SI 618 Project 2 Report

Yifei Sun

1.Motivation

In this project, I will study the used car trade in the United States. Unlike a house, a car is not a fixed asset, so it is difficult to appreciate in value with the additional land resources as property does. Cars depreciate relatively quickly and often exponentially, i.e., they depreciate a lot at first and then stabilize in price. The depreciation of a car is largely influenced by its age and mileage. In addition, the pattern of depreciation varies from make to make and model to model. Researching this issue will help us make the most informed choice when buying a used car. The manufacturer, model, age and mileage of the used car you choose will affect the price when you sell it again.

In this report I will examine the following three questions:

1. Whether there are differences in the age and mileage of different types and different conditions of cars?

   Analysis:

   1. Plot the probability distribution of mileage of each condition and each type of cars of different manufacturers to see if there is a relationship.

2.How can I better predict the price of a used car?

   (1) What is the relationship between mileage and age and the price of a used car?

   Analysis:

   2. Explore the relationship between used prices of cars of the same make and model with age, visualize it, and verify if it is an exponential relationship.

   Explore the relationship between used price and mileage for the same make and model of car, visualize it, and verify if it is exponential.

   (2) Is there a correlation between the various characteristics of used cars? How to predict the price of used cars better by using principal component analysis regression?

   Analysis:

   3. Build a correlation heatmap of different features of secondhand car

   4. Modeling forecasts of secondhand prices using multiple regression analysis (linear or non-linear). Since we already know that mileage and age are exponentially related to used prices, a non-linear transformation may be required. And there may be a linear relationship between mileage and vehicle age, which may require variable selection and some methods to eliminate linearity, such as principal component analysis and then linear combination of features through eigenvectors of the covariance matrix, before analysis.

3. Is there a time of the day or a day in the month when the secondhand car transaction is more frequent?

   5. According to the posting time analysis, observe whether there is a certain time of day used car transactions are more frequent, whether there is a certain day of the week time period used car transactions are more frequent

2.Data Source

The data I used came from Kaggle, a user's crawl of used car transaction data from Craigslist. It is a csv file, which includes about 420,000 pieces of data and is dated from April 4th, 2021, to May 5th ,2021. It includes used cars that were manufactured within the last few decades, mostly in the last 20 years. It includes the location, vehicle's manufacturing company, make, model, mileage, and other vehicle configuration information.

The “id” of the vehicle is in int64 format, “price” is in int64 format. “year” and “odometer” are in floart64 format. “manufacturer”, “model”, “condition”, “cylinders” and other important fields are all in object format, meaning they are strings.

There are many incomplete data with missing values.

The data can be accessed at https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data.

3. Method

For the first question, I first calculate the total number of cars of each manufacturer to figure out what the most popular manufacturers are. I selected the top five most popular manufacturers and for each manufacturer and found out their most frequent types and conditions. Then I drew the violin plots to show the age and mileage distribution of each category. There are a lot of data that do not have a condition or type, so I have to delete those data. And there are some vehicles with extremely big age and mileage. I considered them to be outliers and also removed them. This one is quite easy; I did not encounter any difficulties.

For the second question, I did similar preliminary work to find out the most popular manufacturers. For every manufacturer, I found out the 16 most popular models. Then I continued to perform regression analysis and draw scatterplots. There are also a lot a data with extreme age and mileage value. I deleted most of them and set an upper and lower limit of the value, and most of the data fall in this range. Also, I had to delete the vehicles with price of 0 in order to do log transformation. Before doing PCA, I rescaled and normalized the data to avoid the effect of different units and measurements. I encountered a lot of difficulties in this part, mostly due to a lack of clarity of thought. First of all, because I had too many data points, the points in the drawn scatter plot overlapped together. I adjusted the transparency of the points to a very small number, and drew each model of car separately, so that the scatter plot was clearer, and the density of the points could be indicated by the color shades. Also, I tried a lot of unsuccessful analysis. When I did the logarithmic transformation, at first, I did not do logarithmic transformation for the price but for the age and mileage of the car, and thus the linearity of the drawn scatter plot was not perfect. Later I found this problem and according to general economic theory, I should have done a logarithmic transformation of the prices. The linearity was perfect after this operation. Also, there can be many kinds of regression modeling analysis about price, age, and mileage, and I only chose the one I think is the most reasonable. After OLS regression, I found the coefficient seem to be unstable due to outliers, then I tried robust regression with Huber weighting method in order to get more stable coefficient for variables in the regression model.

For the third question, I first calculated the number of posts per day, per hour, and then did a regression analysis on them, treating seven days a week and 24 hours a day as type variables. The residuals of the regression analysis are treated as the number of posts with the periodicity factor removed. This process was not affected by missing values and outliers, and no other difficulties were encountered.

4.Analysis and Results

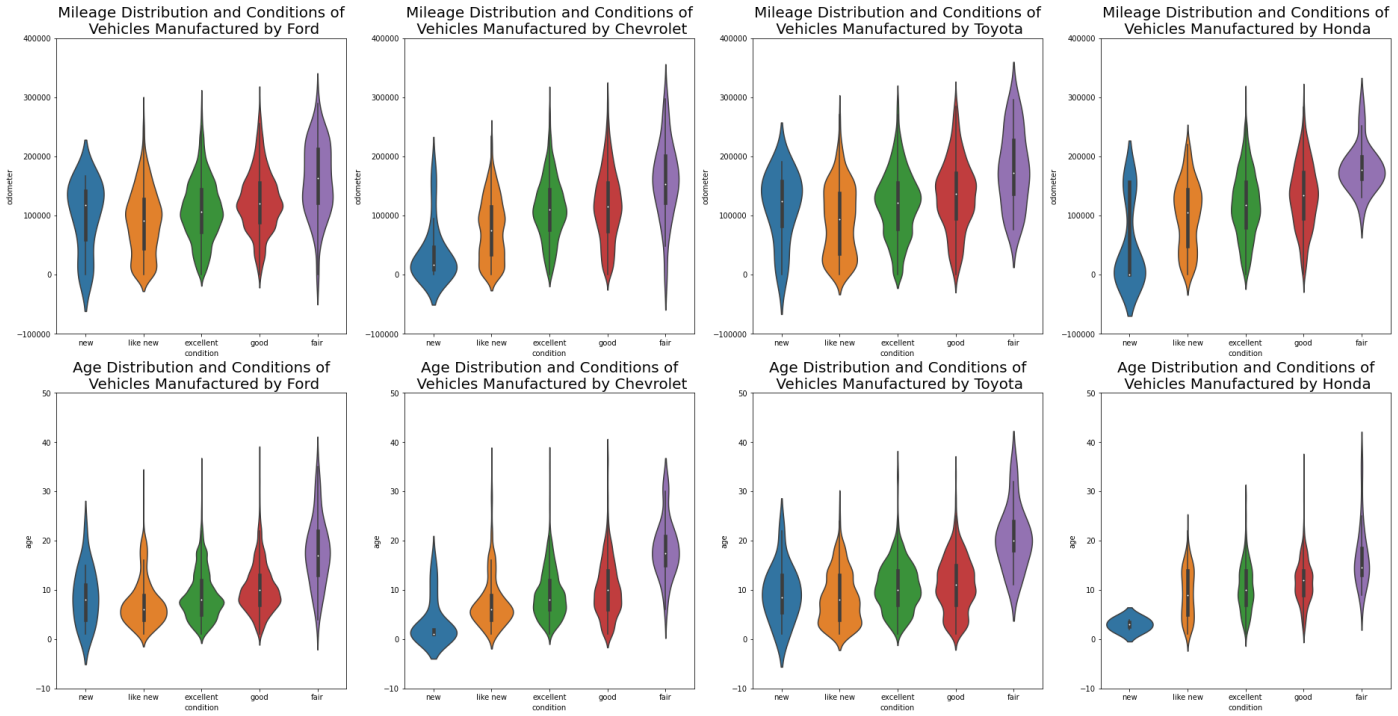
1. Whether there are differences in the age and mileage of different types and different conditions of cars?

Analysis:

1. Plot the probability distribution of mileage of each condition and each type of cars of different manufacturers to see if there is a relationship.

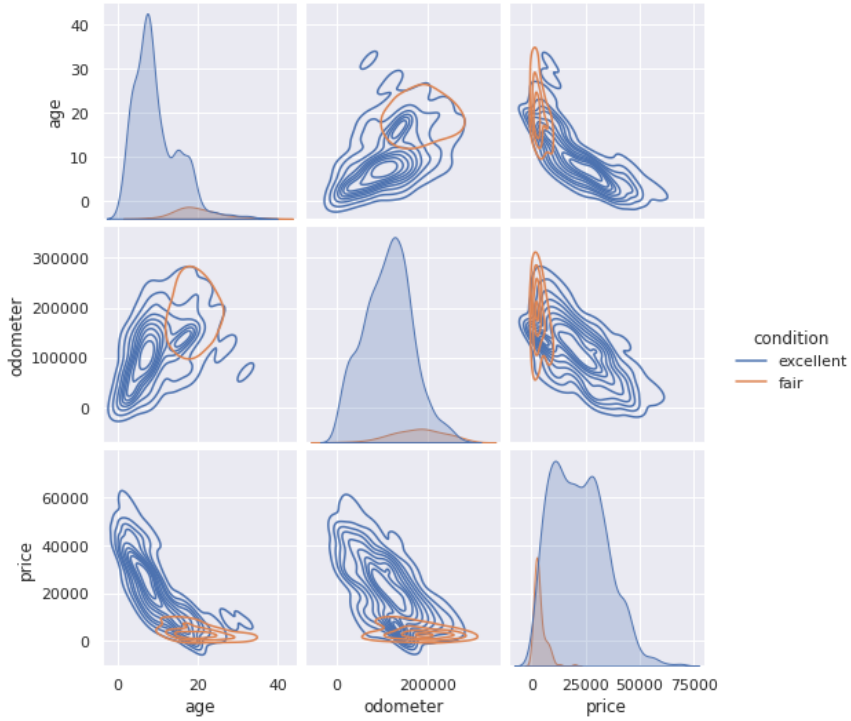
Here I chose violin plot to show the distribution of different categories. I chose the 5 most frequent manufacturers and the respective most frequent types and conditions. I filtered out the data that has extreme and unreasonable age and mileage. Then draw the two plots side by side.

```
n=3
manu_list = [x[0] for x in Counter(list(vehicle_1[vehicle_1["manufacturer"].notna()]["manufacturer"])).most_common()][:n]
fig, axs = plt.subplots(2, n, figsize=(8*n, 8*2))
for i in range(len(manu_list)):
    model=manu_list[i]
    df = vehicle_1[vehicle_1["manufacturer"]==model]
    df["year"] = df["year"].astype(int)
    df["age"] = 2021 - df["year"]
    df["price"] = df["price"].astype(float)
    df=df[["model","manufacturer","condition","type","price","age","odometer"]]
    df = df[(df["odometer"]<=300000) & (df["odometer"]>0) &(df["age"]>0) & (df["age"]<40)]
    df=df[df["condition"].isin([x[0] for x in Counter(list(df["condition"])).most_common()][0:6])]
    my_order = ["new","like new","excellent","good","fair"]
    g1 = sns.violinplot(data=df, x="condition", y="odometer",ax=axs[0,i],\
                        order=my_order)
    g1.set_ylim([-100000,400000])
    g1.set_title("Mileage Distribution and Conditions of\n Vehicles Manufactured by "+model.capitalize(),fontsize=20)
    df=df[df["type"].isin([x[0] for x in Counter(list(df["type"])).most_common()][1:6])]
    g2 = sns.violinplot(data=df, x="condition", y="age",ax=axs[1,i%n],\
                        order=my_order)
    g2.set_ylim([-10,50])
    g2.set_title("Age Distribution and Conditions of\n Vehicles Manufactured by "+model.capitalize(),fontsize=20)
```



Here we can see that there is significant difference in age and mileage across different conditions. The categorical variable of condition seems to be ordinal. Fair cars have longest age and mileage. New cars have the shortest age and mileage. We can see that new cars for Ford seem to have significant longer mileage and age than Chevrolet, which may indicate Ford cars may be more durable. The mileage distribution seems to be different across Ford, Chevrolet. Ford's distribution is left skewed, and Chevrolet's distribution is right skewed. I also did the similar analysis for different types of cars, and the results are in appendix. For types, we can see certain types of cars have slightly longer age and mileage than other, but the different is not significant. The difference of age and mileage across manufacturers are not obvious.

Also, we have seen that all the distributions seem to be skewed with a long right tail, which indicates we may need to do log transformation.



From this plot, we can see that for Ford f-150, “excellent” cars and “fair” cars have very different positions in the space spanned by age, odometer, and price. Fair cars tend to have higher age and odometer and lower price.

```
from scipy import stats
stats.ttest_ind(f_150[f_150["condition"] == 'excellent']['price']\
                ,f_150[f_150["condition"] == 'fair']['price'])
```

```
Ttest_indResult(statistic=15.929288033574391, pvalue=2.5305191438247506e-53)
```

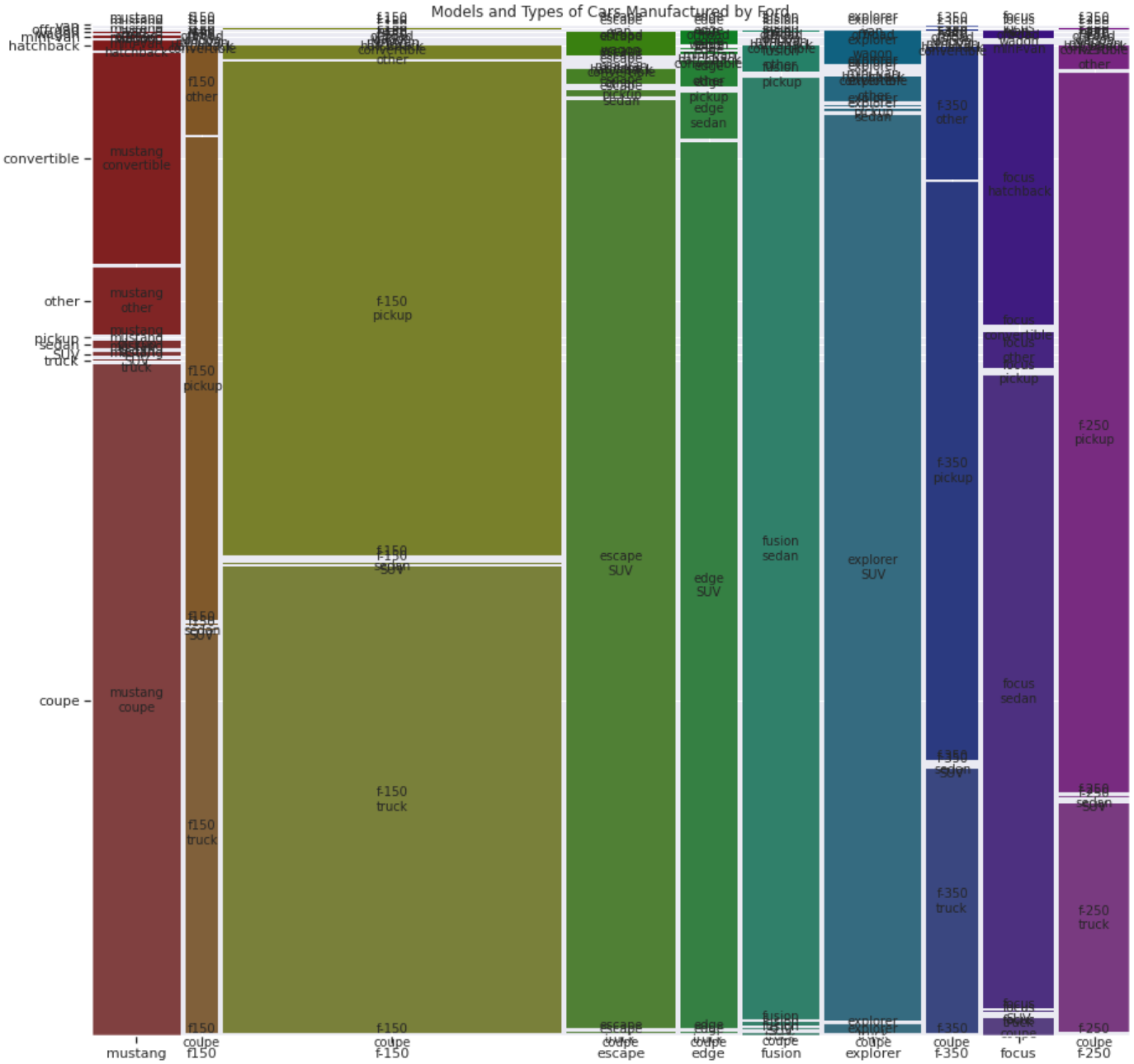
```
from scipy import stats
stats.ttest_ind(f_150[f_150["condition"] == 'excellent']['odometer']\
                ,f_150[f_150["condition"] == 'fair']['odometer'])
```

```
Ttest_indResult(statistic=-12.978899877655225, pvalue=8.769200016272438e-37)
```

```
from scipy import stats
stats.ttest_ind(f_150[f_150["condition"] == 'excellent']['age']\
                ,f_150[f_150["condition"] == 'fair']['age'])
```

```
Ttest_indResult(statistic=-19.804671331624732, pvalue=1.2022274811746099e-78)
```

Using T test, we can see the differences of age, mileage, and price are all very significant between “excellent” and “fair” cars of Ford f-150.



Using a mosaic plot, we can see that for the 10 most popular models of Ford, most models only have one or two major types. For example, for f-150, half of the cars are trucks, and the other half are pickup. Ford escape, explorer and edge are all mainly SUVs. Trucks and pickups seem to be similar, because they tend to appear together, such as f-150, f-250, f-350.

```
from scipy.stats import chi2_contingency
ford_table = ford.pivot_table(index='model', columns='type',
                values='id',aggfunc='count')
ford_table = ford_table.fillna(0)
chi2, p, dof, ex = chi2_contingency(ford_table)
print("chi2 = ", chi2)
print("p-val = ", p)
print("degree of freedom = ",dof)
print("Expected:")
pd.DataFrame(ex)
```

```
chi2 =  56005.287597204995
p-val =  0.0
degree of freedom =  99
Expected:
```

Using chi-squared test, we can see there is significant relationship between the car's type and the car's model.

2. How can I better predict the price of a used car?

(4) What is the relationship between mileage and age and the price of a used car?

Analysis:

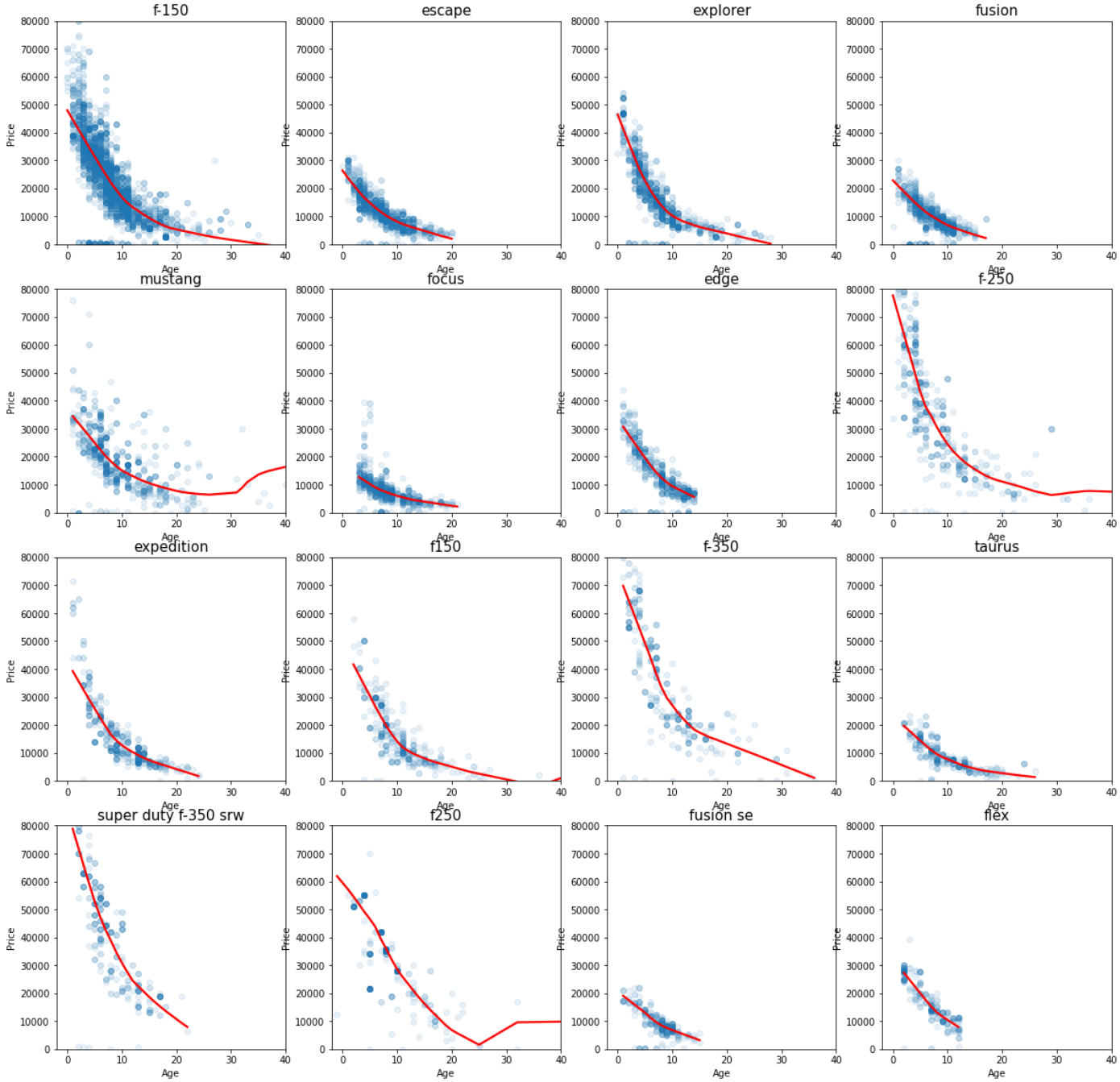
3. Explore the relationship between used prices of cars of the same make and model with age, visualize it, and verify if it is an exponential relationship.

Explore the relationship between used price and mileage for the same make and model of car, visualize it, and verify if it is exponential.

```
model="ford"
n = 4
fig, axs = plt.subplots(n, n, figsize=(20, 20))
fig.suptitle("Logged Price and Age of Cars Manufactured by "+model.capitalize()+" of Different Models", fontsize=30)
print("Price and Age of Cars Manufactured by "+model.capitalize()+" of Different Models")
df = vehicle_1[vehicle_1["manufacturer"]==model]
df["year"] = df["year"].astype(int)
df["age"] = 2021 - df["year"]
df["price"] = df["price"].astype(float)
car_list = [x[0] for x in Counter(list(df[df["model"].notna()]\
                                       ["model"])).most_common()][:n**2]
for i in range(len(car_list)):
    df_1 = vehicle_1[vehicle_1["model"]==car_list[i]]
    df_1["log_price"] = np.log(df_1["price"])
    df_2 = df_1[(df_1["age"]>=2) & (df_1["age"]<=15)]
    model1 = smf.ols('log_price ~ age', data=df_2).fit()
    g1 = sns.regplot(x="age",y=np.log(df_1["price"]),data=df_1,ax=axs[i//n,i%n],scatter_kws={'alpha':0.1},lowess=True,\
                     line_kws={"color": "red"})
    g1.set_title(car_list[i], fontsize=15)
    g1.set_xlim([-2,30])
    g1.set_ylim([7.5,12.5])
    g1.set_xlabel("Age", fontsize=10)
    g1.set_ylabel("Logged Price", fontsize=10)
    g1.text(10,12,"y="+str(round(model1.params["age"],2))+"x+"+str(round(model1.params["Intercept"],2)), fontsize=15)
```

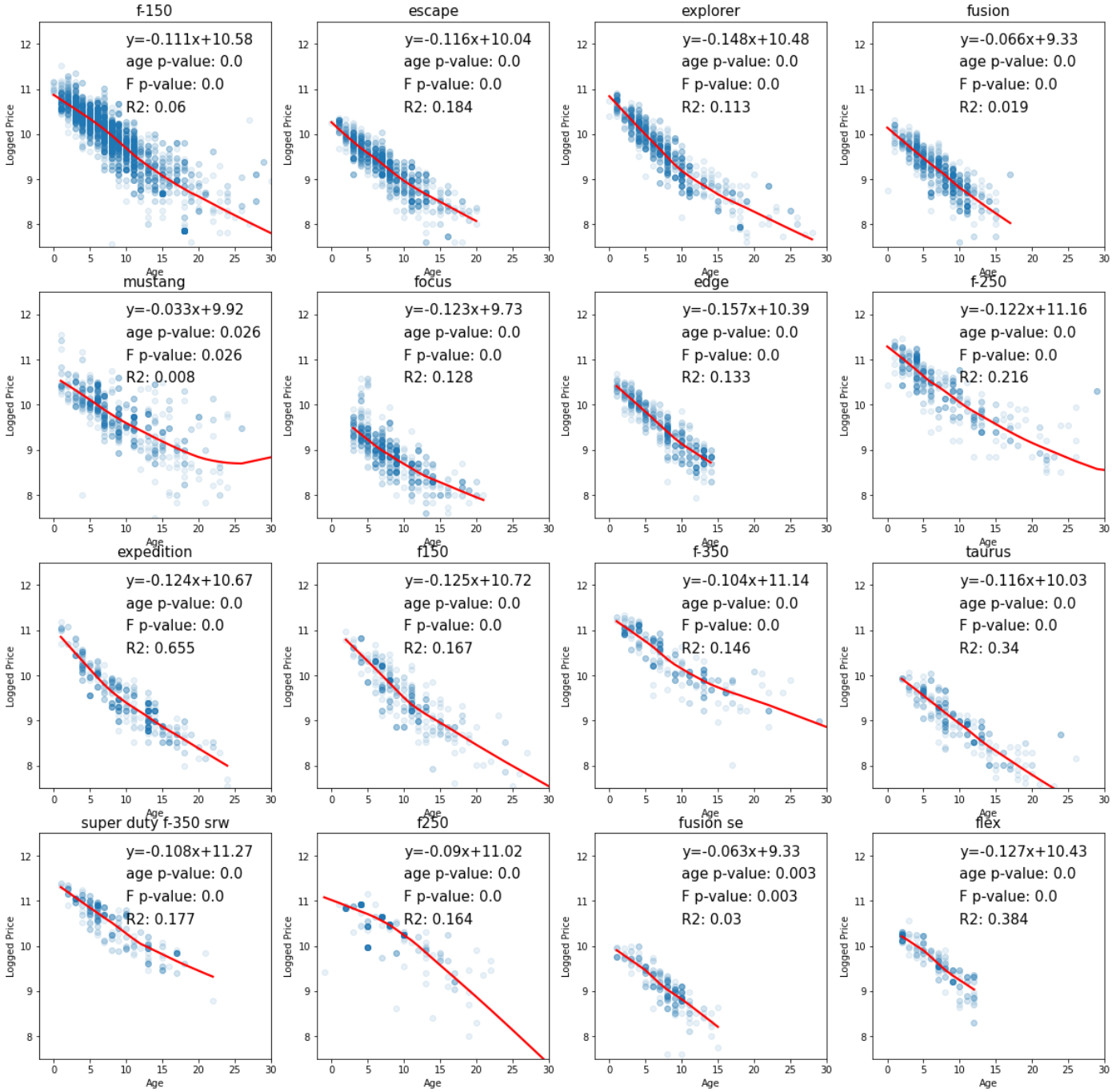
Here is the code I used to visualize the relationship between price and age of cars manufactured by Ford company. I grouped all car models into their manufacturers, because it makes sense to compare different models of vehicles under the same manufacturers, since they are designed for different levels of customers. I used lowess to draw fit lines. And I also performed regression analysis for every model. The code for visualizing car price and car age is similar. I also performed the analysis above with and without logarithm transformation.

Price and Age of Cars Manufactured by Ford of Different Models



Here we can see that the relationship between car age and car price of different models manufactured by Ford are not linear. They are negatively related. The relationship looks like an exponential function. Also, we can see that the slope and the starting price of different models are different. Fusion, escape, and focus have small starting price and the slope is also small.

Logged Price and Age of Cars Manufactured by Ford of Different Models



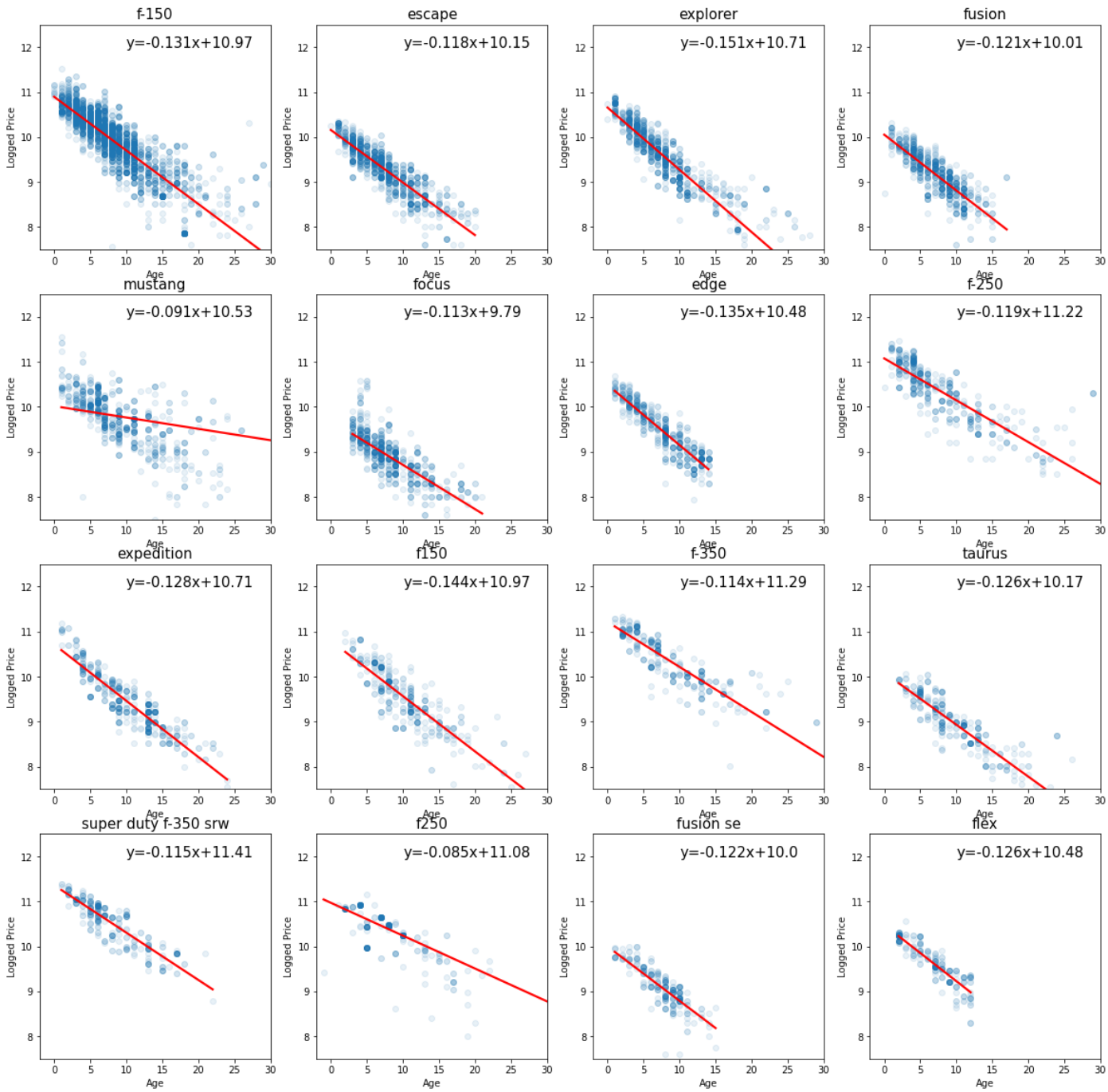
We can see that after log transformation on price, the relationship between ln(prince) and age is very linear for every model of cars. The intercept of every model is different, around 11, but notice that the price is logged, meaning that these models have very different starting prices. However, the slopes are quite similar, all around -0.1, which means that most car prices depreciate at similar rates with respect to age. Nearly all p values and f values are close to zero, meaning that the model and the variable are significant. However the R2 value is small, which means that the model can not explain a large portion of the variance in the price. Also we can see that for some models, their age only span a short time, this could due to they are only manufactured in a short period of time.

By using OLS regression, I found some of the results are weird. Some results have depreciation rates that are positive, which is contradictory to mainstream economic theory. I surmise maybe the outlier is causing the results to be unstable. I try to use robust regression with Huber weighting method, and the results become

more stable and reasonable. Here is the code I used to generate the robust regression plot:

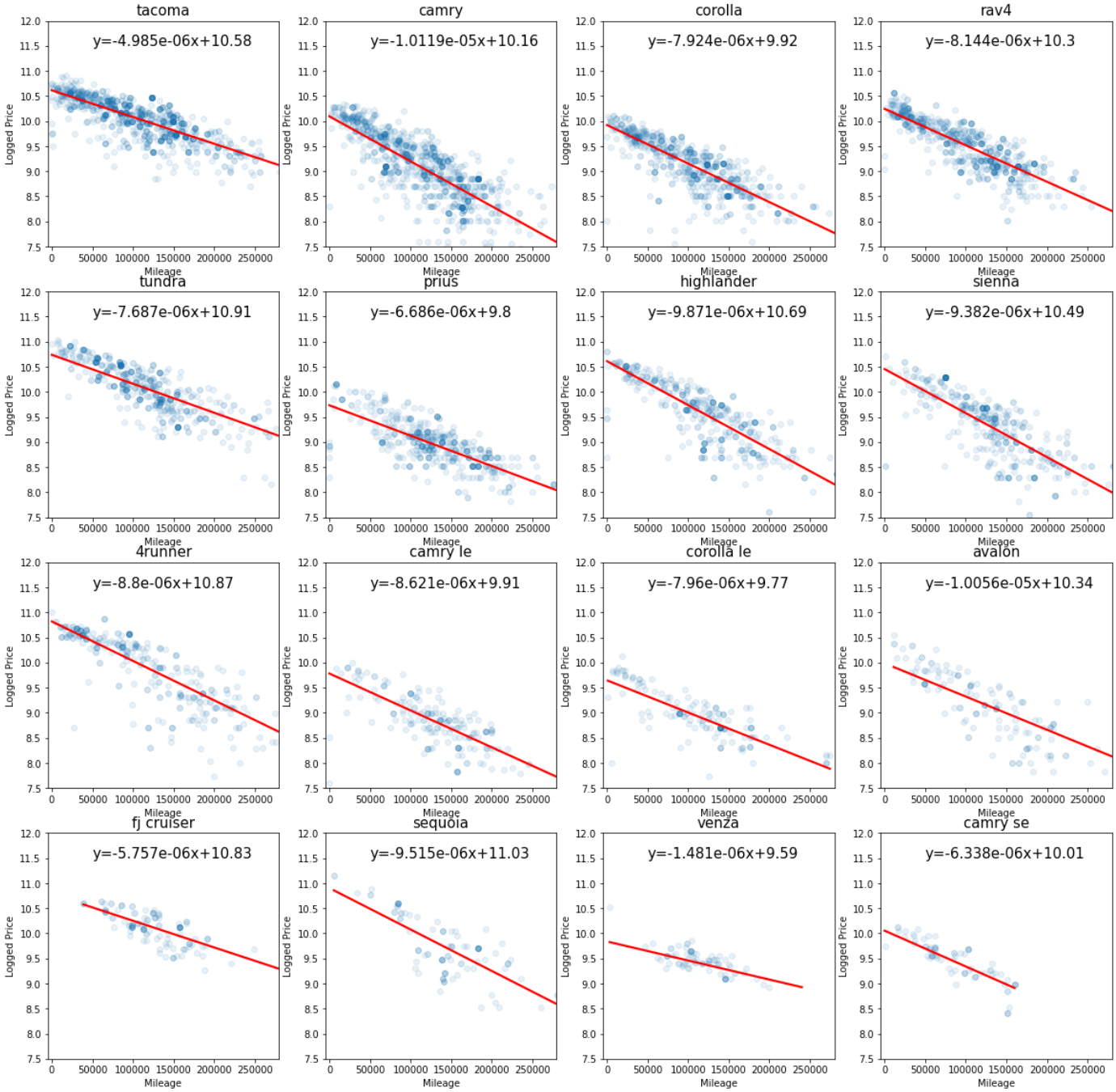
```
n = 4
fig, axs = plt.subplots(n, n, figsize=(20, 20))
fig.suptitle("Logged Price and Age of Cars Manufactured by "+model.capitalize()+\
             " of Different Models\n Robust Regression Huber Method", fontsize=30)
print("Price and Age of Cars Manufactured by "+model.capitalize()+" of Different Models")
df = vehicle_1[vehicle_1["manufacturer"]==model]
df["year"] = df["year"].astype(int)
df["age"] = 2021 - df["year"]
df["price"] = df["price"].astype(float)
car_list = [x[0] for x in Counter(list(df[df["model"].notna()]\
                                       ["model"])).most_common()][:n**2]
for i in range(len(car_list)):
    df_1 = vehicle_1[vehicle_1["model"]==car_list[i]]
    df_1["log_price"] = np.log(df_1["price"])
    df_2 = df_1[(df_1["age"]>=1) & (df_1["age"]<=np.quantile(list(df_1["age"]), .90))]
    data = sm.datasets.stackloss.load()
    data.exog = sm.add_constant(data.exog)
    try:
        model1 = sm.RLM(df_2["log_price"], sm.add_constant(df_2["age"]), M=sm.robust.norms.HuberT(t=2)).fit()
    except:
        pass
    g1 = sns.regplot(x="age",y=np.log(df_2["price"]),data=df_2,ax=axs[i//n,i%n],scatter_kws={'alpha':0.1},lowess=False,\
                     robust=True,ci=None,\
                    line_kws={"color": "red"})
    g1.set_title(car_list[i], fontsize=15)
    g1.set_xlim([-2,30])
    g1.set_ylim([7.5,12.5])
    g1.set_xlabel("Age", fontsize=10)
    g1.set_ylabel("Logged Price", fontsize=10)
    try:
        g1.text(10,12,"y="+str(round(model1.params["age"],3))+"x+"+str(round(model1.params["const"],2)), fontsize=15)
    except:
        pass
```

I used sm.RLM robust linear model and regplot with robust=True here. I also manually eliminate the potential outliers by filtering put the data with age in the top 5%. (The distribution of age is heavily skewed with a long tail to the right.)



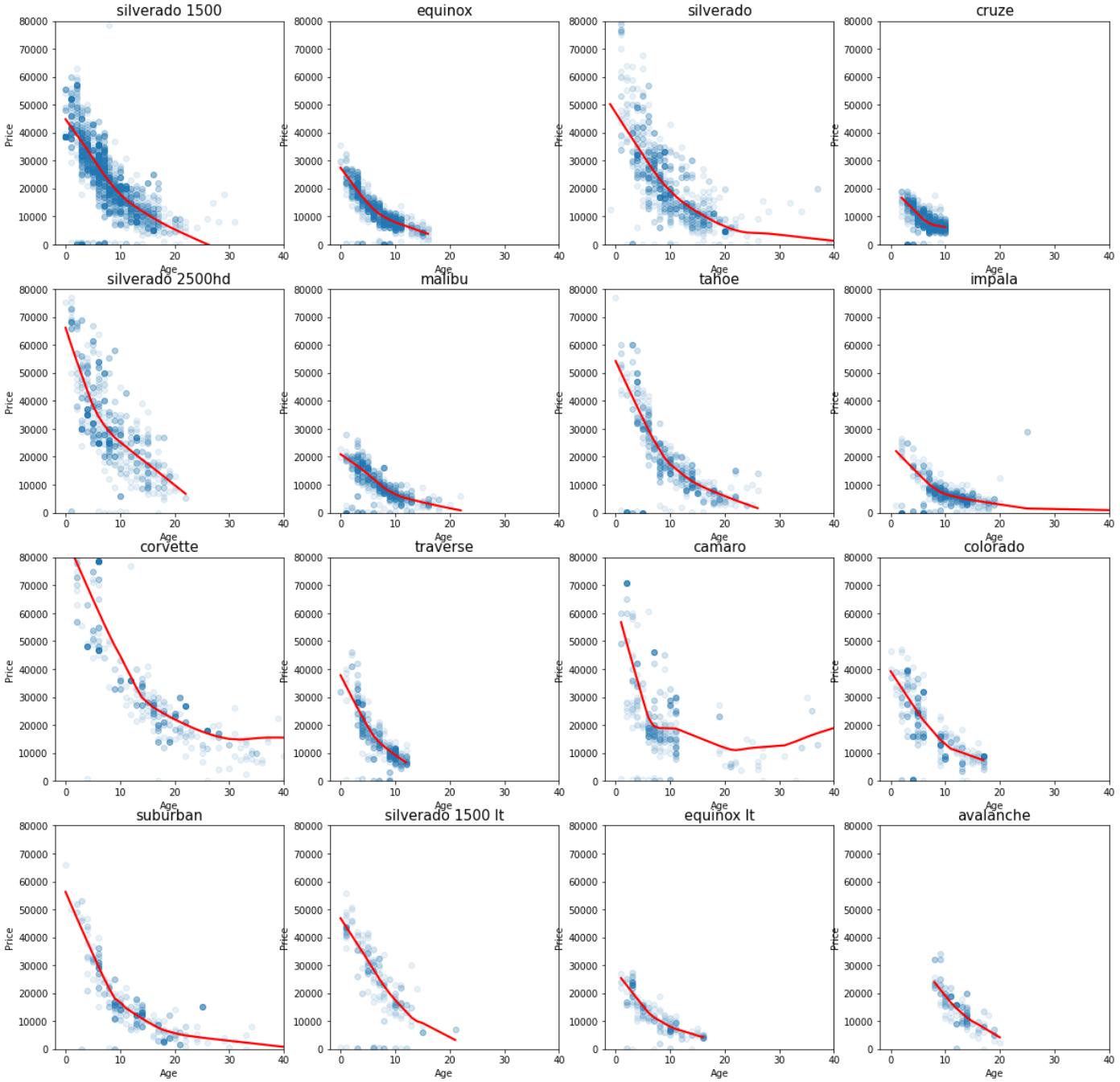
Here we can see that the slope is around -0.1, looks very stable.

Logged Price and Mileage of Cars Manufactured by Toyota of Different Models

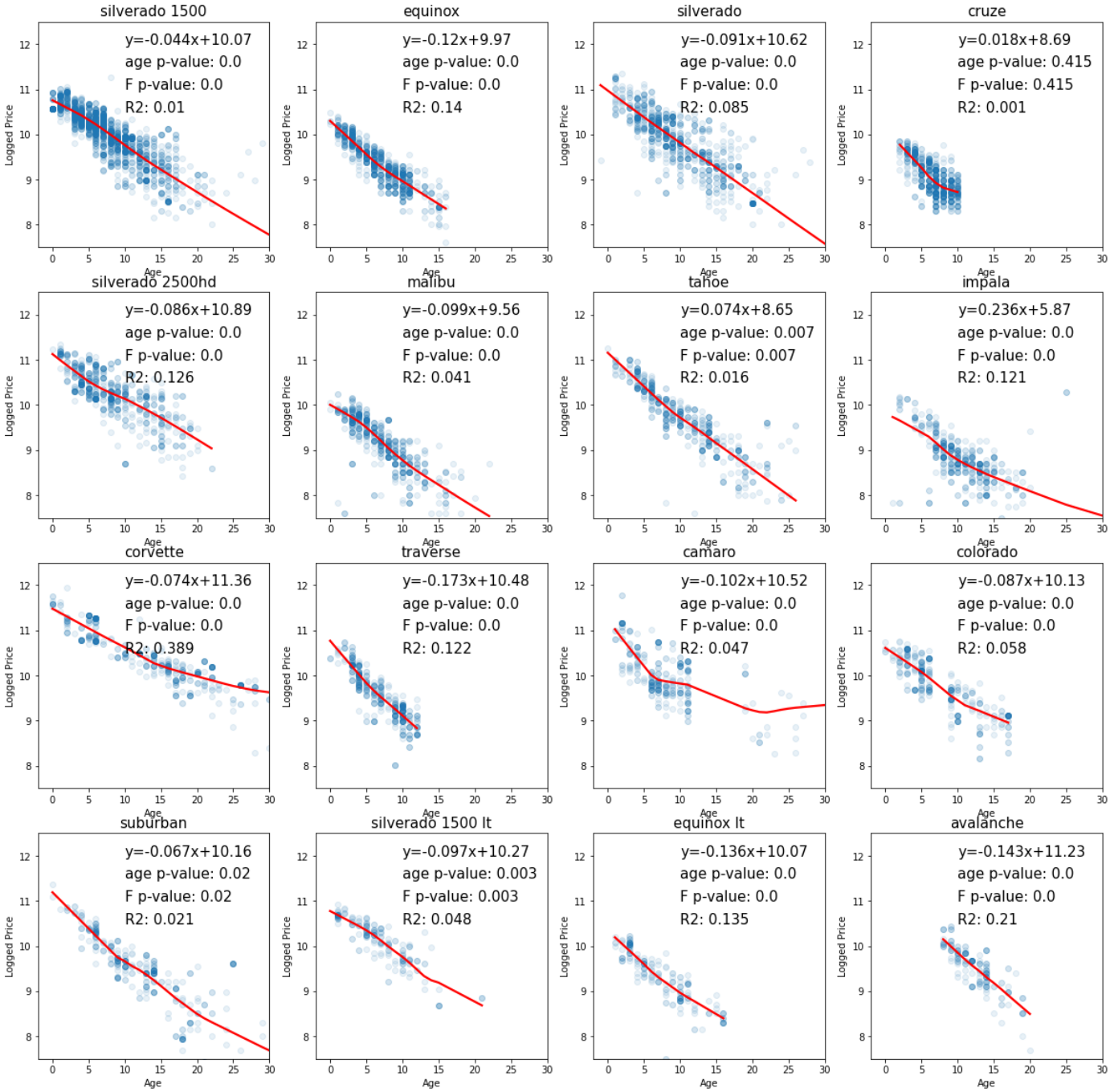


Here are some more manufacturers I have analyzed, they all have similar pattern. Some cars have ages spanning a short period of time, probably due to these models were only manufactured in those years.

Price and Age of Cars Manufactured by Chevrolet of Different Models

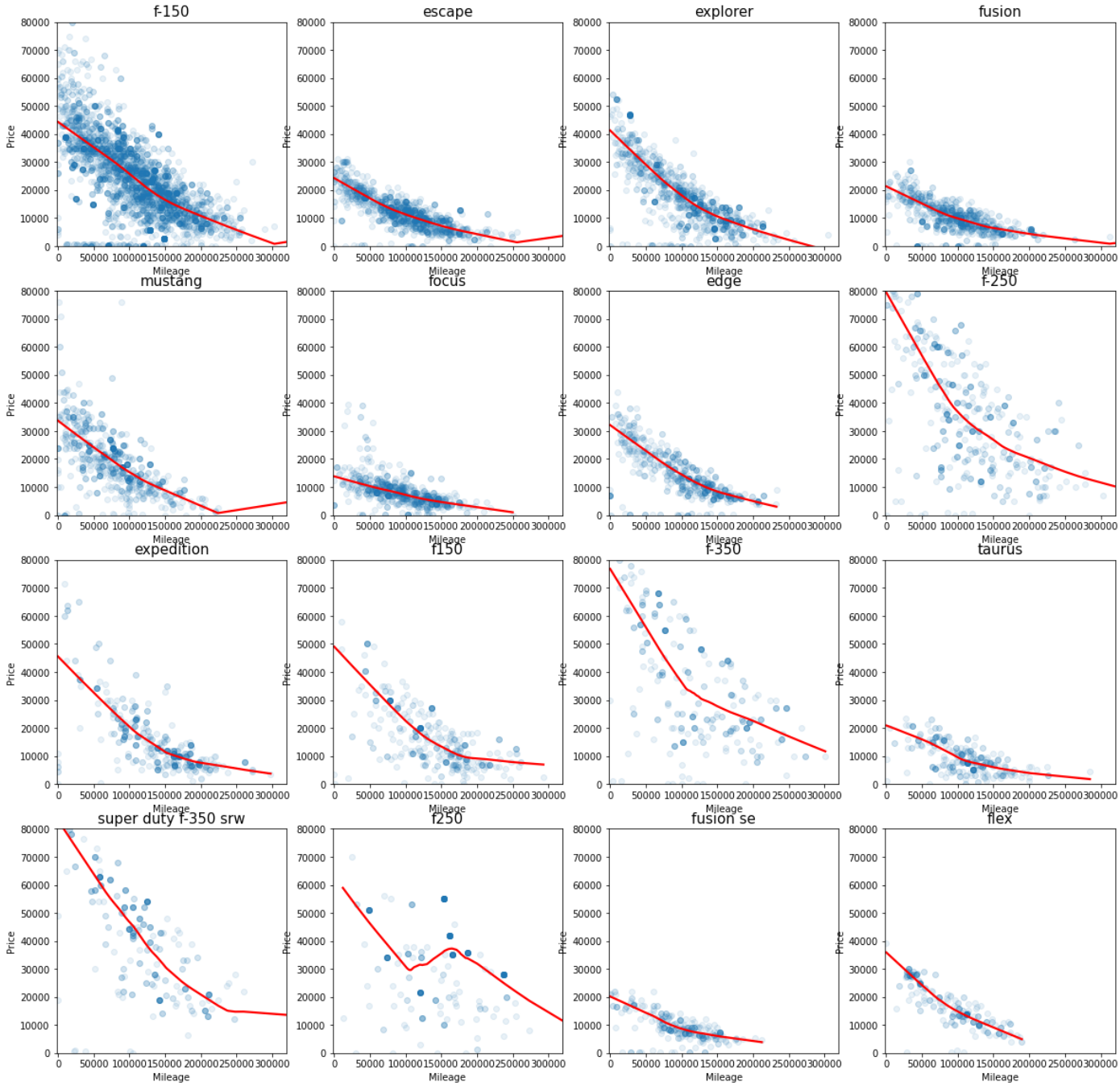


Logged Price and Age of Cars Manufactured by Chevrolet of Different Models



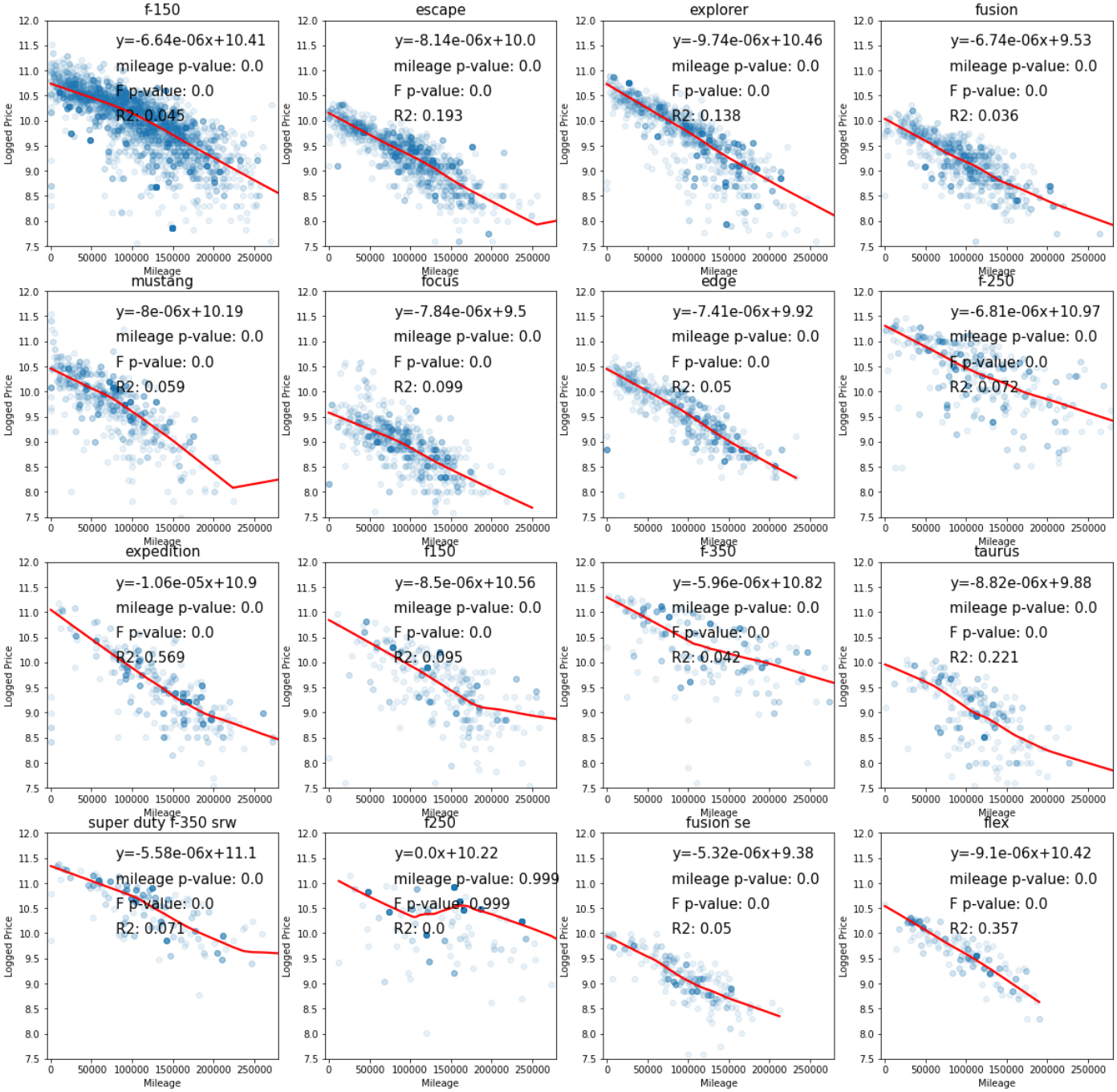
For car price and mileage, I also did similar analysis. Here are the results:

Price and Mileage of Cars Manufactured by Ford of Different Models



Here we can see that for models under Ford, the relationship between car mileage and car price is also not linear, though it looks more linear than the relationship between age and price. Different models have very different starting prices. F-150 seems to have a very linear relationship with slight bending, while for other model the nonlinear relationship is obvious.

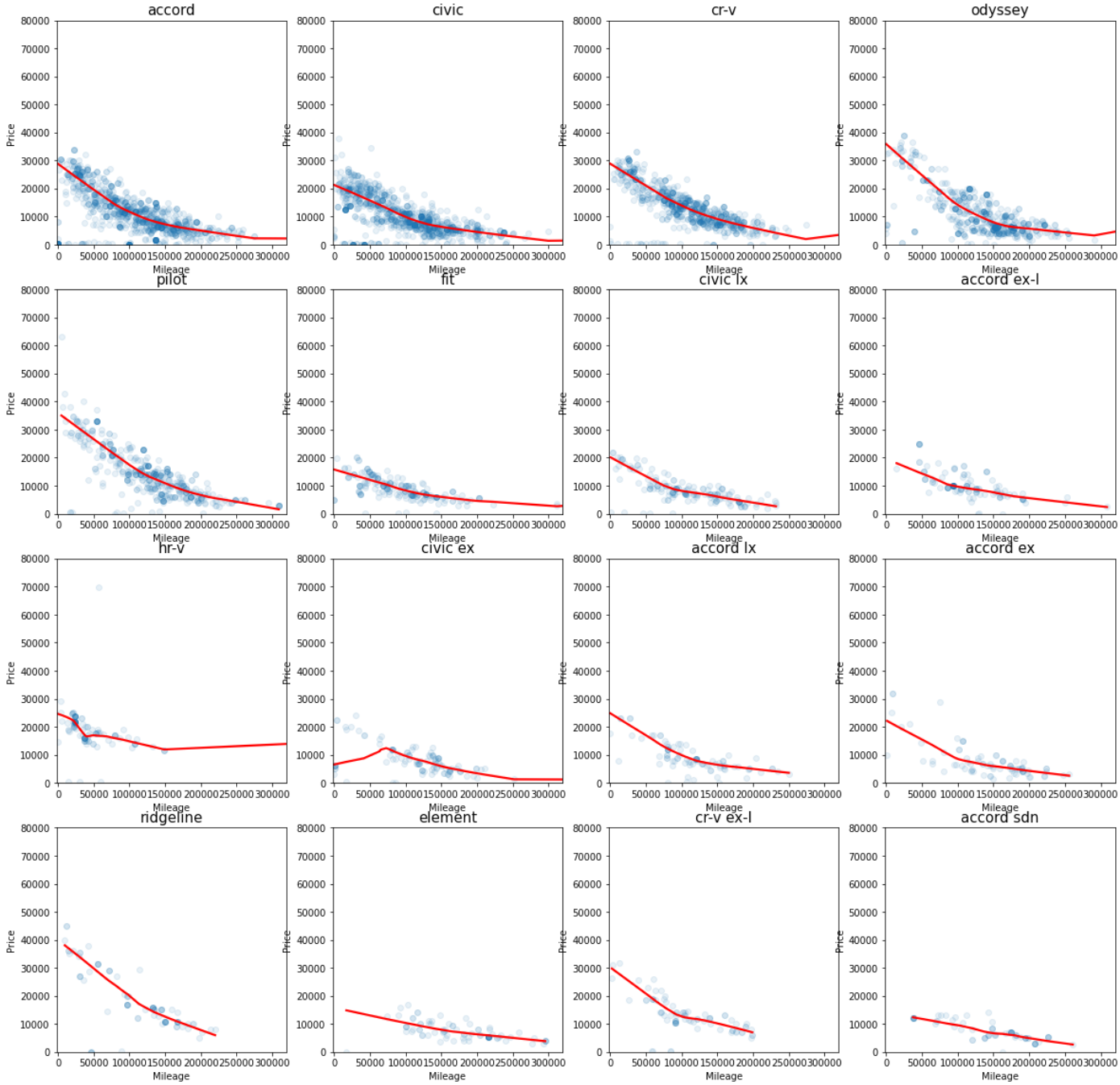
Logged Price and Mileage of Cars Manufactured by Ford of Different Models



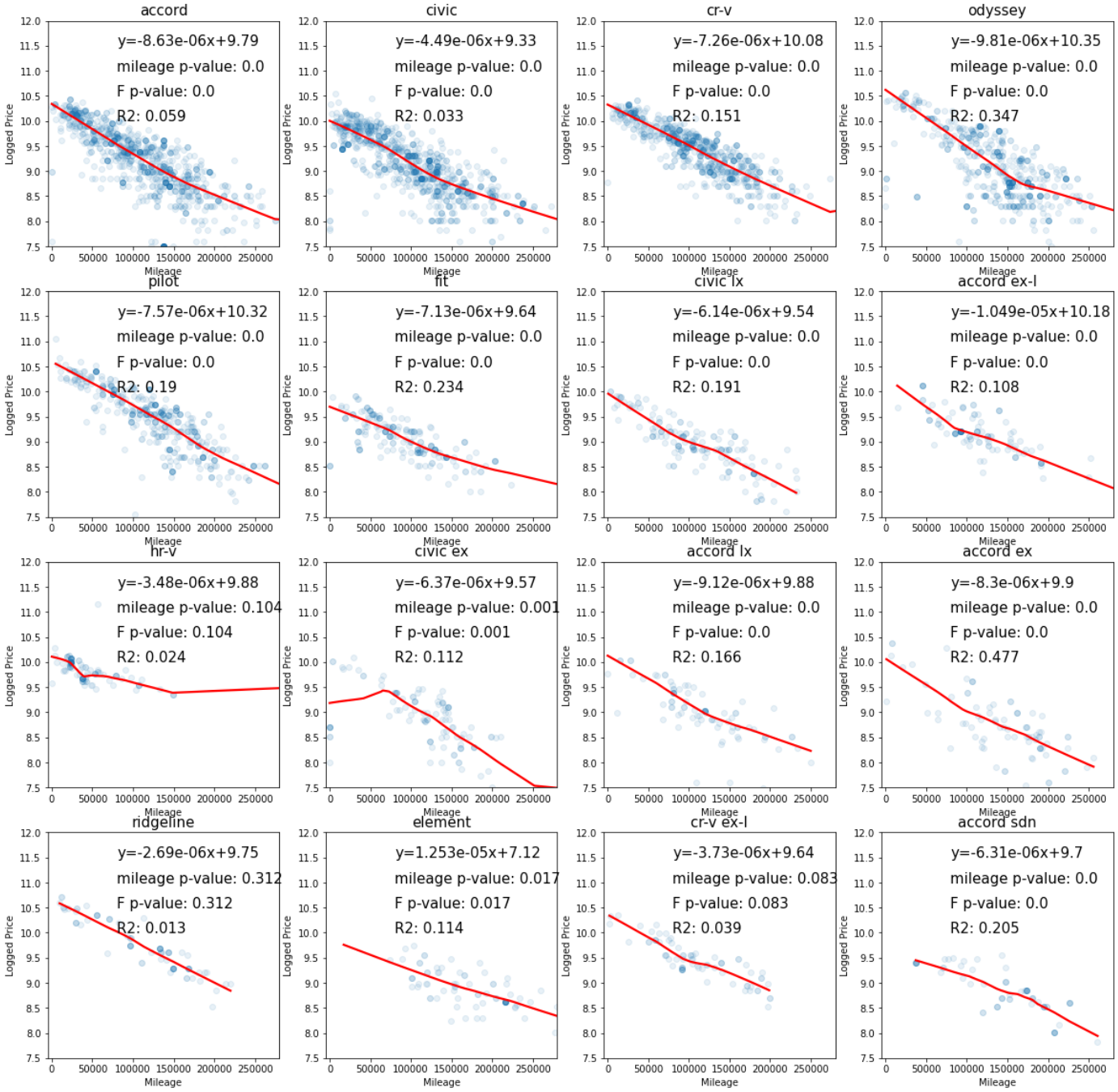
After the logarithm transformation, the curve seems more linear. The increased linearity indicates log transformation is necessary. We can see all curves have similar slope, around -6e-06. The intercepts are around 11, but notice that these are results after log transformation, so the starting price of each model would be quite different. All models depreciate at similar rates with respect to mileage. Most curves here have F p-value and mileage p value close to 0, which means the model and the variable are significant. But the R2 is small, indicating the prediction ability id not good. We can say that both age and mileage are not linearly related to car price. Exponential relationship is more likely.

I have done more analysis for other manufacturers; the patterns are similar. Here are some of the results:

Price and Mileage of Cars Manufactured by Honda of Different Models



Logged Price and Mileage of Cars Manufactured by Honda of Different Models

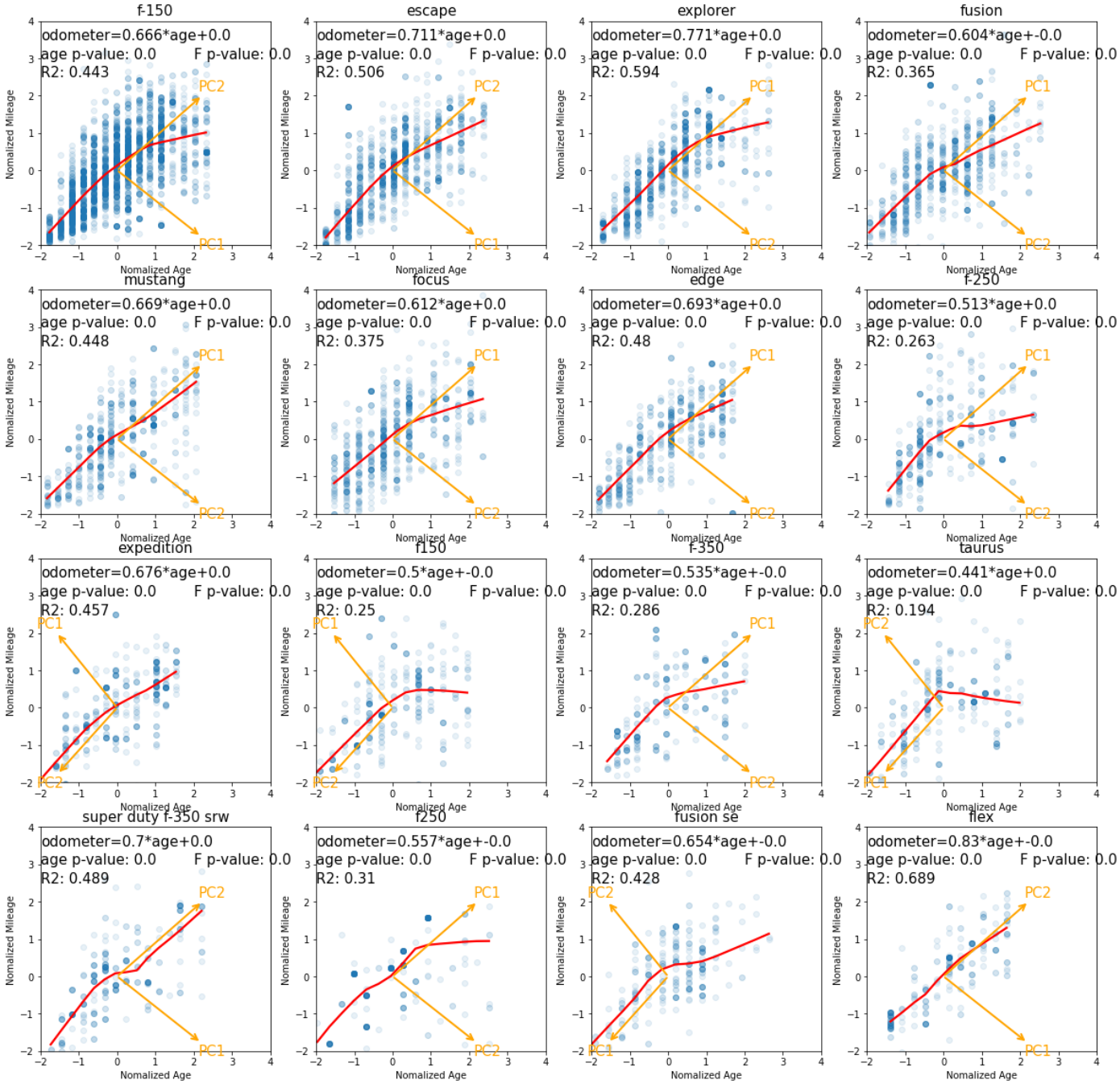


For the relationship between car age and car mileage. We can see that they are positively related. In the first 10 years, the fit curve is almost linear, but after 10 years, the fitting curve bend towards one direction. The curves without logarithm transformation show a big deal of heteroscedasticity. It shows that as the age of cars increases, the uncertainty of their mileage become bigger. The range of mileage becomes bigger. From the p values and f values we can see that the relationship is very significant. Both p values are close to 0. And the R2 is pretty large, indicating that using age to predict mileage is a good method. If we are using both age and mileage in the regression, there could be multicollinearity caused by them. Under this circumstance, PCA is a good method to reduce dimension and avoid multicollinearity. Therefore, I first normalized the value of age and mileage and plotted the direction
of the first PC and the second PC. They are orthogonal.
Here is the code I used to draw the following plots:

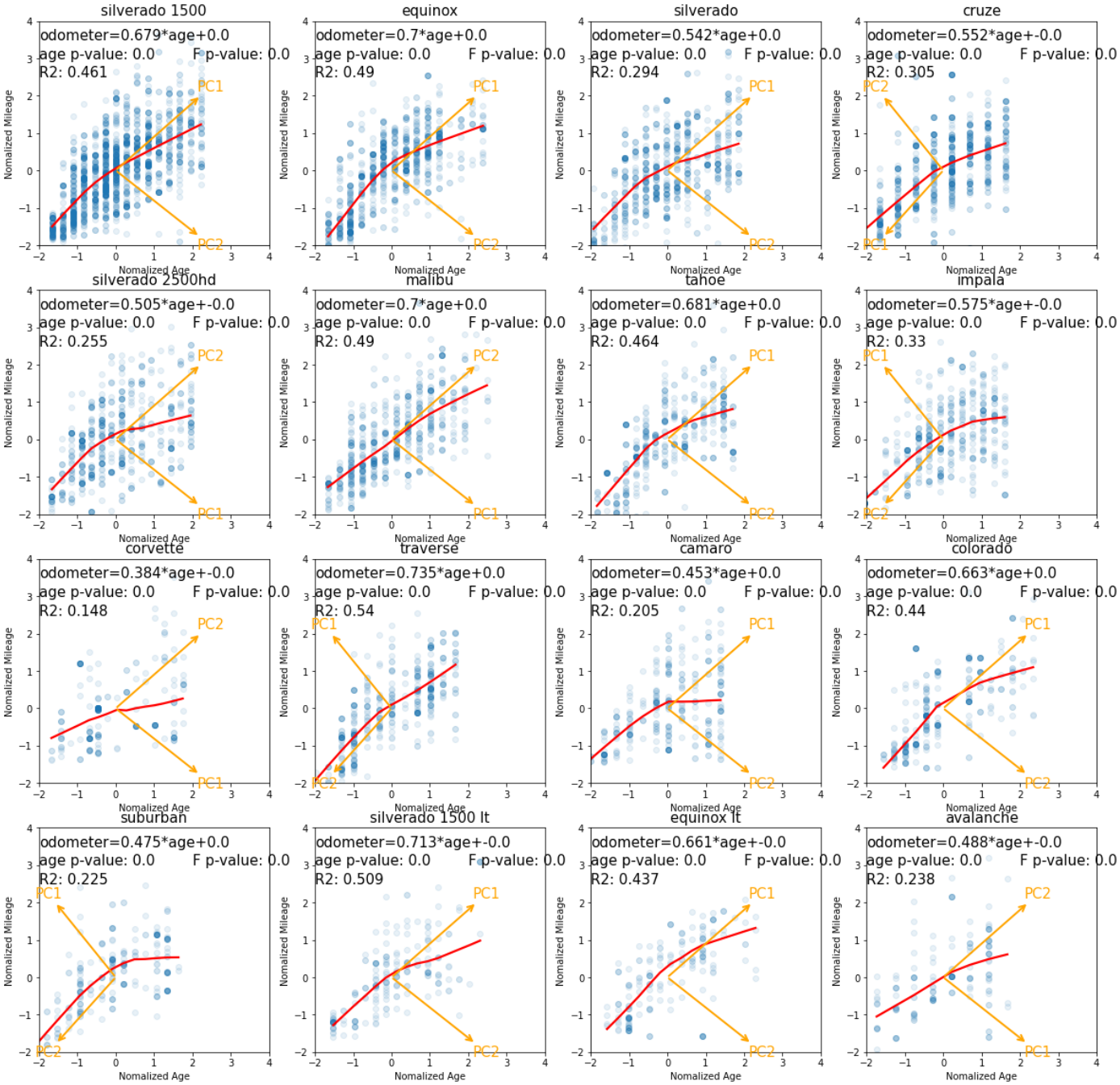
```
import statsmodels.api as sm
import statsmodels.formula.api as smf
from scipy import stats
import warnings
warnings.filterwarnings('ignore')
import sklearn.decomposition as skd
import sklearn.preprocessing as skp

model="nissan"
n = 4
fig, axs = plt.subplots(n, n, figsize=(20, 20))
fig.suptitle("Logged Price, age and Mileage of Cars Manufactured by "+\
             model.capitalize()+" of Different Models\n Using PCA regression", fontsize=30)
df = vehicle_1[vehicle_1["manufacturer"]==model]
df["year"] = df["year"].astype(int)
df["age"] = 2021 - df["year"]
df["price"] = df["price"].astype(float)
car_list = [x[0] for x in Counter(list(df[df["model"].notna()]\
                                       ["model"])).most_common()][:n**2]
for i in range(len(car_list)):
    df_1 = vehicle_1[vehicle_1["model"]==car_list[i]]
    df_1["log_price"] = np.log(df_1["price"])
    #pca_model = skd.PCA().fit(df_hept_norm.iloc[:,:7])
    df_2 = df_1[(df_1["odometer"]>=10) & (df_1["odometer"]<=300000)\
                & (df_1["age"]>=1) & (df_1["age"]<=20)]
    df_2[["price","age","odometer"]] = df_2[["price","age","odometer"]].astype(float)
    df_2[["price","age","odometer"]] = skp.scale(df_2[["price","age","odometer"]])
    df_2 = df_2[["model","price","log_price","age","odometer"]]
    pca_model = skd.PCA().fit(df_2[["age","odometer"]])
    df_2[["PC1","PC2"]] = pca_model.transform(df_2[["age","odometer"]])
    model1 = smf.ols('log_price ~ PC1', data=df_2).fit()
    g1 = sns.regplot(x=[0],y=[0],data=df_1,ax=axs[i//n,i%n],scatter_kws={'alpha':0.1},lowess=True,\
                     line_kws={"color": "red"})
    g1.set_title(car_list[i], fontsize=15)
    g1.set_xlim([0,10])
    g1.set_ylim([0,10])
    g1.set_xlabel("", fontsize=10)
    g1.set_ylabel("", fontsize=10)
    g1.text(1,9,model.capitalize()+"   "+car_list[i].capitalize(), fontsize=15)
    g1.text(1,8,"ln(price)="\
            +str(round(model1.params["PC1"],3))+"*PC1+"\
            +str(round(model1.params["Intercept"],2)), fontsize=15)
    g1.text(1,7,"PC1: ("+str(round(pca_model.components_[0][0],3))\
            +", "+str(round(pca_model.components_[0][1],3))+")", fontsize=15)
    g1.text(1,6,"PC1 p-value: "+str(round(model1.pvalues["PC1"],3)), fontsize=15)
    g1.text(1,5,"F p-value: "+str(round(model1.f_pvalue,3)), fontsize=15)
    g1.text(1,4,"R2: "+str(round(model1.rsquared,3)), fontsize=15)
    g1.text(1,2,"PC1,PC2 variance: \n"+str(round(pca_model.explained_variance_[0],3))\
```

I select the cars that belong to the same manufacturer and grouped them into different models. I used lowess fit line in the regression plot. I also rescaled and normalized the value for age and mileage. Th directions of the two PCs are shown in the plot. We can see the two PCs are orthogonal to each other.

Age and Mileage of Cars Manufactured by Ford of Different Models

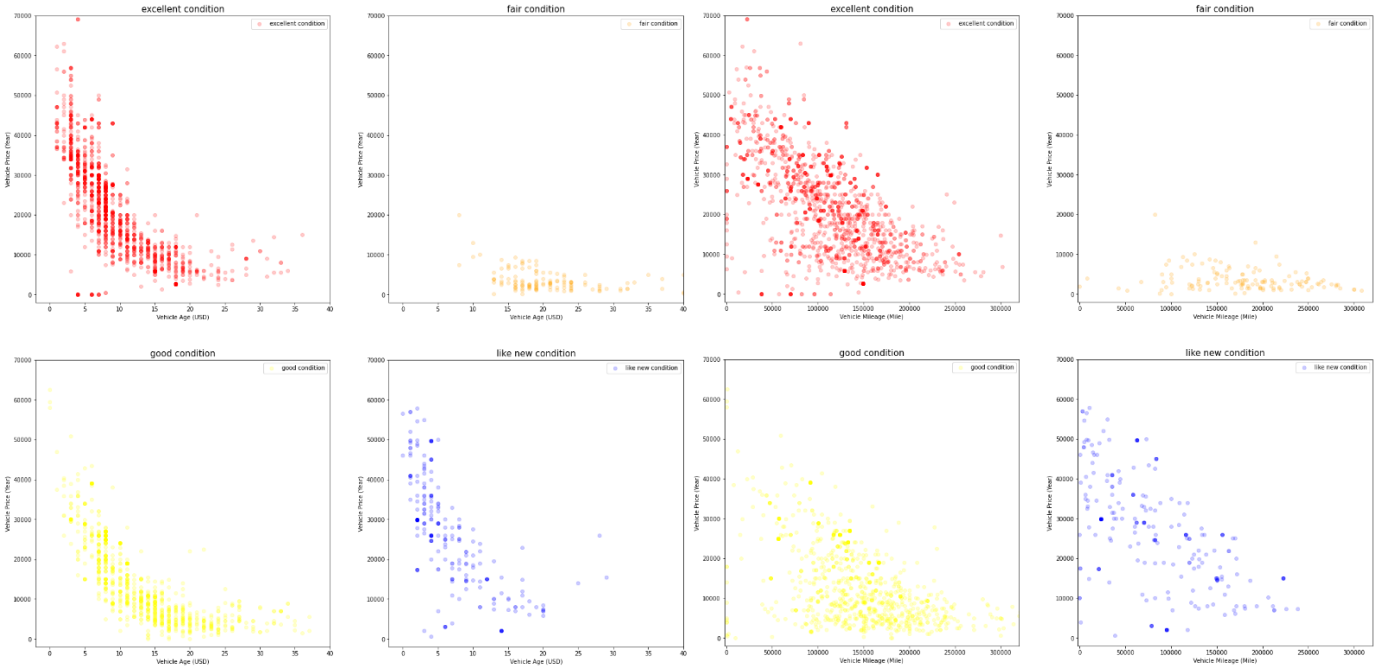


Age and Mileage of Cars Manufactured by Chevrolet of Different Models



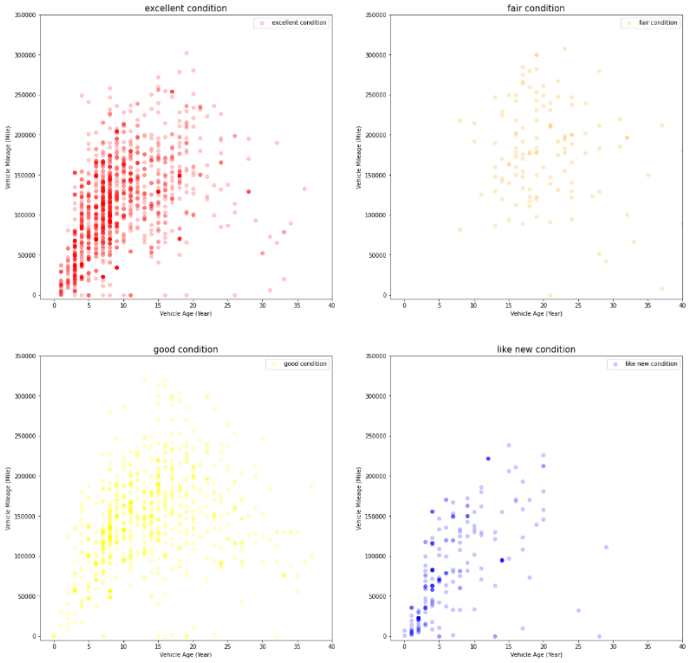
Here I select f-150 as an example to show the relationship between conditions and types of vehicles and prices, ages, mileage. The method and code are similar to previous examples, the extra step is to separate cars of different conditions and types.

Price and Age of Secondhand Ford f-150 Categorized by Condition



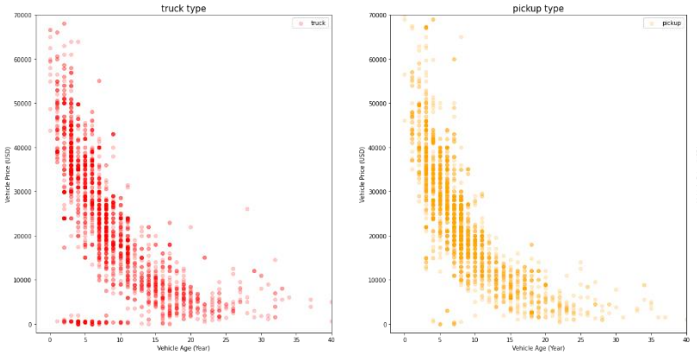
Price and Mileage of Secondhand Ford f-150 Categorized by Condition

Age and Mileage of Secondhand Ford f-150 Categorized by Condition

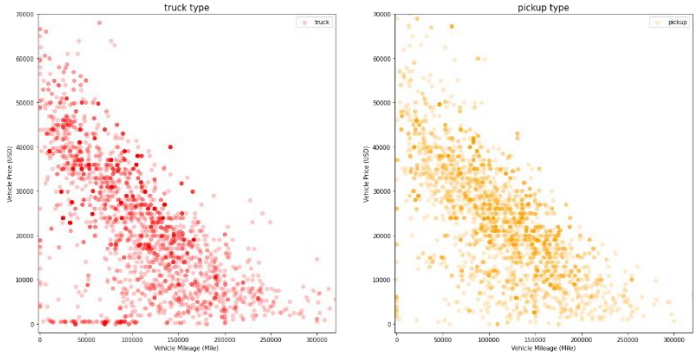


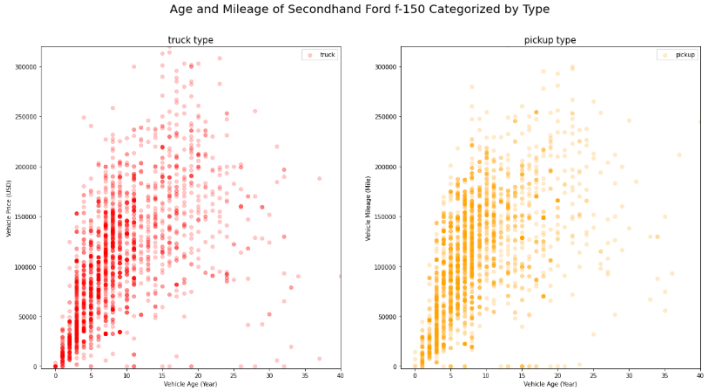
Here we can see that for Ford f-150 cars, cars in good condition generally have lower prices and longer age and mileage. And cars in fair condition have extremely low price and long age and mileage.

Price and Age of Secondhand Ford f-150 Categorized by Type



Price and Mileage of Secondhand Ford f-150 Categorized by Type





However, for pickup and truck of ford f-150, they do not have significant difference in age, mileage and price.

(5) Is there a correlation between the various characteristics of used cars? How to predict the price of used cars better by using principal component analysis regression?

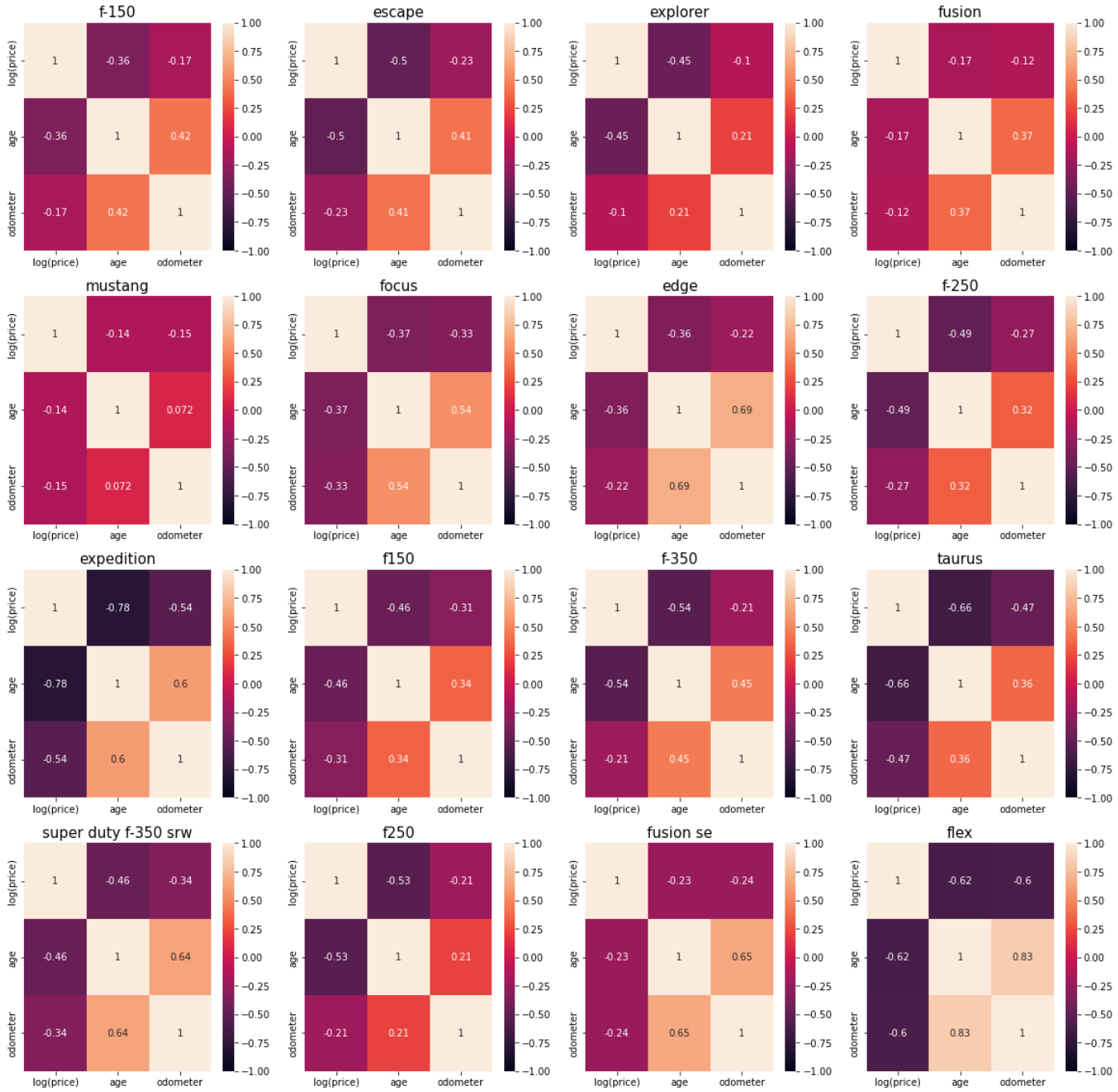
Analysis:

1. Build a correlation heatmap of different features of secondhand car

```
import matplotlib.pyplot as plt
from collections import Counter
import numpy as np
fig, axs = plt.subplots(4, 4, figsize=(20, 20))
model = "honda"
colors = {'excellent':'red', 'fair':'orange', \
          'good':'yellow', 'like new':'blue', \
          'new':'green', 'salvage':'purple'}
fig.suptitle("Correlation Matrix of Cars Manufactured by "+model.capitalize(),\
             fontsize=20)
df = vehicle_1[vehicle_1["manufacturer"]==model]
df["year"] = df["year"].astype(int)
df["age"] = 2021 - df["year"]
df["price"] = df["price"].astype(float)
df["log(age)"] = np.log(df["age"])
df["log(mileage)"] = np.log(df["odometer"])
car_list = [x[0] for x in Counter(list(df[df["model"].notna()]\
                                       ["model"])).most_common()][0:16]
for i in range(len(car_list)):
    matrix_1 = df[df["model"]==car_list[i]]\
    [["price", "log(age)", "log(mileage)"]].corr()
    g1 = sns.heatmap(matrix_1,annot=True,ax=axs[i//4,i%4], vmin=-1, vmax=1)
    g1.set_title(car_list[i], fontsize=15)
```

Above is the code I used to draw the correlation matrix of cars of every model. I grouped every model into their manufacturers. Because same manufacturer will make cars for different audiences, both premium and low-end, it makes sense to put cars made by the same manufacturer together for comparison. For each manufacturer, I selected the top 16 car models in terms of transaction volume. And I conducted the study with and without taking logarithms for mileage and vehicle age, respectively.

Correlation Matrix of Cars Manufactured by Ford



Here we can see that there is a clear negative correlation between the age of the car and the price and the mileage and the price. There is a stronger positive correlation between mileage and age of the car. Different models of the same manufacturer have different laws. For the Ford Mustang and f-350, the negative correlations between price and age and price and mileage are significantly weaker than for the other models. In addition, when logarithms were taken for age and mileage, overall, both negative and positive correlations were lowered. I think this could cause by decreased scale of the value of price through doing log transformation. Theoretically, most economic theory indicates that the price should depreciate exponentially with respect to time, that is $price = Ae^{-\alpha t}$. When assuming this, we should do log transformation to $price$, which means $ln(price) = -\alpha t + lnA$.

We can get similar results by analyzing other manufacturers. I have analyzed a total of 5 manufacturers' cars and got similar results as the three plots above.

When analyzed at the level of the manufacturer rather than the car model, the result is that almost all correlations become weaker. In particular, many manufacturers (Ford, Chevrolet, Toyota, Jeep, Dodge) have correlation coefficients close to 0 for price and mileage and for price and age of the car. This may be due to the fact that these manufacturers have so many different models of cars with different grades, performance and audiences, thus leading to uncertainty when they are all mixed together. This suggests that if predictions are modeled for car prices, it is more appropriate to use the car model rather than the manufacturer as the independent variable.

5. Modeling forecasts of secondhand prices using multiple regression analysis (linear or non-linear). Since we already know that mileage and age are exponentially related to used prices, a non-linear transformation may be required. And there may be a linear relationship between mileage and vehicle age, which may require variable selection and some methods to eliminate linearity, such as principal component analysis and then linear combination of features through eigenvectors of the covariance matrix, before analysis.

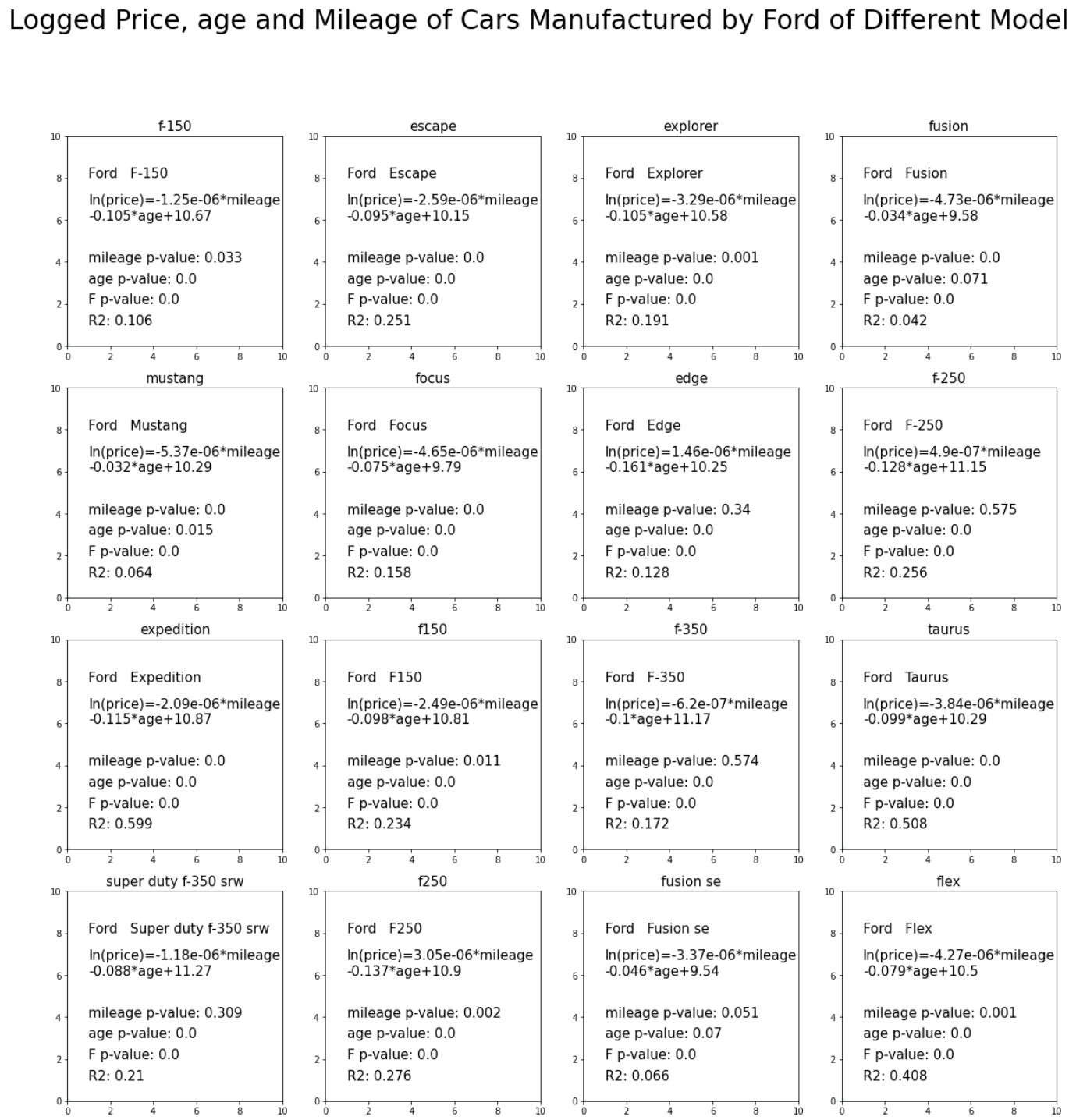
From steps above, we can see there is correlation between mileage and age of cars. Car prices are exponentially correlated with car age and car mileage. Different models of cars have different starting prices. If performing a linear regression, it is necessary to use log transformation and principal component analysis. There could be many potential linear models using this transformation. I will try to perform this following separately on every model of cars, because different model may have different depreciation rate. I think use the same depreciation rate for every model and use model as a categorical variable is not appropriate here.

$$\ln(price) = \beta_0 + \beta_1 PC_1 = \beta_0 + \beta_1(e_1 mileage + e_2 age) \Rightarrow price = Ae^{\alpha mileage} e^{\beta age}$$

Here is some of the code I used to do PCA regression(I have done two regressions and the methods are similar):

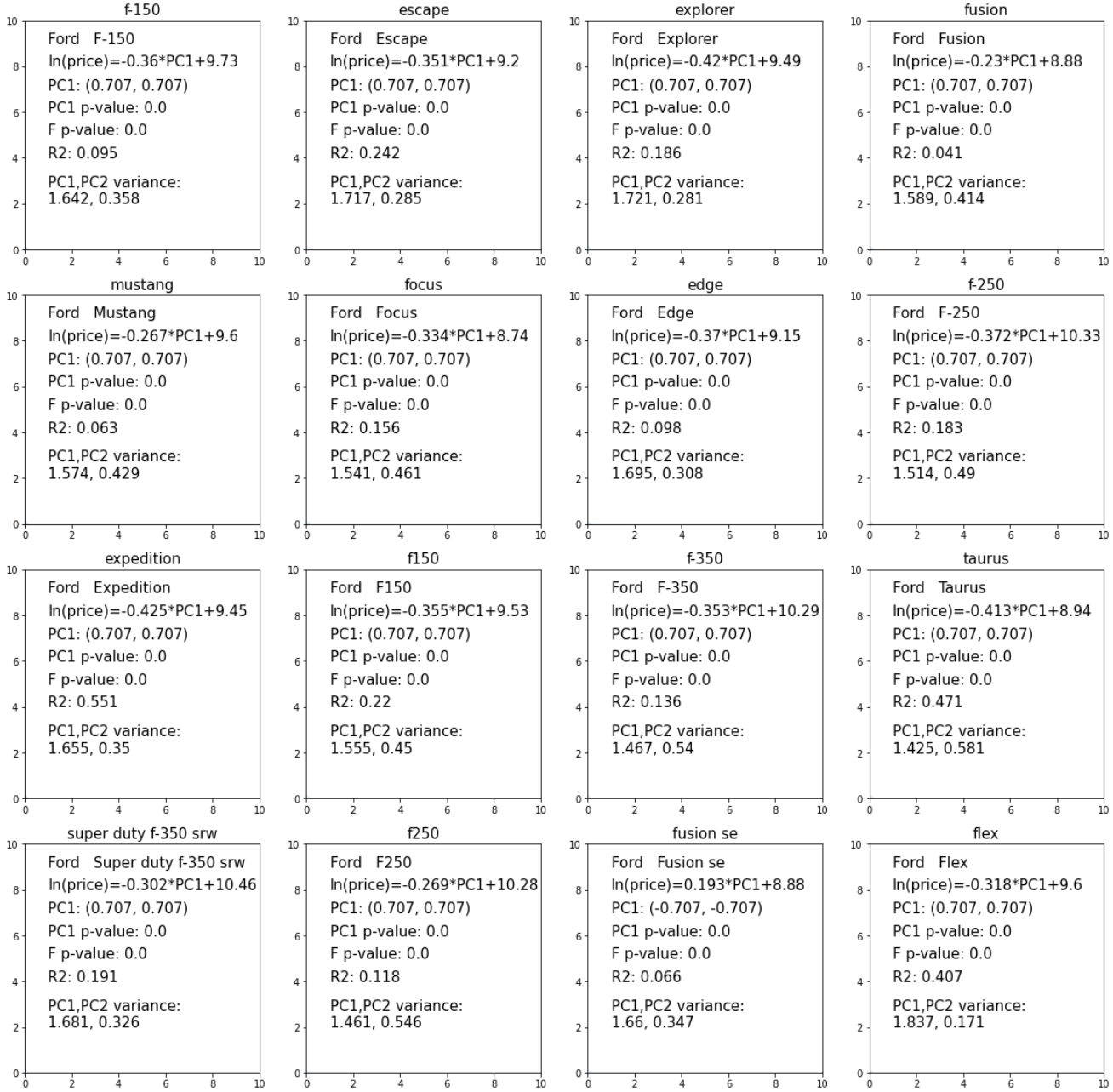
```
import matplotlib.pyplot as plt
from collections import Counter
import numpy as np
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
from scipy import stats
import warnings
warnings.filterwarnings('ignore')
import sklearn.decomposition as skd
import sklearn.preprocessing as skp

model="nissan"
n = 4
fig, axs = plt.subplots(n, n, figsize=(20, 20))
fig.suptitle("Logged Price, age and Mileage of Cars Manufactured by "+\
             model.capitalize()+" of Different Models\n Using PCA regression", fontsize=30)
df = vehicle_1[vehicle_1["manufacturer"]==model]
df["year"] = df["year"].astype(int)
df["age"] = 2021 - df["year"]
df["price"] = df["price"].astype(float)
car_list = [x[0] for x in Counter(list(df[df["model"].notna()]\
                                       ["model"])).most_common()][:n**2]
for i in range(len(car_list)):
    df_1 = vehicle_1[vehicle_1["model"]==car_list[i]]
    df_1["log_price"] = np.log(df_1["price"])
    #pca_model = skd.PCA().fit(df_hept_norm.iloc[:,:7])
    df_2 = df_1[(df_1["odometer"]>=10) & (df_1["odometer"]<=300000)\
                & (df_1["age"]>=1) & (df_1["age"]<=20)]
    df_2[["price","age","odometer"]] = df_2[["price","age","odometer"]].astype(float)
    df_2[["price","age","odometer"]] = skp.scale(df_2[["price","age","odometer"]])
    df_2 = df_2[["model","price","log_price","age","odometer"]]
    pca_model = skd.PCA().fit(df_2[["age","odometer"]])
    df_2[["PC1","PC2"]] = pca_model.transform(df_2[["age","odometer"]])
    model1 = smf.ols('log_price ~ PC1', data=df_2).fit()
    g1 = sns.regplot(x=[0],y=[0],data=df_1,ax=axs[i//n,i%n],scatter_kws={'alpha':0.1},lowess=True,\
                     line_kws={"color": "red"})
    g1.set_title(car_list[i], fontsize=15)
    g1.set_xlim([0,10])
    g1.set_ylim([0,10])
    g1.set_xlabel("", fontsize=10)
    g1.set_ylabel("", fontsize=10)
    g1.text(1,9,model.capitalize()+"   "+car_list[i].capitalize(), fontsize=15)
    g1.text(1,8,"ln(price)="\
            +str(round(model1.params["PC1"],3))+"*PC1+"\
            +str(round(model1.params["Intercept"],2)), fontsize=15)
    g1.text(1,7,"PC1: ("+str(round(pca_model.components_[0][0],3))\
            +", "+str(round(pca_model.components_[0][1],3))+")", fontsize=15)
    g1.text(1,6,"PC1 p-value: "+str(round(model1.pvalues["PC1"],3)), fontsize=15)
    g1.text(1,5,"F p-value: "+str(round(model1.f_pvalue,3)), fontsize=15)
    g1.text(1,4,"R2: "+str(round(model1.rsquared,3)), fontsize=15)
```

I need to filter out the vehicles that belong to the same manufacturer, classify them into their models. And then rescale the value of age and mileage, because PCA is sensitive to scaling. And then do the regression using the PC that has the biggest variance.



Here we can see that by using two variables, the R2 values are bigger than previous models with one variable. Most models have p values of both variables close to 0, but for some models, the p value of one variable is extremely big. This could be the result of multicollinearity, which cause the model to be unstable. Here we can try PCA to combine the two variables into one and use this PC as the variable in the linear regression.

Logged Price, age and Mileage of Cars Manufactured by Ford of Different Models Using PCA regression



Here we can see that all models have p value close to 0, indicating the PC1 is very significant. Fusion se's coefficient for PC1 is positive, which is explained by the unit vector of PC1 being opposite of all others'. Also the variance of PC1 is significantly bigger than PC2. All PC1 has the same direction, probably due to the fact that we have normalized the data.

Also I try to find if there is a difference of depreciation rate between different types of car.

```
dict_mileage_depre_rate = {}
mileage_const = []
for model in [x[0] for x in Counter(list(vehicle_1[vehicle_1["manufacturer"].notna()]\
                                         ["manufacturer"])).\
              most_common()][:20]:
    n = 4
    print("Price and Age of Cars Manufactured by "+model.capitalize()+" of Different Models")
    df = vehicle_1[vehicle_1["manufacturer"]==model]
    df["year"] = df["year"].astype(int)
    df["age"] = 2021 - df["year"]
    df["price"] = df["price"].astype(float)
    car_list = [x[0] for x in Counter(list(df[df["model"].notna()]\
                                           ["model"])).most_common()][:n**2]
    for i in range(len(car_list)):
        df_1 = vehicle_1[vehicle_1["model"]==car_list[i]]
        df_1["log_price"] = np.log(df_1["price"])
        df_2 = df_1[(df_1["age"]>=2) & (df_1["age"]<=15) & (df_1["age"]<=np.quantile\
                                                            (list(df_1["age"]), .90))]
        df_3 = df_1[(df_1["odometer"]>=8000) & (df_1["odometer"]<=300000)\
                & (df_1["odometer"]<=np.quantile(list(df_1["odometer"]), .85))]
        try:
            model1 = sm.RLM(df_2["log_price"], sm.add_constant(df_2["age"]), \
                            M=sm.robust.norms.HuberT(t=2)).fit()
            model2 = sm.RLM(df_3["log_price"], sm.add_constant(df_3["odometer"]), \
                            M=sm.robust.norms.HuberT(t=2)).fit()
            age_const.append(round(model1.params["const"],4))
            mileage_const.append(round(model2.params["const"],4))
            dict_age_depre_rate[model+" "+car_list[i]] = round(model1.params["age"],4)
            dict_mileage_depre_rate[model+" "+car_list[i]] = round(model2.params["odometer"]\
                                                                   *10**6,3)
            manufacturer_list.append(model)
            model_list.append(car_list[i])
        except:
```

```
df_4 = pd.DataFrame({"model and manufacturer":dict_age_depre_rate.keys(),\
              "age_depre_rate":dict_age_depre_rate.values(),\
              "mileage_depre_rate (*10**6)":dict_mileage_depre_rate.values(),\
              "age_const":age_const,\
              "mileage_const":mileage_const,\
              "model":model_list,\
              "manufacturer":manufacturer_list})
df_4["age_0_price"]=np.exp(df_4["age_const"])
df_4["mileage_0_price"]=np.exp(df_4["mileage_const"])
df_4.to_csv("car_depre_rate.csv")
df_4
```

	model and manufacturer	age_depre_rate	mileage_depre_rate (*10**6)	age_const	mileage_const	model	manufacturer	age_0_price	mileag
0	ford f-150	-0.1308	-7.364	10.9655	10.7615	f-150	ford	57843.710160	471
1	ford escape	-0.1191	-8.536	10.1533	10.1316	escape	ford	25675.692341	251
2	ford explorer	-0.1531	-10.454	10.7180	10.7270	explorer	ford	45161.489480	455
3	ford fusion	-0.1274	-8.031	10.0453	9.9310	fusion	ford	23047.210003	205
4	ford mustang	-0.0910	-8.584	10.5313	10.4475	mustang	ford	37470.153935	344
...	...	...	...	...	...	...	...	...	
305	mazda mazda5	-0.0975	-9.329	9.8407	9.8351	mazda5	mazda	18782.859421	186
306	mazda mazda2	-0.1571	-4.706	10.0953	9.2187	mazda2	mazda	24228.865728	100
307	mazda cx-5 grand touring	0.2177	19.387	7.8597	7.8765	cx-5 grand	mazda	2590.743036	26

I used a for loop to do robust linear regression on 310 models of cars from 20 manufacturers, and record the

coefficients in a dataframe.

```
vehicle["model and manufacturer"] = vehicle["manufacturer"]+" "+vehicle["model"]
df_5 = vehicle[vehicle["model and manufacturer"].isin(df_4["model and manufacturer"])].\
pivot_table(index='model and manufacturer',columns='type',values='id',aggfunc=lambda x: len(x.unique())).fillna(0)
df_6 = (df_5.div(df_5.sum(axis=1), axis=0)*100).round(2).astype(str) + '%'
df_6["types"] = np.empty((len(df_6), 0)).tolist()
for i in range(len(df_6.columns)-1):
    for j in range(len(df_6.index)):
        if float(df_6.iloc[j,i][:-1])>=25:
            df_6["types"][j].append(df_6.columns[i])
df_6
```

type	SUV	bus	convertible	coupe	hatchback	mini-van	offroad	other	pickup	sedan	truck	van	wagon	types
model and manufacturer														
audi a3	0.0%	0.0%	2.96%	2.22%	14.81%	0.0%	0.0%	2.96%	0.0%	62.22%	0.0%	0.0%	14.81%	[sedan]
audi a4	0.0%	0.0%	3.97%	0.33%	0.17%	0.0%	0.0%	6.13%	0.0%	84.77%	0.0%	0.0%	4.64%	[sedan]
audi a5	0.74%	0.0%	15.56%	70.37%	7.41%	0.0%	0.0%	2.22%	0.0%	3.7%	0.0%	0.0%	0.0%	[coupe]
audi a6	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.15%	0.0%	97.32%	0.0%	0.0%	1.53%	[sedan]
audi a7	0.0%	0.0%	0.0%	0.0%	36.03%	0.0%	0.0%	1.47%	0.0%	62.5%	0.0%	0.0%	0.0%	[hatchback, sedan]
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
volkswagen passat	0.49%	0.0%	0.0%	0.32%	0.49%	0.0%	0.0%	2.1%	0.0%	92.39%	0.16%	0.0%	4.05%	[sedan]
volkswagen rabbit	0.0%	0.0%	4.05%	16.22%	63.51%	0.0%	0.0%	6.76%	0.0%	8.11%	0.0%	0.0%	1.35%	[hatchback]

```
vehicle["model and manufacturer"] = vehicle["manufacturer"]+" "+vehicle["model"]
df_5 = vehicle[vehicle["model and manufacturer"].isin(df_4["model and manufacturer"])].\
pivot_table(index='model and manufacturer',columns='type',values='id',aggfunc=lambda x: len(x.unique())).fillna(0)
df_6 = (df_5.div(df_5.sum(axis=1), axis=0)*100).round(2).astype(str) + '%'
df_6["types"] = np.empty((len(df_6), 0)).tolist()
df_6.insert(len(df_6.columns),"types_str", "")
for i in range(len(df_6.columns)-2):
    for j in range(len(df_6.index)):
        if float(df_6.iloc[j,i][:-1])>=25:
            df_6["types"][j].append(df_6.columns[i])
for j in range(len(df_6.index)):
    df_6["types_str"][j] = ", ".join(df_6["types"][j])
df_4 = df_4.merge(df_6[["types","types_str"]],right_on="model and manufacturer",left_on=df_6.index)
df_4
```

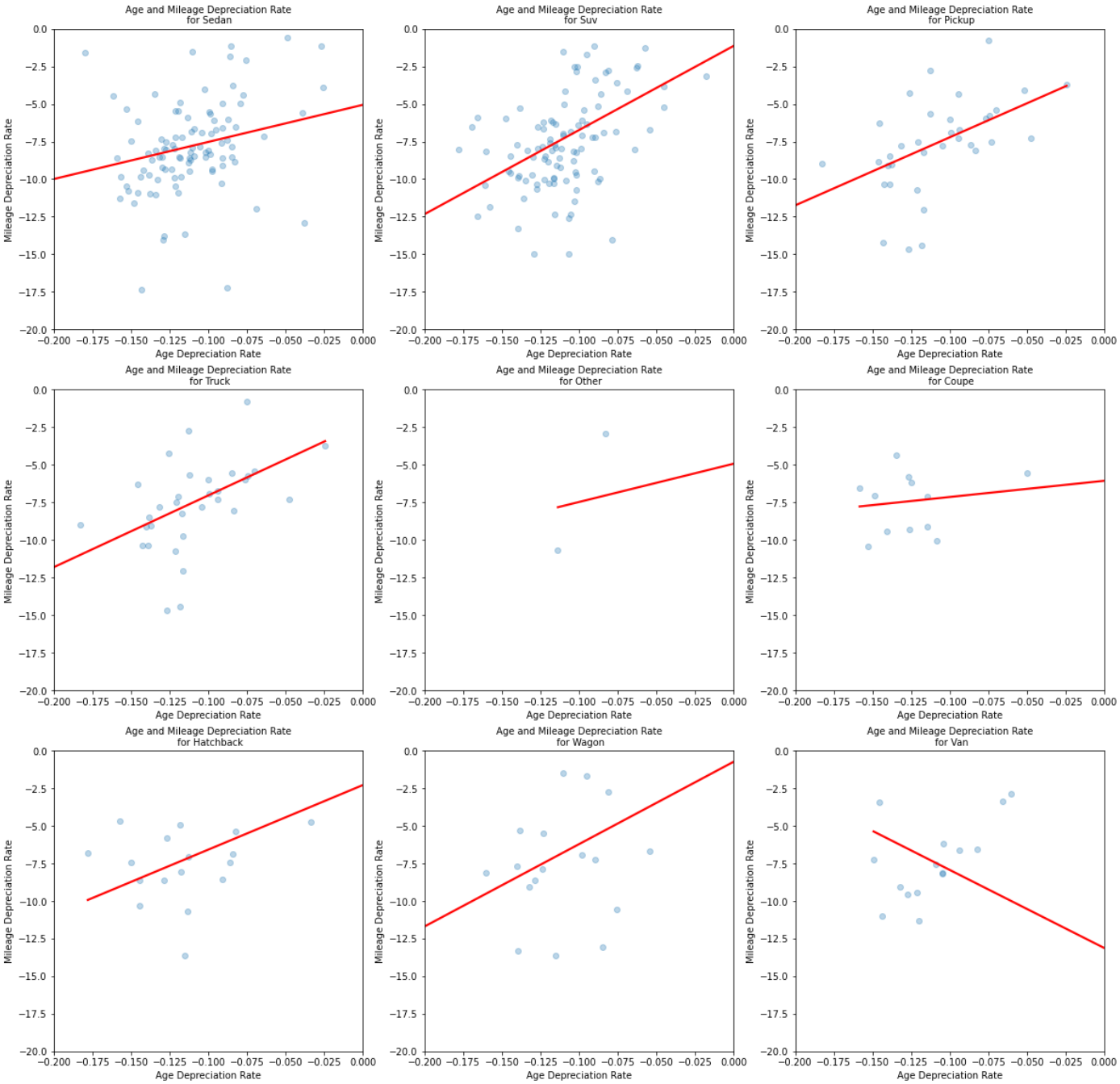
	model and manufacturer	age_depre_rate	mileage_depre_rate (*10**6)	age_const	mileage_const	model	manufacturer	age_0_price	mileage_0_price	types	types_str
0	ford f-150	-0.1308	-7.364	10.9655	10.7615	f-150	ford	57843.710160	47169.369045	[sedan]	sedan
1	ford escape	-0.1191	-8.536	10.1533	10.1316	escape	ford	25675.692341	25124.531540	[sedan]	sedan
2	ford explorer	-0.1531	-10.454	10.7180	10.7270	explorer	ford	45161.489480	45569.777425	[coupe]	coupe
3	ford fusion	-0.1274	-8.031	10.0453	9.9310	fusion	ford	23047.210003	20557.888194	[sedan]	sedan
4	ford mustang	-0.0910	-8.584	10.5313	10.4475	mustang	ford	37470.153935	34458.121634	[hatchback, sedan]	hatchback, sedan
...	...	...	...	...	...	...	...	...	...	...	...
305	mazda	-0.0975	-9.329	9.8407	9.8351	mazda5	mazda	18782.859421	18677.969374	[sedan]	sedan

Then for each car model, I calculate their car type percentage. If more than 25% of that car model belong to a type then we add that car type to this model's type, which means this model has significant amount that belong to this type.

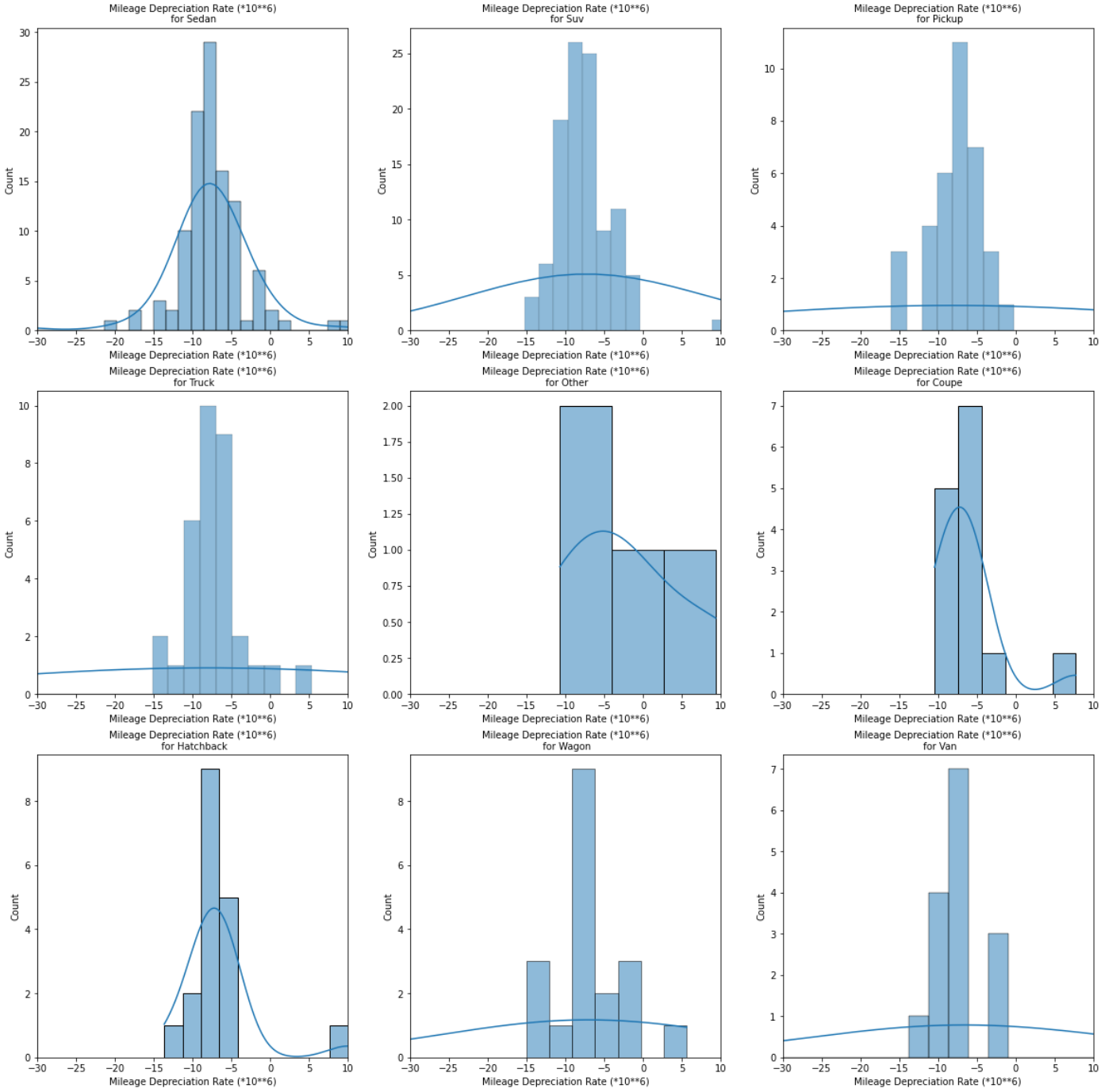
```
from collections import Counter
n = 3
fig, axs = plt.subplots(n, n, figsize=(20, 20))
fig.suptitle("Age and Mileage Depreciation Rate for 310 Models", fontsize=30)
type_list = [x[0] for x in Counter(list(vehicle["type"])).most_common()]
type_list = [x for x in type_list if str(x) != 'nan'][:n**2]
for i in range(len(type_list)):
    car_type = type_list[i]
    new_df = df_4[df_4["types_str"].str.contains(car_type,case=False)]
    g = sns.regplot(x="age_depre_rate",y="mileage_depre_rate (*10**6)",\
    data=new_df,scatter_kws={'alpha':0.3},lowess=False,\
                         robust=True,ci=None,\
                        line_kws={"color": "red"},ax=axs[i//n,i%n])
    g.set_xlim([-0.2,0])
    g.set_ylim([-20,0])
    g.set_title("Age and Mileage Depreciation Rate\n for "+car_type.capitalize(), fontsize=10)
    g.set_xlabel("Age Depreciation Rate", fontsize=10)
    g.set_ylabel("Mileage Depreciation Rate", fontsize=10)
```

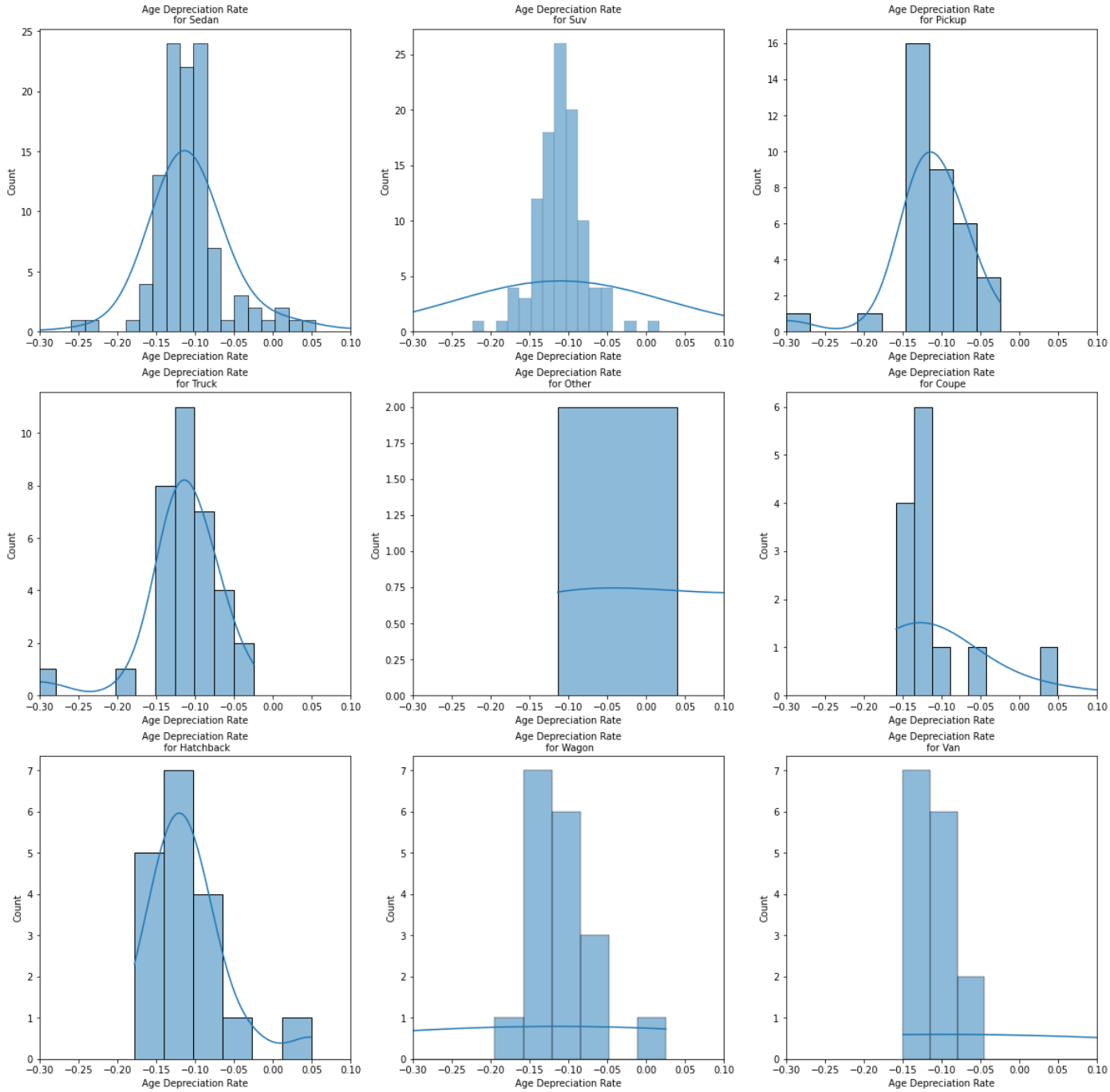
For each car type I did a robust regression and a histogram with kernel density estimation. Here are the results

Age and Mileage Depreciation Rate for 310 Models

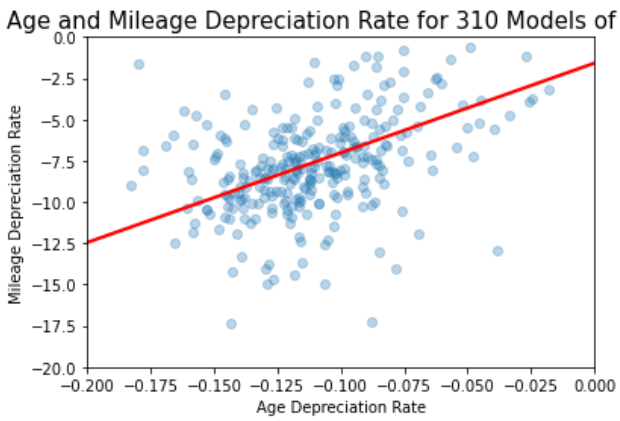


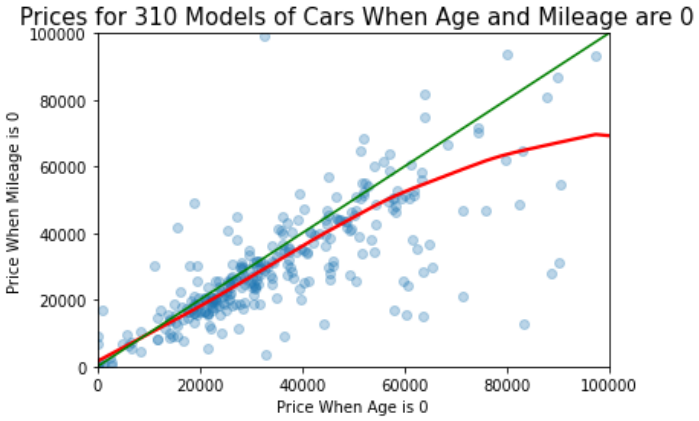
Mileage Depreciation Rate for 310 Models





We can see there does not seem to be a difference in the depreciation rate on age among different types of cars. The distribution is unimodal and center around -0.125 and -0.1. This depreciation rate seems to be consistent among types. The depreciation rate on mileage is centered around 1e-6, which is also unimodal and does not seem to be different among different types of cars. The depreciation rate seem to be a constant with very little variance across all models of cars.





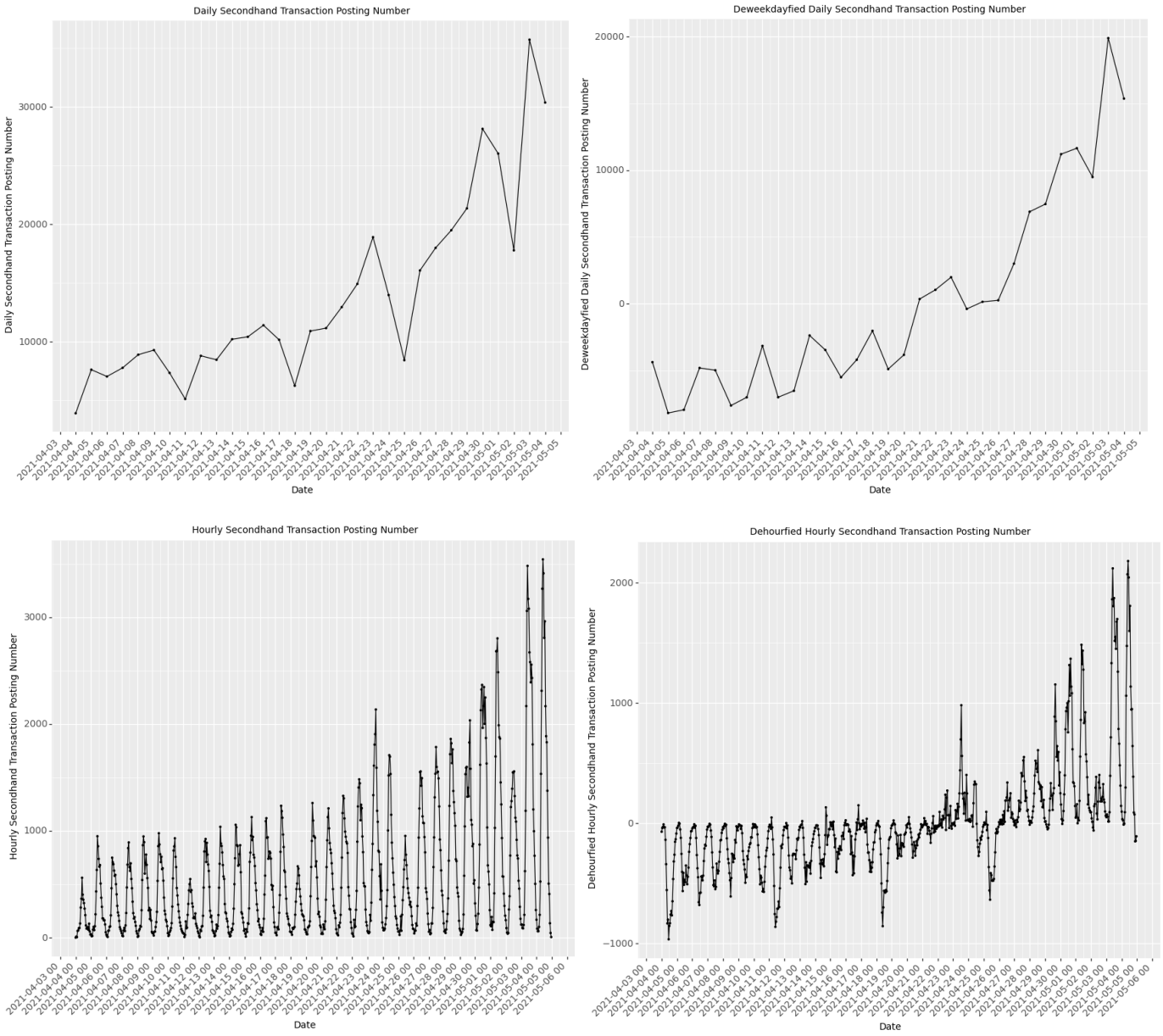
There does seem to be a positive relationship between depreciation rate on car age and depreciation rate on car mileage, and a positive relationship between the price when car age is 0 and price when car mileage is 0. Also we can see the fit line in the second graph is close to the line with slope of 1, indicating that using age and mileage to predict the original price of a car get similar results.

3. Is there a time of the day or a day in the month when the secondhand car transaction is more frequent?

5. According to the posting time analysis, observe whether there is a certain time of day used car transactions are more frequent, whether there is a certain day of the week time period used car transactions are more frequent

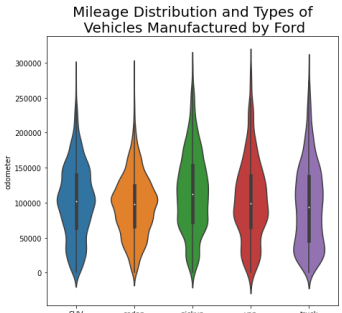
```
from mizani.formatters import comma_format
from plotnine import *
from plotnine.stats import *
from mizani.breaks import date_breaks
from mizani.formatters import date_format
vehicle_2 = vehicle[vehicle["posting_date"].notna()]
vehicle_2["posting_date"] = vehicle_2["posting_date"].str[:-5]
vehicle_2["posting_date"] = pd.to_datetime(vehicle_2["posting_date"],format='%Y-%m-%dT%H:%M:%S')
df = pd.DataFrame(vehicle_2.resample('H', on='posting_date')["id"].count())
df = pd.DataFrame(list(df["id"]),list(df.index))
df.rename(columns={0:"Hourly_Transaction"},inplace=True)
df["hour"]=pd.to_datetime(df.index).strftime('%-H')
model1 = smf.ols('Hourly_Transaction ~ C(hour)', data=df).fit()
df['dehourfied_hourly_Transaction'] = model1.resid
b = (ggplot(df, aes(df.index, df["dehourfied_hourly_Transaction"])) + geom_point(size=0.5) + \
geom_line(aes(group = 1)) \
+ theme(figure_size=(10, 8))\
+ theme(text=element_text(size=10))\
+ theme(axis_text_x=element_text(rotation=45, hjust=1))\
+ labs(x='Date', y='Dehourfied Hourly Secondhand Transaction Posting Number') \
+ ggtitle("Dehourfied Hourly Secondhand Transaction Posting Number")\
+ scale_x_datetime(breaks=date_breaks('1 day'), labels=(date_format('%Y-%m-%d %H')))
b
```

Here is the code I use to analyze the time series. I first get the hourly posting number and then did a regression analysis with hours as categorical variable, and use the residue as the adjusted value, which is removed of the effect of hour. Then I used ggplot to plot the data here.

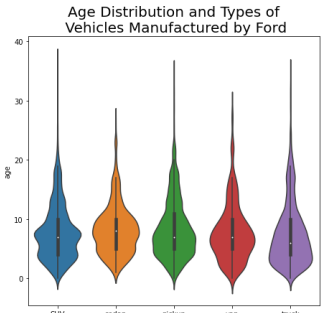


Here we can see the number of postings is periodical. In a week, Sunday has the least postings. From Monday to Sunday, the number of postings increases and then decreases. There is also an apparent daily trend. The number of postings reach the minimum around midnight, increasing to maximum at noon, and then decreasing. The number of posting around midnight is always close to 0. Across this month, we can see a general trend that is the number of posting for secondhand car increased a lot, almost exponentially.

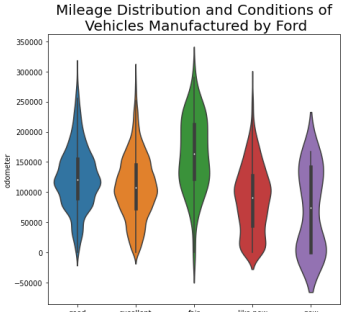
Appendix



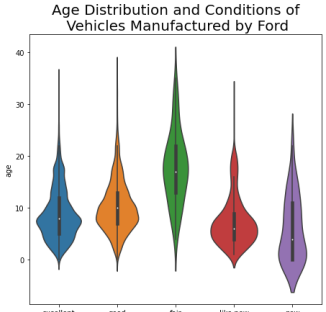
Mileage Distribution and Types of Vehicles Manufactured by Ford



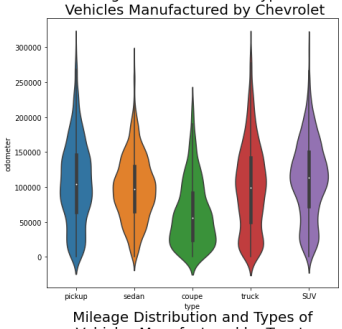
Age Distribution and Types of Vehicles Manufactured by Ford



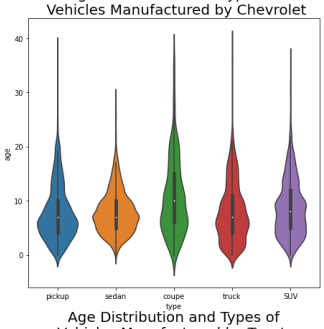
Mileage Distribution and Conditions of Vehicles Manufactured by Ford



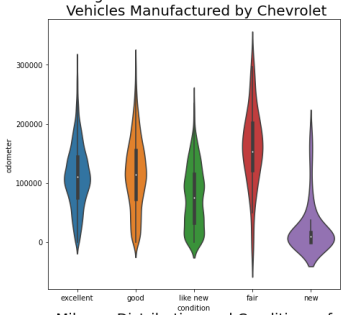
Age Distribution and Conditions of Vehicles Manufactured by Ford



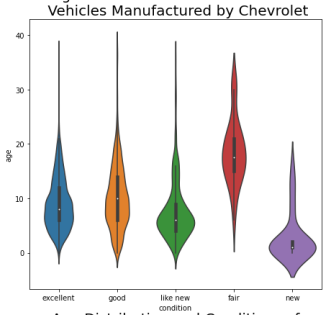
Mileage Distribution and Types of Vehicles Manufactured by Chevrolet



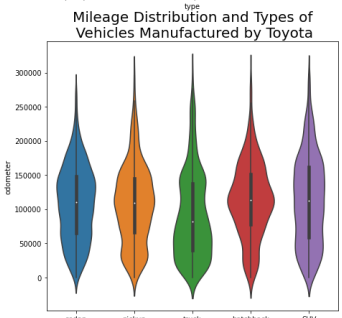
Age Distribution and Types of Vehicles Manufactured by Chevrolet



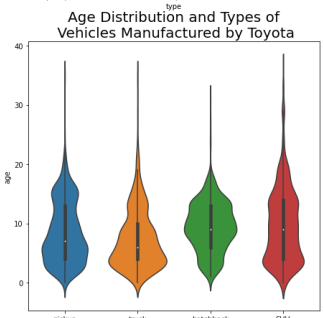
Mileage Distribution and Conditions of Vehicles Manufactured by Chevrolet



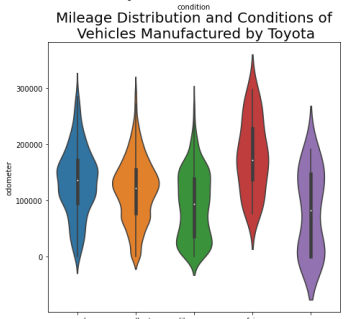
Age Distribution and Conditions of Vehicles Manufactured by Chevrolet



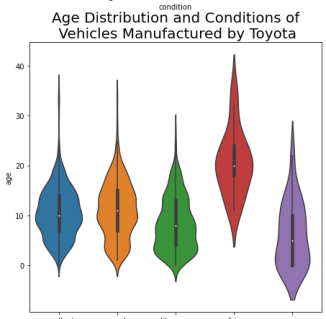
Mileage Distribution and Types of Vehicles Manufactured by Toyota



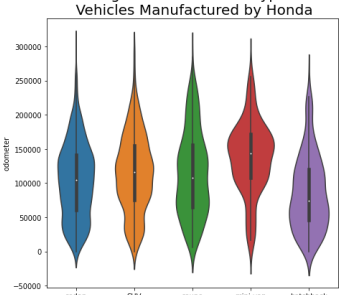
Age Distribution and Types of Vehicles Manufactured by Toyota



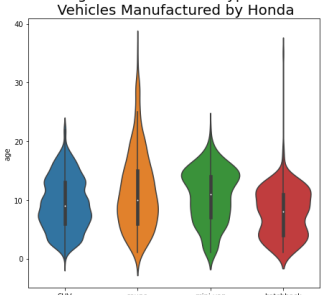
Mileage Distribution and Conditions of Vehicles Manufactured by Toyota



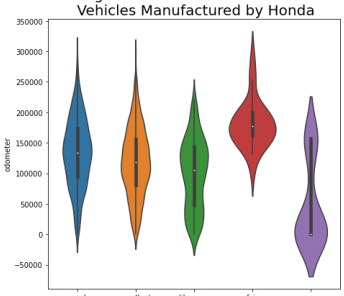
Age Distribution and Conditions of Vehicles Manufactured by Toyota



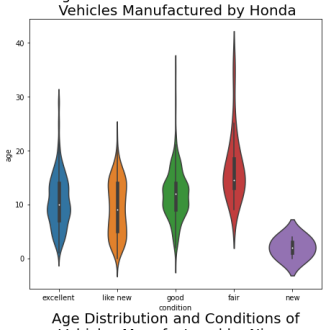
Mileage Distribution and Types of Vehicles Manufactured by Honda



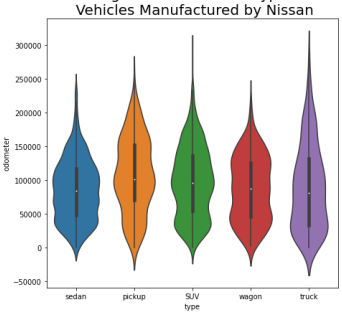
Age Distribution and Types of Vehicles Manufactured by Honda



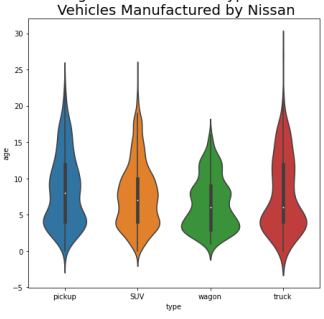
Mileage Distribution and Conditions of Vehicles Manufactured by Honda



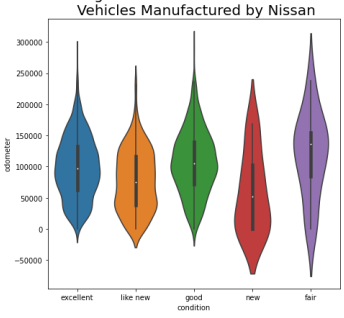
Age Distribution and Conditions of Vehicles Manufactured by Honda



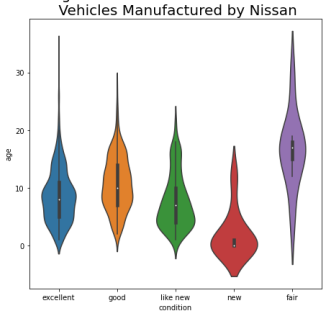
Mileage Distribution and Types of Vehicles Manufactured by Nissan



Age Distribution and Types of Vehicles Manufactured by Nissan



Mileage Distribution and Conditions of Vehicles Manufactured by Nissan



Age Distribution and Conditions of Vehicles Manufactured by Nissan