# Project Hypothesis

Purchasing a used vehicle can be a daunting prospect to any potential buyer. Beyond having preferences for a particular make, model, and car color, a buyer must assess and weigh other features when researching purchase options. These additional features are often traits related specifically to a vehicle (*e.g.*, vehicles’ year, mileage, etc.), but are there outside factors – such as the geographic market in which the car is sold, or perhaps local economic conditions related to wealth and income – that may help drive a vehicle’s listing price?

A 2013 analysis of vehicles listed on CarGurus.com found that “less-expensive markets tend to be the most heavily populated urban areas, where there are more car dealerships and thus more competition.”[[1]](#footnote-0)

The aim of our project is to blend used vehicle listings data with geographic and income data to identify variables that can help predict a used vehicle’s listing price. We also set out to determine whether a significant difference can be found between vehicle listings in metropolitan (*e.g.,* heavily populated urban areas) and non-metropolitan localities.

# DATA SOURCES

## Used Car Listings – MarketCheck.com:

Table Name: “listings”

MarketCheck.com provides online aggregation services for select business sectors, including the automotive industry. As part of their data services, MarketCheck provides access to inventories of new, used and certified car listings gathered from over 50,000 vehicle dealerships. Inventory data is exposed through a fee-based API client.

Our team selected MarketCheck as our primary vehicle data source as it provides comprehensive used car listing data through an accessible and easy to use API platform. Because MarketCheck sources its data directly from dealership websites through a proprietary “polite” web crawler that respects website instructions for bots,[[2]](#footnote-1) it eliminates the need for us to consider our own web-scraping approach.

We had initially set out to query MarketCheck’s Inventory Search API. However, MarketCheck limits data returns to 50 rows per API call and even though a pagination feature is provided, MarketCheck further limits queries by restricting these calls to approximately 1,000 pages. As a result, we were limited to approximately 50,000 data returns per API query of a given set of parameters. This severely restricted our ability to obtain a single vehicle make’s listings (e.g., Honda by itself contains over 400,000 listings), much less MarketCheck’s full listings dataset.

We consulted with MarketCheck’s API team and they offered to provide us a .csv flat file of listing information. We narrowed the scope of our project dataset to the top 15 brands of used car sales, further limiting our request to model years 2000 – 2018. The flat file provided to us includes over 5.2 million rows of data and is over 38 GB in size.

## Bureau of Labor Statistics – Metropolitan and Non-Metropolitan Statistical Areas

Table Name: “locality”

We seek to answer whether a vehicle’s listing price is influenced by the fact that it’s in a metropolitan or non-metropolitan (or urban vs. non-urban) locality. In order to identify whether a given zip code or county is to be considered ‘metropolitan’ or not, we make use of the Bureau of Labor Statistics’ Occupational Employment Statistics (OES) survey[[3]](#footnote-2). The OES program produces employment and wage estimates annually for every metropolitan (MSA) and non-metropolitan statistical area (nMSA). As part of their dataset, BLS identifies every county in the United States and crosswalks it to an MSA or nMSA designation.

## Cities Extended

Table Name: “cities\_extended”

MarketCheck provides geographic details with each used vehicle listing but this is limited to the Vehicle City, Vehicle State, Dealer City, Dealer State, Latitude, and Longitude. Of particular note is that a vehicle listing’s corresponding county is missing. In order to note whether a listing is ‘metropolitan’ or ‘non-metropolitan’, we need to be able to identify a corresponding city and zip code’s county. For this exercise, we make use of an open-source dataset[[4]](#footnote-3) that will allow us to crosswalk zip codes to county designations.

## IRS Statistics of Income (SOI) Individual Income Tax Statistics

Table Name: “soi\_data”

In addition to answering whether vehicle listings in metropolitan areas are higher than those in non-metropolitan areas, we want to assess whether a particular locality’s wealth (measured by income) also affects used vehicle markets. To try and answer this question, we retrieve Internal Revenue Services (IRS) data related to individual income taxes.[[5]](#footnote-4) The information in this dataset is broken out at the zip code level, but we will make use of our “cities\_extended” dataset to find its corresponding county and the “locality” dataset to find its corresponding statistical area.

## Bureau of Economic Analysis (BEA) – Regions

Table Name: “bea\_regions”

We also want to determine whether noticeable differences in prices arise depending upon the region in which a particular vehicle is being sold. We make use of the Bureau of Economic Analysis’ (BEA) regional designations to identify which region a vehicle listing falls under.

Although our economic analysis is limited to IRS statistics of income data, our decision to use BEA regional designations (as opposed to Census regions, for example) is because a future iteration of this project could utilize BEA’s measures of personal income, rather than Census’ median income of families, for further analysis. While utilizing Census’ family-based median income measures could risk limiting a buyer group to only individuals, use of BEA personal income measures would broaden this buyer pool to both individuals and institutions. BEA data helps identify a locality’s economic output through summary measures such as the gross domestic product (GDP).

# DATA STORAGE

A PostgreSQL[[6]](#footnote-5) database has been set up on a DigitalOcean cloud server[[7]](#footnote-6) running Ubuntu[[8]](#footnote-7).

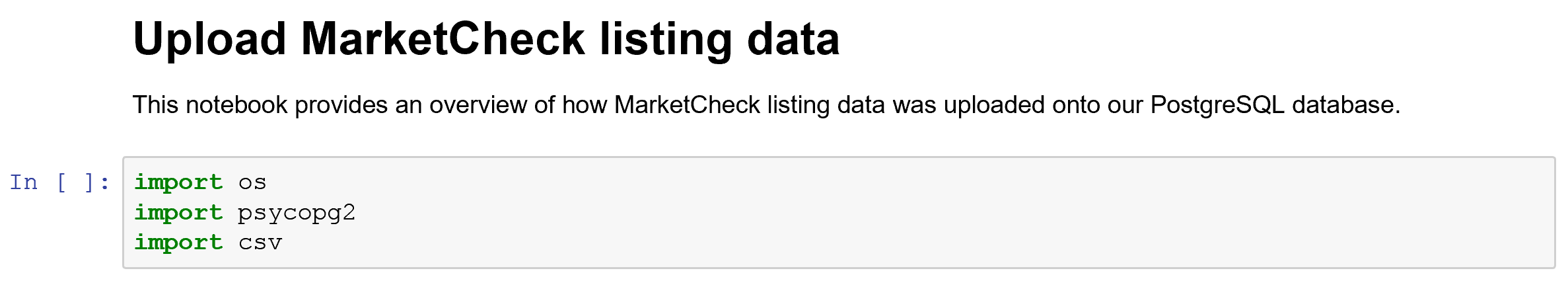
Our database currently consists of five (6) distinct tables from which we retrieve data.

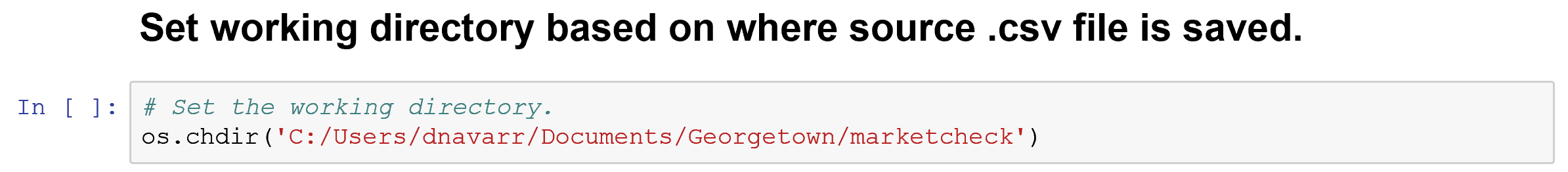
|  |  |
| --- | --- |
| **TABLE NAME** | **DESCRIPTION** |
| listings | MarketCheck Listings |
| locality | Bureau of Labor Statistics dataset containing a FIPS code to MSA crosswalk |
| cities\_extended | Fair use dataset containing city to county crosswalk |
| bea\_regions | Bureau of Economic Analysis regions |
| soi\_data | IRS Statesments of Income datasets |
| census\_fips | Census dataset containing county to FIPS code crosswalk |

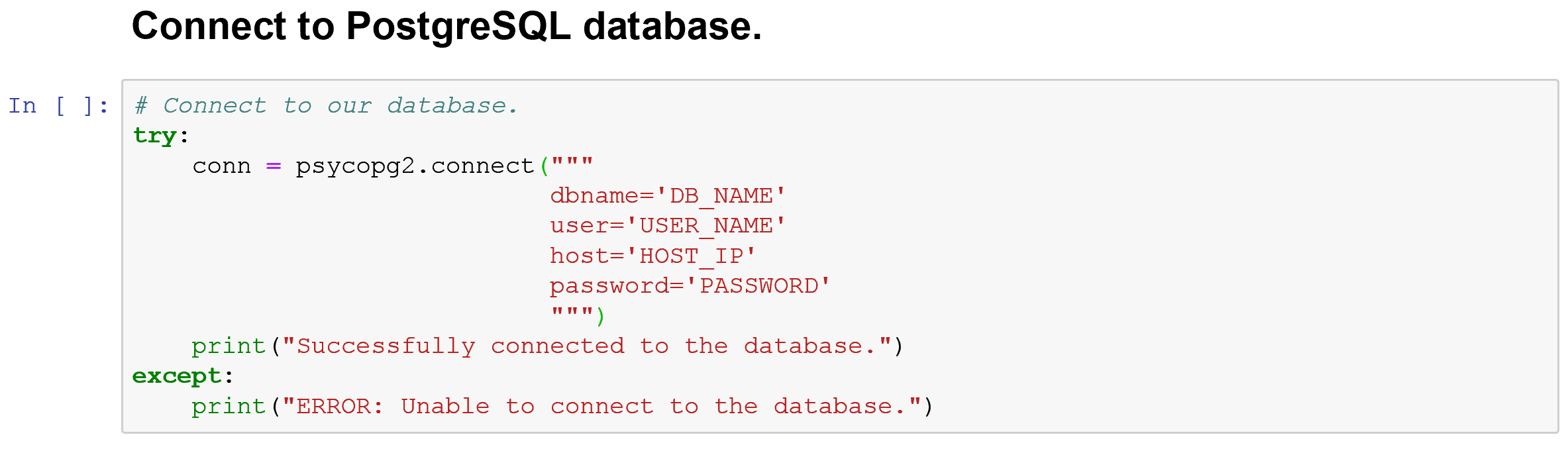
# DATA INGESTION

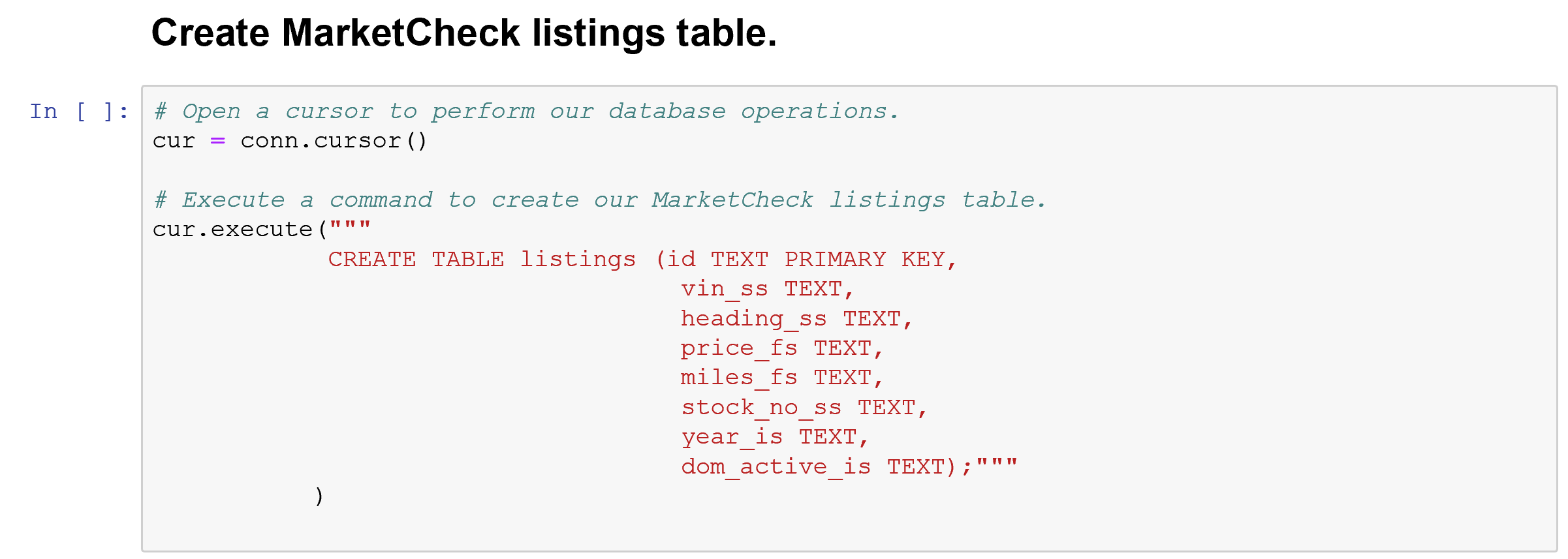
Each distinct dataset utilizes a .csv source file which was used to populate the respective tables in our Postgres database. We made use of Python’s csv package to open and read the source .csv files and we used the psycopg2 package to connect to our database and make changes.

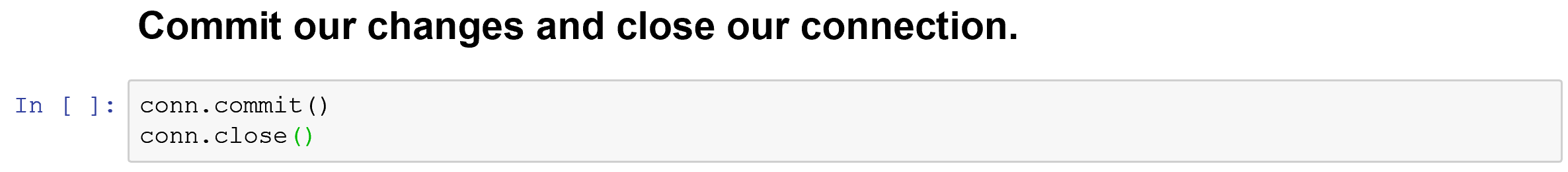
Code used for each of the data uploads is found on our Github repository.[[9]](#footnote-8) For brevity, the following sample code is used to illustrate the general template that each data upload followed:









# DATA WRANGLING

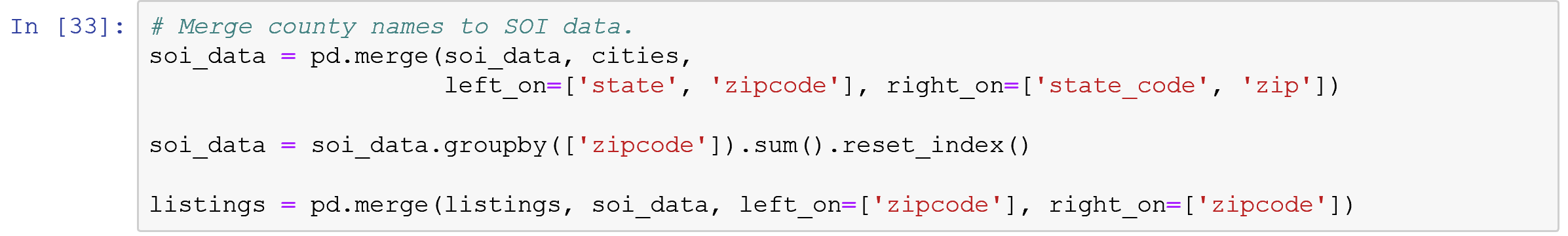
We have uploaded a comprehensive notebook onto our Github repository that shows our data wrangling. Here, we will describe major components of our wrangling process.

## Data Merging

After importing our data, we set out to create a comprehensive data frame upon which we would do our analysis. To accomplish this, we joined our distinct data tables based on locality identifiers such as zip code, state, county name, or FIPS code.

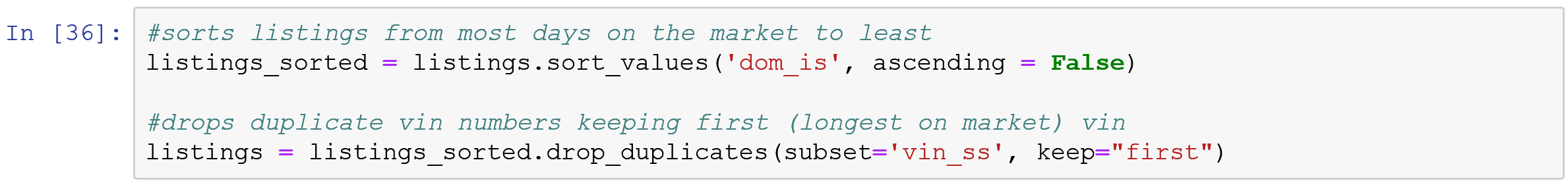


We next merged our consolidated ‘listings’ data frame with our SOI data.



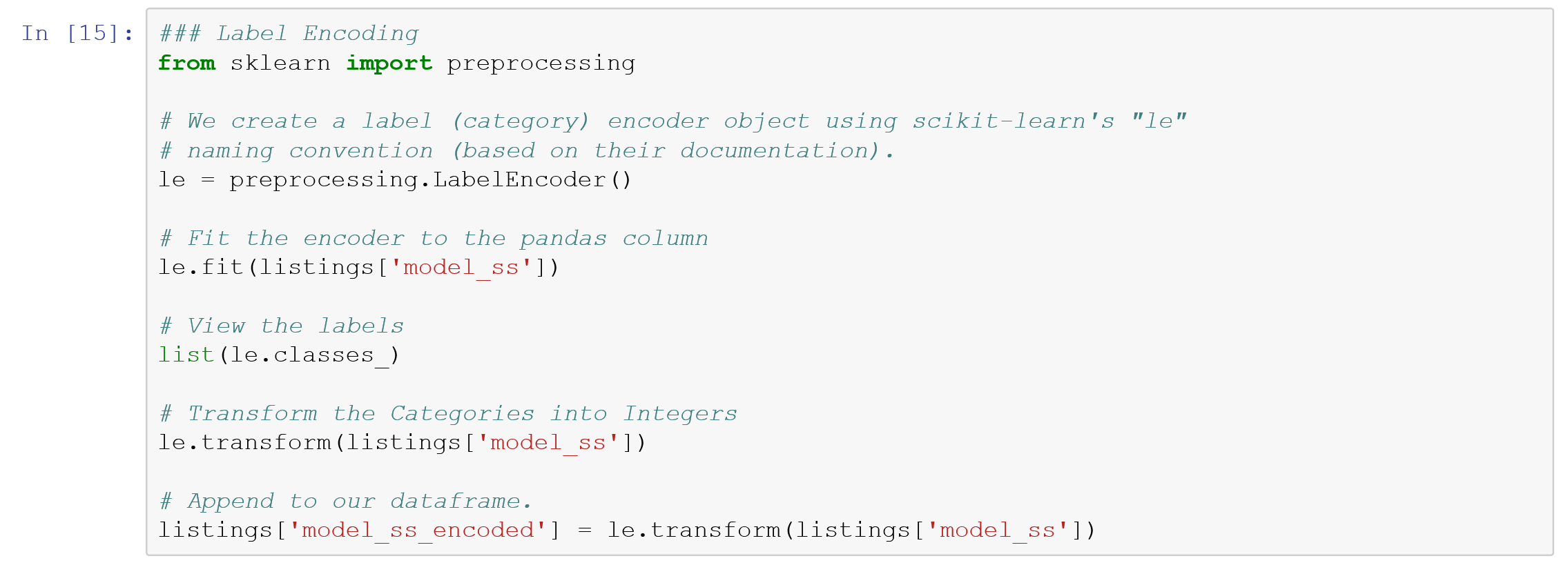
## Drop Duplicates

We noticed that vehicles may be listed across multiple dealerships. To avoid the risk of overtraining our models by feeding it the same vehicle information multiple times, we only kept the oldest vehicle listing in instances in which a unique VIN was listed numerous times.



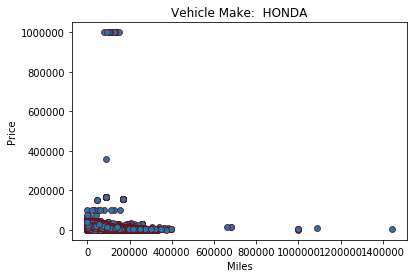
## Label Encoding

We encode our categorical data into numerics so that we can use the various statistical analysis and machine learning packages. The following code snippet provides a general template for the process used across each categorical variable we fed into our models.

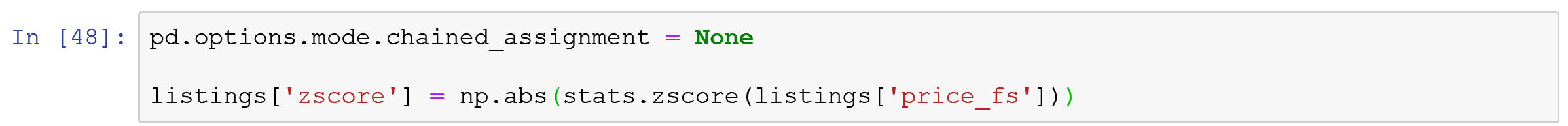


## Outliers

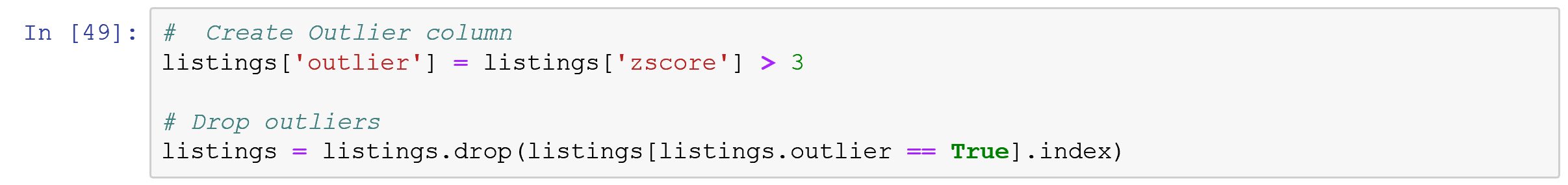
Our initial dataset shows a small number of outliers when viewing vehicle listings’ price.



We believe these may have incorrect data and we have elected to remove these outliers. To do so, we determine the z-score for each listing’s corresponding price value:



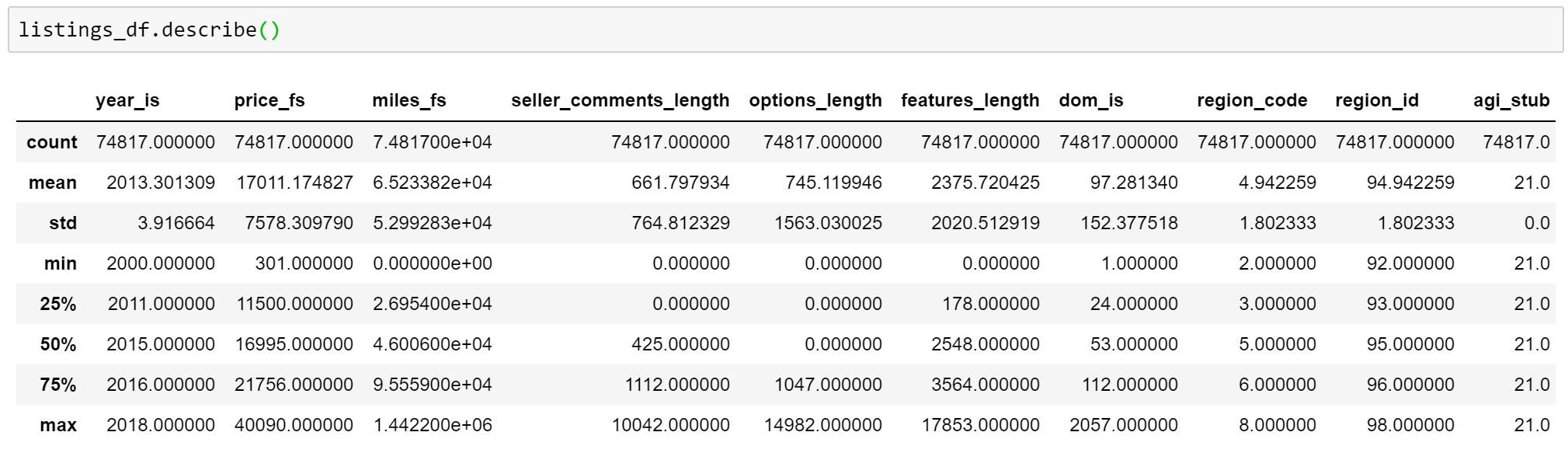
For purposes of our analysis, we drop every instance which has a z-score of greater than 3 as this is our outlier threshold.



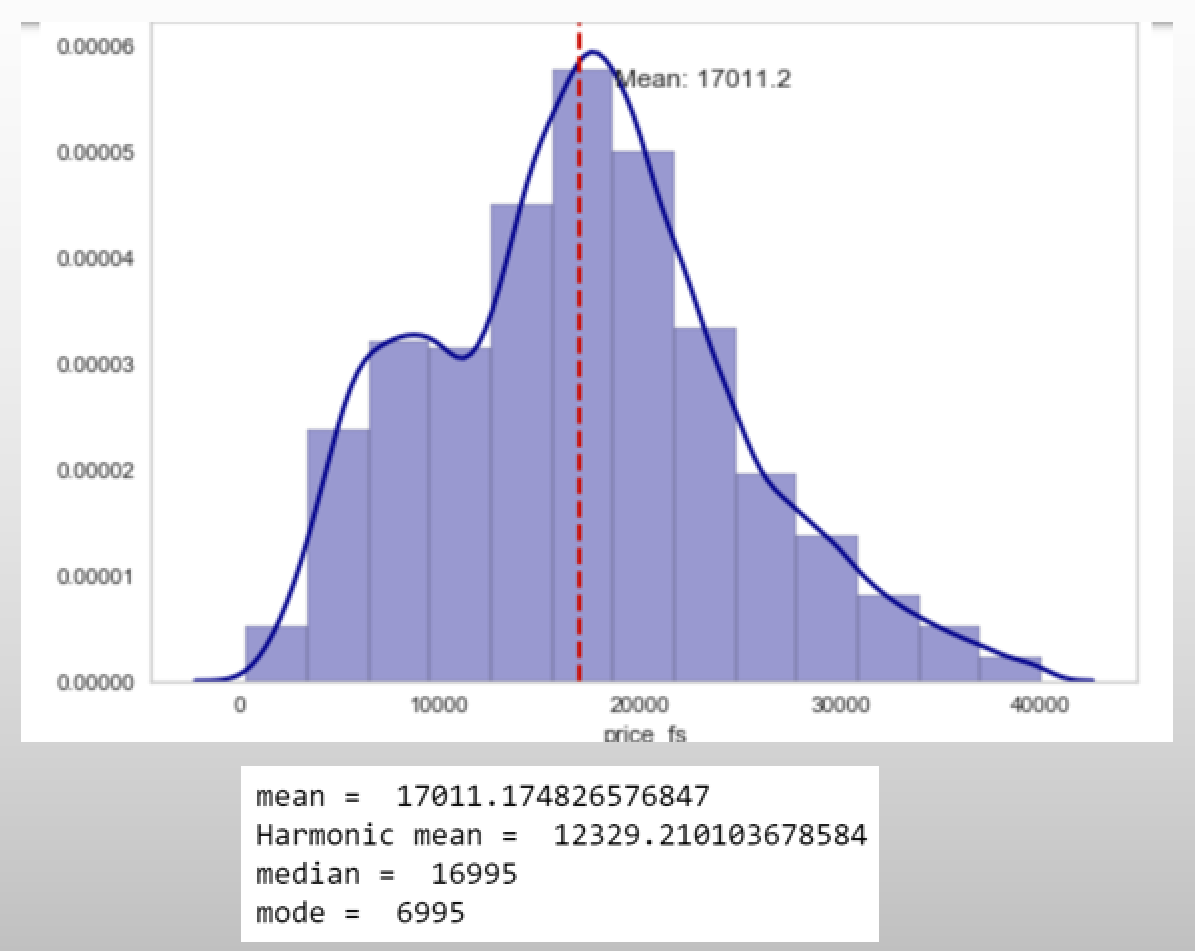
# STATISTICAL ANALYSIS

## Distribution

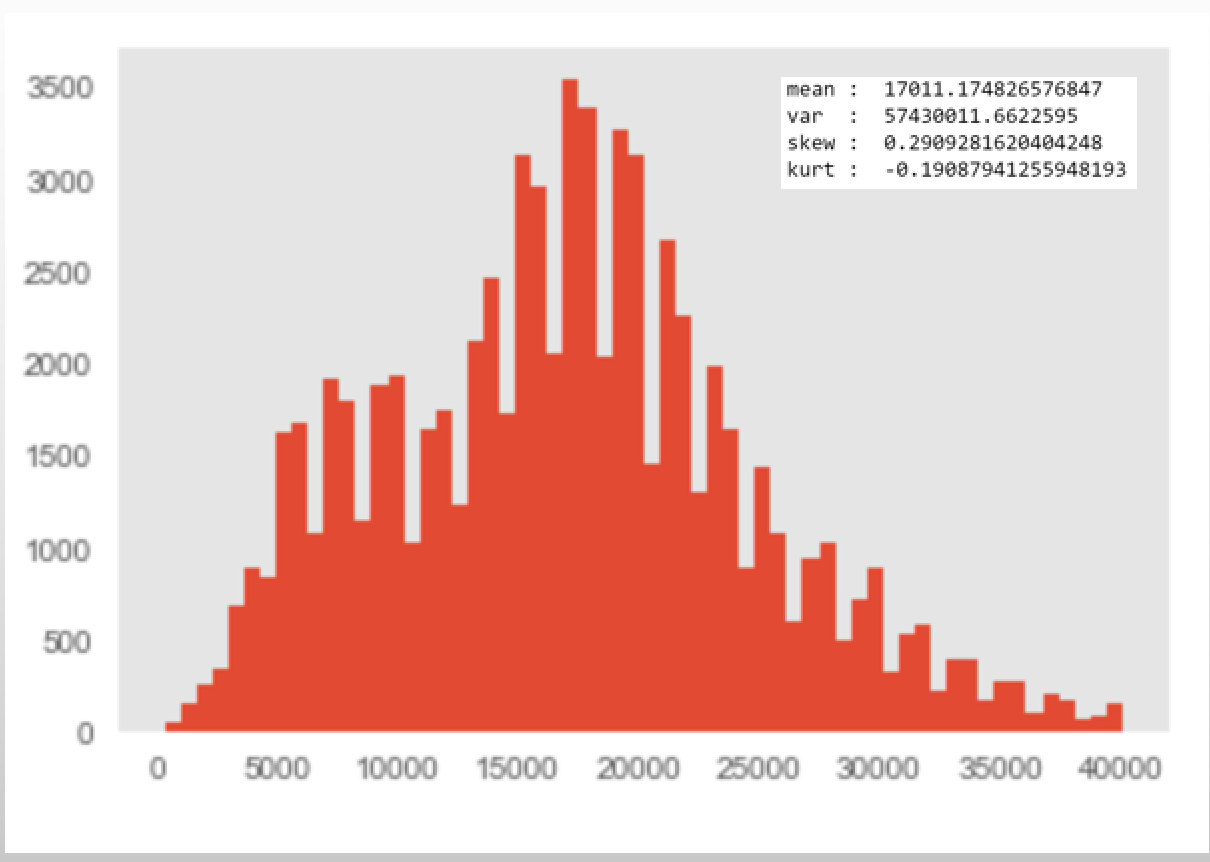
We began the analysis by evaluating the distribution of the Honda dataset. We ran the dataframe.describe() method to view some basic stats.



To further evaluate the distribution, we plot the histogram and density of all prices for Honda. With a mean of 17,011 and standard deviation of 7,578, the distribution appears to be approximately normal. In addition, the mean of 17,011 and median of 16,995 are nearly equal.

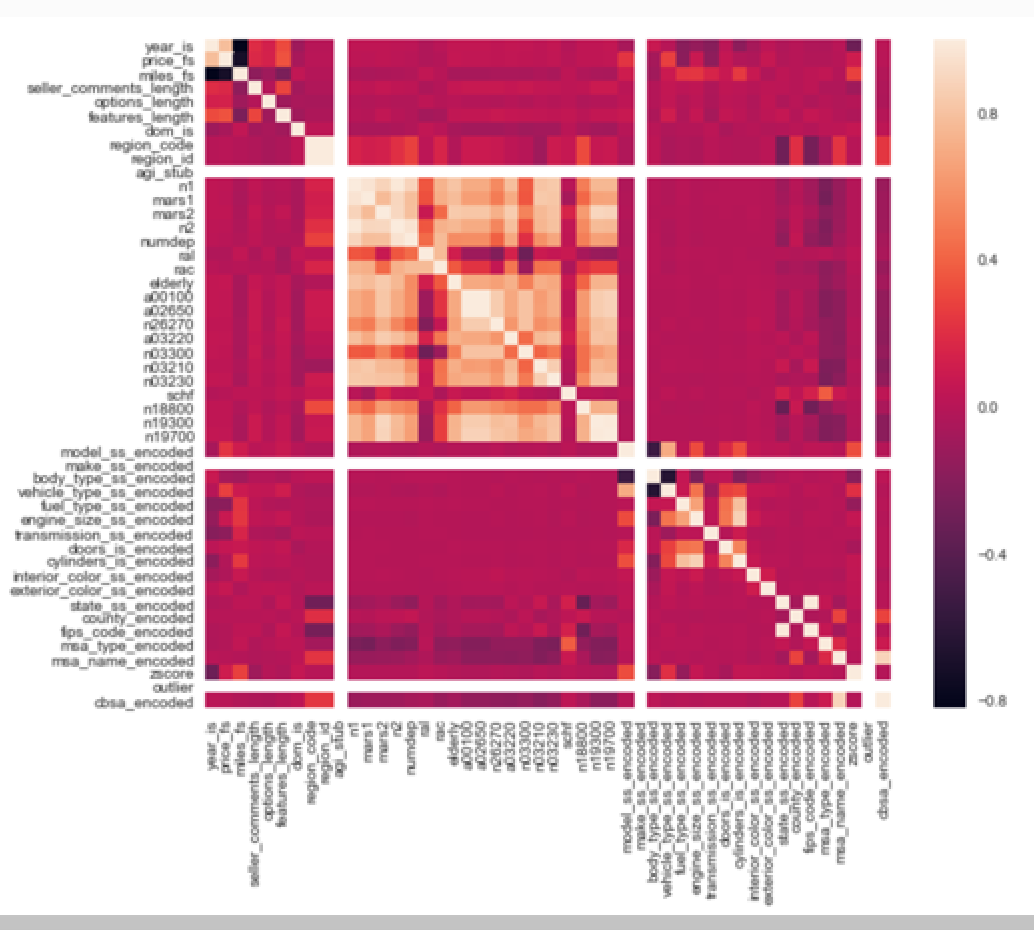


We also calculated skewness and kurtosis. Skewness at 0.29 was within the range of fairly symmetrical. Kurtosis was -0.19, indicating a light-tailed distribution. This gave us an even stronger indication that the data was normal.

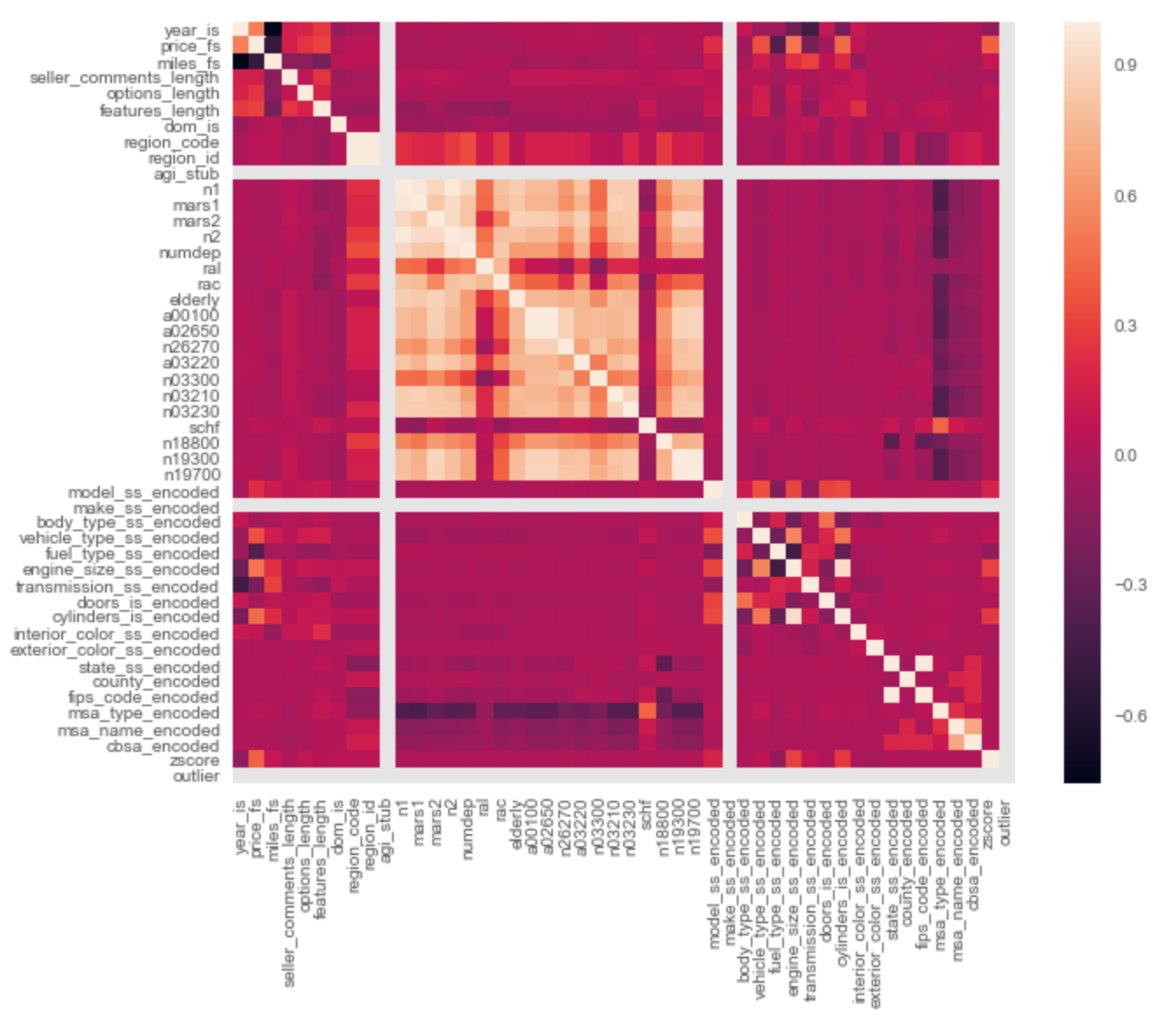


## Correlation

The Correlation Matrix shows that the feature with the greatest impact on price was year of the vehicle. Other features that correlated with price were car attributes such as features length (car features /specs described in the listing), vehicle type, and car model. Does not show a correlation between income attributes and price. Also does not show a strong correlation between price and region/metropolitan vs. non-metropolitan area.



We ran a statistical analysis on the Chevrolet brand as well, to better understand how the results may differ across brands. Results were similar. Year of vehicle and vehicle type impacted price - as well as engine size and cylinders.



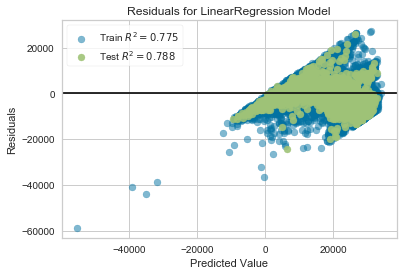
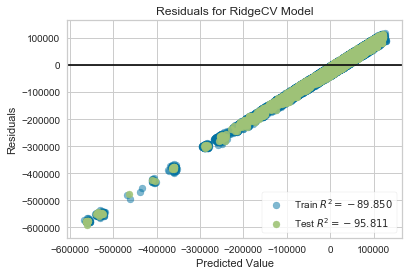
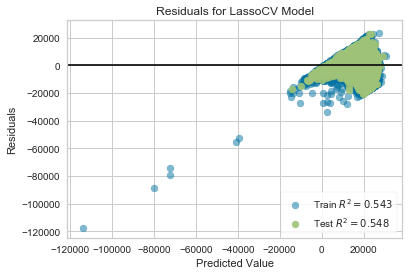
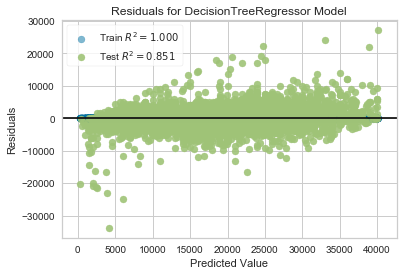
# Modeling

After ingesting and wrangling the data to create a usable instance, we split the data into a training and testing split in order to evaluate the models on data that was not used to train the model. We initially chose four regression models to test. The models chosen were a linear, Ridge, Lasso, and Decision Tree regressions.

After running each model on our entire training and testing dataset, we observed that each models performed fairly well. While the Decision Tree regression performed the best, and the Lasso model performed the worst, we determined that no models stood out enough to warrant an elimination of other models at this point.

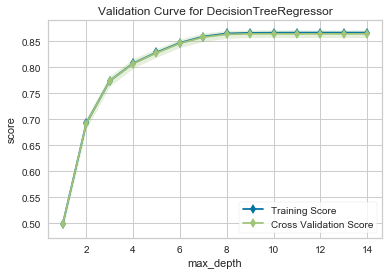
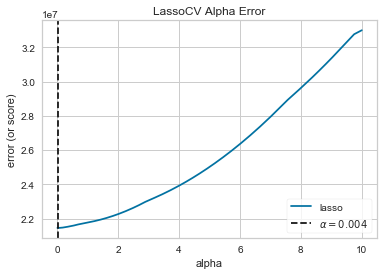
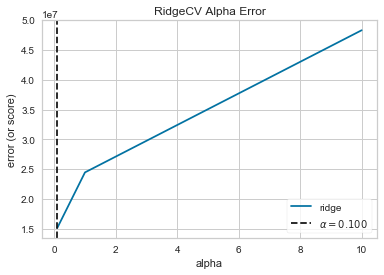
|  |  |
| --- | --- |
| **Model** | **Cross Validation Accuracy** |
| Linear | 0.80773 +/- 0.00626 |
| Ridge | 0.80779 +/- 0.00629 |
| Lasso | 0.80762 +/- 0.00630 |
| DecisionTree | 0.84888 +/- 0.00781 |

In the next step, we plotted the residuals for each model. This helped illuminate some potential problems with certain models. Most clearly, there was an issue with the Ridge model. It had trouble estimating the price of cars on both the high and low end of the prices. This resulted in a model with negative r2 values and no predictive powers. The Linear and Lasso models both suffered from some heteroscedasticity problems. This is shown as there were more error in the higher priced cars. This is due to our underlying data, as there were relatively few “high priced” used Hondas, so the models had a difficult time learning how to predict them. The decision tree did not have any of these problems.

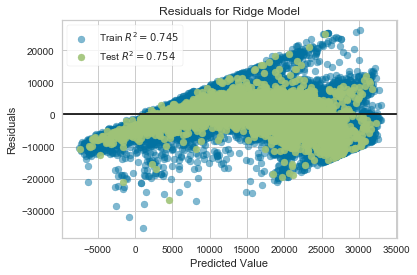
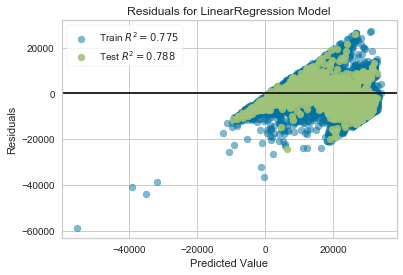


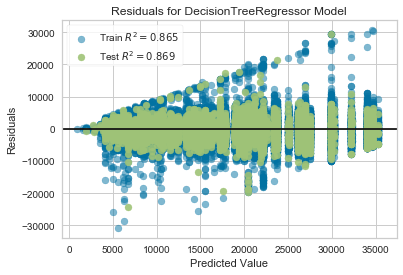
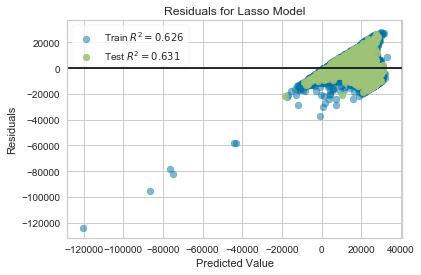
After running our initial models, we decided to eliminate any unnecessary features. We did this by first running a ranked feature to find the ideal number of features for each model. Once we knew the ideal number of features, we were then able to run a recursive feature elimination to find which features to drop. This process gave us several insights. First, the ranked feature suggested we eliminate at least some features for each model. Second, each model had different features that were recommended to be eliminated. While feature selection recommended that different features be eliminated for each model, there were some things that were consistent across all models. Our features can be categorized into two different categories: data describing the car or data describing the location the car is sold in. After feature selection, all models except the decision tree kept some of the features that described the car and the location. The features that were dropped depended on the model, but one thing that determined if it was dropped was the level of correlation to other features in the model. For example, in some models, zipcode was dropped, while in others, the state was dropped, while in others, the BEA region was dropped. These features are highly correlated, and different models preferred using different features. As discussed above, the decision tree model was different from all other models in the features it thought was important. Notably, it decided to keep only two features: year of the car and the make of the car. This extremely simple model is interesting as an opposition to our other models.

After determining the significant features for each model, we tuned our hyper parameters using only the significant features. For the Lasso and Ridge model, we used the AlphaSelection function and for the Decision Tree model, we used the validation curve to find the ideal depth of the decision tree. For the Lasso model, we found an ideal alpha of 0.1. For the Ridge model, we found an ideal alpha of 0.004. For the Decision Tree model, we found an ideal max depth of 8, as at any deeper, there stopped being a distinguishable increase after 8 and any more would leave us at a risk of overfitting.



After we selected the best features and tuned the hyper parameters for each model, we ran the models again with the new features and hyper parameters. After doing this, we found that the Decision Tree performed the best. However, the linear and Ridge model also performed well. The Lasso model performed the least well, which is due to the issues we found earlier in predicting the price of the highest and lowest price models. Given the success of the Linear and Ridge models, which use both location and car based features, we are able to conclude that both the features of a used car and the location it is sold in have an effect on the price the car is sold for. However, the Decision Tree model was the most successful model, which points out that the year and the model of the used car are clearly the most significant determinants of the price.





A potential limitation of our model is that it may be suited for only used Hondas and may not be able to be extrapolated to other cars. In order to see how well the model would apply to other cars, we ran the same process on all of the listings we had for used Chevrolets. This resulted in slightly lower R2 values for the Chevrolet dataset. However, the difference for all models except the Lasso is negligible and is still high enough that we are confident that the same methodology can be used to predict the price of all used cars, not just Hondas.

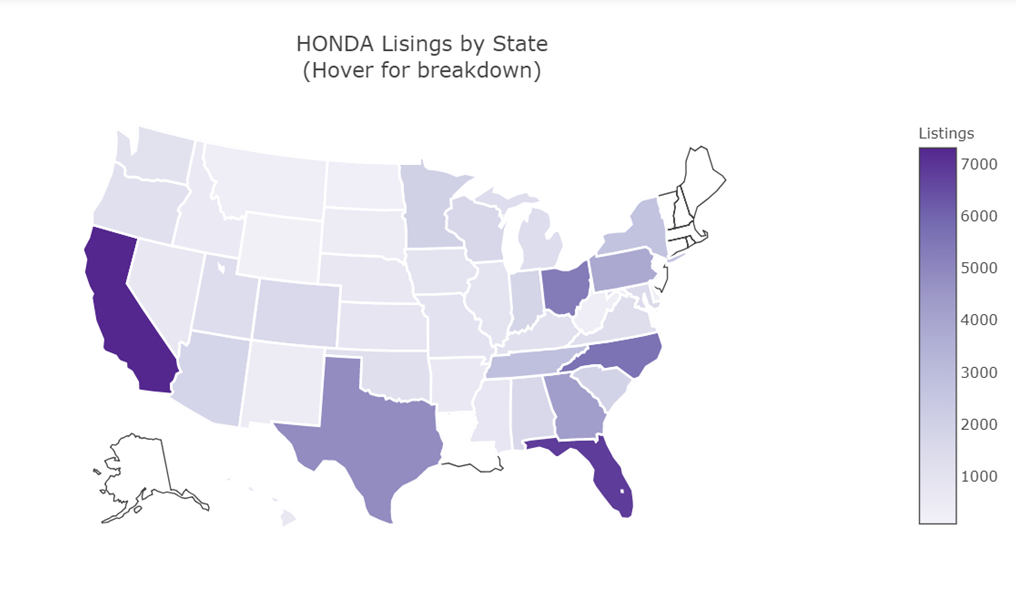
|  |  |
| --- | --- |
| **Model** | **R2** |
| Linear | 0.78 |
| Ridge | 0.74 |
| Lasso | 0.41 |
| DecisionTree | 0.86 |

**Feature Summary**

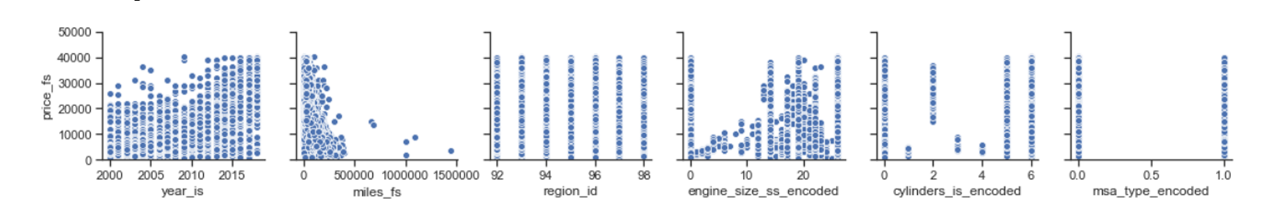
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable Name** | **Description** | **Linear** | **Ridge** | **Lasso** | **Decision Tree** |  | **Variable Name** | **Description** | **Linear** | **Ridge** | **Lasso** | **Decision Tree** |
| year\_is | Model Year of Car | Kept | Kept | Dropped | Kept |  | n03210 | Number of returns with student loan interest deduction | Kept | Kept | Dropped | Dropped |
| miles\_fs | Miles on Car | Kept | Dropped | Kept | Dropped |  | n03230 | Number of returns with tuition and fees deduction | Kept | Kept | Dropped | Dropped |
| seller\_comments\_length | Number of characters used to describe the vehicle listing | Kept | Dropped | Kept | Dropped |  | schf | Number of farm returns | Kept | Kept | Dropped | Dropped |
| option\_length | Number of characters used to describe the vehicle options | Kept | Kept | Kept | Dropped |  | n18800 | Number of returns with Personal property taxes | Kept | Dropped | Kept | Dropped |
| features\_length | Number of characters used to describe the vehicle features | Kept | Kept | Kept | Dropped |  | n19300 | Number of returns with Home mortgage interest paid | Kept | Dropped | Kept | Dropped |
| dom\_is | Days On Market | Kept | Kept | Kept | Dropped |  | n19700 | Number of returns with Total charitable contributions | Kept | Dropped | Kept | Dropped |
| region\_code | Bureau of Economic Analysis region code designation | Kept | Kept | Kept | Dropped |  | model\_ss\_encoded | Encoded variable of Model | Kept | Kept | Kept | Kept |
| region\_id | Bureau of Economic Analysis region identification designation | Kept | Kept | Kept | Dropped |  | make\_ss\_encoded | Encoded variable of Make | Dropped | Dropped | Kept | Dropped |
| agi\_stub | Size of adjusted gross income [1-6] | Dropped | Dropped | Kept | Dropped |  | body\_type\_ss\_encoded | Encoded variable of body type | Kept | Kept | Kept | Dropped |
| n1 | Number of returns | Kept | Kept | Dropped | Dropped |  | vehicle\_type\_ss\_encoded | Encoded variable of vehicle type | Kept | Kept | Dropped | Dropped |
| mars1 | Number of single returns | Dropped | Dropped | Dropped | Dropped |  | fuel\_type\_ss\_encoded | Encoded variable of fuel type | Kept | Kept | Dropped | Dropped |
| mars2 | Number of married returns | Dropped | Dropped | Dropped | Dropped |  | engine\_size\_ss\_encoded | Encoded variable of engine type | Kept | Kept | Dropped | Dropped |
| n2 | Number of exemptions | Kept | Kept | Kept | Dropped |  | transmission\_ss\_encoded | Encoded variabel of transmission | Kept | Kept | Dropped | Dropped |
| numdep | Number of dependents | Kept | Dropped | Dropped | Dropped |  | doors\_is\_encoded | Encoded variabel of number of doors | Kept | Kept | Dropped | Dropped |
| ral | Number of refund anticipation loan returns | Kept | Kept | Dropped | Dropped |  | cylinders\_is\_encoded | Encoded variabel of cylinders | Kept | Kept | Dropped | Dropped |
| rac | Number of refund anticipation check returns | Kept | Dropped | Dropped | Dropped |  | interior\_color\_ss\_encoded | Encoded variable of interior color | Kept | Kept | Dropped | Dropped |
| elderly | Number of elderly returns | Kept | Kept | Dropped | Dropped |  | exterior\_color\_ss\_encoded | Encoded variable of exterior color | Kept | Kept | Dropped | Dropped |
| a00100 | Adjust gross income | Dropped | Dropped | Kept | Dropped |  | state\_ss\_encoded | Encoded variable of state car is listed in | Kept | Kept | Dropped | Dropped |
| a02650 | Total income amount | Dropped | Dropped | Dropped | Dropped |  | county\_encoded | Encoded variable of county car is listed in | Kept | Dropped | Dropped | Dropped |
| n26270 | Number of returns with partnership/S-corp net income (less loss) | Kept | Kept | Dropped | Dropped |  | fips\_code\_encoded | Unique county identifier | Kept | Kept | Dropped | Dropped |
| a03220 | Educator expenses amount | Kept | Kept | Dropped | Dropped |  | msa\_type\_encoded | Encoded variable indicating metropolitan (urban) status | Kept | Kept | Dropped | Dropped |
| n03300 | Number of returns with Self-employed (Keogh) retirement plans | Kept | Kept | Dropped | Dropped |  | msa\_name\_encoded | Encoded variable for each unique Metropolitan Statistical Area | Kept | Kept | Dropped | Dropped |
|  |  |  |  |  |  |  | cbsa\_encoded | Encoded variable for each unique Core-Based Statistical Area | Kept | Kept | Dropped | Dropped |

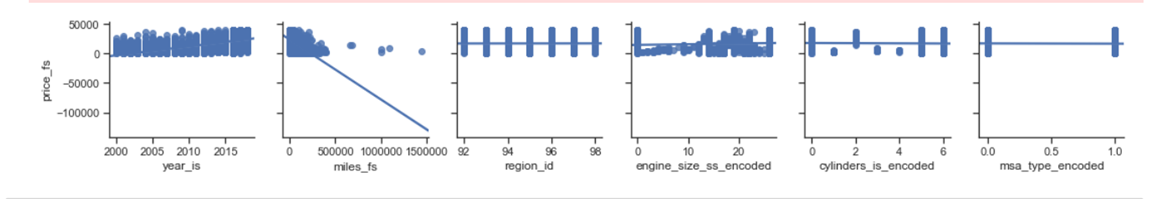
# Visualization

First, we needed to see wanted to see the number of listings per state:

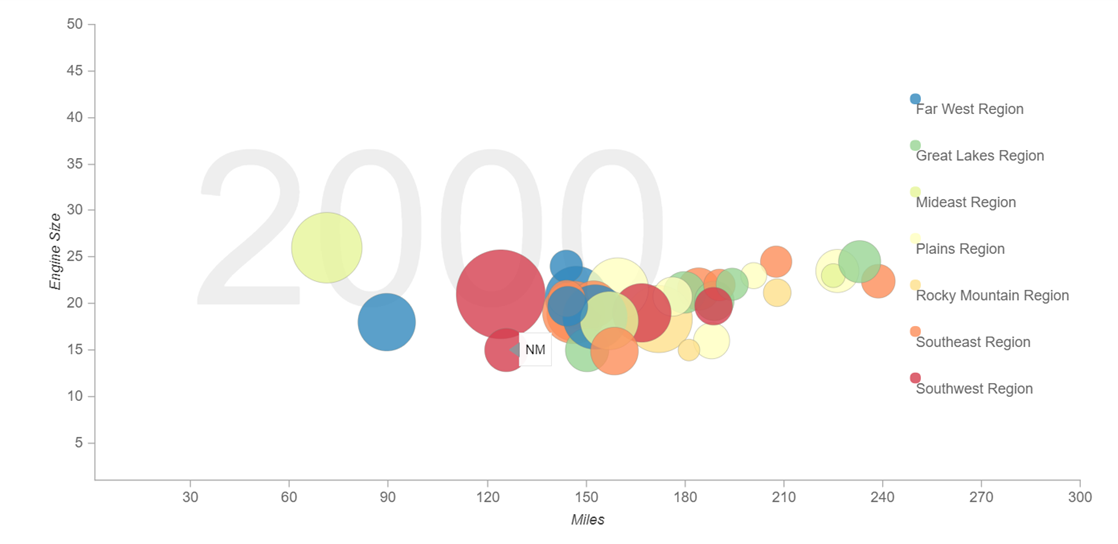


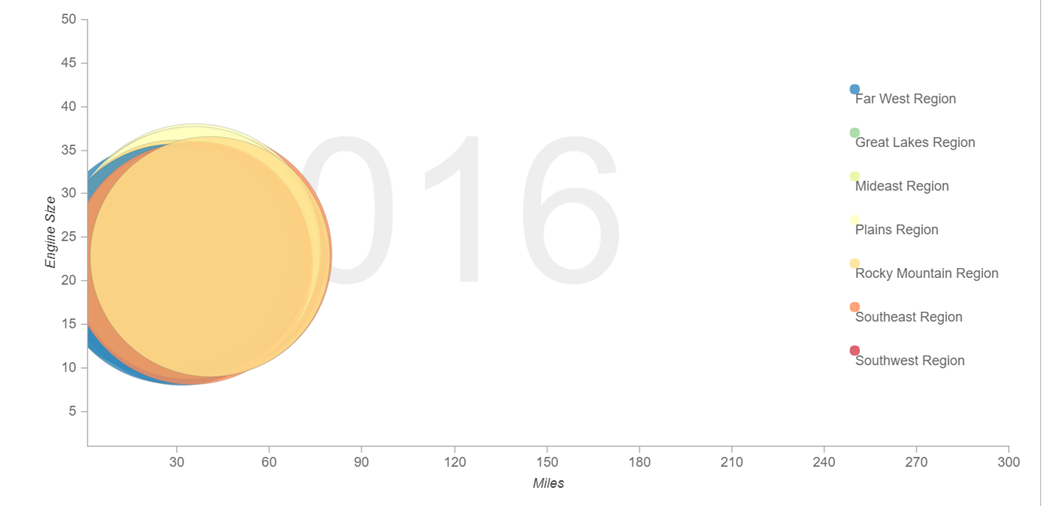
Then, we needed to test our intuition. So, we prepared this pairplot to try to visualize relationships between features. During this phase, we still did not have the results from the data science portion, so we just tried to see the linear relationships.





After looking at how price had a variation according to miles and year, we wanted to see how it changed. So we decided to use this Bokeh graph.





# Dashboard

Using Python library (Dash), we created a data dashboard, assuming it is the most efficient way to track multiple data sources of the price of used vehicles across multiple regions in the US. In essence, the dashboard is used as a recommender for both dealers and buyers of used vehicles to make better choices.

The dashboard has demonstrated to quickly prove or disprove our hypothesis. In the case of our project (Used Cars Analyzer), the dashboard quickly display that used Honda or Chevrolet’s price has less variance across the geographical regions in the United States. With that said, dashboard can also provide a central location for our team to monitor and analyze if there are any change in price based on any of the features we chose in our datasets. Real-time monitoring reduces the hours of analyzing and long line of communication that previously challenged businesses.

Conclusion: Dashboard is the frontend to the team’s analytical Python backend.

# Conclusion

After examining our data and running a variety of statistical and machine learning methods, we have drawn several insights about our data. First, a decision tree model was the best performing model to predict used car prices. However, both a linear and ridge regression worked well in predicting the price. The decision tree regression, which performed the best, took into account only the year of the car and the car’s model. This tells us that these are the two most important determinants of a used car’s price. However, the linear and ridge model performed well and these models took in a wider variety of features into consideration. This shows that both data that describes a used car and data that describes the location the car is sold in has some effect of the used car price. We also were able to conclude that a lasso model was not able to predict the price of a used car.

There was some limitations with our analysis that points to possible next steps to explore. We initially only looked at used Hondas. We then brought in Chevrolets to see if our models could perform well on other used cars and found that they could. However, we still have not considered more luxury car brands like BMW, Mercedes, and Tesla. This could be a point of future analysis. Finally we could also include other descriptive variables such as moonroof, GPS, and other technology that could have an impact on price, but was not included in our data.

1. [https://bucks.blogs.nytimes.com/2013/02/18/used-car-prices-vary-by-market-analysis-finds](https://bucks.blogs.nytimes.com/2013/02/18/used-car-prices-vary-by-market-analysis-finds/) [↑](#footnote-ref-0)
2. <https://www.marketcheck.com/automotive> [↑](#footnote-ref-1)
3. <https://www.bls.gov/oes/> [↑](#footnote-ref-2)
4. <https://www.farinspace.com/us-cities-and-state-sql-dump/> [↑](#footnote-ref-3)
5. <https://www.irs.gov/statistics/soi-tax-stats-individual-income-tax-statistics-zip-code-data-soi> [↑](#footnote-ref-4)
6. PostgreSQL 10.5. [↑](#footnote-ref-5)
7. Our server has 2 GB of memory and a 50 GBs of disk space. [↑](#footnote-ref-6)
8. We are currently running Ubuntu 18.04.1 x64. [↑](#footnote-ref-7)
9. <https://github.com/georgetown-analytics/Used-Cars-Analyzers> [↑](#footnote-ref-8)