
       '20 to 29 Minutes', '30 to 44 Minutes', '45 to 59 Minutes',
       '60 to 89 Minutes', '90 or more Minutes']
```

[37]:
```
for x in education_columns:
    df[x + ' Proportion'] = df[x] / df['Total Education']
for x in income_columns:
    df[x + ' Proportion'] = df[x] / df['Total Income']
for x in race_columns:
    df[x + ' Proportion'] = df[x] / df['Total Race']
for x in commute_columns:
    df[x + ' Proportion'] = df[x] / df['Total Commute']
```

[38]:
```
df.head()
```

[38]:
```
   index Census Block Group Code  County                    id  \
0      0             60570001024  NEVADA  1500000US060570001024
1      5             60570006002  NEVADA  1500000US060570006002
2      6             60570005022  NEVADA  1500000US060570005022
3      7             60570004022  NEVADA  1500000US060570004022
4      8             60570001023  NEVADA  1500000US060570001023

                              Geographic Area Name  Total Education  \
0  Block Group 4, Census Tract 1.02, Nevada Count…           1032.0
1  Block Group 2, Census Tract 6, Nevada County, …            765.0
2  Block Group 2, Census Tract 5.02, Nevada Count…            982.0
3  Block Group 2, Census Tract 4.02, Nevada Count…           1476.0
4  Block Group 3, Census Tract 1.02, Nevada Count…           1887.0

   Total Income  Total Race  White alone  Black or African American alone  \
0         578.0      1476.0       1152.0                              0.0
1         475.0      1104.0        947.0                             24.0
2         582.0      1329.0       1155.0                              0.0
3         811.0      1844.0       1612.0                            210.0
4         979.0      2373.0       2289.0                              0.0

   American Indian and Alaska Native alone  Asian alone  \
0                                      0.0         36.0
1                                     13.0         10.0
2                                      0.0         31.0
3                                      0.0          2.0
4                                      0.0          0.0

   Native Hawaiian and Other Pacific Islander alone  Some other race alone  \
0                                               0.0                    0.0
1                                               0.0                   13.0
2                                               0.0                    0.0
3                                               0.0                    0.0
4                                               0.0                   25.0

   Total Commute  EV Population  Vehicle Population  Proportion EV  \
0          513.0            1.0               526.0       0.001901
1          543.0            1.0               258.0       0.003876
2          344.0            5.0               231.0       0.021645
3          827.0            1.0               243.0       0.004115
4          942.0            3.0               724.0       0.004144

   More College Completed  College Completed  High School Completed  \
0                    77.0              374.0                 1032.0
1                    43.0              209.0                  717.0
2                    70.0              266.0                  857.0
3                   125.0              650.0                 1458.0
4                   201.0              956.0                 1697.0

   Less than $10k  \$10,000 to $24,999  \$25,000 to $49,999  \
0            37.0                 31.0                 76.0
1           101.0                143.0                 81.0
2            33.0                171.0                221.0
3            62.0                 63.0                188.0
4            36.0                 65.0                249.0

   \$50,000 to $99,999  \$100,000 to $199,999  $200,000 or more  \
0                213.0                  187.0              34.0
1                125.0                   25.0               0.0
2                125.0                   15.0              17.0
3                322.0                  114.0              62.0
4                270.0                  268.0              91.0

   Less than 10 Minutes  10 to 19 Minutes  20 to 29 Minutes  30 to 44 Minutes  \
0                  24.0             188.0             163.0              75.0
1                 303.0             184.0               0.0              12.0
2                 152.0             109.0              32.0              51.0
3                 170.0             160.0             164.0             145.0
4                  47.0             445.0             153.0             130.0
```

```
   45 to 59 Minutes  60 to 89 Minutes  90 or more Minutes  \
0               0.0              28.0                35.0   
1               0.0              44.0                 0.0   
2               0.0               0.0                 0.0   
3             128.0               0.0                60.0   
4              39.0              44.0                84.0   

   More College Completed Proportion  College Completed Proportion  \
0                           0.074612                      0.362403   
1                           0.056209                      0.273203   
2                           0.071283                      0.270876   
3                           0.084688                      0.440379   
4                           0.106518                      0.506624   

   High School Completed Proportion  Less than $10k Proportion  \
0                          1.000000                   0.064014   
1                          0.937255                   0.212632   
2                          0.872709                   0.056701   
3                          0.987805                   0.076449   
4                          0.899311                   0.036772   

   \$10,000 to $24,999 Proportion  \$25,000 to $49,999 Proportion  \
0                        0.053633                        0.131488   
1                        0.301053                        0.170526   
2                        0.293814                        0.379725   
3                        0.077682                        0.231813   
4                        0.066394                        0.254341   

   \$50,000 to $99,999 Proportion  \$100,000 to $199,999 Proportion  \
0                        0.368512                          0.323529   
1                        0.263158                          0.052632   
2                        0.214777                          0.025773   
3                        0.397041                          0.140567   
4                        0.275792                          0.273749   

   $200,000 or more Proportion  White alone Proportion  \
0                     0.058824                0.780488   
1                     0.000000                0.857790   
2                     0.029210                0.869074   
3                     0.076449                0.874187   
4                     0.092952                0.964602   

   Black or African American alone Proportion  \
0                                    0.000000   
1                                    0.021739   
2                                    0.000000   
```

```
3                                    0.113883
4                                    0.000000

   American Indian and Alaska Native alone Proportion  Asian alone Proportion  \
0                                            0.000000                0.024390
1                                            0.011775                0.009058
2                                            0.000000                0.023326
3                                            0.000000                0.001085
4                                            0.000000                0.000000

   Native Hawaiian and Other Pacific Islander alone Proportion  \
0                                                0.0
1                                                0.0
2                                                0.0
3                                                0.0
4                                                0.0

   Some other race alone Proportion  Less than 10 Minutes Proportion  \
0                          0.000000                         0.046784
1                          0.011775                         0.558011
2                          0.000000                         0.441860
3                          0.000000                         0.205562
4                          0.010535                         0.049894

   10 to 19 Minutes Proportion  20 to 29 Minutes Proportion  \
0                     0.366472                     0.317739
1                     0.338858                     0.000000
2                     0.316860                     0.093023
3                     0.193470                     0.198307
4                     0.472399                     0.162420

   30 to 44 Minutes Proportion  45 to 59 Minutes Proportion  \
0                     0.146199                     0.000000
1                     0.022099                     0.000000
2                     0.148256                     0.000000
3                     0.175333                     0.154776
4                     0.138004                     0.041401

   60 to 89 Minutes Proportion  90 or more Minutes Proportion
0                     0.054581                       0.068226
1                     0.081031                       0.000000
2                     0.000000                       0.000000
3                     0.000000                       0.072551
4                     0.046709                       0.089172
```

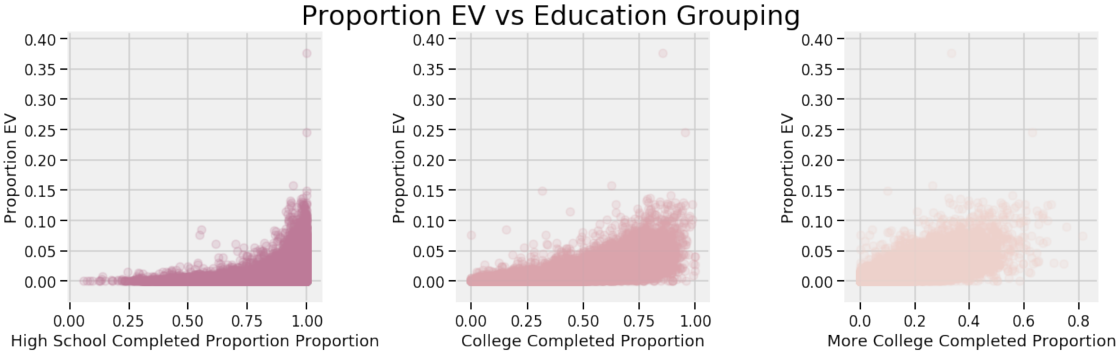
### 1.7.2 EDA

### Education

[45]:

```
#education

y = df["Proportion EV"]
plt.figure(figsize=(18,6))
color =  sns.cubehelix_palette(6)

plt.subplot(1, 3, 3)
plt.scatter(df['More College Completed Proportion'], y, c=color[0], alpha=0.2)
plt.xlabel("More College Completed Proportion")
plt.ylabel("Proportion EV")

plt.subplot(1,3,2)
plt.scatter(df['College Completed Proportion'], y, c=color[1], alpha=0.2)
plt.xlabel('College Completed Proportion')
plt.ylabel('Proportion EV')

plt.subplot(1,3,1)
plt.scatter(df['High School Completed Proportion'], y, c=color[2], alpha=0.2)
plt.xlabel('High School Completed Proportion Proportion')
plt.ylabel('Proportion EV')

plt.tight_layout(pad = 4)

plt.suptitle('Proportion EV vs Education Grouping', fontsize = 30);
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
```



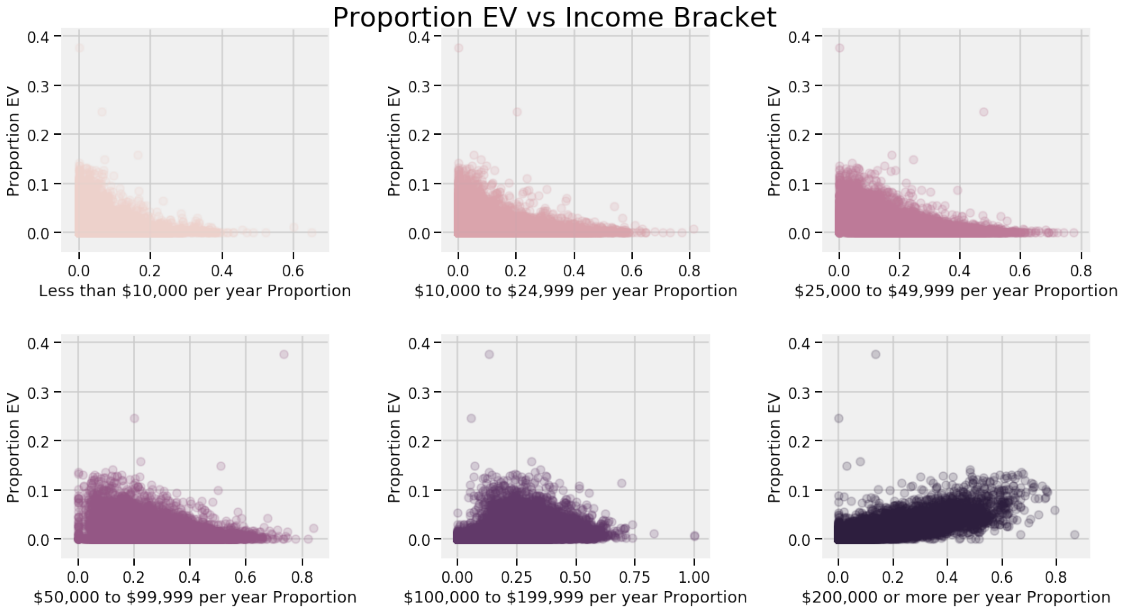
We can see that in areas where a small proportion of the population has completed high school, they tend to have lower levels of EV adoption. Although this holds true for college completion as well, areas with low levels of college completion still have higher levels of EV adoption relative to areas with low levels of high school completion. We can see the same trend hold true with regards to the proportion of people that have completed education beyond college. Overall, we can see that more educated populations tend to have higher levels of EV adoption.

**Income**

[46]:

```
#income

y = df["Proportion EV"]
plt.figure(figsize=(18,10))
color = sns.cubehelix_palette(6)

plt.subplot(2, 3, 1)
plt.scatter(df['Less than $10k Proportion'], y, c=color[0], alpha=0.2)
plt.xlabel("Less than $10,000 per year Proportion")
plt.ylabel("Proportion EV")

plt.subplot(2,3,2)
plt.scatter(df['\$10,000 to $24,999 Proportion'], y, c=color[1], alpha=0.2)
plt.xlabel('\$10,000 to $24,999 per year Proportion')
plt.ylabel('Proportion EV')

plt.subplot(2,3,3)
plt.scatter(df['\$25,000 to $49,999 Proportion'], y, c=color[2], alpha=0.2)
plt.xlabel('\$25,000 to $49,999 per year Proportion')
plt.ylabel('Proportion EV')

plt.subplot(2,3,4)
plt.scatter(df['\$50,000 to $99,999 Proportion'], y, c=color[3], alpha=0.2)
plt.xlabel('\$50,000 to $99,999 per year Proportion')
plt.ylabel('Proportion EV')
plt.subplot(2,3,5)
plt.scatter(df['\$100,000 to $199,999 Proportion'], y, c=color[4], alpha=0.2)
plt.xlabel('\$100,000 to $199,999 per year Proportion')
plt.ylabel('Proportion EV')

plt.subplot(2,3,6)
plt.scatter(df['$200,000 or more Proportion'], y, c=color[5], alpha=0.2)
plt.xlabel('\$200,000 or more per year Proportion')
plt.ylabel('Proportion EV')

plt.tight_layout(pad = 4)

plt.suptitle('Proportion EV vs Income Bracket', fontsize = 30);
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
```



We can see that for all income brackets between \$10k and \$100k annually, the census blocks which have a small proportion of income falling in this range tend to have higher levels of EV adoption. The trends are less clear in the \$100-200k income bracket range. Looking at the $200k+ range, we can see that census blocks wear a larger proportion of the population earns over \$200k tend to have a higher proportion of EVs.

These trends make sense given the fact that EVs have historically been more expeensive than gas vehicles. In addition, most households are unable to have an EV as their only car, due to concerns about taking roadtrips or range anxiety, thus EVs are more likely to be the second car a family owns, which makes it more likely that households who buy EVs are of a higher income, since they can afford to buy multiple cars.

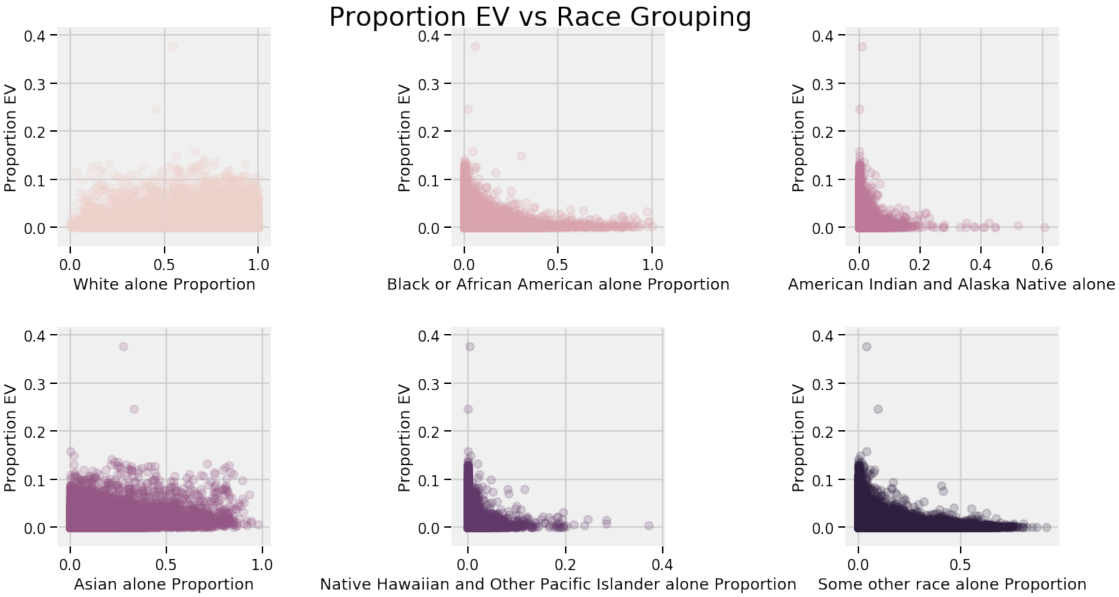
**Race**

[42]:
```
#race

y = df["Proportion EV"]
plt.figure(figsize=(18,10))
color =  sns.cubehelix_palette(6)

plt.subplot(2, 3, 1)
plt.scatter(df['White alone Proportion'], y, c=color[0], alpha=0.2)
plt.xlabel("White alone Proportion")
plt.ylabel("Proportion EV")

plt.subplot(2,3,2)
plt.scatter(df['Black or African American alone Proportion'], y, c=color[1],␣
 ↪alpha=0.2)
```

```
plt.xlabel('Black or African American alone Proportion')
plt.ylabel('Proportion EV')

plt.subplot(2,3,3)
plt.scatter(df['American Indian and Alaska Native alone Proportion'], y,␣
 ↪c=color[2], alpha=0.2)
plt.xlabel('American Indian and Alaska Native alone')
plt.ylabel('Proportion EV')

plt.subplot(2,3,4)
plt.scatter(df['Asian alone Proportion'], y, c=color[3], alpha=0.2)
plt.xlabel('Asian alone Proportion')
plt.ylabel('Proportion EV')

plt.subplot(2,3,5)
plt.scatter(df['Native Hawaiian and Other Pacific Islander alone Proportion'],␣
 ↪y, c=color[4], alpha=0.2)
plt.xlabel('Native Hawaiian and Other Pacific Islander alone Proportion')
plt.ylabel('Proportion EV')

plt.subplot(2,3,6)
plt.scatter(df['Some other race alone Proportion'], y, c=color[5], alpha=0.2)
plt.xlabel('Some other race alone Proportion')
plt.ylabel('Proportion EV')


plt.tight_layout(pad = 4)

plt.suptitle('Proportion EV vs Race Grouping', fontsize = 30);
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
```

```
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
```



We can see there appear to be some interesting relationships between the makeup of race in a census block group and its adoption of EVs. We can see that census block groups with large black, native Hawaiian, or American Indian / Alaskan native populations tend to have the lowest levels of EV adoptions. This makes sense given the impact that systemic racism has had on these communities and their earnings power (we have observed that income is positively correlated with EV adoption.

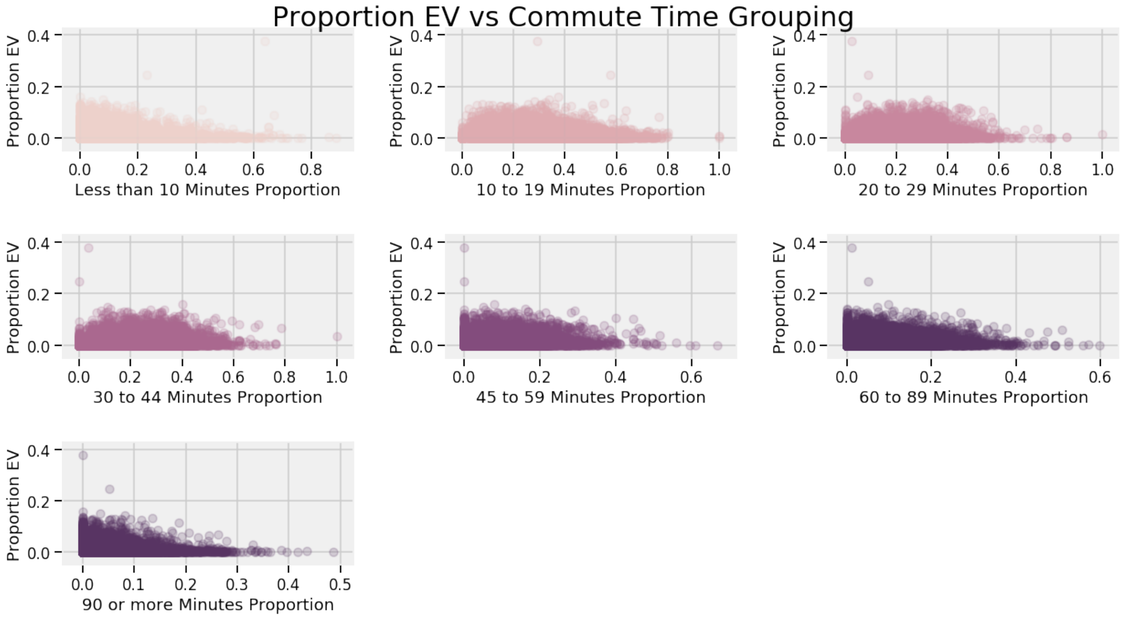
We were surprised to see that populations with a higher share of Asian people had lower levels of EV adoption. This may be due to our own biases with regards to what we've observed in Berkeley / the Bay Area. Finally, with regards to white people, we see a somewhat weak positive trend, demonstrating that towns which are more white have higher levels of EV adoption, but even towns that may only be 30-50% white still have relatively high levels of EV adoption.

### Commute Time

[43]:
```
#commute time

y = df["Proportion EV"]
plt.figure(figsize=(18,10))
color =  sns.cubehelix_palette(7)

plt.subplot(3, 3, 1)
plt.scatter(df['Less than 10 Minutes Proportion'], y, c=color[0], alpha=0.2)
plt.xlabel('Less than 10 Minutes Proportion')
plt.ylabel("Proportion EV")

plt.subplot(3,3,2)
plt.scatter(df['10 to 19 Minutes Proportion'], y, c=color[1], alpha=0.2)
plt.xlabel('10 to 19 Minutes Proportion')
plt.ylabel('Proportion EV')

plt.subplot(3,3,3)
plt.scatter(df['20 to 29 Minutes Proportion'], y, c=color[2], alpha=0.2)
plt.xlabel('20 to 29 Minutes Proportion')
plt.ylabel('Proportion EV')

plt.subplot(3,3,4)
plt.scatter(df['30 to 44 Minutes Proportion'], y, c=color[3], alpha=0.2)
plt.xlabel('30 to 44 Minutes Proportion')
plt.ylabel('Proportion EV')

plt.subplot(3,3,5)
plt.scatter(df['45 to 59 Minutes Proportion'], y, c=color[4], alpha=0.2)
plt.xlabel('45 to 59 Minutes Proportion')
plt.ylabel('Proportion EV')

plt.subplot(3,3,6)
plt.scatter(df['60 to 89 Minutes Proportion'], y, c=color[5], alpha=0.2)
plt.xlabel('60 to 89 Minutes Proportion')
plt.ylabel('Proportion EV')

plt.subplot(3,3,7)
plt.scatter(df['90 or more Minutes Proportion'], y, c=color[5], alpha=0.2)
plt.xlabel('90 or more Minutes Proportion')
plt.ylabel('Proportion EV')


plt.tight_layout(pad = 4)

plt.suptitle('Proportion EV vs Commute Time Grouping', fontsize = 30);
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
'c' argument looks like a single numeric RGB or RGBA sequence, which should be
avoided as value-mapping will have precedence in case its length matches with
'x' & 'y'.  Please use a 2-D array with a single row if you really want to
specify the same RGB or RGBA value for all points.
```

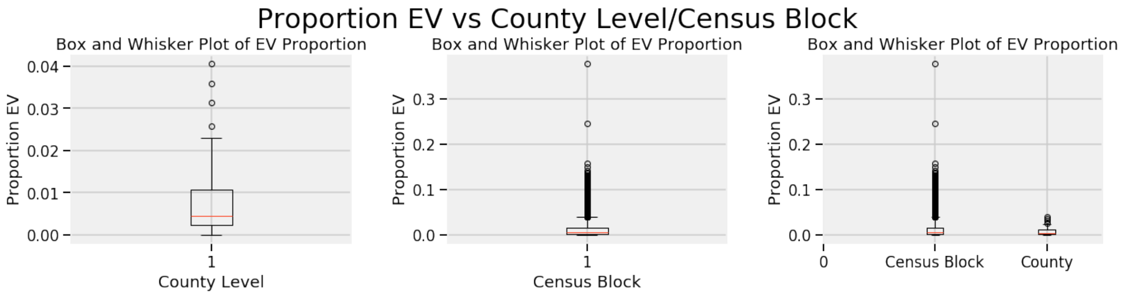


These plots are very interesting. They show the proportion of EV's per commute time bracket. It shows that census blocks with a higher proportion of people with medium length commute times (20-44 minutes) tend to have higher rates of EV adoption than those with short or long commute times. It appears that groups with 20 to 29 minute commute times, and 30 to 44 minute commute times tend to have higher proportions of EV's when they make up .2-.4 of the block's average commute time. These plots also show that when there are no 45 minute to 59 minute, 60 minute to 89 minute, or 90+ minute proportions there are higher rates of EV adoption, which could be indicative of what we call "range anxiety" in which people are less willing to adopt an electric vehicle if they have longer commutes due to the limitations of electric vehicles.

### Census vs County Level

[44]:

```
df_county = df.groupby(df["County"]).sum() #group data by county
df_county["Proportion EV County"] = df_county["EV Population"]/
 ↪df_county["Vehicle Population"]

plt.figure(figsize=(18,5))

plt.subplot(1,3,1)
plt.boxplot(x = df_county["Proportion EV County"])
plt.title("Box and Whisker Plot of EV Proportion")
plt.xlabel("County Level")
plt.ylabel('Proportion EV')

plt.subplot(1,3,2)
plt.boxplot(x = df['Proportion EV'])
plt.title('Box and Whisker Plot of EV Proportion')
plt.xlabel('Census Block')
plt.ylabel('Proportion EV')

plt.subplot(1,3,3)
plt.boxplot([df["Proportion EV"],df_county["Proportion EV County"]])
plt.xticks(range(3),['0','Census Block','County'])
plt.ylabel('Proportion EV')
plt.title("Box and Whisker Plot of EV Proportion")

plt.tight_layout(pad = 4)

plt.suptitle('Proportion EV vs County Level/Census Block', fontsize = 30);
```



There appear to be a lot of outliers in the Census Block box and whisker plots, which is due to the large amount of data points there are within the census block groups dataset. There are not as many outliers at the county level, since there are less data points, but outliers still exist.

Additionally, the census block group data has a much smaller window of EV adoption, whereas at the county level EV adoption appears much higher. This makes sense because the county level is an aggregation of all of the EV's in that county.

## 1.8 Forecasting and Prediction Modeling (25 points)

**Prediction Problems:**

1. Predicting adoption at the census block level
2. Predicting adoption using aggregated data at the county level
3. Generating a feature from our existing data (predicting categorical commute time at some threshold)

### 1.8.1 1. Predicting EV Adoption at the Census Block Level

*We chose to exclude two observations in this section because their Proportion EV values are significantly larger than the rest of the observations and we didn't want two outliers to skew our overall predicting power.*

[47]:
```
df1 = df.loc[df["Proportion EV"]<0.2]
df1.head()
```

[47]:
```
   index Census Block Group Code  County                     id  \
0      0             60570001024  NEVADA  1500000US060570001024
1      5             60570006002  NEVADA  1500000US060570006002
2      6             60570005022  NEVADA  1500000US060570005022
3      7             60570004022  NEVADA  1500000US060570004022
4      8             60570001023  NEVADA  1500000US060570001023

                                Geographic Area Name  Total Education  \
0  Block Group 4, Census Tract 1.02, Nevada Count…           1032.0
1  Block Group 2, Census Tract 6, Nevada County, …            765.0
2  Block Group 2, Census Tract 5.02, Nevada Count…            982.0
3  Block Group 2, Census Tract 4.02, Nevada Count…           1476.0
4  Block Group 3, Census Tract 1.02, Nevada Count…           1887.0

   Total Income  Total Race  White alone  Black or African American alone  \
0         578.0      1476.0       1152.0                              0.0
1         475.0      1104.0        947.0                             24.0
2         582.0      1329.0       1155.0                              0.0
3         811.0      1844.0       1612.0                            210.0
4         979.0      2373.0       2289.0                              0.0

   American Indian and Alaska Native alone  Asian alone  \
0                                      0.0         36.0
1                                     13.0         10.0
2                                      0.0         31.0
3                                      0.0          2.0
4                                      0.0          0.0
```

```
   Native Hawaiian and Other Pacific Islander alone  Some other race alone  \
0                                               0.0                    0.0   
1                                               0.0                   13.0   
2                                               0.0                    0.0   
3                                               0.0                    0.0   
4                                               0.0                   25.0   

   Total Commute  EV Population  Vehicle Population  Proportion EV  \
0          513.0            1.0               526.0       0.001901   
1          543.0            1.0               258.0       0.003876   
2          344.0            5.0               231.0       0.021645   
3          827.0            1.0               243.0       0.004115   
4          942.0            3.0               724.0       0.004144   

   More College Completed  College Completed  High School Completed  \
0                    77.0              374.0                 1032.0   
1                    43.0              209.0                  717.0   
2                    70.0              266.0                  857.0   
3                   125.0              650.0                 1458.0   
4                   201.0              956.0                 1697.0   

   Less than $10k  \$10,000 to $24,999  \$25,000 to $49,999  \
0            37.0                 31.0                 76.0   
1           101.0                143.0                 81.0   
2            33.0                171.0                221.0   
3            62.0                 63.0                188.0   
4            36.0                 65.0                249.0   

   \$50,000 to $99,999  \$100,000 to $199,999  $200,000 or more  \
0                213.0                  187.0              34.0   
1                125.0                   25.0               0.0   
2                125.0                   15.0              17.0   
3                322.0                  114.0              62.0   
4                270.0                  268.0              91.0   

   Less than 10 Minutes  10 to 19 Minutes  20 to 29 Minutes  30 to 44 Minutes  \
0                  24.0             188.0             163.0              75.0   
1                 303.0             184.0               0.0              12.0   
2                 152.0             109.0              32.0              51.0   
3                 170.0             160.0             164.0             145.0   
4                  47.0             445.0             153.0             130.0   

   45 to 59 Minutes  60 to 89 Minutes  90 or more Minutes  \
0               0.0              28.0                35.0   
1               0.0              44.0                 0.0   
2               0.0               0.0                 0.0   
```

```
3             128.0              0.0                60.0
4              39.0             44.0                84.0

   More College Completed Proportion  College Completed Proportion  \
0                           0.074612                      0.362403
1                           0.056209                      0.273203
2                           0.071283                      0.270876
3                           0.084688                      0.440379
4                           0.106518                      0.506624

   High School Completed Proportion  Less than $10k Proportion  \
0                          1.000000                   0.064014
1                          0.937255                   0.212632
2                          0.872709                   0.056701
3                          0.987805                   0.076449
4                          0.899311                   0.036772

   \$10,000 to $24,999 Proportion  \$25,000 to $49,999 Proportion  \
0                        0.053633                        0.131488
1                        0.301053                        0.170526
2                        0.293814                        0.379725
3                        0.077682                        0.231813
4                        0.066394                        0.254341

   \$50,000 to $99,999 Proportion  \$100,000 to $199,999 Proportion  \
0                        0.368512                          0.323529
1                        0.263158                          0.052632
2                        0.214777                          0.025773
3                        0.397041                          0.140567
4                        0.275792                          0.273749

   $200,000 or more Proportion  White alone Proportion  \
0                     0.058824                0.780488
1                     0.000000                0.857790
2                     0.029210                0.869074
3                     0.076449                0.874187
4                     0.092952                0.964602

   Black or African American alone Proportion  \
0                                    0.000000
1                                    0.021739
2                                    0.000000
3                                    0.113883
4                                    0.000000

   American Indian and Alaska Native alone Proportion  Asian alone Proportion  \
0                                            0.000000                0.024390
```

```
1                                           0.011775                 0.009058
2                                           0.000000                 0.023326
3                                           0.000000                 0.001085
4                                           0.000000                 0.000000

   Native Hawaiian and Other Pacific Islander alone Proportion  \
0                                                0.0
1                                                0.0
2                                                0.0
3                                                0.0
4                                                0.0

   Some other race alone Proportion  Less than 10 Minutes Proportion  \
0                          0.000000                         0.046784
1                          0.011775                         0.558011
2                          0.000000                         0.441860
3                          0.000000                         0.205562
4                          0.010535                         0.049894

   10 to 19 Minutes Proportion  20 to 29 Minutes Proportion  \
0                     0.366472                     0.317739
1                     0.338858                     0.000000
2                     0.316860                     0.093023
3                     0.193470                     0.198307
4                     0.472399                     0.162420

   30 to 44 Minutes Proportion  45 to 59 Minutes Proportion  \
0                     0.146199                     0.000000
1                     0.022099                     0.000000
2                     0.148256                     0.000000
3                     0.175333                     0.154776
4                     0.138004                     0.041401

   60 to 89 Minutes Proportion  90 or more Minutes Proportion
0                     0.054581                       0.068226
1                     0.081031                       0.000000
2                     0.000000                       0.000000
3                     0.000000                       0.072551
4                     0.046709                       0.089172
```

**Data Standardization** We chose to standardize the values for all of our features, given that we plan to use Lasso and Ridge regression methods. It is worth noting that using proportions does somewhat normalize all of the data to be values between 0 and 1, however, we chose to use the standard scaler to ensure our data was normalized with mean = 0 and variance = 1.

```
[48]: X = df1.loc[:, "More College Completed Proportion":
                     "90 or more Minutes Proportion"]
```

```
y = df1[["Proportion EV"]]
```

[49]:

```
scaler = StandardScaler() #initializ scaler
#standardize features
scaler.fit(X)
X_stnd = scaler.transform(X)
#standardize response variables
scaler.fit(y)
y_stnd = scaler.transform(y)
```

[50]:

```
X_train, X_test, y_train, y_test = train_test_split(X_stnd, y_stnd, test_size=0.
 ↪2, random_state=1)
```

**Prediction Model 1.1 - MLR** We build a multiple linear regression model on the training data and use those parameter estimates to predict the values of our target in the testing data and training data and use those predictions to calculate a test RMSE and a train RMSE. The test RMSE is constructed to evaluate the performance of this model relative to other models we will build (Ridge, Lasso, and Regression Tree).

[51]:

```
MLR_model = LinearRegression()
MLR_fit = MLR_model.fit(X_train, y_train)
y_pred_test = MLR_fit.predict(X_test)
y_pred_train = MLR_fit.predict(X_train)
print("test RMSE:", mean_squared_error(y_test, y_pred_test, squared = False))
print("train RMSE:", mean_squared_error(y_train, y_pred_train, squared = False))
print("R^2 test:", r2_score(y_test, y_pred_test))
print("R^2 train:", r2_score(y_train, y_pred_train))
MLR_coefs = MLR_fit.coef_
MLR_coefs
```

```
test RMSE: 0.5974473359333997
train RMSE: 0.5923059816377686
R^2 test: 0.6363543342540623
R^2 train: 0.6507756764946591
```

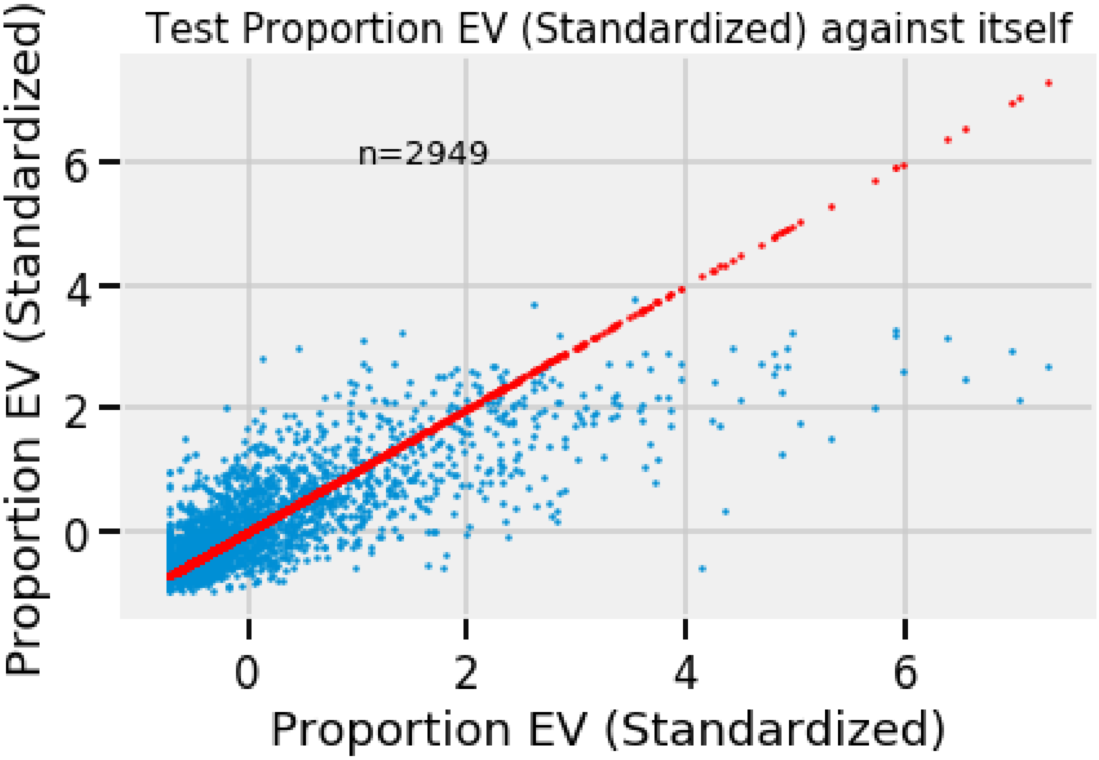
[51]:

```
array([[ 2.96815016e-01,  8.63948171e-02, -3.28113683e-02,
        -4.25408383e-02, -6.90373193e-02, -9.18287549e-02,
        -1.05907185e-01, -1.40275091e-01,  3.95695410e-01,
         4.47537370e-02, -5.50815450e-03, -1.50301605e-04,
         6.96975006e-02, -2.20982570e-04,  3.43574333e-02,
         1.48719336e-03, -1.99357866e-02,  1.67914621e-02,
         1.13960051e-02,  1.38346005e-02, -6.16564356e-03,
        -2.38942134e-02]])
```

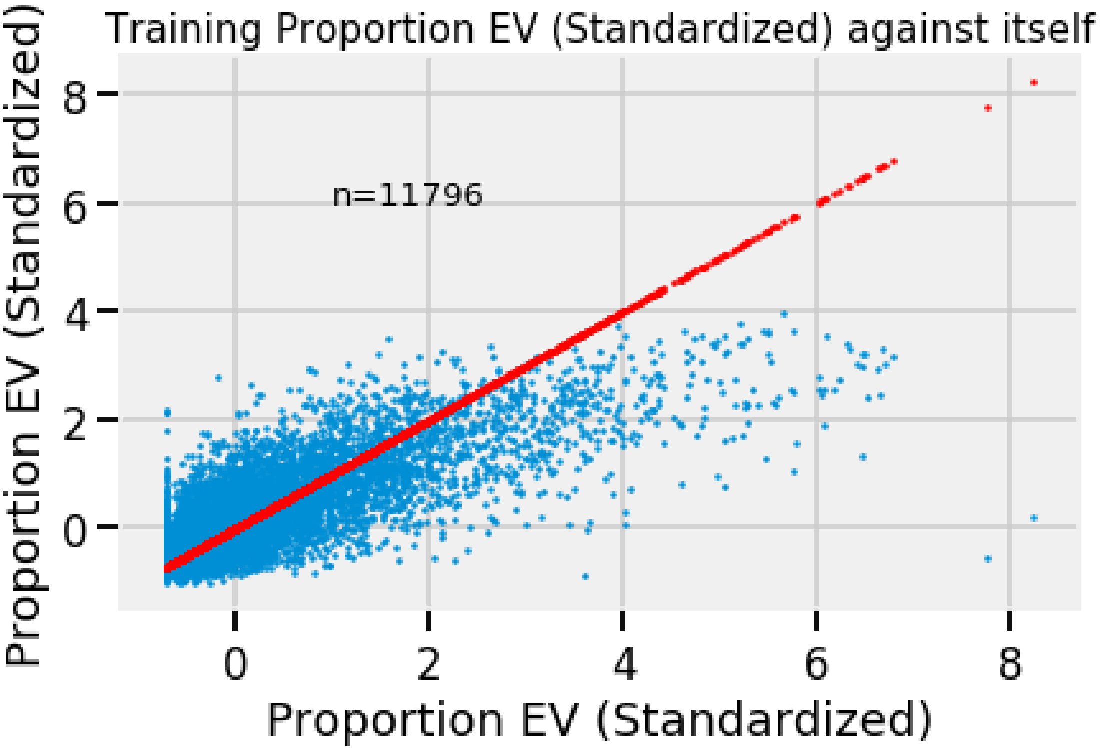
We observe that our train RMSE and our test RMSE are already very similar, before even implementing any sort of regularization approach. We see that the train RMSE is slightly lower than the test RMSE, but we expected this difference to be larger.

*Let's visualize how far off each predictions is from the actual obervation. These are essentialy graphs of the residuals. We will continue to display this type of graph throughout the modeling section of our ntebook to visually illustrate the residuals of our models.*

[52]:
```
# Scatter test predictions on test truth
plt.scatter(y_test, y_pred_test, s=1)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test, y_test, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)")
plt.ylabel("Proportion EV (Standardized)")
plt.title("Test Proportion EV (Standardized) against itself", size=15)
plt.text(1, 6, "n=2949", size=13);
```



[53]:
```
# Scatter train predictions on train truth
plt.scatter(y_train, y_pred_train, s=1)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train, y_train, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)")
plt.ylabel("Proportion EV (Standardized)")
plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(1, 6, "n=11796", size=13);
```



It looks like we're systematically over predicting for small values of y and systematically under predicting for large values of y. This may be evidence that linear regression is simply not appropriate for this prediction question. We will now turn to two regularization methods to address the potential problem over data overfit: Ridge and Lasso.

**Prediction Model 1.2 - Ridge Regression** We move now to the first of our regularization methods. We include this model to check to see whether the multiple linear regression model suffered from a problem of overfitting data, a result of including too many features.

**5-fold CV** We first implemented a 5-fold cross validation to detmine the optimal size of the penalty coefficient, $\lambda$ (referred to as alpha here). We obtained an alpha value of 78.48. When applying the Ridge model to the test data set, we observe a slightly higher test RMSE for our ridge regression with 5-fold CV than we do for our original linear regression. As of this point, linear regression is our best model.

We chose 5-fold because we understood that to be standard pratice, but to see whether or not the number of folds would affect our results, we decided to run another Ridge model using 15-fold CV to determine the optimal alpha.

**NOTE**: *this cell will take awhile to load*

```
[54]: kf = KFold(n_splits = 5, shuffle = True, random_state = 1)
      grid = np.linspace(0.01, 100, 1000)
      ridge_model = RidgeCV(cv = kf, alphas = grid)
      ridge_fit = ridge_model.fit(X_train, y_train)


      y_pred_ridge_test_5 = ridge_fit.predict(X_test)
      y_pred_ridge_train_5 = ridge_fit.predict(X_train)
      print("test RMSE:", mean_squared_error(y_test, y_pred_ridge_test_5, squared =␣
       ↪False))
      print("train RMSE:", mean_squared_error(y_train, y_pred_ridge_train_5, squared␣
       ↪= False))
      print("R2 test:", r2_score(y_test, y_pred_ridge_test_5))
      print("R2 train:", r2_score(y_train, y_pred_ridge_train_5))
      print("alpha:", ridge_fit.alpha_)
      ridge_coefs = ridge_fit.coef_
      ridge_coefs
```

```
test RMSE: 0.5974504814612387
train RMSE: 0.5923251523678588
R2 test: 0.6363505050945448
R2 train: 0.6507530699575023
alpha: 78.48063063063063
```

```
[54]: array([[ 0.29293678,  0.09100529, -0.03303619, -0.04222182, -0.06823353,
              -0.09124812, -0.10597162, -0.13852995,  0.39263336,  0.03168366,
              -0.01192523, -0.00165595,  0.05979496, -0.00132003,  0.02553477,
               0.00115809, -0.01998584,  0.0168765 ,  0.01157747,  0.01392109,
              -0.00612818, -0.023895  ]])
```

**15-fold CV** We also tried a 15-fold CV to see if the output alpha would give us a better test RMSE. The only notable difference between 5-fold CV and 15-fold CV, though, is that we obtain a higher alpha value of 93.89 (compared to 78.48). This means that we are penalizing additional features more heavily in our model building process than we did in the case of 5-fold CV. The test RMSE is larger with 15-fold CV by an extremely small order of magnitude (10^-6).

**NOTE**: *this cell will take awhile to load*

```
[55]: kf = KFold(n_splits = 15, shuffle = True, random_state = 1)
      grid = np.linspace(0.01, 100, 1000)
      ridge_model = RidgeCV(cv = kf, alphas = grid)
      ridge_fit = ridge_model.fit(X_train, y_train)


      y_pred_ridge_test_15 = ridge_fit.predict(X_test)
      y_pred_ridge_train_15 = ridge_fit.predict(X_train)
```

```
print("test RMSE:", mean_squared_error(y_test, y_pred_ridge_test_15, squared =␣
 ↪False))
print("train RMSE:", mean_squared_error(y_train, y_pred_ridge_train_15, squared␣
 ↪= False))
print("R2 test:", r2_score(y_test, y_pred_ridge_test_15))
print("R2 train:", r2_score(y_train, y_pred_ridge_train_15))
print("alpha:", ridge_fit.alpha_)
ridge_coefs = ridge_fit.coef_
ridge_coefs
```

```
test RMSE: 0.5974520510257225
train RMSE: 0.5923324364614463
R2 test: 0.6363485944020297
R2 train: 0.6507444802058922
alpha: 93.8945045045045
```

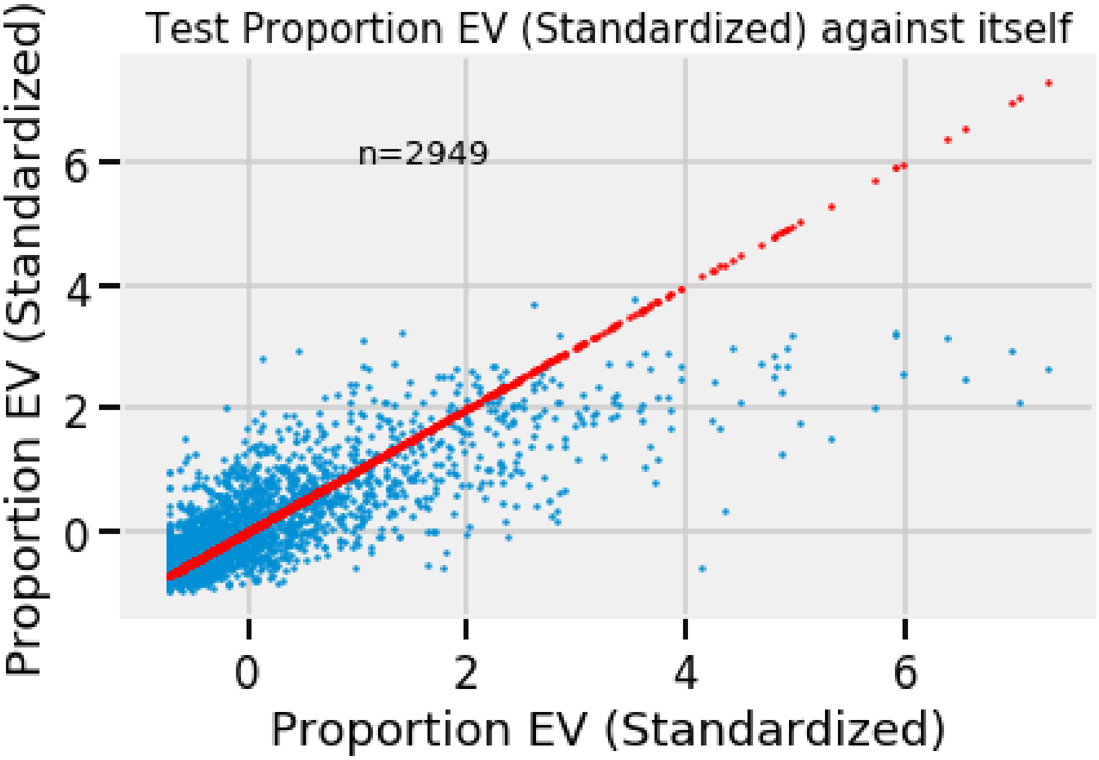
[55]:

```
array([[ 0.29221638,  0.09183542, -0.03301985, -0.04216166, -0.06807882,
        -0.0911365 , -0.10598241, -0.1381885 ,  0.39203745,  0.02990957,
        -0.01281216, -0.00186545,  0.05844207, -0.00147402,  0.02432825,
         0.00109582, -0.01999475,  0.01689218,  0.01160823,  0.01393699,
        -0.00611774, -0.02389173]])
```

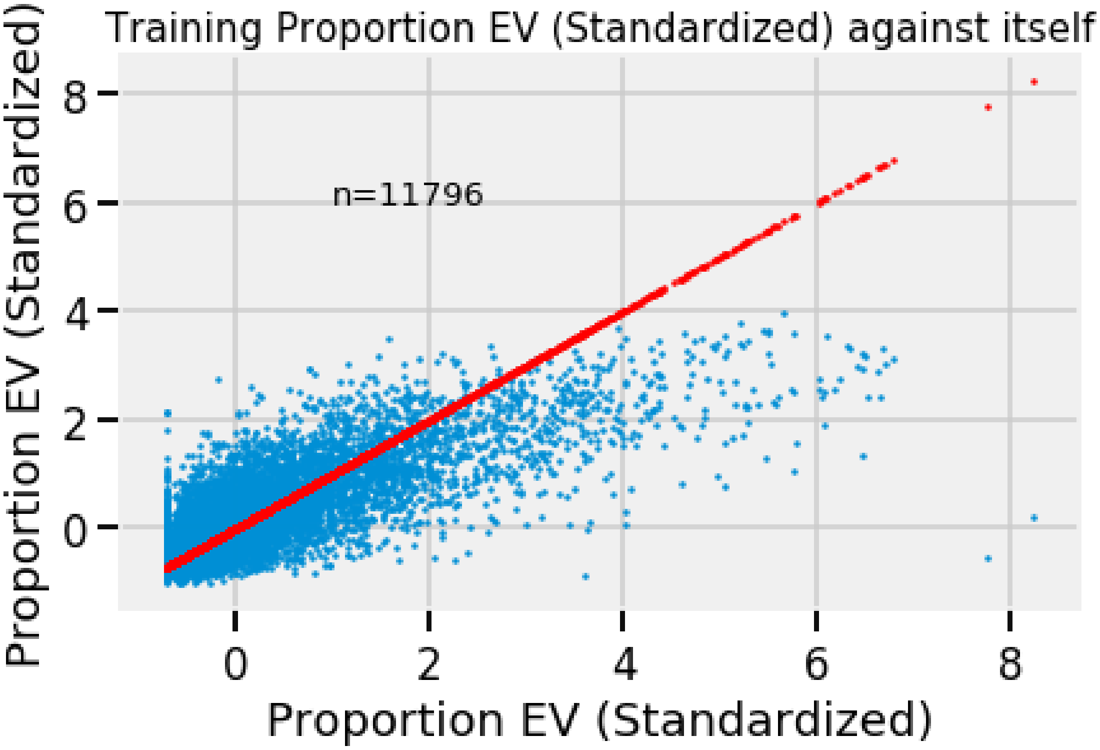
*Let's visualize how far off each predictions is from the actual obervation just for the caase of 5-fold CV.*

[56]:

```
# Scatter test predictions on test truth
plt.scatter(y_test, y_pred_ridge_test_5, s=1)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test, y_test, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)")
plt.ylabel("Proportion EV (Standardized)")
plt.title("Test Proportion EV (Standardized) against itself", size=15)
plt.text(1, 6, "n=2949", size=13);
```



[57]:
```
# Scatter train predictions on train truth
plt.scatter(y_train, y_pred_ridge_train_5, s=1)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train, y_train, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)")
plt.ylabel("Proportion EV (Standardized)")
plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(1, 6, "n=11796", size=13);
```



We continue to observe that we have a problem of underpredicting the value of our target variable (Proportion EV) for large values of this variable.

**Prediction Model 1.3 - Lasso Regression** We turn now to our second regularization model, Lasso. As was the case for Ridge, we will need to begin by determining our ideal hyerparameter alpha. To do this, we try one model which utilizes 5-fold CV and another model which utilizes 15-fold CV. In each model, we use the alpha found through CV to build the model on the training data, then we apply the model to our test data to see how it performs relative to linear regression and ridge.

**5-fold CV**

[58]:
```
kf = KFold(n_splits = 5, shuffle = True, random_state = 1)
grid = np.linspace(0.01, 100, 1000)
lasso_model = LassoCV(cv = kf, alphas = grid)
lasso_fit = lasso_model.fit(X_train, y_train)


y_pred_lasso_test = lasso_fit.predict(X_test)
y_pred_lasso_train = lasso_fit.predict(X_train)
print("test RMSE:", mean_squared_error(y_test, y_pred_lasso_test, squared =␣
 ↪False))
```

```
print("train RMSE:", mean_squared_error(y_train, y_pred_lasso_train, squared =␣
 ↪False))
print("R2 test:", r2_score(y_test, y_pred_lasso_test))
print("R2 train:", r2_score(y_train, y_pred_lasso_train))
print("alpha:", lasso_fit.alpha_)
lasso_coefs = lasso_fit.coef_
lasso_coefs
```

```
/opt/conda/lib/python3.8/site-packages/sklearn/utils/validation.py:72:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  return f(**kwargs)
```

```
test RMSE: 0.5972230187271781
train RMSE: 0.5937147696225569
R2 test: 0.636627351343292
R2 train: 0.6491124546730251
alpha: 0.01
```

```
[58]: array([ 0.30747385,  0.03704383, -0.00459413,  0.        ,  0.00472323,
              -0.        , -0.01174558, -0.02125518,  0.49850594, -0.        ,
              -0.01559882, -0.        ,  0.03297258, -0.        ,  0.        ,
              -0.        , -0.01616737,  0.0075399 ,  0.00499849,  0.00194969,
              -0.        , -0.02008017])
```

We determine the ideal alpha to be 0.01 for 5-fold CV Lasso Regression, making this model nearly identical to our linear regression model (which can be thought of as having an alpha of 0). We see that Lasso with 5-fold CV returns our smallest test RMSE yet! As of this point, Lasso is now our best model.

**15-fold CV**

```
[59]: kf = KFold(n_splits = 15, shuffle = True, random_state = 1)
grid = np.linspace(0.01, 100, 1000)
lasso_model = LassoCV(cv = kf, alphas = grid)
lasso_fit = lasso_model.fit(X_train, y_train)


y_pred_lasso_test = lasso_fit.predict(X_test)
y_pred_lasso_train = lasso_fit.predict(X_train)
print("test RMSE:", mean_squared_error(y_test, y_pred_lasso_test, squared =␣
 ↪False))
print("train RMSE:", mean_squared_error(y_train, y_pred_lasso_train, squared =␣
 ↪False))
print("R2 test:", r2_score(y_test, y_pred_lasso_test))
print("R2 train:", r2_score(y_train, y_pred_lasso_train))
print("alpha:", lasso_fit.alpha_)
```

```
lasso_coefs = lasso_fit.coef_
lasso_coefs
```

```
/opt/conda/lib/python3.8/site-packages/sklearn/utils/validation.py:72:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  return f(**kwargs)
```

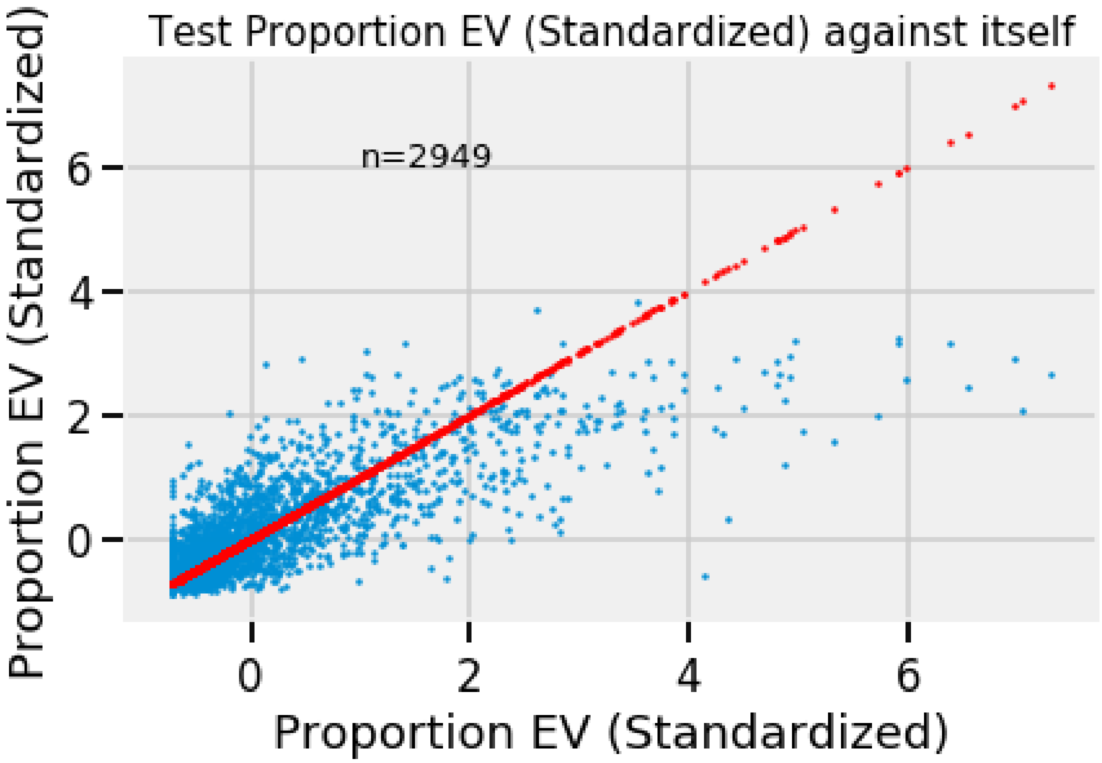
```
test RMSE: 0.5972230187271781
train RMSE: 0.5937147696225569
R2 test: 0.636627351343292
R2 train: 0.6491124546730251
alpha: 0.01
```

```
[59]: array([ 0.30747385,  0.03704383, -0.00459413,  0.        ,  0.00472323,
              -0.        , -0.01174558, -0.02125518,  0.49850594, -0.        ,
              -0.01559882, -0.        ,  0.03297258, -0.        ,  0.        ,
              -0.        , -0.01616737,  0.0075399 ,  0.00499849,  0.00194969,
              -0.        , -0.02008017])
```

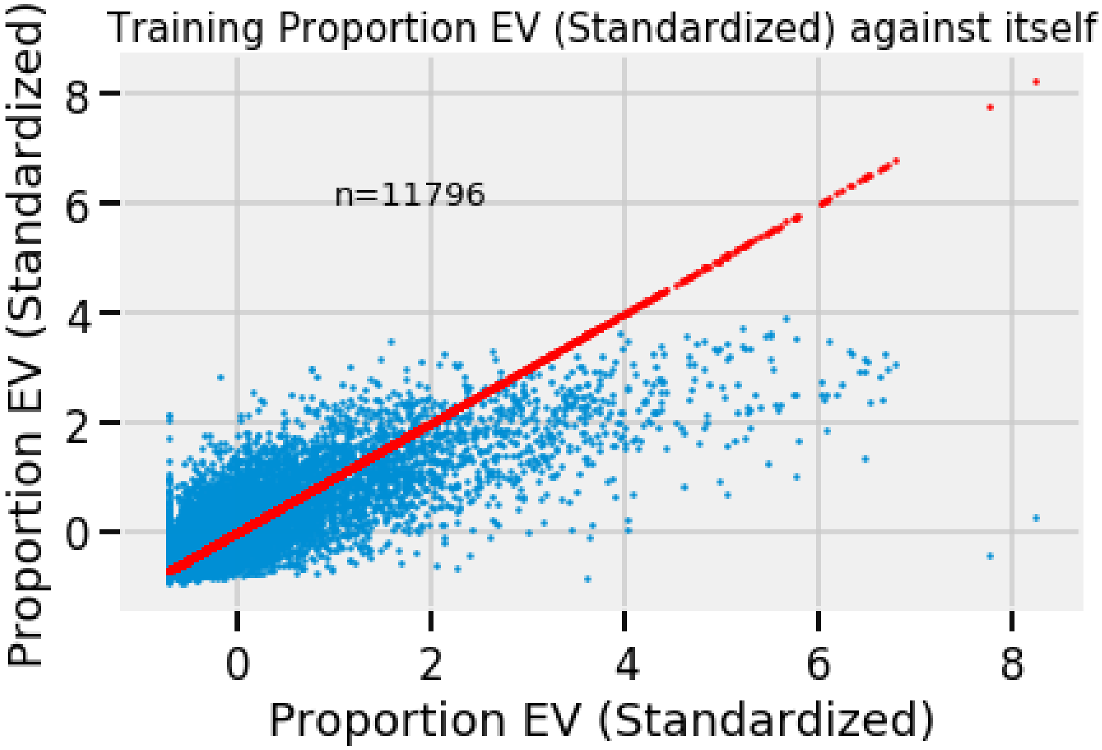
There is no difference when implementing 15-fold CV compared to 5-fold CV.

*Let's visualize how far off each predictions is from the actual obervation.*

```
[60]: # Scatter test predictions on test truth
plt.scatter(y_test, y_pred_lasso_test, s=1)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test, y_test, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)")
plt.ylabel("Proportion EV (Standardized)")
plt.title("Test Proportion EV (Standardized) against itself", size=15)
plt.text(1, 6, "n=2949", size=13);
```



[61]:
```
# Scatter train predictions on train truth
plt.scatter(y_train, y_pred_lasso_train, s=1)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train, y_train, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)")
plt.ylabel("Proportion EV (Standardized)")
plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(1, 6, "n=11796", size=13);
```



We continue to observe the problem of under predicting for large values of Proportion EV.

**Prediction Model 1.4 - Regression Tree** For our final model we chose to implement a regression tree in the hope that its non-parametric nature would correct our issue of systematically under predicting for large values of Proportion EV.

[62]:

```
from sklearn.tree import DecisionTreeRegressor

regressor = DecisionTreeRegressor(random_state = 1)
regressor.fit(X_train, y_train)
y_pred_tree_test = regressor.predict(X_test)
y_pred_tree_train = regressor.predict(X_train)

print("test RMSE:", mean_squared_error(y_test, y_pred_tree_test, squared =␣
 ↪False))
print("train RMSE:", mean_squared_error(y_train, y_pred_tree_train, squared =␣
 ↪False))
print("R2 test:", r2_score(y_test, y_pred_tree_test))
print("R2 train:", r2_score(y_train, y_pred_tree_train))
```

```
test RMSE: 0.8108215178080813
train RMSE: 6.051831938670744e-17
R2 test: 0.3302238341483078
```

```
R2 train: 1.0
```

Using default values for `max_leaf_nodes`, `max_features`, and `max_depth`, we obtain an extremely small train RMSE and more importantly a slightly larger test RMSE, 0.81702, than we do for each of our first three models (0.59744, 0.59745, and 0.59722 respecitvely). We will now decide what values to set for each of these hyperparameters using a gridsearch.

### Determine ideal hyperparameters

**NOTE**: *this cell will take awhile to load*

```
[63]: from sklearn.model_selection import RandomizedSearchCV
      from scipy.stats import randint

      param_dist = {'max_leaf_nodes': randint(3, 100),
                    'max_features': randint(2, 22),
                    'max_depth': randint(1, 10)}

      rnd_search = RandomizedSearchCV(regressor, param_distributions=param_dist,
                                      cv=10, n_iter=200, random_state = 2020)
      rnd_search.fit(X_train, y_train)

      print(rnd_search.best_score_)
      print(rnd_search.best_params_)
```

```
0.6363022708853932
{'max_depth': 7, 'max_features': 15, 'max_leaf_nodes': 20}
```

```
[64]: regressor = DecisionTreeRegressor(random_state = 1, max_depth = 7, max_features␣
       ↪= 15,
                                         max_leaf_nodes = 20)
      regressor.fit(X_train, y_train)
      y_pred_tree_test = regressor.predict(X_test)
      y_pred_tree_train = regressor.predict(X_train)

      print("test RMSE:", mean_squared_error(y_test, y_pred_tree_test, squared =␣
       ↪False))
      print("train RMSE:", mean_squared_error(y_train, y_pred_tree_train, squared =␣
       ↪False))
      print("R2 test:", r2_score(y_test, y_pred_tree_test))
      print("R2 train:", r2_score(y_train, y_pred_tree_train))
```

```
test RMSE: 0.5970587266664085
train RMSE: 0.574282717424998
R2 test: 0.6368272466184982
R2 train: 0.6717053985967233
```

It looks like a decision tree with tuned hyperparameters gives us our best model yet! We obtain a test RMSE of 0.59705 which is lower than any of our other models.

Let's take a shot at visualizing what this model gives us.

[65]:
```
from sklearn import tree
import graphviz
```

Acquire the input for webgraphviz

[66]:
```
print(tree.export_graphviz(regressor, feature_names=X.columns))
```

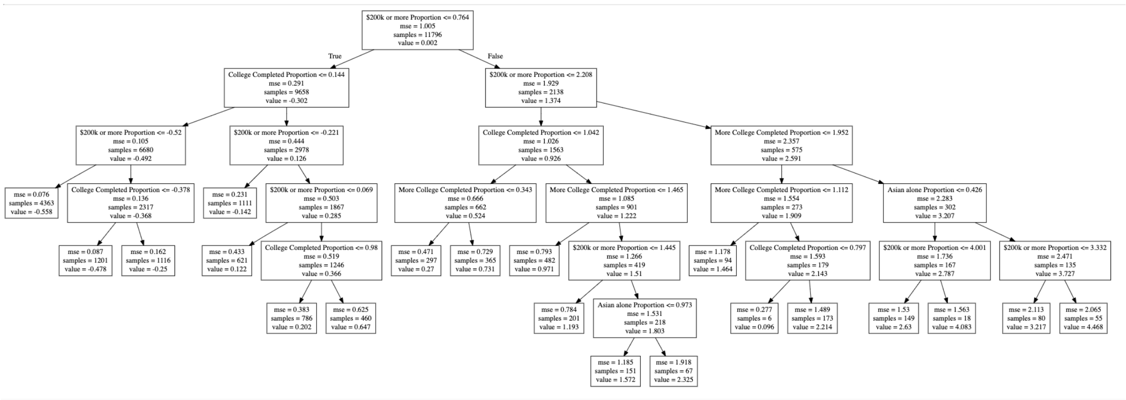
```
digraph Tree {
node [shape=box] ;
0 [label="$200,000 or more Proportion <= 0.764\nmse = 1.005\nsamples =
11796\nvalue = 0.002"] ;
1 [label="College Completed Proportion <= 0.144\nmse = 0.291\nsamples =
9658\nvalue = -0.302"] ;
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
5 [label="$200,000 or more Proportion <= -0.52\nmse = 0.105\nsamples =
6680\nvalue = -0.492"] ;
1 -> 5 ;
17 [label="mse = 0.076\nsamples = 4363\nvalue = -0.558"] ;
5 -> 17 ;
18 [label="College Completed Proportion <= -0.378\nmse = 0.136\nsamples =
2317\nvalue = -0.368"] ;
5 -> 18 ;
27 [label="mse = 0.087\nsamples = 1201\nvalue = -0.478"] ;
18 -> 27 ;
28 [label="mse = 0.162\nsamples = 1116\nvalue = -0.25"] ;
18 -> 28 ;
6 [label="$200,000 or more Proportion <= -0.221\nmse = 0.444\nsamples =
2978\nvalue = 0.126"] ;
1 -> 6 ;
11 [label="mse = 0.231\nsamples = 1111\nvalue = -0.142"] ;
6 -> 11 ;
12 [label="$200,000 or more Proportion <= 0.069\nmse = 0.503\nsamples =
1867\nvalue = 0.285"] ;
6 -> 12 ;
35 [label="mse = 0.433\nsamples = 621\nvalue = 0.122"] ;
12 -> 35 ;
36 [label="College Completed Proportion <= 0.98\nmse = 0.519\nsamples =
1246\nvalue = 0.366"] ;
12 -> 36 ;
37 [label="mse = 0.383\nsamples = 786\nvalue = 0.202"] ;
36 -> 37 ;
38 [label="mse = 0.625\nsamples = 460\nvalue = 0.647"] ;
36 -> 38 ;
2 [label="$200,000 or more Proportion <= 2.208\nmse = 1.929\nsamples =
2138\nvalue = 1.374"] ;
0 -> 2 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;
```

```
3 [label="College Completed Proportion <= 1.042\nmse = 1.026\nsamples =
1563\nvalue = 0.926"] ;
2 -> 3 ;
9 [label="More College Completed Proportion <= 0.343\nmse = 0.666\nsamples =
662\nvalue = 0.524"] ;
3 -> 9 ;
23 [label="mse = 0.471\nsamples = 297\nvalue = 0.27"] ;
9 -> 23 ;
24 [label="mse = 0.729\nsamples = 365\nvalue = 0.731"] ;
9 -> 24 ;
10 [label="More College Completed Proportion <= 1.465\nmse = 1.085\nsamples =
901\nvalue = 1.222"] ;
3 -> 10 ;
15 [label="mse = 0.793\nsamples = 482\nvalue = 0.971"] ;
10 -> 15 ;
16 [label="$200,000 or more Proportion <= 1.445\nmse = 1.266\nsamples =
419\nvalue = 1.51"] ;
10 -> 16 ;
21 [label="mse = 0.784\nsamples = 201\nvalue = 1.193"] ;
16 -> 21 ;
22 [label="Asian alone Proportion <= 0.973\nmse = 1.531\nsamples = 218\nvalue =
1.803"] ;
16 -> 22 ;
31 [label="mse = 1.185\nsamples = 151\nvalue = 1.572"] ;
22 -> 31 ;
32 [label="mse = 1.918\nsamples = 67\nvalue = 2.325"] ;
22 -> 32 ;
4 [label="More College Completed Proportion <= 1.952\nmse = 2.357\nsamples =
575\nvalue = 2.591"] ;
2 -> 4 ;
7 [label="More College Completed Proportion <= 1.112\nmse = 1.554\nsamples =
273\nvalue = 1.909"] ;
4 -> 7 ;
29 [label="mse = 1.178\nsamples = 94\nvalue = 1.464"] ;
7 -> 29 ;
30 [label="College Completed Proportion <= 0.797\nmse = 1.593\nsamples =
179\nvalue = 2.143"] ;
7 -> 30 ;
33 [label="mse = 0.277\nsamples = 6\nvalue = 0.096"] ;
30 -> 33 ;
34 [label="mse = 1.489\nsamples = 173\nvalue = 2.214"] ;
30 -> 34 ;
8 [label="Asian alone Proportion <= 0.426\nmse = 2.283\nsamples = 302\nvalue =
3.207"] ;
4 -> 8 ;
13 [label="$200,000 or more Proportion <= 4.001\nmse = 1.736\nsamples =
167\nvalue = 2.787"] ;
8 -> 13 ;
```

```
25 [label="mse = 1.53\nsamples = 149\nvalue = 2.63"] ;
13 -> 25 ;
26 [label="mse = 1.563\nsamples = 18\nvalue = 4.083"] ;
13 -> 26 ;
14 [label="$200,000 or more Proportion <= 3.332\nmse = 2.471\nsamples =
135\nvalue = 3.727"] ;
8 -> 14 ;
19 [label="mse = 2.113\nsamples = 80\nvalue = 3.217"] ;
14 -> 19 ;
20 [label="mse = 2.065\nsamples = 55\nvalue = 4.468"] ;
14 -> 20 ;
}
```

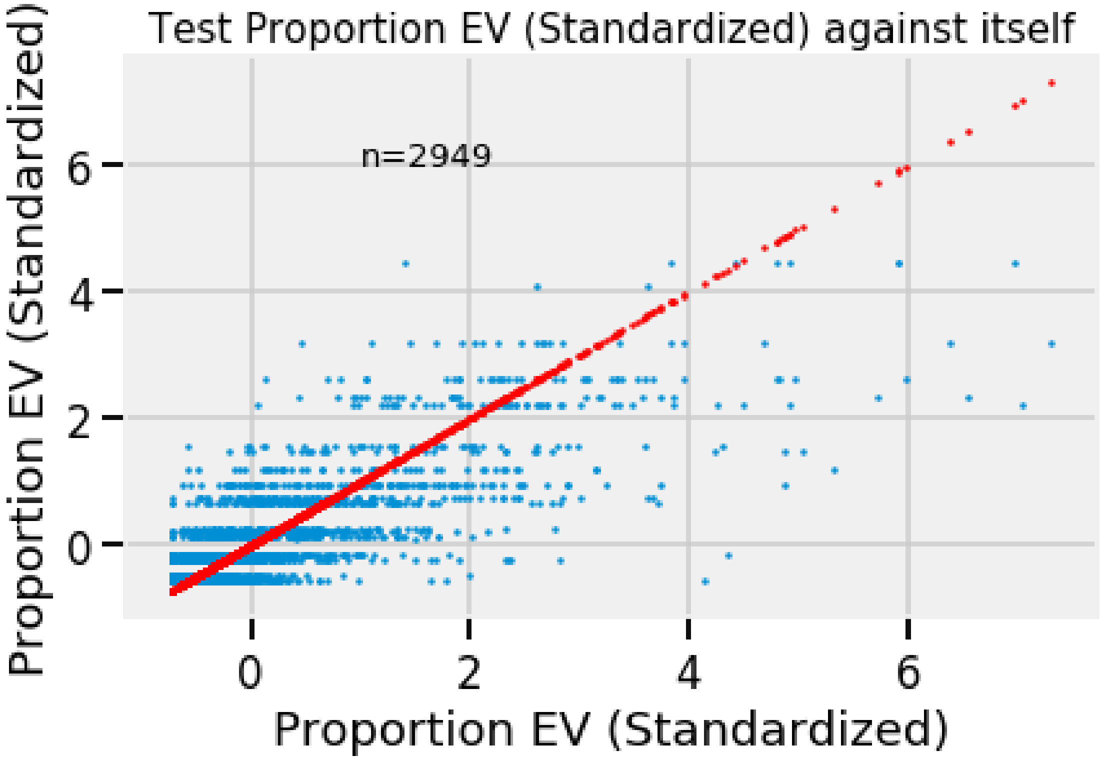
[67]:
```python
from IPython.display import Image
Image(filename='Output2.png')
```

[67]:

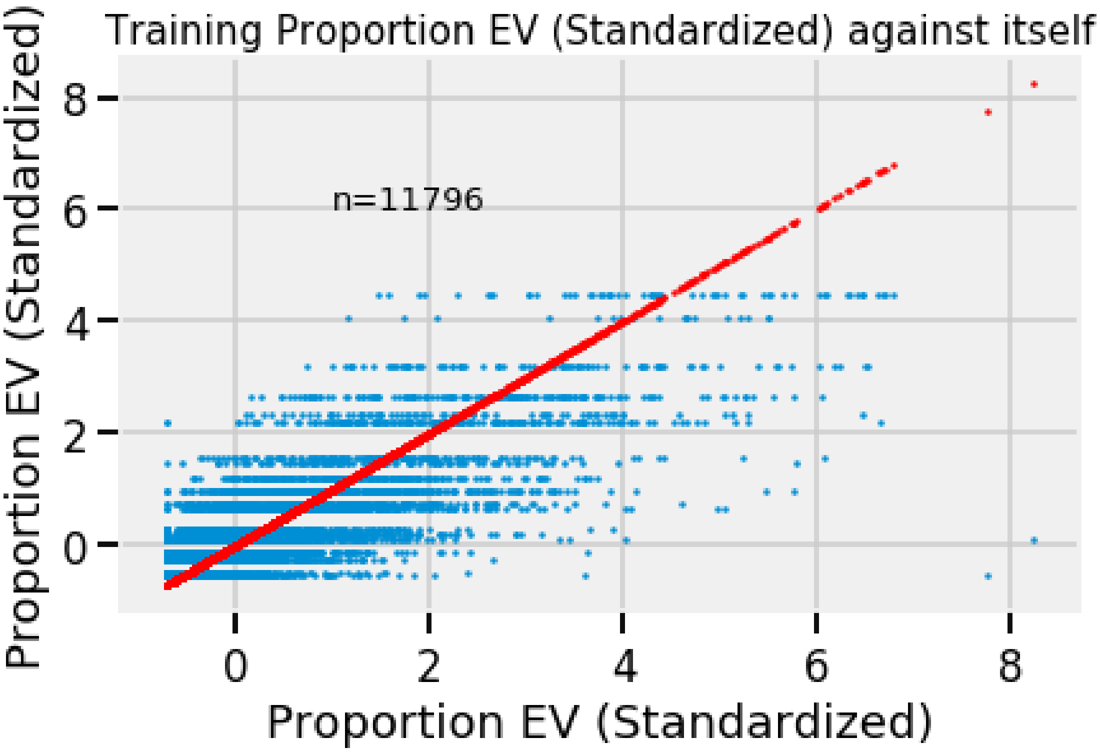


Below, we utilize the same residual visualization that we've been using throughout.

[68]:
```python
# Scatter test predictions on test truth
plt.scatter(y_test, y_pred_tree_test, s=1)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test, y_test, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)")
plt.ylabel("Proportion EV (Standardized)")
plt.title("Test Proportion EV (Standardized) against itself", size=15)
plt.text(1, 6, "n=2949", size=13);
```



[69]:
```
# Scatter train predictions on train truth
plt.scatter(y_train, y_pred_tree_train, s=1)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train, y_train, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)")
plt.ylabel("Proportion EV (Standardized)")
plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(1, 6, "n=11796", size=13);
```



We observe the same issue of under predicting for large values of Proportion EV using regression trees. The predictions using the tree are stratefied in horizontal lines because regression trees output discontinuous predictions. Each horizontal grouping corresponds to a different terminal node in the tree.

### 1.8.2 Training on NorCal and Testing on SoCal

To simulate if this model can be performed on other states, we'll split of California into 2 regions: Northern and Southern.

Let's split our data between those located in counties in Northern California versus those located in Southern California and made a train test split, with Northern CA census block group data being the training set, and Southern CA census block group data as the test set.

```
[70]: norcal_counties = ['DEL NORTE', 'SISKIYOU', 'MODOC', 'HUMBOLDT', 'TRINITY',␣
       ↪'SHASTA', 'LASSEN', 'MENDOCINO', 'TEHAMA', 'PLUMAS',
                         'GLENN', 'BUTTE', 'SIERRA', 'LAKE', 'COLUSA', 'YUBA',␣
       ↪'NEVADA', 'PLACER', 'SUTTER', 'SONOMA', 'NAPA', 'YOLO',
                         'SACRAMENTO', 'EL DORADO', 'ALPINE', 'AMADOR', 'SOLANO',␣
       ↪'MARIN', 'CONTRA COSTA', 'SAN JOAQUIN', 'CALAVERAS',
                         'TUOLUMNE', 'MONO', 'STANISLAUS', 'ALAMEDA', 'SAN FRANCISCO',␣
       ↪'SAN MATEO', 'SANTA CLARA', 'SANTA CRUZ',
                  'MERCED', 'MARIPOSA', 'MADERA', 'SAN BENITO', 'MONTEREY',␣
 ↪'FRESNO', 'KINGS', 'TULARE', 'INYO']
socal_counties = ['SAN LUIS OBISPO', 'SANTA BARBARA', 'KERN', 'VENTURA', 'LOS␣
 ↪ANGELES', 'SAN BERNARDINO', 'ORANGE',
                  'RIVERSIDE', 'SAN DIEGO', 'IMPERIAL']
```

[71]:

```
X = df1.loc[:, ["County",'More College Completed',
       'College Completed', 'High School Completed', 'Less than $10k',
       '\$10,000 to $24,999', '\$25,000 to $49,999', '\$50,000 to $99,999',
       '\$100,000 to $199,999', '$200,000 or more', 'Less than 10 Minutes',
       '10 to 19 Minutes', '20 to 29 Minutes', '30 to 44 Minutes',
       '45 to 59 Minutes', '60 to 89 Minutes', '90 or more Minutes',
       'More College Completed Proportion', 'College Completed Proportion',
       'High School Completed Proportion', 'Less than $10k Proportion',
       '\$10,000 to $24,999 Proportion', '\$25,000 to $49,999 Proportion',
       '\$50,000 to $99,999 Proportion', '\$100,000 to $199,999 Proportion',
       '$200,000 or more Proportion', 'White alone Proportion',
       'Black or African American alone Proportion',
       'American Indian and Alaska Native alone Proportion',
       'Asian alone Proportion',
       'Native Hawaiian and Other Pacific Islander alone Proportion',
       'Some other race alone Proportion', 'Less than 10 Minutes Proportion',
       '10 to 19 Minutes Proportion', '20 to 29 Minutes Proportion',
       '30 to 44 Minutes Proportion', '45 to 59 Minutes Proportion',
       '60 to 89 Minutes Proportion', '90 or more Minutes Proportion']]
y = df1.loc[:, ["County", 'Proportion EV']]
```

[72]:

```
#training on norcal and testing on socal
X_train = X[X['County'].isin(norcal_counties)].drop(columns='County')
X_test = X[X['County'].isin(socal_counties)].drop(columns='County')
y_train = y[y['County'].isin(norcal_counties)].drop(columns='County')
y_test = y[y['County'].isin(socal_counties)].drop(columns='County')
```

With this split, Northern California has 6503 datapoints, and Southern California has 8242 datapoints. The following code will be very similar to the process as above: using Linear and Lasso Regression.

[73]:

```
#scaling the test and training data (trained on the original X and y)
scaler = StandardScaler()
scaler.fit(X.drop(columns='County'))
X_train_stnd = scaler.transform(X_train)
X_test_stnd = scaler.transform(X_test)
scaler.fit(y.drop(columns='County'))
y_train_stnd = scaler.transform(y_train)
y_test_stnd = scaler.transform(y_test)
```

### Linear Regression

[74]:

```
MLR_model = LinearRegression()
MLR_fit = MLR_model.fit(X_train_stnd, y_train_stnd)
y_pred_test = MLR_fit.predict(X_test_stnd)
y_pred_train = MLR_fit.predict(X_train_stnd)
print("test RMSE:", mean_squared_error(y_test_stnd, y_pred_test, squared =␣
 ↪False))
print("train RMSE:", mean_squared_error(y_train, y_pred_train, squared = False))
print("R^2 test:", r2_score(y_test_stnd, y_pred_test))
print("R^2 train:", r2_score(y_train_stnd, y_pred_train))
MLR_coefs = MLR_fit.coef_
MLR_coefs
```

```
test RMSE: 0.5415629871735279
train RMSE: 0.9620806333140902
R^2 test: 0.5606293854446297
R^2 train: 0.6797211949652378
```

[74]:

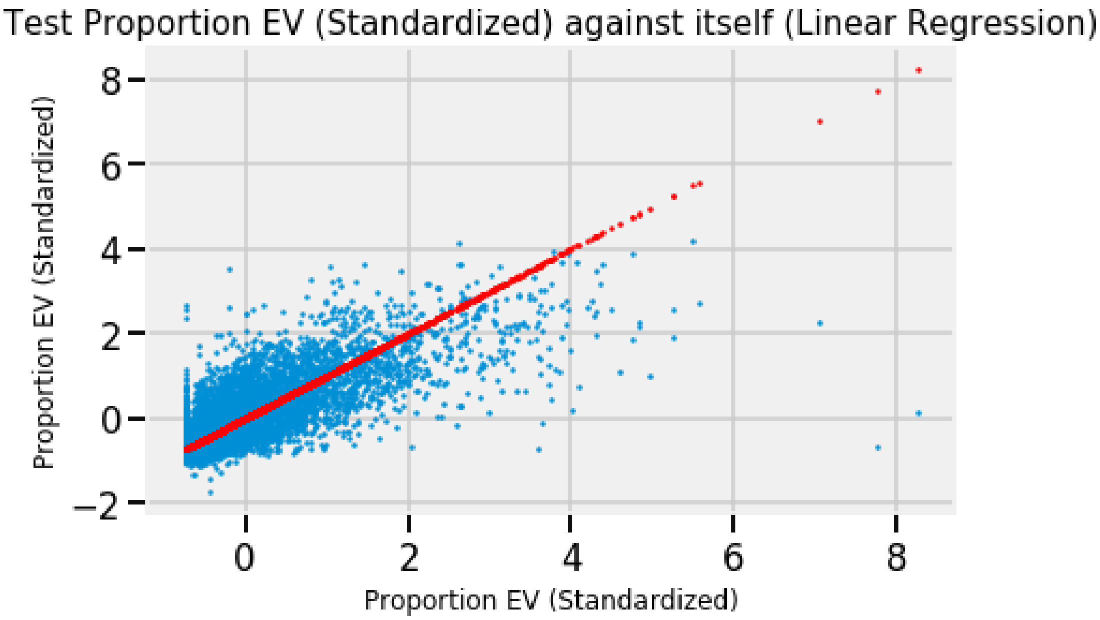
```
array([[ 0.09899852, -0.11054697,  0.17550865, -0.01667815, -0.01837829,
         0.03887218, -0.03789287,  0.02365891, -0.10221602, -0.00507533,
        -0.03211741, -0.04241756, -0.09798565, -0.03596612,  0.0140383 ,
        -0.01545628,  0.28758893,  0.05071867, -0.0525688 , -0.05417545,
        -0.08209632, -0.14658876, -0.12067752, -0.17408912,  0.5075715 ,
         0.08413841,  0.01311597,  0.0098667 ,  0.1395486 ,  0.00466904,
         0.0862681 ,  0.00343908, -0.02878554,  0.02472815,  0.03132999,
         0.0163892 , -0.04087685, -0.01913269]])
```

When we originally ran the model with a random train-test split, we got the following values [test RMSE: 0.5974473359333997, train RMSE: 0.5923059816377686, $R^2$ test: 0.6363543342540623, $R^2$ train: 0.6507756764946592]. It appears that the model performed on the NorCal/SoCal split does slightly better, however our training RMSE is incredibly high. This is may be due to chance, but it makes sense to conclude MLR as a bad model.

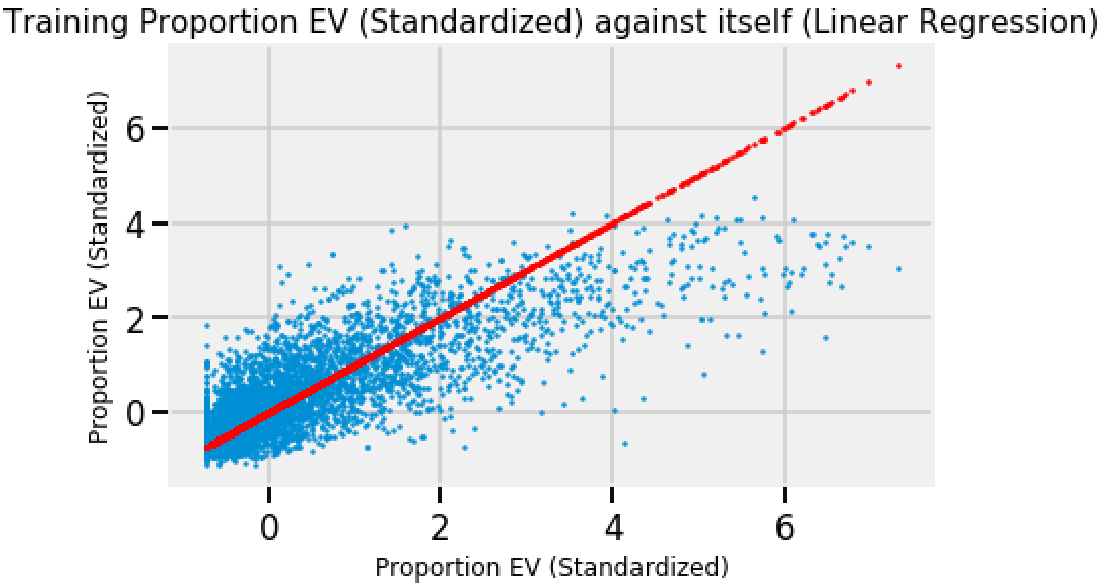
*Now let's visualize how far off each predictions is from the actual obervation.*

[75]:

```
# Scatter test predictions on test truth
plt.scatter(y_test_stnd, y_pred_test, s=1)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test_stnd, y_test_stnd, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
plt.title("Test Proportion EV (Standardized) against itself (Linear␣
 ↪Regression)", size=15);
```



[76]:
```
# Scatter train predictions on train truth
plt.scatter(y_train_stnd, y_pred_train, s=1)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train_stnd, y_train_stnd, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
plt.title("Training Proportion EV (Standardized) against itself (Linear␣
 ↪Regression)", size=15);
```



*These visualizations look very similar to the ones we made when doing our first prediction problem!*

### 1.8.3 KFold's Lasso Model

Now we will be doing a Lasso model, which worked the best on our first prediction problem.

```
[77]: kf = KFold(n_splits = 5, shuffle = True, random_state = 1)
      grid = np.linspace(0.01, 100, 1000)
      lasso_model = LassoCV(cv = kf, alphas = grid)
      lasso_fit = lasso_model.fit(X_train_stnd, y_train_stnd)


      y_pred_lasso_test = lasso_fit.predict(X_test_stnd)
      y_pred_lasso_train = lasso_fit.predict(X_train_stnd)
      print("test RMSE:", mean_squared_error(y_test_stnd, y_pred_lasso_test, squared␣
       ↪= False))
      print("train RMSE:", mean_squared_error(y_train_stnd, y_pred_lasso_train,␣
       ↪squared = False))
      print("R2 test:", r2_score(y_test_stnd, y_pred_lasso_test))
      print("R2 train:", r2_score(y_train_stnd, y_pred_lasso_train))
      print("alpha:", lasso_fit.alpha_)
      lasso_coefs = lasso_fit.coef_
      lasso_coefs
```

```
/opt/conda/lib/python3.8/site-packages/sklearn/utils/validation.py:72:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  return f(**kwargs)
```

```
test RMSE: 0.5299523394906483
train RMSE: 0.6711684229268619
R2 test: 0.5792668955443778
R2 train: 0.675683826482044
alpha: 0.01
```

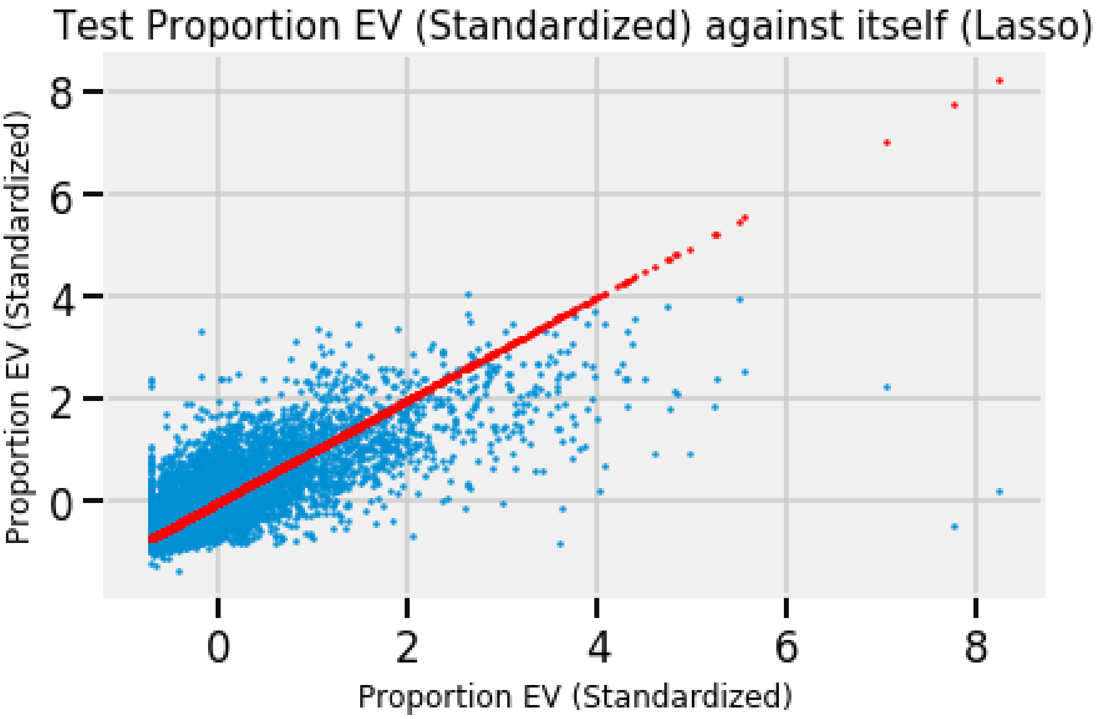
```
[77]: array([ 0.        , -0.        , -0.        , -0.        ,  0.        ,
              0.        , -0.        , -0.        , -0.        , -0.        ,
             -0.02984387, -0.        , -0.03616143, -0.00602279, -0.00437527,
             -0.        ,  0.32516567, -0.        , -0.00712405, -0.        ,
              0.00854898,  0.        , -0.0125975 , -0.00954526,  0.5729909 ,
             -0.        , -0.01512858,  0.        ,  0.06854569, -0.        ,
              0.00505354,  0.        , -0.01562682,  0.00624645,  0.        ,
              0.        , -0.01543041, -0.01838603])
```

The values we got when we ran this model with a random train-test split were: [test RMSE: 0.5972230187271781, train RMSE: 0.5937147696225569, $R^2$ test: 0.636627351343292, $R^2$ train: 0.6491124546730251, alpha: 0.01]. As you can see, we have a similar test RMSE, as well as a much closer train RMSE, as well as similar $R^2$ test & train values!

*Again, let's visualize how far off each predictions is from the actual obervation.*

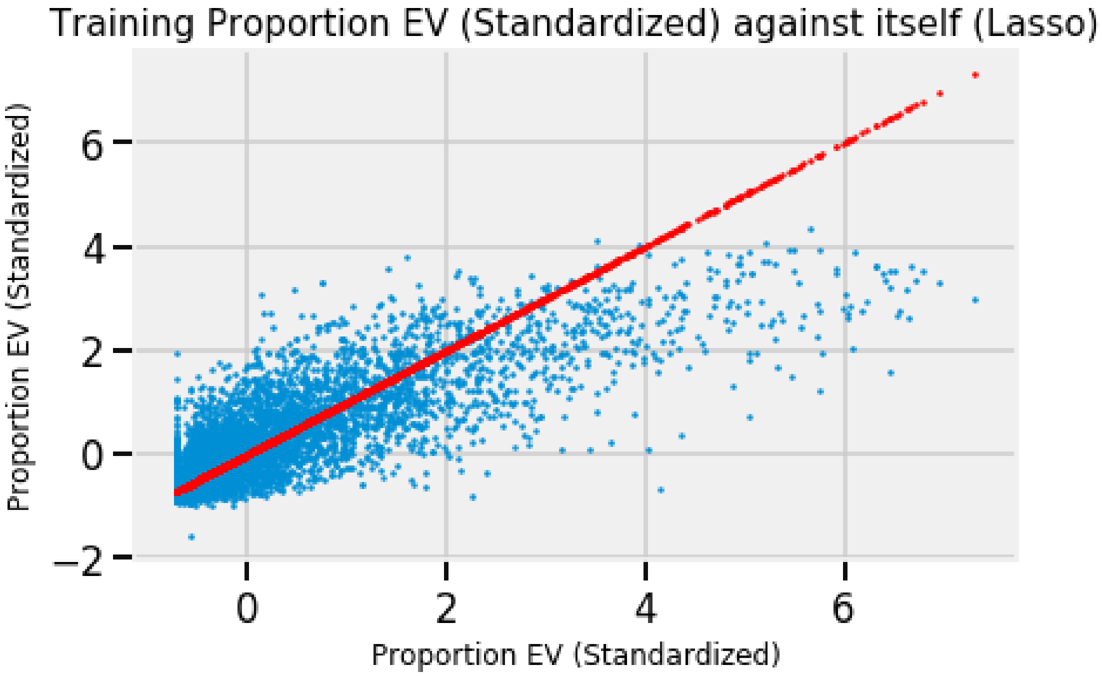
[78]:

```
# Scatter test predictions on test truth
plt.scatter(y_test_stnd, y_pred_lasso_test, s=1)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test_stnd, y_test_stnd, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
plt.title("Test Proportion EV (Standardized) against itself (Lasso)", size=15);
```



[79]:

```
# Scatter train predictions on train truth
plt.scatter(y_train_stnd, y_pred_lasso_train, s=1)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train_stnd, y_train_stnd, s=1, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
plt.title("Training Proportion EV (Standardized) against itself (Lasso)",␣
↪size=15);
```



Since our model seems to be predicting fairly similar RMSE and $R^2$ values, we can get a decent sense of how this model could potentially be applied to different states.

### 1.8.4 2. Predicting EV Adoption at the County Level

[80]:
```
pd.set_option('display.max_columns', None)
```

**Group data by county** In the first prediction problem, we looked at predicting at the census block level. We're now going to look at the data on a county level and maake predictions based on this aggregated dataset.

[81]:
```
df_county = df.groupby(df["County"]).sum() #start by grouping the original data␣
↪by county
df_county.head()
```

[81]:
```
              index  Total Education  Total Income  Total Race  White alone  \
County
ALAMEDA     6874021         707835.0      350409.0   1004894.0     421776.0
AMADOR       178075          19064.0        8795.0     24401.0      20886.0
BUTTE       1272687          95563.0       57063.0    147851.0     120725.0
```

```
CALAVERAS   154743          19055.0        9299.0     26386.0      24069.0
COLUSA      125374          10288.0        5167.0     16100.0      14024.0

           Black or African American alone  \
County
ALAMEDA                           113081.0
AMADOR                               750.0
BUTTE                               2557.0
CALAVERAS                            185.0
COLUSA                               265.0

           American Indian and Alaska Native alone  Asian alone  \
County
ALAMEDA                                     7025.0     287122.0
AMADOR                                       202.0        236.0
BUTTE                                       1638.0       7261.0
CALAVERAS                                    239.0        369.0
COLUSA                                       115.0        290.0

           Native Hawaiian and Other Pacific Islander alone  \
County
ALAMEDA                                              7637.0
AMADOR                                                 54.0
BUTTE                                                 221.0
CALAVERAS                                             114.0
COLUSA                                                 53.0

           Some other race alone  Total Commute  EV Population  \
County
ALAMEDA                 103175.0       441870.0         9198.0
AMADOR                    1190.0         7096.0           21.0
BUTTE                     6251.0        56110.0          117.0
CALAVERAS                  205.0         8744.0           10.0
COLUSA                     920.0         7015.0            0.0

           Vehicle Population  Proportion EV  More College Completed  \
County
ALAMEDA              358154.0      14.873580                134496.0
AMADOR                 6359.0       0.062305                  1265.0
BUTTE                 39237.0       0.361166                  8329.0
CALAVERAS              4741.0       0.060644                  1057.0
COLUSA                 2647.0       0.000000                   425.0

           College Completed  High School Completed  Less than $10k  \
County
ALAMEDA             366866.0               620005.0         14838.0
AMADOR                5643.0                16836.0           368.0
```

```
BUTTE                33777.0                84904.0          4819.0
CALAVERAS             5468.0                17060.0           435.0
COLUSA                2612.0                 6958.0           148.0

           \$10,000 to $24,999  \$25,000 to $49,999  \$50,000 to $99,999  \
County
ALAMEDA                33308.0              51067.0              89149.0
AMADOR                  1362.0               2109.0               2775.0
BUTTE                  11786.0              14092.0              15189.0
CALAVERAS               1357.0               2220.0               2576.0
COLUSA                   861.0               1185.0               1898.0

           \$100,000 to $199,999  $200,000 or more  Less than 10 Minutes  \
County
ALAMEDA                 104452.0           57595.0               27478.0
AMADOR                    1811.0             370.0                1041.0
BUTTE                     8865.0            2312.0               13202.0
CALAVERAS                 2251.0             460.0                1223.0
COLUSA                     882.0             193.0                1556.0

           10 to 19 Minutes  20 to 29 Minutes  30 to 44 Minutes  \
County
ALAMEDA            104435.0           75553.0          101434.0
AMADOR               1598.0            1612.0             966.0
BUTTE               21389.0            7550.0            8381.0
CALAVERAS            1402.0             743.0            1867.0
COLUSA               1843.0            1166.0            1184.0

           45 to 59 Minutes  60 to 89 Minutes  90 or more Minutes  \
County
ALAMEDA             57644.0           56029.0            19297.0
AMADOR                619.0             791.0              469.0
BUTTE                2485.0            1407.0             1696.0
CALAVERAS            1348.0            1368.0              793.0
COLUSA                465.0             524.0              277.0

           More College Completed Proportion  College Completed Proportion  \
County
ALAMEDA                           124.729090                    337.315551
AMADOR                              1.278942                      5.843792
BUTTE                              11.308776                     45.504865
CALAVERAS                           0.807279                      4.768486
COLUSA                              0.544218                      3.555583

           High School Completed Proportion  Less than $10k Proportion  \
County
ALAMEDA                          571.477170                  27.757573
```

```
AMADOR                      16.326510                   0.708266
BUTTE                      114.952245                  10.846003
CALAVERAS                   14.321842                   0.958093
COLUSA                       9.186771                   0.455497

           \$10,000 to $24,999 Proportion  \$25,000 to $49,999 Proportion  \
County
ALAMEDA                         64.808111                       99.569912
AMADOR                           2.823958                        4.344597
BUTTE                           27.269027                       32.119173
CALAVERAS                        2.735701                        4.087452
COLUSA                           2.275792                        2.914379

           \$50,000 to $99,999 Proportion  \$100,000 to $199,999 Proportion  \
County
ALAMEDA                        169.176756                         192.215668
AMADOR                           5.624919                           3.761287
BUTTE                           35.017217                          19.502183
CALAVERAS                        4.103414                           3.293779
COLUSA                           4.531522                           2.312655

           $200,000 or more Proportion  White alone Proportion  \
County
ALAMEDA                     103.471981              284.975869
AMADOR                        0.736974               16.038905
BUTTE                         5.246398              106.434357
CALAVERAS                     0.821561               14.722489
COLUSA                        0.510155               11.215092

           Black or African American alone Proportion  \
County
ALAMEDA                                     81.429975
AMADOR                                       0.235559
BUTTE                                        2.003277
CALAVERAS                                    0.059944
COLUSA                                       0.319549

           American Indian and Alaska Native alone Proportion  \
County
ALAMEDA                                              4.694494
AMADOR                                               0.143752
BUTTE                                                1.647032
CALAVERAS                                            0.125763
COLUSA                                               0.105092

           Asian alone Proportion  \
County
```

```
ALAMEDA                171.097102
AMADOR                   0.194074
BUTTE                    6.153083
CALAVERAS                0.173193
COLUSA                   0.261970

           Native Hawaiian and Other Pacific Islander alone Proportion  \
County
ALAMEDA                                             4.881193
AMADOR                                              0.048812
BUTTE                                               0.223335
CALAVERAS                                           0.061432
COLUSA                                              0.058116

           Some other race alone Proportion  Less than 10 Minutes Proportion  \
County
ALAMEDA                           66.299734                        40.154756
AMADOR                             0.692588                         2.723881
BUTTE                              5.265180                        29.529421
CALAVERAS                          0.144863                         2.215991
COLUSA                             0.710203                         2.979140

           10 to 19 Minutes Proportion  20 to 29 Minutes Proportion  \
County
ALAMEDA                     155.311371                   113.295933
AMADOR                        3.888953                     4.029832
BUTTE                        46.981999                    18.276657
CALAVERAS                     3.329259                     1.740178
COLUSA                        3.418083                     1.815158

           30 to 44 Minutes Proportion  45 to 59 Minutes Proportion  \
County
ALAMEDA                     153.287258                    85.607938
AMADOR                        2.656412                     1.594473
BUTTE                        21.855234                     5.991326
CALAVERAS                     3.505458                     1.775060
COLUSA                        2.011706                     1.046439

           60 to 89 Minutes Proportion  90 or more Minutes Proportion
County
ALAMEDA                      81.368776                      27.973968
AMADOR                        1.838971                       1.267478
BUTTE                         3.349639                       4.015725
CALAVERAS                     1.881623                       1.552430
COLUSA                        1.248202                       0.481272
```

One of the issues that we run into when aggrgating the data this way is that the proportions get summed together, thus, we need to create new proportion columns based on the county aggregated data.

```
[82]: df_county["Proportion EV County"] = df_county["EV Population"]/
       ↪df_county["Vehicle Population"]
      df_county.head()
```

```
[82]:             index  Total Education  Total Income  Total Race  White alone  \
      County
      ALAMEDA    6874021         707835.0      350409.0   1004894.0     421776.0
      AMADOR      178075          19064.0        8795.0     24401.0      20886.0
      BUTTE      1272687          95563.0       57063.0    147851.0     120725.0
      CALAVERAS   154743          19055.0        9299.0     26386.0      24069.0
      COLUSA      125374          10288.0        5167.0     16100.0      14024.0

                 Black or African American alone  \
      County
      ALAMEDA                           113081.0
      AMADOR                               750.0
      BUTTE                               2557.0
      CALAVERAS                            185.0
      COLUSA                               265.0

                 American Indian and Alaska Native alone  Asian alone  \
      County
      ALAMEDA                                     7025.0     287122.0
      AMADOR                                       202.0        236.0
      BUTTE                                       1638.0       7261.0
      CALAVERAS                                    239.0        369.0
      COLUSA                                       115.0        290.0

                 Native Hawaiian and Other Pacific Islander alone  \
      County
      ALAMEDA                                              7637.0
      AMADOR                                                 54.0
      BUTTE                                                 221.0
      CALAVERAS                                             114.0
      COLUSA                                                 53.0

                 Some other race alone  Total Commute  EV Population  \
      County
      ALAMEDA                 103175.0       441870.0         9198.0
      AMADOR                    1190.0         7096.0           21.0
      BUTTE                     6251.0        56110.0          117.0
      CALAVERAS                  205.0         8744.0           10.0
      COLUSA                     920.0         7015.0            0.0
```

```
           Vehicle Population  Proportion EV  More College Completed  \
County
ALAMEDA              358154.0      14.873580                134496.0
AMADOR                 6359.0       0.062305                  1265.0
BUTTE                 39237.0       0.361166                  8329.0
CALAVERAS              4741.0       0.060644                  1057.0
COLUSA                 2647.0       0.000000                   425.0

           College Completed  High School Completed  Less than $10k  \
County
ALAMEDA             366866.0               620005.0         14838.0
AMADOR                5643.0                16836.0           368.0
BUTTE                33777.0                84904.0          4819.0
CALAVERAS             5468.0                17060.0           435.0
COLUSA                2612.0                 6958.0           148.0

           \$10,000 to $24,999  \$25,000 to $49,999  \$50,000 to $99,999  \
County
ALAMEDA                33308.0              51067.0              89149.0
AMADOR                  1362.0               2109.0               2775.0
BUTTE                  11786.0              14092.0              15189.0
CALAVERAS               1357.0               2220.0               2576.0
COLUSA                   861.0               1185.0               1898.0

           \$100,000 to $199,999  $200,000 or more  Less than 10 Minutes  \
County
ALAMEDA                 104452.0           57595.0               27478.0
AMADOR                    1811.0             370.0                1041.0
BUTTE                     8865.0            2312.0               13202.0
CALAVERAS                 2251.0             460.0                1223.0
COLUSA                     882.0             193.0                1556.0

           10 to 19 Minutes  20 to 29 Minutes  30 to 44 Minutes  \
County
ALAMEDA            104435.0           75553.0          101434.0
AMADOR               1598.0            1612.0             966.0
BUTTE               21389.0            7550.0            8381.0
CALAVERAS            1402.0             743.0            1867.0
COLUSA               1843.0            1166.0            1184.0

           45 to 59 Minutes  60 to 89 Minutes  90 or more Minutes  \
County
ALAMEDA             57644.0           56029.0             19297.0
AMADOR                619.0             791.0               469.0
BUTTE                2485.0            1407.0              1696.0
CALAVERAS            1348.0            1368.0               793.0
COLUSA                465.0             524.0               277.0
```

```
           More College Completed Proportion  College Completed Proportion  \
County
ALAMEDA                           124.729090                    337.315551
AMADOR                              1.278942                      5.843792
BUTTE                              11.308776                     45.504865
CALAVERAS                           0.807279                      4.768486
COLUSA                              0.544218                      3.555583

           High School Completed Proportion  Less than $10k Proportion  \
County
ALAMEDA                          571.477170                  27.757573
AMADOR                            16.326510                   0.708266
BUTTE                            114.952245                  10.846003
CALAVERAS                         14.321842                   0.958093
COLUSA                             9.186771                   0.455497

           \$10,000 to $24,999 Proportion  \$25,000 to $49,999 Proportion  \
County
ALAMEDA                         64.808111                       99.569912
AMADOR                           2.823958                        4.344597
BUTTE                           27.269027                       32.119173
CALAVERAS                        2.735701                        4.087452
COLUSA                           2.275792                        2.914379

           \$50,000 to $99,999 Proportion  \$100,000 to $199,999 Proportion  \
County
ALAMEDA                        169.176756                        192.215668
AMADOR                           5.624919                          3.761287
BUTTE                           35.017217                         19.502183
CALAVERAS                        4.103414                          3.293779
COLUSA                           4.531522                          2.312655

           $200,000 or more Proportion  White alone Proportion  \
County
ALAMEDA                     103.471981              284.975869
AMADOR                        0.736974               16.038905
BUTTE                         5.246398              106.434357
CALAVERAS                     0.821561               14.722489
COLUSA                        0.510155               11.215092

           Black or African American alone Proportion  \
County
ALAMEDA                                     81.429975
AMADOR                                       0.235559
BUTTE                                        2.003277
CALAVERAS                                    0.059944
```

```
COLUSA                                       0.319549

           American Indian and Alaska Native alone Proportion  \
County
ALAMEDA                                              4.694494
AMADOR                                               0.143752
BUTTE                                                1.647032
CALAVERAS                                            0.125763
COLUSA                                               0.105092

           Asian alone Proportion  \
County
ALAMEDA                171.097102
AMADOR                   0.194074
BUTTE                    6.153083
CALAVERAS                0.173193
COLUSA                   0.261970

           Native Hawaiian and Other Pacific Islander alone Proportion  \
County
ALAMEDA                                              4.881193
AMADOR                                               0.048812
BUTTE                                                0.223335
CALAVERAS                                            0.061432
COLUSA                                               0.058116

           Some other race alone Proportion  Less than 10 Minutes Proportion  \
County
ALAMEDA                           66.299734                        40.154756
AMADOR                             0.692588                         2.723881
BUTTE                              5.265180                        29.529421
CALAVERAS                          0.144863                         2.215991
COLUSA                             0.710203                         2.979140

           10 to 19 Minutes Proportion  20 to 29 Minutes Proportion  \
County
ALAMEDA                     155.311371                   113.295933
AMADOR                        3.888953                     4.029832
BUTTE                        46.981999                    18.276657
CALAVERAS                     3.329259                     1.740178
COLUSA                        3.418083                     1.815158

           30 to 44 Minutes Proportion  45 to 59 Minutes Proportion  \
County
ALAMEDA                     153.287258                    85.607938
AMADOR                        2.656412                     1.594473
BUTTE                        21.855234                     5.991326
```

```
CALAVERAS                       3.505458                     1.775060
COLUSA                          2.011706                     1.046439

           60 to 89 Minutes Proportion  90 or more Minutes Proportion  \
County
ALAMEDA                      81.368776                      27.973968
AMADOR                        1.838971                       1.267478
BUTTE                         3.349639                       4.015725
CALAVERAS                     1.881623                       1.552430
COLUSA                        1.248202                       0.481272

           Proportion EV County
County
ALAMEDA                0.025682
AMADOR                 0.003302
BUTTE                  0.002982
CALAVERAS              0.002109
COLUSA                 0.000000
```

Now that we've updated Proportion EV County to use aggregated data, we want to do the same for the rest of the features that we're working with, so that our new proportions reflect the aggregated sums.

[83]:

```
#We started by updating all of the race proportions
df_county["White alone county prop"] = df_county["White alone"]/
 ↪df_county["Total Race"]
df_county["Black alone county prop"] = df_county["Black or African American␣
 ↪alone"]/df_county["Total Race"]
df_county["American Indian and Alaskan Native alone county prop"] =␣
 ↪df_county["American Indian and Alaska Native alone"]/df_county["Total Race"]
df_county["Asian alone county prop"] = df_county["Asian alone"]/
 ↪df_county["Total Race"]
df_county["Hawaiian and Islander alone county prop"] = df_county["Native␣
 ↪Hawaiian and Other Pacific Islander alone"]/df_county["Total Race"]
df_county["Some other race alone county prop"] = df_county["Some other race␣
 ↪alone"]/df_county["Total Race"]

#then we update all of the commute proportions
df_county["<10 min county prop"] = df_county["Less than 10 Minutes"]/
 ↪df_county["Total Commute"]
df_county["10-19 min county prop"] = df_county["10 to 19 Minutes"]/
 ↪df_county["Total Commute"]
df_county["20-29 min county prop"] = df_county["20 to 29 Minutes"]/
 ↪df_county["Total Commute"]
df_county["30-44 min county prop"] = df_county["30 to 44 Minutes"]/
 ↪df_county["Total Commute"]
```

```
df_county["45-59 min county prop"] = df_county["45 to 59 Minutes"]/
 ↪df_county["Total Commute"]
df_county["60-89 min county prop"] = df_county["60 to 89 Minutes"]/
 ↪df_county["Total Commute"]
df_county["90+ min county prop"] = df_county["90 or more Minutes"]/
 ↪df_county["Total Commute"]

#then we update all of the education proportions
df_county["high school county prop"] = df_county['High School Completed']/
 ↪df_county['Total Education']
df_county["college county prop"] = df_county['College Completed']/
 ↪df_county['Total Education']
df_county["more college county prop"] = df_county['More College Completed']/
 ↪df_county['Total Education']

#finally, we update all of the income proportions
df_county["<10k county prop"] = df_county['Less than $10k']/df_county["Total␣
 ↪Income"]
df_county["10-25k county prop"] = df_county['\$10,000 to $24,999']/
 ↪df_county["Total Income"]
df_county["25-50k county prop"] = df_county['\$25,000 to $49,999']/
 ↪df_county["Total Income"]
df_county["50-100k county prop"] = df_county['\$50,000 to $99,999']/
 ↪df_county["Total Income"]
df_county["100-200k county prop"] = df_county['\$100,000 to $199,999']/
 ↪df_county["Total Income"]
df_county["200k or more county prop"] = df_county['$200,000 or more']/
 ↪df_county["Total Income"]
```

[84]:

```
df_county.head()
```

[84]:

```
             index  Total Education  Total Income  Total Race  White alone  \
County
ALAMEDA    6874021         707835.0      350409.0   1004894.0     421776.0
AMADOR      178075          19064.0        8795.0     24401.0      20886.0
BUTTE      1272687          95563.0       57063.0    147851.0     120725.0
CALAVERAS   154743          19055.0        9299.0     26386.0      24069.0
COLUSA      125374          10288.0        5167.0     16100.0      14024.0

           Black or African American alone  \
County
ALAMEDA                           113081.0
AMADOR                               750.0
BUTTE                               2557.0
CALAVERAS                            185.0
COLUSA                               265.0
```

```
           American Indian and Alaska Native alone  Asian alone  \
County
ALAMEDA                                     7025.0     287122.0
AMADOR                                       202.0        236.0
BUTTE                                       1638.0       7261.0
CALAVERAS                                    239.0        369.0
COLUSA                                       115.0        290.0

           Native Hawaiian and Other Pacific Islander alone  \
County
ALAMEDA                                              7637.0
AMADOR                                                 54.0
BUTTE                                                 221.0
CALAVERAS                                             114.0
COLUSA                                                 53.0

           Some other race alone  Total Commute  EV Population  \
County
ALAMEDA                 103175.0       441870.0         9198.0
AMADOR                    1190.0         7096.0           21.0
BUTTE                     6251.0        56110.0          117.0
CALAVERAS                  205.0         8744.0           10.0
COLUSA                     920.0         7015.0            0.0

           Vehicle Population  Proportion EV  More College Completed  \
County
ALAMEDA              358154.0      14.873580                134496.0
AMADOR                 6359.0       0.062305                  1265.0
BUTTE                 39237.0       0.361166                  8329.0
CALAVERAS              4741.0       0.060644                  1057.0
COLUSA                 2647.0       0.000000                   425.0

           College Completed  High School Completed  Less than $10k  \
County
ALAMEDA             366866.0               620005.0         14838.0
AMADOR                5643.0                16836.0           368.0
BUTTE                33777.0                84904.0          4819.0
CALAVERAS             5468.0                17060.0           435.0
COLUSA                2612.0                 6958.0           148.0

           \$10,000 to $24,999  \$25,000 to $49,999  \$50,000 to $99,999  \
County
ALAMEDA                33308.0              51067.0              89149.0
AMADOR                  1362.0               2109.0               2775.0
BUTTE                  11786.0              14092.0              15189.0
CALAVERAS               1357.0               2220.0               2576.0
```

```
COLUSA                   861.0               1185.0               1898.0

           \$100,000 to $199,999  $200,000 or more  Less than 10 Minutes  \
County
ALAMEDA                 104452.0           57595.0               27478.0
AMADOR                    1811.0             370.0                1041.0
BUTTE                     8865.0            2312.0               13202.0
CALAVERAS                 2251.0             460.0                1223.0
COLUSA                     882.0             193.0                1556.0

           10 to 19 Minutes  20 to 29 Minutes  30 to 44 Minutes  \
County
ALAMEDA            104435.0           75553.0          101434.0
AMADOR               1598.0            1612.0             966.0
BUTTE               21389.0            7550.0            8381.0
CALAVERAS            1402.0             743.0            1867.0
COLUSA               1843.0            1166.0            1184.0

           45 to 59 Minutes  60 to 89 Minutes  90 or more Minutes  \
County
ALAMEDA             57644.0           56029.0             19297.0
AMADOR                619.0             791.0               469.0
BUTTE                2485.0            1407.0              1696.0
CALAVERAS            1348.0            1368.0               793.0
COLUSA                465.0             524.0               277.0

           More College Completed Proportion  College Completed Proportion  \
County
ALAMEDA                           124.729090                    337.315551
AMADOR                              1.278942                      5.843792
BUTTE                              11.308776                     45.504865
CALAVERAS                           0.807279                      4.768486
COLUSA                              0.544218                      3.555583

           High School Completed Proportion  Less than $10k Proportion  \
County
ALAMEDA                          571.477170                  27.757573
AMADOR                            16.326510                   0.708266
BUTTE                            114.952245                  10.846003
CALAVERAS                         14.321842                   0.958093
COLUSA                             9.186771                   0.455497

           \$10,000 to $24,999 Proportion  \$25,000 to $49,999 Proportion  \
County
ALAMEDA                         64.808111                       99.569912
AMADOR                           2.823958                        4.344597
BUTTE                           27.269027                       32.119173
```

```
CALAVERAS                        2.735701                        4.087452
COLUSA                           2.275792                        2.914379

           \$50,000 to $99,999 Proportion  \$100,000 to $199,999 Proportion  \
County
ALAMEDA                        169.176756                        192.215668
AMADOR                           5.624919                          3.761287
BUTTE                           35.017217                         19.502183
CALAVERAS                        4.103414                          3.293779
COLUSA                           4.531522                          2.312655

           $200,000 or more Proportion  White alone Proportion  \
County
ALAMEDA                     103.471981              284.975869
AMADOR                        0.736974               16.038905
BUTTE                         5.246398              106.434357
CALAVERAS                     0.821561               14.722489
COLUSA                        0.510155               11.215092

           Black or African American alone Proportion  \
County
ALAMEDA                                     81.429975
AMADOR                                       0.235559
BUTTE                                        2.003277
CALAVERAS                                    0.059944
COLUSA                                       0.319549

           American Indian and Alaska Native alone Proportion  \
County
ALAMEDA                                             4.694494
AMADOR                                              0.143752
BUTTE                                               1.647032
CALAVERAS                                           0.125763
COLUSA                                              0.105092

           Asian alone Proportion  \
County
ALAMEDA                171.097102
AMADOR                   0.194074
BUTTE                    6.153083
CALAVERAS                0.173193
COLUSA                   0.261970

           Native Hawaiian and Other Pacific Islander alone Proportion  \
County
ALAMEDA                                             4.881193
AMADOR                                              0.048812
```

```
BUTTE                                               0.223335
CALAVERAS                                           0.061432
COLUSA                                              0.058116

           Some other race alone Proportion  Less than 10 Minutes Proportion  \
County
ALAMEDA                           66.299734                        40.154756
AMADOR                             0.692588                         2.723881
BUTTE                              5.265180                        29.529421
CALAVERAS                          0.144863                         2.215991
COLUSA                             0.710203                         2.979140

           10 to 19 Minutes Proportion  20 to 29 Minutes Proportion  \
County
ALAMEDA                     155.311371                   113.295933
AMADOR                        3.888953                     4.029832
BUTTE                        46.981999                    18.276657
CALAVERAS                     3.329259                     1.740178
COLUSA                        3.418083                     1.815158

           30 to 44 Minutes Proportion  45 to 59 Minutes Proportion  \
County
ALAMEDA                     153.287258                    85.607938
AMADOR                        2.656412                     1.594473
BUTTE                        21.855234                     5.991326
CALAVERAS                     3.505458                     1.775060
COLUSA                        2.011706                     1.046439

           60 to 89 Minutes Proportion  90 or more Minutes Proportion  \
County
ALAMEDA                      81.368776                      27.973968
AMADOR                        1.838971                       1.267478
BUTTE                         3.349639                       4.015725
CALAVERAS                     1.881623                       1.552430
COLUSA                        1.248202                       0.481272

           Proportion EV County  White alone county prop  \
County
ALAMEDA                0.025682                 0.419722
AMADOR                 0.003302                 0.855949
BUTTE                  0.002982                 0.816532
CALAVERAS              0.002109                 0.912188
COLUSA                 0.000000                 0.871056

           Black alone county prop  \
County
ALAMEDA                   0.112530
```

```
AMADOR                    0.030736
BUTTE                     0.017294
CALAVERAS                 0.007011
COLUSA                    0.016460

           American Indian and Alaskan Native alone county prop  \
County
ALAMEDA                                              0.006991
AMADOR                                               0.008278
BUTTE                                                0.011079
CALAVERAS                                            0.009058
COLUSA                                               0.007143

           Asian alone county prop  Hawaiian and Islander alone county prop  \
County
ALAMEDA                   0.285724                                 0.007600
AMADOR                    0.009672                                 0.002213
BUTTE                     0.049110                                 0.001495
CALAVERAS                 0.013985                                 0.004320
COLUSA                    0.018012                                 0.003292

           Some other race alone county prop  <10 min county prop  \
County
ALAMEDA                             0.102673             0.062186
AMADOR                              0.048768             0.146702
BUTTE                               0.042279             0.235288
CALAVERAS                           0.007769             0.139867
COLUSA                              0.057143             0.221810

           10-19 min county prop  20-29 min county prop  \
County
ALAMEDA                 0.236348               0.170985
AMADOR                  0.225197               0.227170
BUTTE                   0.381198               0.134557
CALAVERAS               0.160339               0.084973
COLUSA                  0.262723               0.166215

           30-44 min county prop  45-59 min county prop  \
County
ALAMEDA                 0.229556               0.130455
AMADOR                  0.136133               0.087232
BUTTE                   0.149367               0.044288
CALAVERAS               0.213518               0.154163
COLUSA                  0.168781               0.066287

           60-89 min county prop  90+ min county prop  \
County
```

```
ALAMEDA                 0.126800             0.043671
AMADOR                  0.111471             0.066094
BUTTE                   0.025076             0.030226
CALAVERAS               0.156450             0.090691
COLUSA                  0.074697             0.039487

           high school county prop  college county prop  \
County
ALAMEDA                   0.875917             0.518293
AMADOR                    0.883131             0.296003
BUTTE                     0.888461             0.353453
CALAVERAS                 0.895303             0.286959
COLUSA                    0.676322             0.253888

           more college county prop  <10k county prop  10-25k county prop  \
County
ALAMEDA                    0.190010          0.042345            0.095055
AMADOR                     0.066355          0.041842            0.154861
BUTTE                      0.087157          0.084451            0.206544
CALAVERAS                  0.055471          0.046779            0.145930
COLUSA                     0.041310          0.028643            0.166634

           25-50k county prop  50-100k county prop  100-200k county prop  \
County
ALAMEDA              0.145735             0.254414              0.298086
AMADOR               0.239795             0.315520              0.205912
BUTTE                0.246955             0.266179              0.155355
CALAVERAS            0.238735             0.277019              0.242069
COLUSA               0.229340             0.367331              0.170699

           200k or more county prop
County
ALAMEDA                    0.164365
AMADOR                     0.042069
BUTTE                      0.040517
CALAVERAS                  0.049468
COLUSA                     0.037352
```

We will now drop the columns containing the incorrectly aggregated proportions

```
[85]: df_county.drop(['More College Completed Proportion', 'College Completed␣
       ↪Proportion',
             'High School Completed Proportion', 'Less than $10k Proportion',
             '\$10,000 to $24,999 Proportion', '\$25,000 to $49,999 Proportion',
             '\$50,000 to $99,999 Proportion', '\$100,000 to $199,999 Proportion',
             '$200,000 or more Proportion', 'White alone Proportion',
             'Black or African American alone Proportion',
             'American Indian and Alaska Native alone Proportion',
             'Asian alone Proportion',
             'Native Hawaiian and Other Pacific Islander alone Proportion',
             'Some other race alone Proportion', 'Less than 10 Minutes Proportion',
             '10 to 19 Minutes Proportion', '20 to 29 Minutes Proportion',
             '30 to 44 Minutes Proportion', '45 to 59 Minutes Proportion',
             '60 to 89 Minutes Proportion', '90 or more Minutes Proportion'],axis =␣
     ↪1,inplace = True)
```

[86]:
```
df_county.head()
```

[86]:
```
             index  Total Education  Total Income  Total Race  White alone  \
County
ALAMEDA    6874021         707835.0      350409.0   1004894.0     421776.0
AMADOR      178075          19064.0        8795.0     24401.0      20886.0
BUTTE      1272687          95563.0       57063.0    147851.0     120725.0
CALAVERAS   154743          19055.0        9299.0     26386.0      24069.0
COLUSA      125374          10288.0        5167.0     16100.0      14024.0

           Black or African American alone  \
County
ALAMEDA                           113081.0
AMADOR                               750.0
BUTTE                               2557.0
CALAVERAS                            185.0
COLUSA                               265.0

           American Indian and Alaska Native alone  Asian alone  \
County
ALAMEDA                                     7025.0     287122.0
AMADOR                                       202.0        236.0
BUTTE                                       1638.0       7261.0
CALAVERAS                                    239.0        369.0
COLUSA                                       115.0        290.0

           Native Hawaiian and Other Pacific Islander alone  \
County
ALAMEDA                                              7637.0
AMADOR                                                 54.0
BUTTE                                                 221.0
CALAVERAS                                             114.0
COLUSA                                                 53.0

           Some other race alone  Total Commute  EV Population  \
County
ALAMEDA                 103175.0       441870.0         9198.0
AMADOR                    1190.0         7096.0           21.0
```

```
BUTTE                     6251.0        56110.0          117.0
CALAVERAS                  205.0         8744.0           10.0
COLUSA                     920.0         7015.0            0.0

           Vehicle Population  Proportion EV  More College Completed  \
County
ALAMEDA              358154.0      14.873580                134496.0
AMADOR                 6359.0       0.062305                  1265.0
BUTTE                 39237.0       0.361166                  8329.0
CALAVERAS              4741.0       0.060644                  1057.0
COLUSA                 2647.0       0.000000                   425.0

           College Completed  High School Completed  Less than $10k  \
County
ALAMEDA             366866.0               620005.0         14838.0
AMADOR                5643.0                16836.0           368.0
BUTTE                33777.0                84904.0          4819.0
CALAVERAS             5468.0                17060.0           435.0
COLUSA                2612.0                 6958.0           148.0

           \$10,000 to $24,999  \$25,000 to $49,999  \$50,000 to $99,999  \
County
ALAMEDA                33308.0              51067.0              89149.0
AMADOR                  1362.0               2109.0               2775.0
BUTTE                  11786.0              14092.0              15189.0
CALAVERAS               1357.0               2220.0               2576.0
COLUSA                   861.0               1185.0               1898.0

           \$100,000 to $199,999  $200,000 or more  Less than 10 Minutes  \
County
ALAMEDA                 104452.0           57595.0               27478.0
AMADOR                    1811.0             370.0                1041.0
BUTTE                     8865.0            2312.0               13202.0
CALAVERAS                 2251.0             460.0                1223.0
COLUSA                     882.0             193.0                1556.0

           10 to 19 Minutes  20 to 29 Minutes  30 to 44 Minutes  \
County
ALAMEDA            104435.0           75553.0          101434.0
AMADOR               1598.0            1612.0             966.0
BUTTE               21389.0            7550.0            8381.0
CALAVERAS            1402.0             743.0            1867.0
COLUSA               1843.0            1166.0            1184.0

           45 to 59 Minutes  60 to 89 Minutes  90 or more Minutes  \
County
ALAMEDA             57644.0           56029.0             19297.0
```

```
AMADOR                619.0             791.0               469.0
BUTTE                2485.0            1407.0              1696.0
CALAVERAS            1348.0            1368.0               793.0
COLUSA                465.0             524.0               277.0

           Proportion EV County  White alone county prop  \
County
ALAMEDA                0.025682                 0.419722
AMADOR                 0.003302                 0.855949
BUTTE                  0.002982                 0.816532
CALAVERAS              0.002109                 0.912188
COLUSA                 0.000000                 0.871056

           Black alone county prop  \
County
ALAMEDA                   0.112530
AMADOR                    0.030736
BUTTE                     0.017294
CALAVERAS                 0.007011
COLUSA                    0.016460

           American Indian and Alaskan Native alone county prop  \
County
ALAMEDA                                             0.006991
AMADOR                                              0.008278
BUTTE                                               0.011079
CALAVERAS                                           0.009058
COLUSA                                              0.007143

           Asian alone county prop  Hawaiian and Islander alone county prop  \
County
ALAMEDA                   0.285724                                 0.007600
AMADOR                    0.009672                                 0.002213
BUTTE                     0.049110                                 0.001495
CALAVERAS                 0.013985                                 0.004320
COLUSA                    0.018012                                 0.003292

           Some other race alone county prop  <10 min county prop  \
County
ALAMEDA                             0.102673             0.062186
AMADOR                              0.048768             0.146702
BUTTE                               0.042279             0.235288
CALAVERAS                           0.007769             0.139867
COLUSA                              0.057143             0.221810

           10-19 min county prop  20-29 min county prop  \
County
```

```
ALAMEDA                 0.236348               0.170985
AMADOR                  0.225197               0.227170
BUTTE                   0.381198               0.134557
CALAVERAS               0.160339               0.084973
COLUSA                  0.262723               0.166215

           30-44 min county prop  45-59 min county prop  \
County
ALAMEDA                 0.229556               0.130455
AMADOR                  0.136133               0.087232
BUTTE                   0.149367               0.044288
CALAVERAS               0.213518               0.154163
COLUSA                  0.168781               0.066287

           60-89 min county prop  90+ min county prop  \
County
ALAMEDA                 0.126800             0.043671
AMADOR                  0.111471             0.066094
BUTTE                   0.025076             0.030226
CALAVERAS               0.156450             0.090691
COLUSA                  0.074697             0.039487

           high school county prop  college county prop  \
County
ALAMEDA                   0.875917             0.518293
AMADOR                    0.883131             0.296003
BUTTE                     0.888461             0.353453
CALAVERAS                 0.895303             0.286959
COLUSA                    0.676322             0.253888

           more college county prop  <10k county prop  10-25k county prop  \
County
ALAMEDA                    0.190010          0.042345            0.095055
AMADOR                     0.066355          0.041842            0.154861
BUTTE                      0.087157          0.084451            0.206544
CALAVERAS                  0.055471          0.046779            0.145930
COLUSA                     0.041310          0.028643            0.166634

           25-50k county prop  50-100k county prop  100-200k county prop  \
County
ALAMEDA              0.145735             0.254414              0.298086
AMADOR               0.239795             0.315520              0.205912
BUTTE                0.246955             0.266179              0.155355
CALAVERAS            0.238735             0.277019              0.242069
COLUSA               0.229340             0.367331              0.170699

           200k or more county prop
```

```
County
ALAMEDA                     0.164365
AMADOR                      0.042069
BUTTE                       0.040517
CALAVERAS                   0.049468
COLUSA                      0.037352
```

For this section of the notebook, we will be using **Proportion EV County** as our response variable. As we've discussed earlier, this variable represents the proportion of EVs registered in a county divided by the total number of passenger vehicles in the same county.

For illlustrative purposes, we wanted to show some information about the distribution of this variable.

```
[87]: df_county["Proportion EV County"].describe()
```

```
[87]: count    56.000000
      mean      0.008043
      std       0.008942
      min       0.000000
      25%       0.002435
      50%       0.004562
      75%       0.010816
      max       0.040570
      Name: Proportion EV County, dtype: float64
```

Based on this output, we can see that the average county has a 0.8% proportion of EVs. In addition, we can see that there are certain counties that have zero electric vehicle adoption. Meaanwhile, the country with the highest rate of adoption has a ~ 4% proportion of EVs.

Before building the model, we wanted to dig further into the counties with the smallest and largest proportions of EVs.

```
[88]: df_county.nsmallest(10,"Proportion EV County").loc[:,"Proportion EV County":
       ↪"200k or more county prop"]
```

```
[88]:            Proportion EV County  White alone county prop  \
      County
      COLUSA                 0.000000                 0.871056
      LASSEN                 0.000000                 0.893091
      MODOC                  0.000000                 0.905880
      MONO                   0.000000                 0.827771
      TRINITY                0.000000                 0.821377
      GLENN                  0.000618                 0.834270
      IMPERIAL               0.000704                 0.666029
      YUBA                   0.001396                 0.751891
      TEHAMA                 0.001697                 0.856640
```

```
CALAVERAS              0.002109                 0.912188

           Black alone county prop  \
County
COLUSA                    0.016460
LASSEN                    0.005419
MODOC                     0.004505
MONO                      0.012275
TRINITY                   0.030846
GLENN                     0.004729
IMPERIAL                  0.021305
YUBA                      0.031674
TEHAMA                    0.004384
CALAVERAS                 0.007011

           American Indian and Alaskan Native alone county prop  \
County
COLUSA                                                 0.007143
LASSEN                                                 0.025401
MODOC                                                  0.035799
MONO                                                   0.067127
TRINITY                                                0.012912
GLENN                                                  0.028095
IMPERIAL                                               0.009930
YUBA                                                   0.013260
TEHAMA                                                 0.026923
CALAVERAS                                              0.009058

           Asian alone county prop  Hawaiian and Islander alone county prop  \
County
COLUSA                    0.018012                                 0.003292
LASSEN                    0.012192                                 0.016934
MODOC                     0.002371                                 0.000000
MONO                      0.009973                                 0.000000
TRINITY                   0.076040                                 0.000000
GLENN                     0.044951                                 0.002837
IMPERIAL                  0.006696                                 0.002100
YUBA                      0.061432                                 0.003522
TEHAMA                    0.013313                                 0.000206
CALAVERAS                 0.013985                                 0.004320

           Some other race alone county prop  <10 min county prop  \
County
COLUSA                              0.057143             0.221810
LASSEN                              0.017498             0.262437
MODOC                               0.007349             0.298913
MONO                                0.050249             0.145702
```

```
TRINITY                             0.007174             0.122642
GLENN                               0.059694             0.255860
IMPERIAL                            0.246475             0.203552
YUBA                                0.056200             0.102312
TEHAMA                              0.063847             0.197684
CALAVERAS                           0.007769             0.139867

           10-19 min county prop  20-29 min county prop  \
County
COLUSA                  0.262723               0.166215
LASSEN                  0.362228               0.200477
MODOC                   0.419384               0.153986
MONO                    0.354298               0.250524
TRINITY                 0.316038               0.127358
GLENN                   0.257290               0.196255
IMPERIAL                0.371264               0.198109
YUBA                    0.291169               0.159231
TEHAMA                  0.275411               0.192586
CALAVERAS               0.160339               0.084973

           30-44 min county prop  45-59 min county prop  \
County
COLUSA                  0.168781               0.066287
LASSEN                  0.096217               0.039321
MODOC                   0.062500               0.041667
MONO                    0.244235               0.005241
TRINITY                 0.186321               0.096698
GLENN                   0.218553               0.042882
IMPERIAL                0.135054               0.028742
YUBA                    0.183907               0.131302
TEHAMA                  0.218957               0.047769
CALAVERAS               0.213518               0.154163

           60-89 min county prop  90+ min county prop  \
County
COLUSA                  0.074697             0.039487
LASSEN                  0.020554             0.018767
MODOC                   0.014493             0.009058
MONO                    0.000000             0.000000
TRINITY                 0.014151             0.136792
GLENN                   0.006146             0.023013
IMPERIAL                0.031575             0.031703
YUBA                    0.090363             0.041717
TEHAMA                  0.025489             0.042105
CALAVERAS               0.156450             0.090691

           high school county prop  college county prop  \
County
COLUSA                    0.676322             0.253888
LASSEN                    0.930802             0.328238
MODOC                     0.882008             0.189048
MONO                      0.901728             0.367407
TRINITY                   0.893028             0.225406
GLENN                     0.768179             0.248875
IMPERIAL                  0.692120             0.212923
YUBA                      0.804675             0.271418
TEHAMA                    0.831506             0.231526
CALAVERAS                 0.895303             0.286959

           more college county prop  <10k county prop  10-25k county prop  \
County
COLUSA                     0.041310          0.028643            0.166634
LASSEN                     0.071992          0.055297            0.174677
MODOC                      0.023142          0.060850            0.187141
MONO                       0.107654          0.036520            0.080559
TRINITY                    0.015282          0.055046            0.282569
GLENN                      0.031077          0.095080            0.187896
IMPERIAL                   0.047521          0.072379            0.243500
YUBA                       0.042287          0.056523            0.183538
TEHAMA                     0.041965          0.070740            0.236119
CALAVERAS                  0.055471          0.046779            0.145930

           25-50k county prop  50-100k county prop  100-200k county prop  \
County
COLUSA               0.229340             0.367331              0.170699
LASSEN               0.188372             0.305685              0.258656
MODOC                0.273249             0.374282              0.104478
MONO                 0.203008             0.368421              0.265306
TRINITY              0.271560             0.247706              0.143119
GLENN                0.226834             0.316481              0.157410
IMPERIAL             0.235534             0.279002              0.139946
YUBA                 0.257826             0.305036              0.178308
TEHAMA               0.250854             0.290866              0.126026
CALAVERAS            0.238735             0.277019              0.242069

           200k or more county prop
County
COLUSA                     0.037352
LASSEN                     0.017313
MODOC                      0.000000
MONO                       0.046187
TRINITY                    0.000000
GLENN                      0.016299
IMPERIAL                   0.029639
```

```
YUBA                       0.018769
TEHAMA                     0.025397
CALAVERAS                  0.049468
```

Based on these results, we can see that there are 5 counties (Colusa, Lassen, Modoc, Mono, and Trinity) that had no EV registrations in 2018. Just from a cursory look at the names of these counties, these are majority white, largely rural counties that are likely to have far different charcteristics to more urban counties or other counties with higher levels of EV adoption.

We then looked at the top 10 counties with the highest proportions of electric vehicles.

```
[89]: df_county.nlargest(10,"Proportion EV County").loc[:,"Proportion EV County":
       ↪"200k or more county prop"]
```

```
[89]:                Proportion EV County  White alone county prop  \
      County
      SANTA CLARA                0.040570                 0.449802
      MARIN                      0.035930                 0.796220
      SAN MATEO                  0.031368                 0.516236
      ALAMEDA                    0.025682                 0.419722
      SAN FRANCISCO              0.022887                 0.464016
      SANTA CRUZ                 0.017866                 0.768085
      SONOMA                     0.017822                 0.747782
      ORANGE                     0.016476                 0.615581
      CONTRA COSTA               0.015127                 0.575338
      NAPA                       0.013399                 0.713482

                     Black alone county prop  \
      County
      SANTA CLARA                   0.022942
      MARIN                         0.020072
      SAN MATEO                     0.019854
      ALAMEDA                       0.112530
      SAN FRANCISCO                 0.050457
      SANTA CRUZ                    0.008767
      SONOMA                        0.016063
      ORANGE                        0.017949
      CONTRA COSTA                  0.084898
      NAPA                          0.024587

                     American Indian and Alaskan Native alone county prop  \
      County
      SANTA CLARA                                             0.005152
      MARIN                                                   0.002819
      SAN MATEO                                               0.003743
      ALAMEDA                                                 0.006991
```

```
SAN FRANCISCO                                           0.003195
SANTA CRUZ                                              0.004925
SONOMA                                                  0.009915
ORANGE                                                  0.004713
CONTRA COSTA                                            0.005456
NAPA                                                    0.006808

               Asian alone county prop  \
County
SANTA CLARA                   0.361642
MARIN                         0.061255
SAN MATEO                     0.289880
ALAMEDA                       0.285724
SAN FRANCISCO                 0.347974
SANTA CRUZ                    0.036752
SONOMA                        0.039168
ORANGE                        0.207773
CONTRA COSTA                  0.147214
NAPA                          0.077773

               Hawaiian and Islander alone county prop  \
County
SANTA CLARA                                   0.004303
MARIN                                         0.001280
SAN MATEO                                     0.010604
ALAMEDA                                       0.007600
SAN FRANCISCO                                 0.003054
SANTA CRUZ                                    0.001118
SONOMA                                        0.003285
ORANGE                                        0.002920
CONTRA COSTA                                  0.005170
NAPA                                          0.002562

               Some other race alone county prop  <10 min county prop  \
County
SANTA CLARA                             0.106218             0.066056
MARIN                                   0.068940             0.117297
SAN MATEO                               0.102064             0.076897
ALAMEDA                                 0.102673             0.062186
SAN FRANCISCO                           0.076192             0.037973
SANTA CRUZ                              0.137498             0.150943
SONOMA                                  0.126560             0.153787
ORANGE                                  0.110040             0.080997
CONTRA COSTA                            0.110251             0.074841
NAPA                                    0.134469             0.184931

               10-19 min county prop  20-29 min county prop  \
County
SANTA CLARA                 0.253488               0.241305
MARIN                       0.248424               0.152739
SAN MATEO                   0.252934               0.199517
ALAMEDA                     0.236348               0.170985
SAN FRANCISCO               0.181426               0.210924
SANTA CRUZ                  0.312316               0.161267
SONOMA                      0.323584               0.201814
ORANGE                      0.274022               0.225874
CONTRA COSTA                0.220202               0.140356
NAPA                        0.339529               0.164817

               30-44 min county prop  45-59 min county prop  \
County
SANTA CLARA                 0.250162               0.092031
MARIN                       0.194022               0.122356
SAN MATEO                   0.254005               0.114507
ALAMEDA                     0.229556               0.130455
SAN FRANCISCO               0.304152               0.116439
SANTA CRUZ                  0.166168               0.086116
SONOMA                      0.167357               0.056363
ORANGE                      0.245268               0.079431
CONTRA COSTA                0.196865               0.124095
NAPA                        0.156752               0.056935

               60-89 min county prop  90+ min county prop  \
County
SANTA CLARA                 0.070285             0.026675
MARIN                       0.120834             0.044327
SAN MATEO                   0.084334             0.017806
ALAMEDA                     0.126800             0.043671
SAN FRANCISCO               0.112478             0.036609
SANTA CRUZ                  0.088480             0.034711
SONOMA                      0.058868             0.038227
ORANGE                      0.066704             0.027705
CONTRA COSTA                0.161612             0.082030
NAPA                        0.064073             0.032963

               high school county prop  college county prop  \
County
SANTA CLARA                   0.882955             0.589518
MARIN                         0.944249             0.679386
SAN MATEO                     0.894299             0.571238
ALAMEDA                       0.875917             0.518293
SAN FRANCISCO                 0.883701             0.625228
SANTA CRUZ                    0.857490             0.480897
SONOMA                        0.880175             0.442620
```

```
ORANGE                        0.852650             0.478522
CONTRA COSTA                  0.884032             0.486210
NAPA                          0.843803             0.429290

               more college county prop  <10k county prop  10-25k county prop  \
County
SANTA CLARA                    0.244866          0.033423            0.068334
MARIN                          0.266620          0.035959            0.075291
SAN MATEO                      0.210821          0.030359            0.069072
ALAMEDA                        0.190010          0.042345            0.095055
SAN FRANCISCO                  0.230125          0.049530            0.113109
SANTA CRUZ                     0.155729          0.053537            0.118868
SONOMA                         0.128017          0.038014            0.102954
ORANGE                         0.142515          0.042930            0.089916
CONTRA COSTA                   0.147351          0.034663            0.084863
NAPA                           0.114050          0.030536            0.089157

               25-50k county prop  50-100k county prop  100-200k county prop  \
County
SANTA CLARA              0.115066             0.214981              0.309931
MARIN                    0.122804             0.221234              0.277614
SAN MATEO                0.119791             0.221453              0.302024
ALAMEDA                  0.145735             0.254414              0.298086
SAN FRANCISCO            0.114420             0.201070              0.273241
SANTA CRUZ               0.161121             0.278424              0.251227
SONOMA                   0.171282             0.313537              0.274963
ORANGE                   0.155273             0.278521              0.289175
CONTRA COSTA             0.153337             0.266865              0.299757
NAPA                     0.168715             0.288681              0.292460

               200k or more county prop
County
SANTA CLARA                    0.258266
MARIN                          0.267098
SAN MATEO                      0.257302
ALAMEDA                        0.164365
SAN FRANCISCO                  0.248630
SANTA CRUZ                     0.136823
SONOMA                         0.099251
ORANGE                         0.144185
CONTRA COSTA                   0.160514
NAPA                           0.130451
```

**Building the model** We will begin by first splitting our data into a test and training dataset. Then, we will test out a range of different models in order to determine, which has the best performance. It is worth noting that when we aggregate our data, we only have 56 observations - one for each county (3 counties are dropped, since they don't publish any EV data). While this is a large enough sample size to run the models, we wanted to raise this caveat as it may affect the model's performance. The small number of observations also means that our model is more significantly impacted by factors like the size of the train test split or the randomness associated with such a process.

[90]:
```
df_county.shape
```

[90]:
```
(56, 53)
```

[91]:
```
df_county.head()
```

[91]:
```
             index  Total Education  Total Income  Total Race  White alone  \
County
ALAMEDA    6874021         707835.0      350409.0   1004894.0     421776.0
AMADOR      178075          19064.0        8795.0     24401.0      20886.0
BUTTE      1272687          95563.0       57063.0    147851.0     120725.0
CALAVERAS   154743          19055.0        9299.0     26386.0      24069.0
COLUSA      125374          10288.0        5167.0     16100.0      14024.0

           Black or African American alone  \
County
ALAMEDA                           113081.0
AMADOR                               750.0
BUTTE                               2557.0
CALAVERAS                            185.0
COLUSA                               265.0

           American Indian and Alaska Native alone  Asian alone  \
County
ALAMEDA                                     7025.0     287122.0
AMADOR                                       202.0        236.0
BUTTE                                       1638.0       7261.0
CALAVERAS                                    239.0        369.0
COLUSA                                       115.0        290.0

           Native Hawaiian and Other Pacific Islander alone  \
County
ALAMEDA                                              7637.0
AMADOR                                                 54.0
BUTTE                                                 221.0
CALAVERAS                                             114.0
COLUSA                                                 53.0

           Some other race alone  Total Commute  EV Population  \
County
ALAMEDA                 103175.0       441870.0         9198.0
```

```
AMADOR                    1190.0         7096.0           21.0
BUTTE                     6251.0        56110.0          117.0
CALAVERAS                  205.0         8744.0           10.0
COLUSA                     920.0         7015.0            0.0

           Vehicle Population  Proportion EV  More College Completed  \
County
ALAMEDA              358154.0      14.873580                134496.0
AMADOR                 6359.0       0.062305                  1265.0
BUTTE                 39237.0       0.361166                  8329.0
CALAVERAS              4741.0       0.060644                  1057.0
COLUSA                 2647.0       0.000000                   425.0

           College Completed  High School Completed  Less than $10k  \
County
ALAMEDA             366866.0               620005.0         14838.0
AMADOR                5643.0                16836.0           368.0
BUTTE                33777.0                84904.0          4819.0
CALAVERAS             5468.0                17060.0           435.0
COLUSA                2612.0                 6958.0           148.0

           \$10,000 to $24,999  \$25,000 to $49,999  \$50,000 to $99,999  \
County
ALAMEDA                33308.0              51067.0              89149.0
AMADOR                  1362.0               2109.0               2775.0
BUTTE                  11786.0              14092.0              15189.0
CALAVERAS               1357.0               2220.0               2576.0
COLUSA                   861.0               1185.0               1898.0

           \$100,000 to $199,999  $200,000 or more  Less than 10 Minutes  \
County
ALAMEDA                 104452.0           57595.0               27478.0
AMADOR                    1811.0             370.0                1041.0
BUTTE                     8865.0            2312.0               13202.0
CALAVERAS                 2251.0             460.0                1223.0
COLUSA                     882.0             193.0                1556.0

           10 to 19 Minutes  20 to 29 Minutes  30 to 44 Minutes  \
County
ALAMEDA            104435.0           75553.0          101434.0
AMADOR               1598.0            1612.0             966.0
BUTTE               21389.0            7550.0            8381.0
CALAVERAS            1402.0             743.0            1867.0
COLUSA               1843.0            1166.0            1184.0

           45 to 59 Minutes  60 to 89 Minutes  90 or more Minutes  \
County
```

```
ALAMEDA             57644.0           56029.0             19297.0
AMADOR                619.0             791.0               469.0
BUTTE                2485.0            1407.0              1696.0
CALAVERAS            1348.0            1368.0               793.0
COLUSA                465.0             524.0               277.0

           Proportion EV County  White alone county prop  \
County
ALAMEDA                0.025682                 0.419722
AMADOR                 0.003302                 0.855949
BUTTE                  0.002982                 0.816532
CALAVERAS              0.002109                 0.912188
COLUSA                 0.000000                 0.871056

           Black alone county prop  \
County
ALAMEDA                   0.112530
AMADOR                    0.030736
BUTTE                     0.017294
CALAVERAS                 0.007011
COLUSA                    0.016460

           American Indian and Alaskan Native alone county prop  \
County
ALAMEDA                                             0.006991
AMADOR                                              0.008278
BUTTE                                               0.011079
CALAVERAS                                           0.009058
COLUSA                                              0.007143

           Asian alone county prop  Hawaiian and Islander alone county prop  \
County
ALAMEDA                   0.285724                                 0.007600
AMADOR                    0.009672                                 0.002213
BUTTE                     0.049110                                 0.001495
CALAVERAS                 0.013985                                 0.004320
COLUSA                    0.018012                                 0.003292

           Some other race alone county prop  <10 min county prop  \
County
ALAMEDA                             0.102673             0.062186
AMADOR                              0.048768             0.146702
BUTTE                               0.042279             0.235288
CALAVERAS                           0.007769             0.139867
COLUSA                              0.057143             0.221810

           10-19 min county prop  20-29 min county prop  \
County                                        
ALAMEDA                 0.236348               0.170985
AMADOR                  0.225197               0.227170
BUTTE                   0.381198               0.134557
CALAVERAS               0.160339               0.084973
COLUSA                  0.262723               0.166215

           30-44 min county prop  45-59 min county prop  \
County                                                    
ALAMEDA                 0.229556               0.130455
AMADOR                  0.136133               0.087232
BUTTE                   0.149367               0.044288
CALAVERAS               0.213518               0.154163
COLUSA                  0.168781               0.066287

           60-89 min county prop  90+ min county prop  \
County                                                  
ALAMEDA                 0.126800             0.043671
AMADOR                  0.111471             0.066094
BUTTE                   0.025076             0.030226
CALAVERAS               0.156450             0.090691
COLUSA                  0.074697             0.039487

           high school county prop  college county prop  \
County                                                    
ALAMEDA                   0.875917             0.518293
AMADOR                    0.883131             0.296003
BUTTE                     0.888461             0.353453
CALAVERAS                 0.895303             0.286959
COLUSA                    0.676322             0.253888

           more college county prop  <10k county prop  10-25k county prop  \
County                                                                      
ALAMEDA                    0.190010          0.042345            0.095055
AMADOR                     0.066355          0.041842            0.154861
BUTTE                      0.087157          0.084451            0.206544
CALAVERAS                  0.055471          0.046779            0.145930
COLUSA                     0.041310          0.028643            0.166634

           25-50k county prop  50-100k county prop  100-200k county prop  \
County                                                                     
ALAMEDA              0.145735             0.254414              0.298086
AMADOR               0.239795             0.315520              0.205912
BUTTE                0.246955             0.266179              0.155355
CALAVERAS            0.238735             0.277019              0.242069
COLUSA               0.229340             0.367331              0.170699
```

```
           200k or more county prop
County
ALAMEDA                    0.164365
AMADOR                     0.042069
BUTTE                      0.040517
CALAVERAS                  0.049468
COLUSA                     0.037352
```

[92]:

```
#set random state = 1
random = 1
```

The above dataframe still cotnains certain variables like the county name and other non-proportion variables that we don't want to include in our training data. Hence, we remove all of the columns containing data that's not a proportion.

[93]:

```
#we drop all the columns that aren't proportion variables
y = df_county[["Proportion EV County"]] #select reeponse variable
X = df_county.loc[:,"White alone county prop":"200k or more county prop"]
    #selecting all the columns between this range gives us only the proportion␣
 ↪variables
```

[94]:

```
X.head()
```

[94]:

```
           White alone county prop  Black alone county prop  \
County
ALAMEDA                   0.419722                 0.112530
AMADOR                    0.855949                 0.030736
BUTTE                     0.816532                 0.017294
CALAVERAS                 0.912188                 0.007011
COLUSA                    0.871056                 0.016460

           American Indian and Alaskan Native alone county prop  \
County
ALAMEDA                                             0.006991
AMADOR                                              0.008278
BUTTE                                               0.011079
CALAVERAS                                           0.009058
COLUSA                                              0.007143

           Asian alone county prop  Hawaiian and Islander alone county prop  \
County
ALAMEDA                   0.285724                                 0.007600
AMADOR                    0.009672                                 0.002213
BUTTE                     0.049110                                 0.001495
CALAVERAS                 0.013985                                 0.004320
COLUSA                    0.018012                                 0.003292
```

```
           Some other race alone county prop  <10 min county prop  \
County
ALAMEDA                             0.102673             0.062186
AMADOR                              0.048768             0.146702
BUTTE                               0.042279             0.235288
CALAVERAS                           0.007769             0.139867
COLUSA                              0.057143             0.221810

           10-19 min county prop  20-29 min county prop  \
County
ALAMEDA                 0.236348               0.170985
AMADOR                  0.225197               0.227170
BUTTE                   0.381198               0.134557
CALAVERAS               0.160339               0.084973
COLUSA                  0.262723               0.166215

           30-44 min county prop  45-59 min county prop  \
County
ALAMEDA                 0.229556               0.130455
AMADOR                  0.136133               0.087232
BUTTE                   0.149367               0.044288
CALAVERAS               0.213518               0.154163
COLUSA                  0.168781               0.066287

           60-89 min county prop  90+ min county prop  \
County
ALAMEDA                 0.126800             0.043671
AMADOR                  0.111471             0.066094
BUTTE                   0.025076             0.030226
CALAVERAS               0.156450             0.090691
COLUSA                  0.074697             0.039487

           high school county prop  college county prop  \
County
ALAMEDA                   0.875917             0.518293
AMADOR                    0.883131             0.296003
BUTTE                     0.888461             0.353453
CALAVERAS                 0.895303             0.286959
COLUSA                    0.676322             0.253888

           more college county prop  <10k county prop  10-25k county prop  \
County
ALAMEDA                    0.190010          0.042345            0.095055
AMADOR                     0.066355          0.041842            0.154861
BUTTE                      0.087157          0.084451            0.206544
CALAVERAS                  0.055471          0.046779            0.145930
```

```
COLUSA                     0.041310          0.028643            0.166634

           25-50k county prop  50-100k county prop  100-200k county prop  \
County
ALAMEDA              0.145735             0.254414              0.298086
AMADOR               0.239795             0.315520              0.205912
BUTTE                0.246955             0.266179              0.155355
CALAVERAS            0.238735             0.277019              0.242069
COLUSA               0.229340             0.367331              0.170699

           200k or more county prop
County
ALAMEDA                    0.164365
AMADOR                     0.042069
BUTTE                      0.040517
CALAVERAS                  0.049468
COLUSA                     0.037352
```

**Data Standardization** Like in the previous prediction question, we want to standardize the values for all of our features before we perform Lasso and Ridge regression.

[95]:
```python
scaler = StandardScaler() #initializ scaler
#standardize features
scaler.fit(X)
X_stnd = scaler.transform(X)
#standardize response variables
scaler.fit(y)
y_stnd = scaler.transform(y)
```

[96]:
```python
X_train, X_test, y_train, y_test = train_test_split(X_stnd,y_stnd,test_size = 0.
 ↪3, random_state = random)
```

**Prediction Model 1.1 - MLR** We started by using a MLR model to determine a baseline for our model, prior to introducing any sort of regularized or non-parametric methods.

[97]:
```python
MLR_model = LinearRegression()
MLR_fit = MLR_model.fit(X_train, y_train)
y_pred_test = MLR_fit.predict(X_test)
y_pred_train = MLR_fit.predict(X_train)

print("test RMSE:", mean_squared_error(y_test, y_pred_test, squared = False))
print("train RMSE:", mean_squared_error(y_train, y_pred_train, squared = False))
print("R^2 test:", r2_score(y_test, y_pred_test))
print("R^2 train:", r2_score(y_train, y_pred_train))
```

```
#MLR_coefs = MLR_fit.coef_
#MLR_coefs
```

```
test RMSE: 0.5784911640954038
train RMSE: 0.14960310586437423
R^2 test: 0.8329758542993702
R^2 train: 0.9430630973025429
```

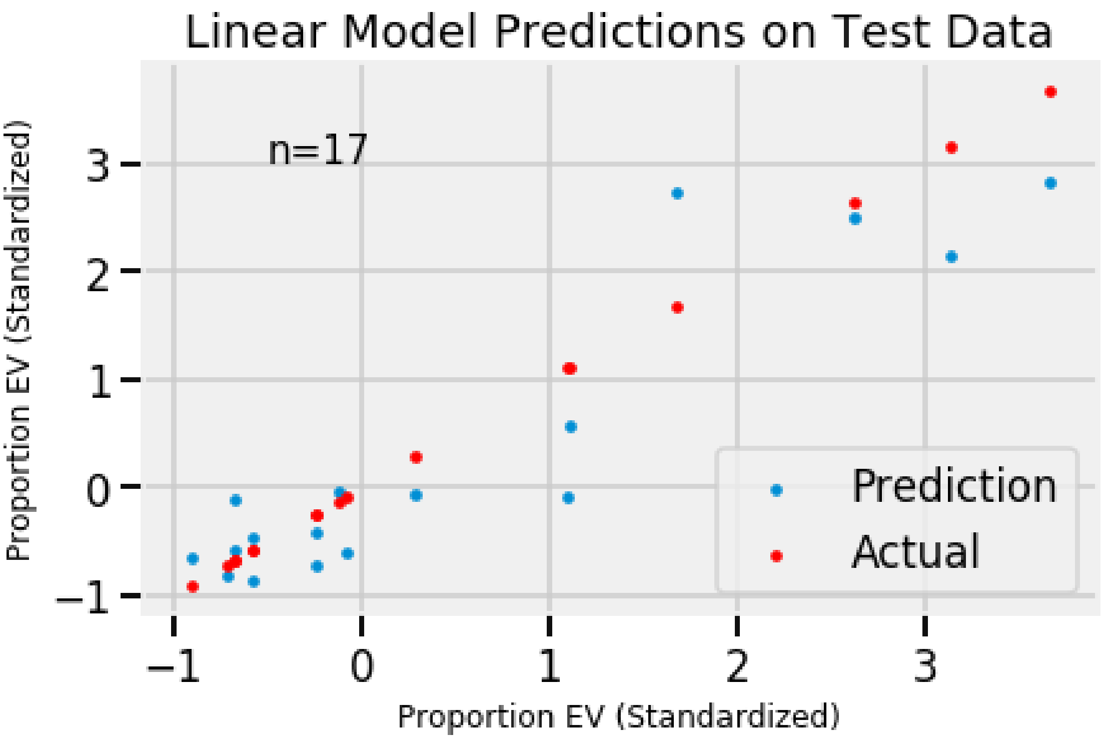
We can see that our model performs fairly well on the training data with a training RMSE of .05 and and a test RMSE of 0.535. The higher test RMSE relative to the training RMSE does imply that we are likley to be seeing some overfit in the data. Another concern that comes up with these models is the fact that we only have 56 observation, since we've aggregated the data at the ccounty level. The lack of data hurts the effectiveness of our model, since it's more likley to be impacted by the randomness in the train-test-split process.

*Let's visualize how far off each prediction is from the actual obervations in our linear model.*

Visualizing the model's performance on the test data

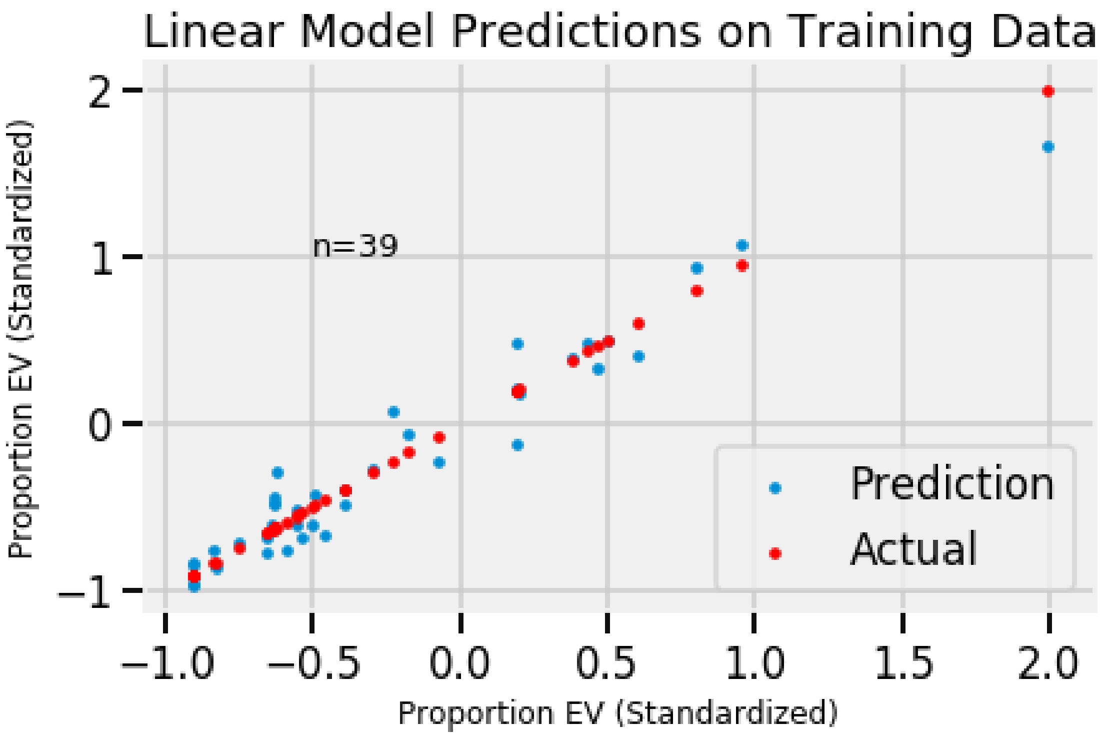
[98]:

```
# Scatter test predictions on test truth
plt.scatter(y_test, y_pred_test, s=10)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test, y_test, s=10, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
#plt.title("Test Proportion EV (Standardized) against itself", size=15)
plt.title("Linear Model Predictions on Test Data")
plt.legend(['Prediction', 'Actual'], loc=4)
plt.text(-.5, 3, "n=" + str(len(y_test)), size=15);
```



[99]:

```
# Scatter train predictions on train truth
plt.scatter(y_train, y_pred_train, s=10)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train, y_train, s=10, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
plt.title("Linear Model Predictions on Training Data")
plt.legend(['Prediction', 'Actual'], loc=4)

#plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(-.5, 1, "n=" + str(len(y_train)), size=13);
```



Unsurprisignly, we can see that our residuals (delta between our predicted and actual values) are far smaller for the training data. The larger residuals that we see in the test data are likely an indicator of overfitting, which is someething that we will tackle in the latter stages of this notebook. Generally, however, we can see that our predictions are fairly close, even on the test data. Another important caveat is the fact that sample sizes in both the test and training data very small, which is likely to further limit the predictive power of these county level models.

Following the linear model and our concerns of potential overfit, we wanted to test out regularized models such as Lasso and Ridge.

### Prediction Model 2.2 - Lasso

```
[100]: kf = KFold(n_splits = 5, shuffle = True, random_state = 1)
grid = np.linspace(0.0001, 100, 10000)
lasso_model = LassoCV(cv = kf, alphas = grid,max_iter=100000)
lasso_fit = lasso_model.fit(X_train, y_train.ravel())

y_pred_lasso_test = lasso_fit.predict(X_test)
y_pred_lasso_train = lasso_fit.predict(X_train)
print("test RMSE:", mean_squared_error(y_test, y_pred_lasso_test, squared =␣
 ↪False))
```

```
print("train RMSE:", mean_squared_error(y_train, y_pred_lasso_train, squared =␣
 ↪False))
print("R2 test:", r2_score(y_test, y_pred_lasso_test))
print("R2 train:", r2_score(y_train, y_pred_lasso_train))
print("alpha:", lasso_fit.alpha_)
lasso_coefs = lasso_fit.coef_
lasso_coefs
```

```
test RMSE: 0.5990736378324224
train RMSE: 0.17832832731892373
R2 test: 0.8208791187890304
R2 train: 0.9190991116358326
alpha: 0.0201019801980198
```

```
[100]: array([-0.        ,  0.        ,  0.        ,  0.20876083, -0.0356515 ,
         0.        , -0.0062517 ,  0.        ,  0.        ,  0.0123173 ,
         0.01757933,  0.        , -0.02055262,  0.        ,  0.06003837,
         0.11803656,  0.        , -0.        , -0.        , -0.02196521,
         0.        ,  0.45263894])
```

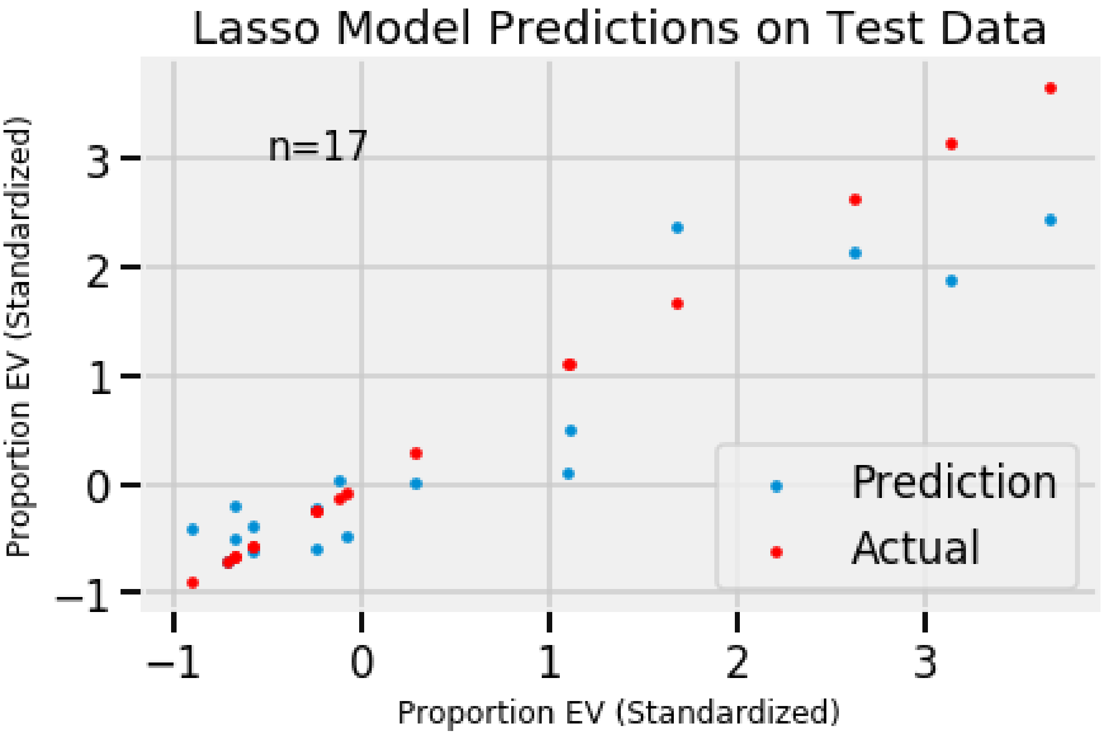
In order to try and reduce the overfit that we saw in the linear model, we chose to use a regularized model such as Lasso. First, we needed to tune the hyperparameter alpha (lambda in class), which determines the extent to which we penalized overfit in our model. When we tuned the model, our alpha parameter came out to be very close to zero. This implies that the lasso model is perfoming in a very similar manner to linear regression. However, looking at the coefficients that the model oupputs, we can still see that a number of the coefficients are still being driven down to zero. Though an Alpha parameter of 0 in a lasso regression would give a result identical to linear regression, our alpha parameter = 0.02 creates a model that performs better than our linear moodel from above. The small Alpha value also suggests that the OLS model is fairly close to fitting the optimum number and combination of features. We chose to use this Alpha parameter because it was the optimal based on 5 fold cross validation.

The test RMSE is now 0.525 vs. 0.578 in the linear model. The test $R^2$ = 0.806 vs. 0.833 in the linear model. The reason for the higher $R^2$ in the linear model is $R^2$ increases as you include more features, so when lasso drivs features down to 0, it makees sense that our $R^2$ decreasees. Overall, based on the RMSE, the Lasso model performs better than the linear model.

*Let's visualize how far off each prediction is from the actual obervations in our lasso model.*

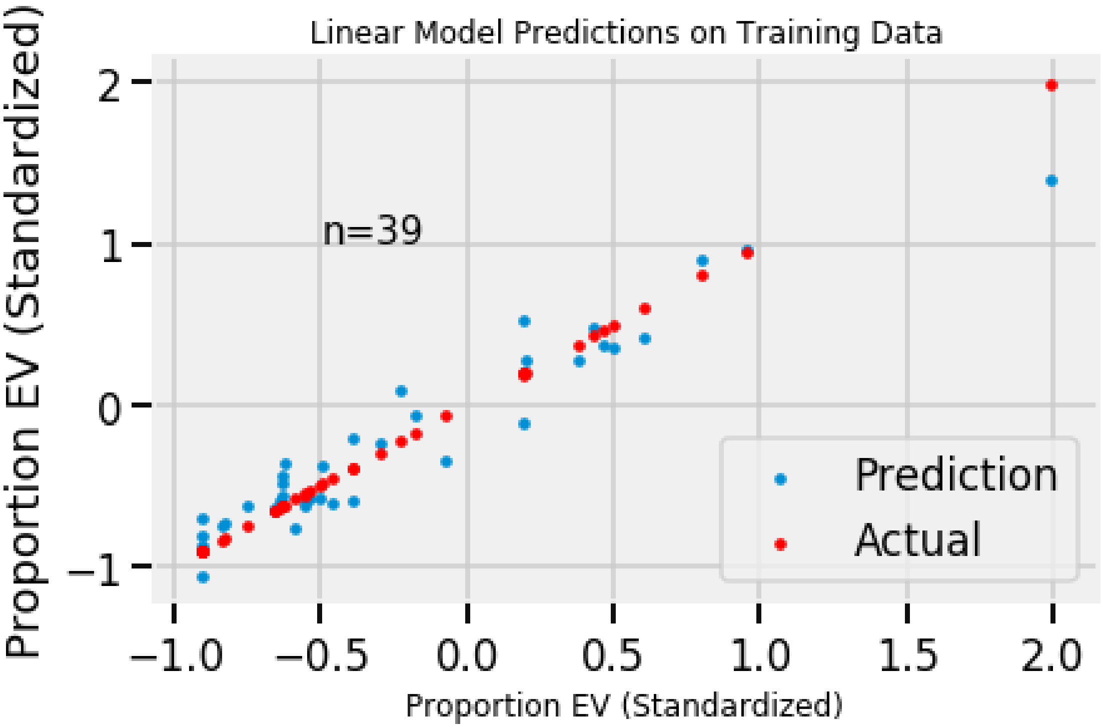
Visualizing the model's performance on the test data

```
[101]: # Scatter test predictions on test truth
plt.scatter(y_test, y_pred_lasso_test, s=10)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test, y_test, s=10, color="r")
plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
#plt.title("Test Proportion EV (Standardized) against itself", size=15)
plt.title("Lasso Model Predictions on Test Data")
plt.legend(['Prediction', 'Actual'], loc=4)
plt.text(-.5, 3, "n=" + str(len(y_test)), size=15);
```



Visualizing the model's performance on the training data

```
[102]: # Scatter train predictions on train truth
plt.scatter(y_train, y_pred_lasso_train, s=10)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train, y_train, s=10, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)")
plt.title("Linear Model Predictions on Training Data", size=12)
plt.legend(['Prediction', 'Actual'], loc=4)

#plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(-.5, 1, "n=" + str(len(y_train)), size=15);
```



Compared to the linear model plots from above, we can see that the residuals in both the test and training data are smaller with the Lasso model. This makes seense, given the lower RMSE that we observe in the Lasso model. The same caveat remains that our small sample size does hurt the performance of our model at the county level (this fact is actually the main reason why we chose to initially make census block level predictions in our first prediction problem. Of the two models that we've tried, Lasso has performed the best so far.

**Prediction Model 2.3 - Ridge** As a model, Lasso is known to perform worse that Ridge in situations where there might be collinear features. Given that many of the features we include in our model, such as education and income tend to be highly correlated, we also ran a Ridge regression in order to see how it performed.

[103]:
```
kf = KFold(n_splits = 5, shuffle = True, random_state = 1)
grid = np.linspace(0.0001, 100, 100)
ridge_model = RidgeCV(cv = kf, alphas = grid)
ridge_fit = ridge_model.fit(X_train, y_train)

y_pred_ridge_test = ridge_fit.predict(X_test)
y_pred_ridge_train = ridge_fit.predict(X_train)
print("test RMSE:", mean_squared_error(y_test, y_pred_ridge_test, squared =␣
 ↪False))
```

```
print("train RMSE:", mean_squared_error(y_train, y_pred_ridge_train, squared =␣
 ↪False))
print("R2 test:", r2_score(y_test, y_pred_ridge_test))
print("R2 train:", r2_score(y_train, y_pred_ridge_train))
print("alpha:", ridge_fit.alpha_)
ridge_coefs = ridge_fit.coef_
ridge_coefs
```

```
test RMSE: 0.7982438027968789
train RMSE: 0.20999203774246122
R2 test: 0.6819781860481351
R2 train: 0.8878192667372232
alpha: 19.192
```

```
[103]: array([[-0.05915915,  0.01398487,  0.00404004,  0.09795364, -0.05628286,
          0.00183899, -0.02971969,  0.00773119, -0.00425365,  0.036076  ,
          0.02895631,  0.01539551, -0.04487915,  0.02792649,  0.09660437,
          0.10772014, -0.00599556, -0.06796747, -0.07154748, -0.06889496,
          0.03864505,  0.10288057]])
```

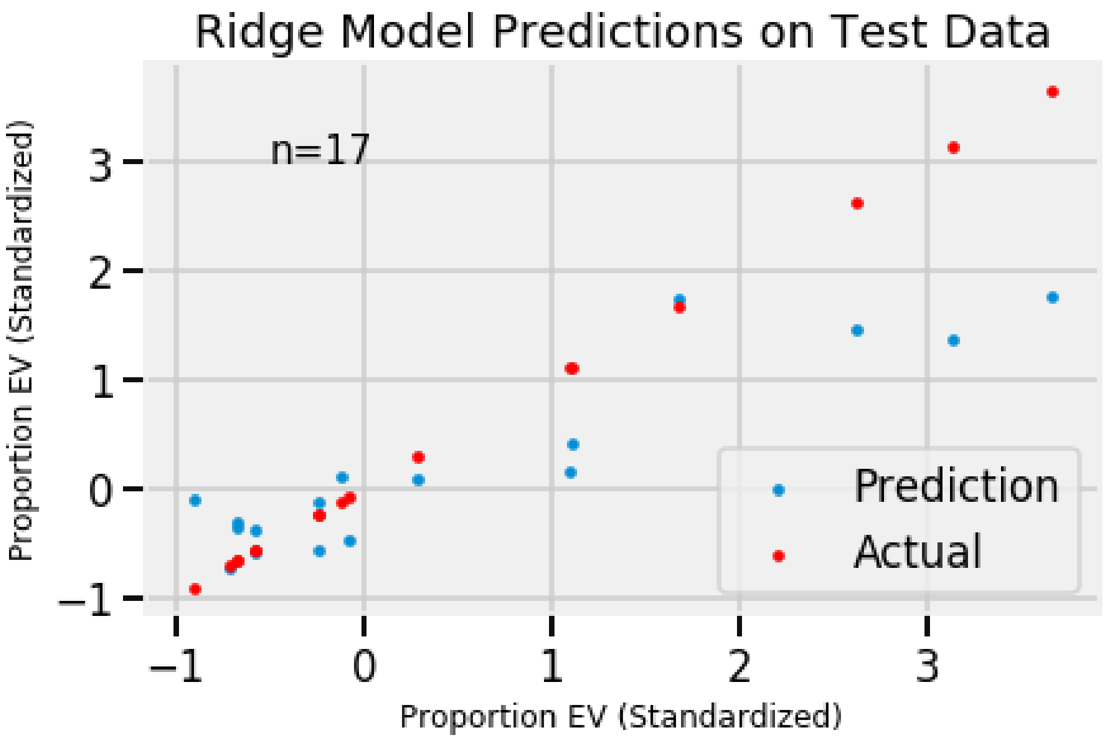
We followed a very similar process to the Lasso regression, starting by tuning the hyperparameter Alpha (lambda in class) and then fitting a model with the tuned hyperparameter. Overall, the Ridge model performs worse than Lasso and OLS on both the training and test data. This may be due to the fact that, unlike Lasso, ridge doesn't drive any parameters down to 0.

The test RMSE with Ridge is now 0.798 vs. 0.525 in the Lasso model. The test $R^2$ = 0.68 vs. 0.806 in the lasso model. Overall, ridge regression performs worse than both our Linear and Lasso models.

*Let's visualize how far off each prediction is from the actual obervations in our Ridge model.*

Visualizing the model's performance on the test data

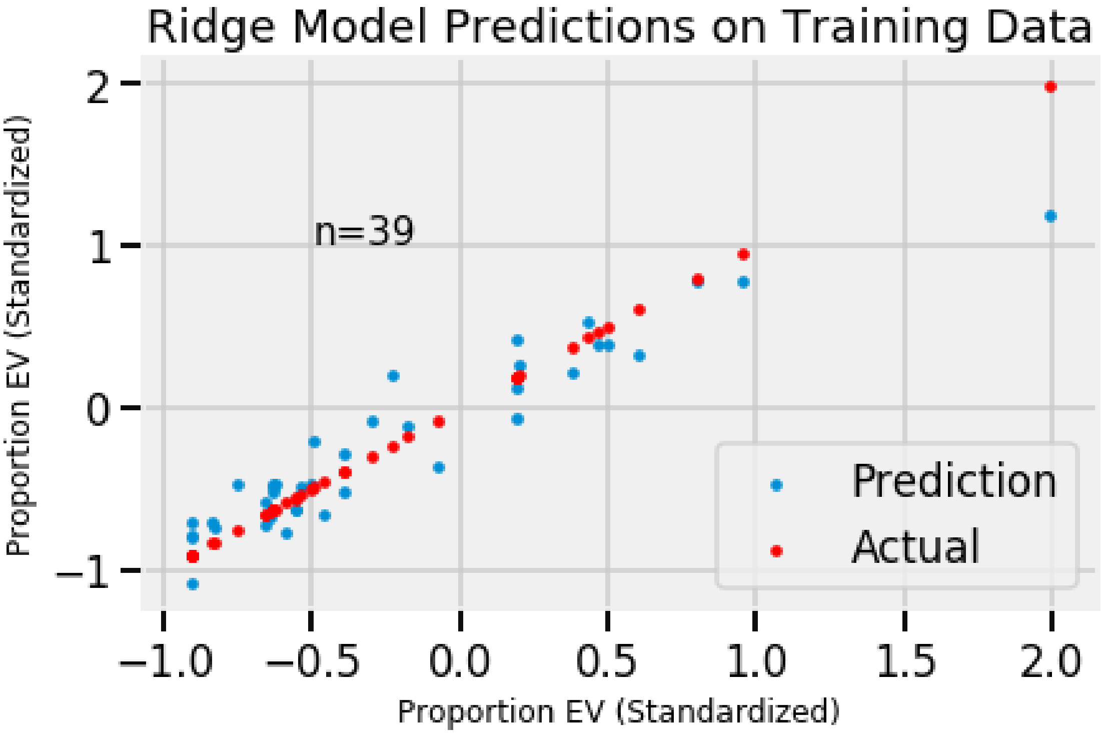
```
[104]: # Scatter test predictions on test truth
plt.scatter(y_test, y_pred_ridge_test, s=10)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test, y_test, s=10, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
#plt.title("Test Proportion EV (Standardized) against itself", size=15)
plt.title("Ridge Model Predictions on Test Data")
plt.legend(['Prediction', 'Actual'], loc=4)
plt.text(-.5, 3, "n=" + str(len(y_test)), size=15);
```



Visualizing the model's performance on the training data

[105]:

```
# Scatter train predictions on train truth
plt.scatter(y_train, y_pred_ridge_train, s=10)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train, y_train, s=10, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
plt.title("Ridge Model Predictions on Training Data")
plt.legend(['Prediction', 'Actual'], loc=4)

#plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(-.5, 1, "n=" + str(len(y_train)), size=15);
```



The plots of residuals confirm the significantly worse performance of the Ridge model relative to both the Lasso and Linear models. This is particularly evident in the plot of the Ridge model's prediction on the test data. The ridge model appears to systematically underpredict for larger values of Y, however, this claim should be caveated by the small sample size of our data.

**Prediction Model 1.4 - Regression Trees (Non-parametric)** Although we largely focussed on parametric models, we wanted to also look at how non-parametric models might be used to address our prediction problem.

[106]:
```
from sklearn.tree import DecisionTreeRegressor

regressor = DecisionTreeRegressor(random_state = 1)
regressor.fit(X_train, y_train)
y_pred_tree_test = regressor.predict(X_test)
y_pred_tree_train = regressor.predict(X_train)

print("test RMSE:", mean_squared_error(y_test, y_pred_tree_test, squared =␣
 ↪False))
print("train RMSE:", mean_squared_error(y_train, y_pred_tree_train, squared =␣
 ↪False))
print("R2 test:", r2_score(y_test, y_pred_tree_test))
print("R2 train:", r2_score(y_train, y_pred_tree_train))
```

```
test RMSE: 0.839597898768418
train RMSE: 0.0
R2 test: 0.6481735507270435
R2 train: 1.0
```

This is a base line model using an untuned tree. We can see that this model performs basically perfectly on the training data (RMSE = 0) due to the fact that we don't place any contstraints on the depth or number of features in the tree. However this model is likely to be overfitting to our test data, hence, the high test RMSE, especially when comparaed to the other prior models we've created. In order to overcome the overfit, we created a tuned version of our decision tree; we start by determinning the ideal hyperparameters.

[107]:

```python
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint

param_dist = {'max_leaf_nodes': randint(3, 100),
              'max_features': randint(2, 22),
              'max_depth': randint(1, 10)}

rnd_search = RandomizedSearchCV(regressor, param_distributions=param_dist,
                                cv=10, n_iter=200, random_state = 2020)
rnd_search.fit(X_train, y_train)

print(rnd_search.best_score_)
print(rnd_search.best_params_)
```

```
0.687428884611719
{'max_depth': 4, 'max_features': 19, 'max_leaf_nodes': 52}
```

We chose to use max features between 2 and 22, since 22 is the total number of features in our training data. We chose to use max leaf nodes beetween 3 and 100 since we felt that this would give us ample coverage and flexibility to fit an effective model; we followed a similar philosophy for the tree dept, especially given that we have fairly small sample sizes in the county data.

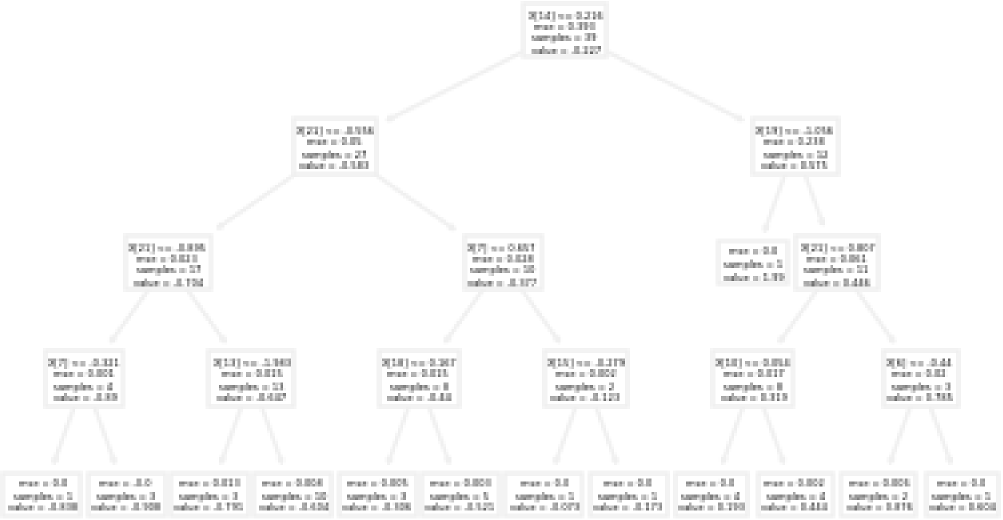
[108]:

```python
regressor = DecisionTreeRegressor(random_state = random, max_depth = 4,␣
 ↪max_features = 19,
                                  max_leaf_nodes = 52)
regressor.fit(X_train, y_train)
y_pred_tree_test = regressor.predict(X_test)
y_pred_tree_train = regressor.predict(X_train)

print("test RMSE:", mean_squared_error(y_test, y_pred_tree_test, squared =␣
 ↪False))
print("train RMSE:", mean_squared_error(y_train, y_pred_tree_train, squared =␣
 ↪False))
print("R2 test:", r2_score(y_test, y_pred_tree_test))
print("R2 train:", r2_score(y_train, y_pred_tree_train))
```

```
test RMSE: 0.6931673228247061
train RMSE: 0.06550953714815651
R2 test: 0.7601929418008542
R2 train: 0.9890825382122047
```

Now that we have tuned the tree, we can see that the test RMSE has decreased significantly relative to the untuned tree. In addition, the test RMSE produced by this tuned regression tree is lower than the ridge model, but still higher than the Lasso and Linear regression.

[109]:
```
from sklearn import tree
tree.plot_tree(regressor);
```



Though this plot isn't the most interpretable, it provides a general sense for the shape of the tree. In order to see a clearer version of the plot, we used http://www.jdolivet.byethost13.com/Logiciels/WebGraphviz/ and inserted the output of the print statement into the site to generate the tree diagram for the tuned tree below.

[110]:
```
import graphviz
print(tree.export_graphviz(regressor, feature_names=X.columns))
```

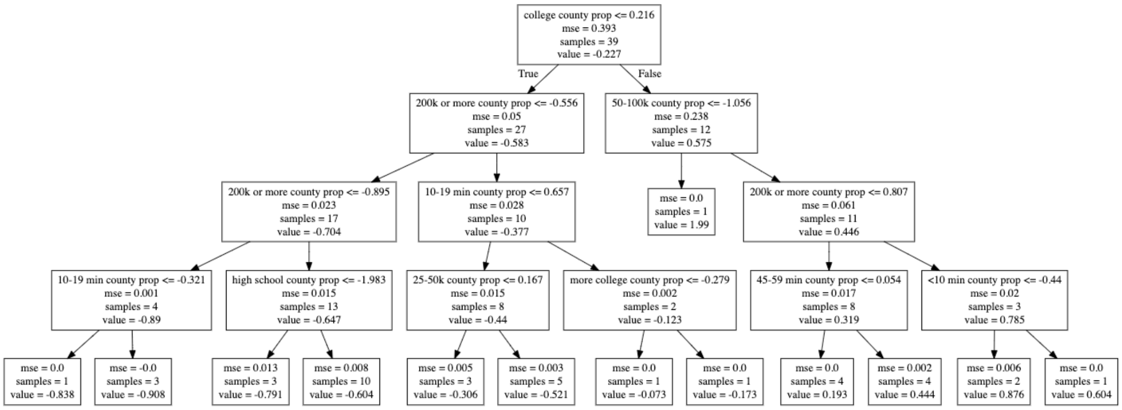
```
digraph Tree {
node [shape=box] ;
0 [label="college county prop <= 0.216\nmse = 0.393\nsamples = 39\nvalue =
-0.227"] ;
```

```
1 [label="200k or more county prop <= -0.556\nmse = 0.05\nsamples = 27\nvalue =
-0.583"] ;
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
5 [label="200k or more county prop <= -0.895\nmse = 0.023\nsamples = 17\nvalue =
-0.704"] ;
1 -> 5 ;
9 [label="10-19 min county prop <= -0.321\nmse = 0.001\nsamples = 4\nvalue =
-0.89"] ;
5 -> 9 ;
23 [label="mse = 0.0\nsamples = 1\nvalue = -0.838"] ;
9 -> 23 ;
24 [label="mse = -0.0\nsamples = 3\nvalue = -0.908"] ;
9 -> 24 ;
10 [label="high school county prop <= -1.983\nmse = 0.015\nsamples = 13\nvalue =
-0.647"] ;
5 -> 10 ;
17 [label="mse = 0.013\nsamples = 3\nvalue = -0.791"] ;
10 -> 17 ;
18 [label="mse = 0.008\nsamples = 10\nvalue = -0.604"] ;
10 -> 18 ;
6 [label="10-19 min county prop <= 0.657\nmse = 0.028\nsamples = 10\nvalue =
-0.377"] ;
1 -> 6 ;
11 [label="25-50k county prop <= 0.167\nmse = 0.015\nsamples = 8\nvalue =
-0.44"] ;
6 -> 11 ;
15 [label="mse = 0.005\nsamples = 3\nvalue = -0.306"] ;
11 -> 15 ;
16 [label="mse = 0.003\nsamples = 5\nvalue = -0.521"] ;
11 -> 16 ;
12 [label="more college county prop <= -0.279\nmse = 0.002\nsamples = 2\nvalue =
-0.123"] ;
6 -> 12 ;
21 [label="mse = 0.0\nsamples = 1\nvalue = -0.073"] ;
12 -> 21 ;
22 [label="mse = 0.0\nsamples = 1\nvalue = -0.173"] ;
12 -> 22 ;
2 [label="50-100k county prop <= -1.056\nmse = 0.238\nsamples = 12\nvalue =
0.575"] ;
0 -> 2 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;
3 [label="mse = 0.0\nsamples = 1\nvalue = 1.99"] ;
2 -> 3 ;
4 [label="200k or more county prop <= 0.807\nmse = 0.061\nsamples = 11\nvalue =
0.446"] ;
2 -> 4 ;
7 [label="45-59 min county prop <= 0.054\nmse = 0.017\nsamples = 8\nvalue =
0.319"] ;
4 -> 7 ;
```

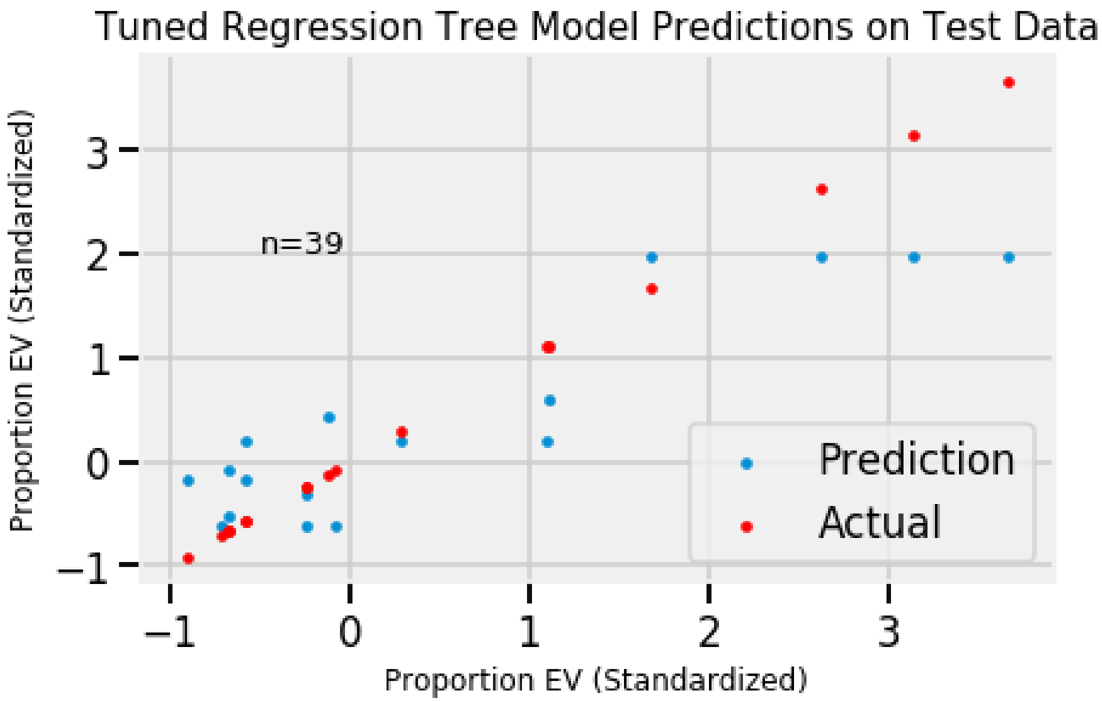
```
13 [label="mse = 0.0\nsamples = 4\nvalue = 0.193"] ;
7 -> 13 ;
14 [label="mse = 0.002\nsamples = 4\nvalue = 0.444"] ;
7 -> 14 ;
8 [label="<10 min county prop <= -0.44\nmse = 0.02\nsamples = 3\nvalue = 0.785"]
;
4 -> 8 ;
19 [label="mse = 0.006\nsamples = 2\nvalue = 0.876"] ;
8 -> 19 ;
20 [label="mse = 0.0\nsamples = 1\nvalue = 0.604"] ;
8 -> 20 ;
}
```

[111]:
```python
from IPython.display import Image
Image(filename='output1.png')
```

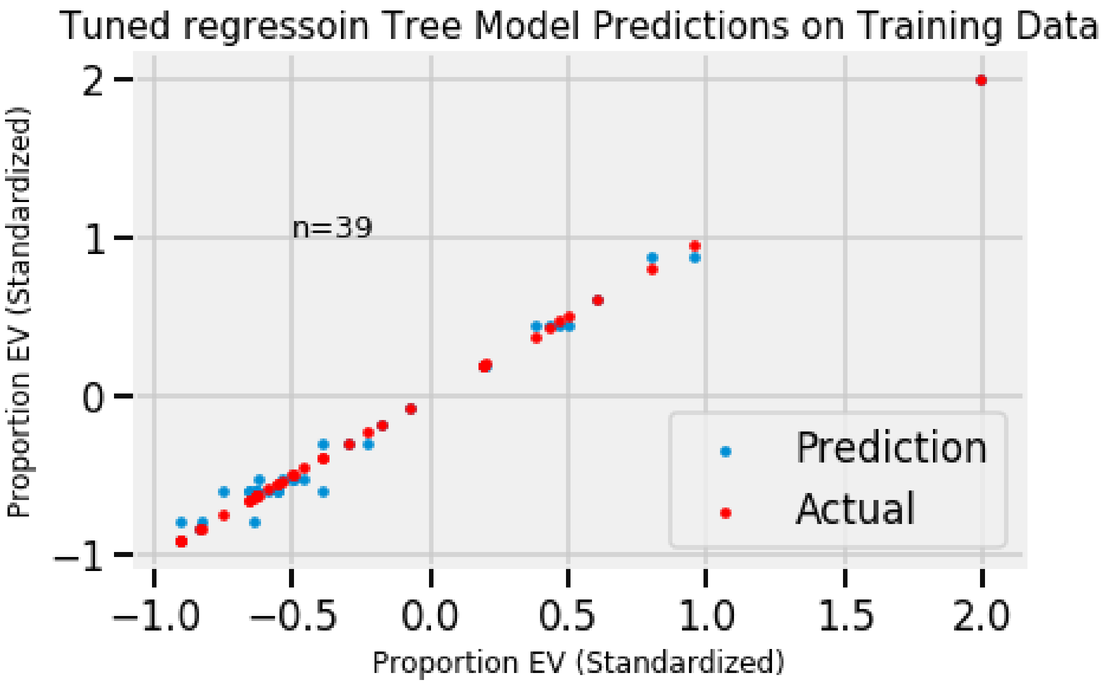
[111]:


college county prop <= 0.216 mse = 0.393 samples = 39 value = -0.227
True
False
200k or more county prop <= -0.556 mse = 0.05 samples = 27 value = -0.583
50-100k county prop <= -1.056 mse = 0.238 samples = 12 value = 0.575
200k or more county prop <= -0.895 mse = 0.023 samples = 17 value = -0.704
10-19 min county prop <= 0.657 mse = 0.028 samples = 10 value = -0.377
mse = 0.0 samples = 1 value = 1.99
200k or more county prop <= 0.807 mse = 0.061 samples = 11 value = 0.446
10-19 min county prop <= -0.321 mse = 0.001 samples = 4 value = -0.89
high school county prop <= -1.983 mse = 0.015 samples = 13 value = -0.647
25-50k county prop <= 0.167 mse = 0.015 samples = 8 value = -0.44
more college county prop <= -0.279 mse = 0.002 samples = 2 value = -0.123
45-59 min county prop <= 0.054 mse = 0.017 samples = 8 value = 0.319
<10 min county prop <= -0.44 mse = 0.02 samples = 3 value = 0.785
mse = 0.0 samples = 1 value = -0.838
mse = -0.0 samples = 3 value = -0.908
mse = 0.013 samples = 3 value = -0.791
mse = 0.008 samples = 10 value = -0.604
mse = 0.005 samples = 3 value = -0.306
mse = 0.003 samples = 5 value = -0.521
mse = 0.0 samples = 1 value = -0.073
mse = 0.0 samples = 1 value = -0.173
mse = 0.0 samples = 4 value = 0.193
mse = 0.002 samples = 4 value = 0.444
mse = 0.006 samples = 2 value = 0.876
mse = 0.0 samples = 1 value = 0.604

[112]:
```python
# Scatter test predictions on test truth
plt.scatter(y_test, y_pred_tree_test, s=10)
# Scatter test truth on test truth to make line y=x
plt.scatter(y_test, y_test, s=10, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
plt.title("Tuned Regression Tree Model Predictions on Test Data", size=15)
plt.legend(['Prediction', 'Actual'], loc=4)

#plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(-.5, 2, "n=" + str(len(y_train)), size=13);
```



[113]:
```
# Scatter train predictions on train truth
plt.scatter(y_train, y_pred_tree_train, s=10)
# Scatter train truth on train truth to make line y=x
plt.scatter(y_train, y_train, s=10, color="r")

plt.xlabel("Proportion EV (Standardized)", size=12)
plt.ylabel("Proportion EV (Standardized)", size=12)
plt.title("Tuned regressoin Tree Model Predictions on Training Data", size=15)
plt.legend(['Prediction', 'Actual'], loc=4)

#plt.title("Training Proportion EV (Standardized) against itself", size=15)
plt.text(-.5, 1, "n=" + str(len(y_train)), size=13);
```



The plots above reaffirm the findings that we saw when computing the RMSE for the tuned regression tree. We can see that similar to our Ridge model, the residuals in the test data are fairly large and seem to be somewhat systematiclaly underpredicting for larger values of Y.

### 1.8.5 3. Predicting Short vs Long Commute Time at the Census Block Level

To follow along the same thread of our main prediction questions (above), we wanted to do something else related to transportation: commute time. Because of the way our data was collected, though, it was a bit difficult to perform normal categorical analysis. Thus, we decided to predict whether or not an observation's commute time would be short or long, which we classified as either >30 minutes or <= 30 minutes, respectively.

In the following cell, we compute the proportion of short commute times versus long commutes times. We compare these proportions and create an indicator/response variable based on which proportion is larger. The variable will be 1 if the majority average commute time is short, and 0 if the majority average commute time is long.

```
[114]: df = df.reset_index()
       df.head()
```

```
[114]:    level_0  index Census Block Group Code  County                       id  \
       0        0      0             60570001024  NEVADA  1500000US060570001024
       1        1      5             60570006002  NEVADA  1500000US060570006002
```

```
2        2      6             60570005022  NEVADA  1500000US060570005022
3        3      7             60570004022  NEVADA  1500000US060570004022
4        4      8             60570001023  NEVADA  1500000US060570001023

                                Geographic Area Name  Total Education  \
0  Block Group 4, Census Tract 1.02, Nevada Count…           1032.0
1  Block Group 2, Census Tract 6, Nevada County, …            765.0
2  Block Group 2, Census Tract 5.02, Nevada Count…            982.0
3  Block Group 2, Census Tract 4.02, Nevada Count…           1476.0
4  Block Group 3, Census Tract 1.02, Nevada Count…           1887.0

   Total Income  Total Race  White alone  Black or African American alone  \
0         578.0      1476.0       1152.0                              0.0
1         475.0      1104.0        947.0                             24.0
2         582.0      1329.0       1155.0                              0.0
3         811.0      1844.0       1612.0                            210.0
4         979.0      2373.0       2289.0                              0.0

   American Indian and Alaska Native alone  Asian alone  \
0                                      0.0         36.0
1                                     13.0         10.0
2                                      0.0         31.0
3                                      0.0          2.0
4                                      0.0          0.0

   Native Hawaiian and Other Pacific Islander alone  Some other race alone  \
0                                               0.0                    0.0
1                                               0.0                   13.0
2                                               0.0                    0.0
3                                               0.0                    0.0
4                                               0.0                   25.0

   Total Commute  EV Population  Vehicle Population  Proportion EV  \
0          513.0            1.0               526.0       0.001901
1          543.0            1.0               258.0       0.003876
2          344.0            5.0               231.0       0.021645
3          827.0            1.0               243.0       0.004115
4          942.0            3.0               724.0       0.004144

   More College Completed  College Completed  High School Completed  \
0                    77.0              374.0                 1032.0
1                    43.0              209.0                  717.0
2                    70.0              266.0                  857.0
3                   125.0              650.0                 1458.0
4                   201.0              956.0                 1697.0

   Less than $10k  \$10,000 to $24,999  \$25,000 to $49,999  \
0            37.0                 31.0                 76.0
1           101.0                143.0                 81.0
2            33.0                171.0                221.0
3            62.0                 63.0                188.0
4            36.0                 65.0                249.0

   \$50,000 to $99,999  \$100,000 to $199,999  $200,000 or more  \
0                213.0                  187.0              34.0
1                125.0                   25.0               0.0
2                125.0                   15.0              17.0
3                322.0                  114.0              62.0
4                270.0                  268.0              91.0

   Less than 10 Minutes  10 to 19 Minutes  20 to 29 Minutes  30 to 44 Minutes  \
0                  24.0             188.0             163.0              75.0
1                 303.0             184.0               0.0              12.0
2                 152.0             109.0              32.0              51.0
3                 170.0             160.0             164.0             145.0
4                  47.0             445.0             153.0             130.0

   45 to 59 Minutes  60 to 89 Minutes  90 or more Minutes  \
0               0.0              28.0                35.0
1               0.0              44.0                 0.0
2               0.0               0.0                 0.0
3             128.0               0.0                60.0
4              39.0              44.0                84.0

   More College Completed Proportion  College Completed Proportion  \
0                           0.074612                      0.362403
1                           0.056209                      0.273203
2                           0.071283                      0.270876
3                           0.084688                      0.440379
4                           0.106518                      0.506624

   High School Completed Proportion  Less than $10k Proportion  \
0                          1.000000                   0.064014
1                          0.937255                   0.212632
2                          0.872709                   0.056701
3                          0.987805                   0.076449
4                          0.899311                   0.036772

   \$10,000 to $24,999 Proportion  \$25,000 to $49,999 Proportion  \
0                        0.053633                        0.131488
1                        0.301053                        0.170526
2                        0.293814                        0.379725
3                        0.077682                        0.231813
4                        0.066394                        0.254341
```

```
   \$50,000 to $99,999 Proportion  \$100,000 to $199,999 Proportion  \
0                    0.368512                          0.323529
1                    0.263158                          0.052632
2                    0.214777                          0.025773
3                    0.397041                          0.140567
4                    0.275792                          0.273749

   $200,000 or more Proportion  White alone Proportion  \
0                     0.058824                0.780488
1                     0.000000                0.857790
2                     0.029210                0.869074
3                     0.076449                0.874187
4                     0.092952                0.964602

   Black or African American alone Proportion  \
0                                    0.000000
1                                    0.021739
2                                    0.000000
3                                    0.113883
4                                    0.000000

   American Indian and Alaska Native alone Proportion  Asian alone Proportion  \
0                                           0.000000                0.024390
1                                           0.011775                0.009058
2                                           0.000000                0.023326
3                                           0.000000                0.001085
4                                           0.000000                0.000000

   Native Hawaiian and Other Pacific Islander alone Proportion  \
0                                                0.0
1                                                0.0
2                                                0.0
3                                                0.0
4                                                0.0

   Some other race alone Proportion  Less than 10 Minutes Proportion  \
0                          0.000000                         0.046784
1                          0.011775                         0.558011
2                          0.000000                         0.441860
3                          0.000000                         0.205562
4                          0.010535                         0.049894

   10 to 19 Minutes Proportion  20 to 29 Minutes Proportion  \
0                     0.366472                     0.317739
1                     0.338858                     0.000000
2                     0.316860                     0.093023
```

```
3                     0.193470                     0.198307
4                     0.472399                     0.162420

   30 to 44 Minutes Proportion  45 to 59 Minutes Proportion  \
0                     0.146199                     0.000000
1                     0.022099                     0.000000
2                     0.148256                     0.000000
3                     0.175333                     0.154776
4                     0.138004                     0.041401

   60 to 89 Minutes Proportion  90 or more Minutes Proportion
0                     0.054581                       0.068226
1                     0.081031                       0.000000
2                     0.000000                       0.000000
3                     0.000000                       0.072551
4                     0.046709                       0.089172
```

[115]:

```
short = ['Less than 10 Minutes', '10 to 19 Minutes', '20 to 29 Minutes']
long = ['45 to 59 Minutes', '60 to 89 Minutes', '90 or more Minutes', '30 to 44␣
 ↪Minutes']

prop_short = (df['Less than 10 Minutes'] + df['10 to 19 Minutes'] + df['20 to␣
 ↪29 Minutes'])/df['Total Commute']
prop_long = (df['45 to 59 Minutes'] + df['60 to 89 Minutes'] + df['90 or more␣
 ↪Minutes'] + df['30 to 44 Minutes'])/df['Total Commute']
```

[116]:

```
#make an indicator variable that outputs 1 if the commute time is short and 0␣
 ↪if long
short = []
for x in np.arange(len(prop_short)):
    if prop_short[x] > prop_long[x]:
        short.append(1)
    else:
        short.append(0)

#adding the indicator variable to our df
df['Short Commute Length Indicator'] = short
```

[117]:

```
df[['Less than 10 Minutes Proportion', '10 to 19 Minutes Proportion', '20 to 29␣
 ↪Minutes Proportion', '30 to 44 Minutes Proportion',
   '45 to 59 Minutes Proportion', '60 to 89 Minutes Proportion', '90 or more␣
 ↪Minutes Proportion', 'Short Commute Length Indicator']].head()
```

[117]:

```
   Less than 10 Minutes Proportion  10 to 19 Minutes Proportion  \
0                         0.046784                     0.366472
1                         0.558011                     0.338858
2                         0.441860                     0.316860
```

```
3                         0.205562                     0.193470
4                         0.049894                     0.472399

   20 to 29 Minutes Proportion  30 to 44 Minutes Proportion  \
0                     0.317739                     0.146199
1                     0.000000                     0.022099
2                     0.093023                     0.148256
3                     0.198307                     0.175333
4                     0.162420                     0.138004

   45 to 59 Minutes Proportion  60 to 89 Minutes Proportion  \
0                     0.000000                     0.054581
1                     0.000000                     0.081031
2                     0.000000                     0.000000
3                     0.154776                     0.000000
4                     0.041401                     0.046709

   90 or more Minutes Proportion  Short Commute Length Indicator
0                       0.068226                               1
1                       0.000000                               1
2                       0.000000                               1
3                       0.072551                               1
4                       0.089172                               1
```

For our features, we make sure to only use the relevant proportions such as our education, income, and race data. Proportions are used for the same reason stated earlier.

```
[118]: X = df[['More College Completed Proportion', 'College Completed Proportion',␣
        ↪'High School Completed Proportion',
               'Less than $10k Proportion', '\$10,000 to $24,999 Proportion',␣
        ↪'\$25,000 to $49,999 Proportion',
               '\$50,000 to $99,999 Proportion', '\$100,000 to $199,999 Proportion',␣
        ↪'$200,000 or more Proportion',
               'White alone Proportion', 'Black or African American alone Proportion',␣
        ↪'American Indian and Alaska Native alone Proportion',
               'Asian alone Proportion', 'Native Hawaiian and Other Pacific Islander␣
        ↪alone Proportion', 'Some other race alone Proportion',
               'Proportion EV']]
       y = df['Short Commute Length Indicator']
```

```
[119]: X_train, X_test, y_train, y_test = train_test_split(X, y)
```

**Model 3.1: Logistic Regression** Since our response variable takes on the value of 0 or 1, we figured that a Logsitic Model should be our first choice to cater towards this classification problem.

[120]:

```python
from sklearn.linear_model import LogisticRegression

clf = LogisticRegression().fit(X_train, y_train)
y_pred = clf.predict(X_test)
clf.score(X_test, y_test)
```

[120]:

```
0.6634119880661785
```

Also, since we're dealing with a classification problem, we'd want to evaluate further than the mean accuracy on the test data (as noted above). Thus, we want to look at the confusion matrix.

[121]:

```python
from sklearn.metrics import confusion_matrix

y_vals = y_test.values
confusion_matrix(y_vals, y_pred)
```

[121]:

```
array([[ 226, 1045],
       [ 196, 2220]])
```

As you can see from the confusion matrix, we have very high rates of false positives and false negatives when using logistic regression, and our accuracy on our test set was very low (~67%).

### Model 3.2: Decision Tree

[122]:

```python
tree = DecisionTreeClassifier()
tree.fit(X_train, y_train)

print("Number of features: {}".format(tree.tree_.n_features))
print("Number of nodes (leaves): {}".format(tree.tree_.node_count),"\n")

train_score = tree.score(X_train, y_train)
test_score = tree.score(X_test, y_test)

print('Train Score: ', train_score)
print('Test Score: ', test_score)
```

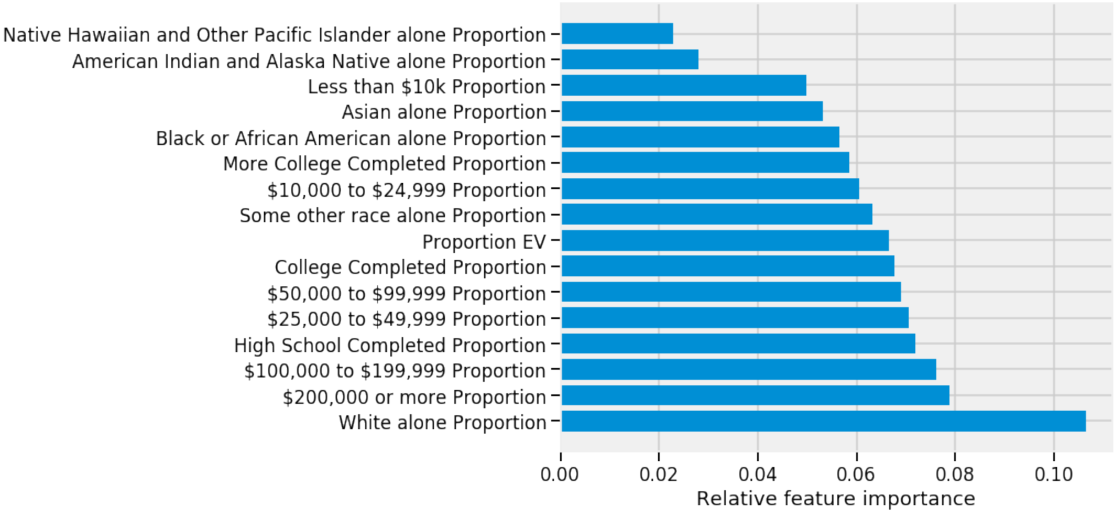
```
Number of features: 16
Number of nodes (leaves): 3939

Train Score:  1.0
Test Score:  0.6159479251423922
```

[123]:

```python
def importance_plot(tree):
    # YOUR CODE HERE
    feature_importance = tree.feature_importances_ # get the importance of each␣
 ↪feature
```

```
    #calculate the relative feature importance and save to a dataframe with two␣
↪columns:
    # One holding the names of the features, and one holding the associated␣
↪relative importance
    # of each feature.
    relative_importance = feature_importance/max(feature_importance) * 100
    feat_df = pd.DataFrame({'Feature': X.columns, 'Importance': tree.
↪feature_importances_})

    # Sort feat_df in order of importance
    feat_df = feat_df.sort_values(by='Importance', ascending=False)

    plt.figure(figsize=(8, 7.5))
    plt.barh(width=feat_df['Importance'], y=feat_df['Feature'])
    plt.xlabel('Relative feature importance');
```

[124]:
```
importance_plot(tree)
```



Hmm… doesn't look like there's not that many features that stand out in terms of predicting short vs long commute times. Let's see if we have a better prediction if we were to not use features with <5% importance

[125]:
```
new_X = df[['College Completed Proportion',
       'High School Completed Proportion',
       '\$10,000 to $24,999 Proportion', '\$25,000 to $49,999 Proportion',
       '\$50,000 to $99,999 Proportion', '\$100,000 to $199,999 Proportion',
       '$200,000 or more Proportion', 'White alone Proportion',
       'Black or African American alone Proportion',
       'Asian alone Proportion',
       'Some other race alone Proportion', 'Proportion EV']]
y = df['Short Commute Length Indicator']

new_X_train, new_X_test, y_train, y_test = train_test_split(new_X, y)

new_tree = DecisionTreeClassifier()
new_tree.fit(new_X_train, y_train)

print("Number of features: {}".format(new_tree.tree_.n_features))
print("Number of nodes (leaves): {}".format(new_tree.tree_.node_count),"\n")

new_train_score = new_tree.score(new_X_train, y_train)
new_test_score = new_tree.score(new_X_test, y_test)

print('Train Score: ', new_train_score)
print('Test Score: ', new_test_score)
```

```
Number of features: 12
Number of nodes (leaves): 4143

Train Score:  1.0
Test Score:  0.6145918090588555
```

It doesn't seem to be like there's that big of a difference. This is most likely due to how we formatted the prediction itself. We'll go more in depth on this in the last section, but first let's try one more model to see if we'll get a better prediction with random forests.

### Model 3.3: Random Forest

[126]:
```
random_forest_model = ensemble.RandomForestClassifier(n_estimators = 20,␣
 ↪random_state = 42)
random_forest_model.fit(X_train, y_train)
rf_train_accuracy = random_forest_model.score(X_train, y_train)
print(f'training accuracy = {rf_train_accuracy}')

rf_test_accuracy = random_forest_model.score(X_test, y_test)
print(f'test accuracy = {rf_test_accuracy}')

cv_rf = np.mean(cross_val_score(random_forest_model, X_train, y_train, cv = 5))
print(f'cross validation = {cv_rf}')

cnf_matrix = confusion_matrix(y_test, random_forest_model.predict(X_test))
tn = cnf_matrix[0,0]
fn = cnf_matrix[1,0]
tp = cnf_matrix[1,1]
fp = cnf_matrix[0,1]
```

```
precision = tp / (tp+fp)
recall = tp / (tp+fn)

print(f'precision = {precision:.4f}')
print(f'recall = {recall:.4f}')
cnf_matrix
```

```
training accuracy = 0.9854430379746836
test accuracy = 0.6099810143748304
cross validation = 0.6144665461121157
precision = 0.6614
recall = 0.8447
```

```
[126]: array([[ 182, 1058],
               [ 380, 2067]])
```

As we can see from all 3 models, despite the extensive efforts made, there's not really a difference in test accuracy (~60%). Although an interesting feature to predict, it's a bit difficult to make accurate predictions by grouping census block groups in short vs long commute times. The data we have and how we chose to predict commute time isn't truly representative of that census block group's actual commute times. Prehaps the model could be bettter if we were to find data collected by household so we could use one-hot encoding.

### 1.9 Interpretation and Conclusions (20 points)

The resource allocation question we set out to answer was to determine which areas are in the greatest need for economic incentives to adopt electric vehicles. We decided that the proportion of electric vehicles in an area would be a good proxy for determining the degree to which an area is in need of economic incentives. We built several different models with the goal of effectively predicting the proportion of electric vehicles in a geographic region (our main model restricted the size of this region to the census block group level). We determined that among our parametric models, Lasso Regression performs the best in terms of minimizing test RMSE; however, among all the models we built and tested, Regression Trees returned the lowest test RMSE. This finding suggests that there may be some underlying non-linear data-generating process at play.

Our model could be used by policy makers in other states striving to determine how best to allocate economic incentives for EV adoption, assuming they have census block group level data on file for our predictors: education, income, race, and commute time. We would advise that if our model returns a proportion of electric vehicles in a given region that is lower than what that the level of adoption that region is striving for, then greater economic incentives should be implemented in those regions. Economic incentives may not be the only factor that leads to low levels of EV adoption, policy makers may also want to consider factors like the number of EV charging stations or EV showrooms in locations where there are low levels of adoption. Ultimately, our model will help local leaders understand not only how best to distribute funds to potential economic incentives, but also best to allocate their time in order to understand the drivers behind the lack of adoption in each specific census block.

Throughout the course of this notebook, we have noted several caveats that may influence the performance of our model. One such caveat is the fact that we chose to group the data into certain categorical variables, such as the proportion of people who complete high school, which includes both those who graduate from high school and those who complete the GED. In the case of California, we felt like grouping these variables together is unlikely to drastically skew the data, however, there may be other states where such decisions may materially affect the model. In a similar vein, our model only included race for people who reported to be only 1 race, thus, we don't account for mixed race people. This is a simplifying assumption that we made, however, it could impact our model if we use the same groupings in areas with large populations of mixed race people.

During our census block level predictions, we found that in areas with higher proportions of electric vehicles, our model would systematically underestimate the proportion of electric vehicles. This may be driven by the fact that the lion share of observations in our data tend to have really low proportions of electric vehicles, thus, our model is biased towards predicting lower proportions of electric vehicles. This is something that is likely to be less of an issue in other states, since we think that California is one of the states that has the highest levels of variance when it comes to EV adoption, since there are pockets of California, such as Menlo Park and Palo Alto that are flooded with Teslas, while many rural areas have next to no electric vehicle adoption. We would argue that other states are less likely to see the same scale of variance as is present in California.

Given some of these caveats and many of the potential underlying state-level factors that influence EV adoption (e.g. political views, EV charger availability, tax incentives), which may not be captured in our model, we wanted to test the external validity of our methods. We simulated this by dividing California into Northern California and Southern California, since we felt that there was significant enough variation between these two regions that could serve as a proxy for simulating inter-state variation. We built a model that was trained on only Northern California data, which we then tested on Southern California data and found that the Lasso test RMSE to be quite similar (0.529 - Norcal / Socal vs. 0.597 - random split) to our model when we tested it using a random train-test-split. This demonstrates that our model is flexible enough to potentially deal with underlying state-level variation and the similar RMSE is more a function of the model itself than any state-specific characteristics. While our model did hold up to our preliminary external validity testing, there are certain states where our model may be unable to capture significant underlying differences between states. For example, if we were to apply our model to a state like Hawaii, it may be necessary to include certain state-level control variables.

Ultimately, it is always important to exercise caution when applying a model to a different state with potentially different underlying characteristics, however, this model can serve as a strong first order approximation of the proportion of EVs that one would find in a given census block.