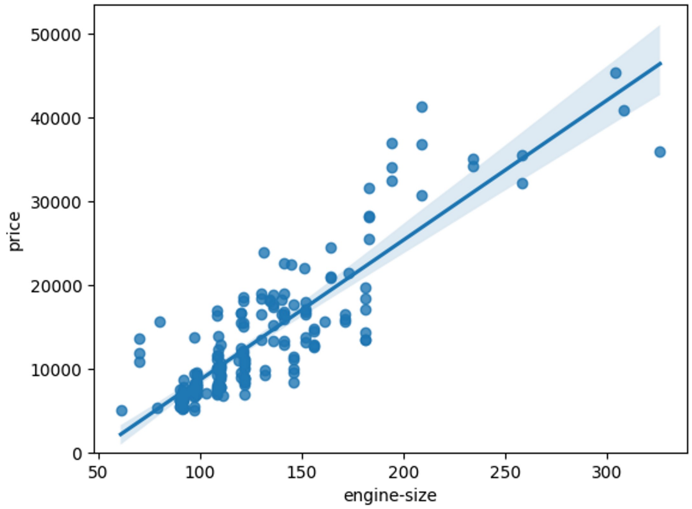
WEEK3



As the engine-size goes up, the price goes up: this indicates a positive direct correlation between these two variables. Engine size seems like a pretty good predictor of price since the regression line is almost a perfect diagonal line.
Code:

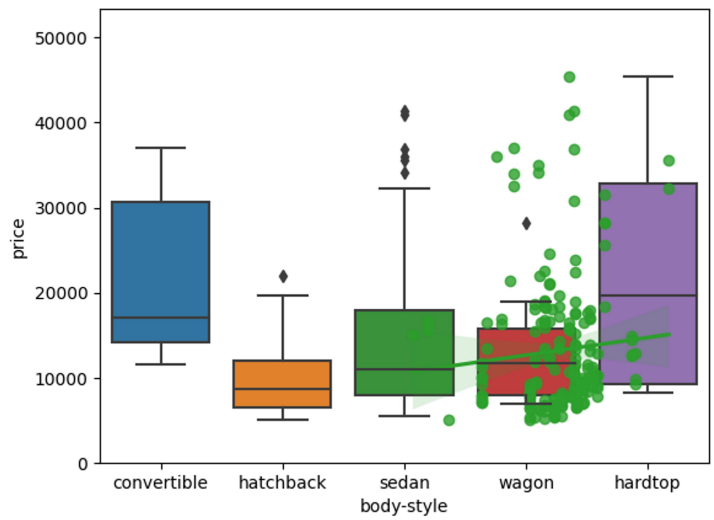
```
sns.regplot(x="engine-size", y="price", data=df)
plt.ylim(0,)
plt.show()
```

We can examine the correlation between 'engine-size' and 'price' and see that it's approximately 0.87.
Code:

```
print(df[['engine-size','price']].corr())
```

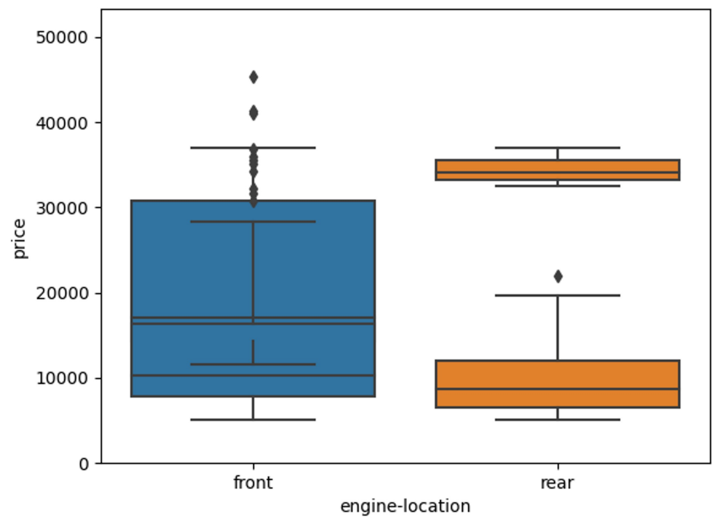
Or

```
print(df['engine-size'].corr(df['price']))
```



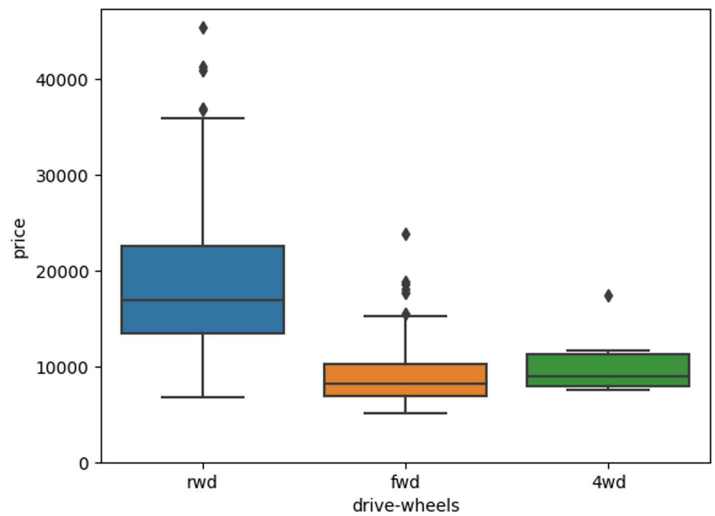
We see that the distributions of price between the different body-style categories have a significant overlap, so body-style would not be a good predictor of price. Let's examine engine "engine-location" and "price"
Code:

```
sns.boxplot(x="body-style", y="price", data=df)
plt.ylim(0,)
plt.show()
```



Here we see that the distribution of price between these two engine-location categories, front and rear, are distinct enough to take engine-location as a potential good predictor of price.
Code:

```
sns.boxplot(x="engine-location", y="price", data=df)
plt.ylim(0,)
plt.show()
```



Here we see that the distribution of price between the different drive-wheels categories differs. As such, drive-wheels could potentially be a predictor of price.
Code:

```
sns.boxplot(x="drive-wheels", y="price", data=df)
plt.ylim(0,)
plt.show()
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    JUPYTER    TERMINAL                Python

   drive-wheels   body-style         price
0           4wd    hatchback   7603.000000
1           4wd        sedan  12647.333333
2           4wd        wagon   9095.750000
3           fwd  convertible  11595.000000
4           fwd      hardtop   8249.000000
5           fwd    hatchback   8396.387755
6           fwd        sedan   9811.800000
7           fwd        wagon   9997.333333
8           rwd  convertible  23949.600000
9           rwd      hardtop  24202.714286
10          rwd    hatchback  14337.777778
11          rwd        sedan  21711.833333
12          rwd        wagon  16994.222222
PS C:\Users\msii\OneDrive\git\Python\dataAnalysis\week3>
```

This grouped data is much easier to visualize when it is made into a pivot table. A pivot table is like an Excel spreadsheet, with one variable along the column and another along the row. We can convert the dataframe to a pivot table using the method "pivot" to create a pivot table from the groups.
Code:

```
df_gptest = df[['drive-wheels','body-style','price']]
grouped_test1 = df_gptest.groupby(['drive-wheels','body-style'],as_index=False).mean()
print(grouped_test1)
```

```
body-style    convertible       hardtop     hatchback         sedan         wagon
drive-wheels
4wd                   0.0      0.000000   7603.000000  12647.333333   9095.750000
fwd               11595.0   8249.000000   8396.387755   9811.800000   9997.333333
rwd               23949.6  24202.714286  14337.777778  21711.833333  16994.222222
PS C:\Users\msii\OneDrive\git\Python\dataAnalysis\week3>
```

n this case, we will leave the drive-wheels variable as the rows of the table, and pivot body-style to become the columns of the table:
Code:

```
grouped_pivot = grouped_test1.pivot(index='drive-wheels',columns='body-style')
grouped_pivot = grouped_pivot.fillna(0) #fill missing values with 0
grouped_pivot
print(grouped_pivot)
```