stat_1

February 12, 2023

[ ]:
```python
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[ ]:
```python
# read data file and create data frame
df = pd.read_csv('cars_clean.csv')
```

[ ]:
```python
# check size
df.shape
```

[ ]:
```python
# check head
df.head(10)
```

[ ]:
```python
# list data type for each column
print (df.dtypes)
```

[ ]:
```python
# what is the type of peak-rpm
print (df['peak-rpm'].dtype)
```

[ ]:
```python
# find correlation between all numeric values
df.corr(numeric_only = True)
```

[ ]:
```python
# find correlation between bore, stroke, compression-ratio,
# and horsepower
df[['bore', 'stroke', 'compression-ratio', 'horsepower']].corr(numeric_only =␣
↪True)
```

Relationship Among Data

[ ]:
```python
# check relationship between engine-size and price
sns.regplot (x = 'engine-size', y = 'price', data = df)
plt.ylim(0,)
```

[ ]:
```python
df[["engine-size", "price"]].corr()
```

```
[ ]: # find relationship between 'highway-mpg' and 'price'
sns.regplot(x="highway-mpg", y="price", data=df)
```

```
[ ]: # find correlation between 'highway-mpg' and 'price'
df[['highway-mpg', 'price']].corr()
```

```
[ ]: # weak linear relationship
sns.regplot(x="peak-rpm", y="price", data=df)
```

```
[ ]: # find correlation
df[['peak-rpm','price']].corr()
```

```
[ ]: # find correlation
df[['stroke','price']].corr()
```

```
[ ]: # weak linear relationship
sns.regplot(x="stroke", y="price", data=df)
```

Categorical Variables

```
[ ]: sns.boxplot(x="body-style", y="price", data=df)
```

```
[ ]: sns.boxplot(x="engine-location", y="price", data=df)
```

```
[ ]: # drive-wheels
sns.boxplot(x="drive-wheels", y="price", data=df)
```

Descriptive Statistical Analysis

- count of that variable
- mean
- standard deviation
- minimum value
- IQR (interquartile range: 25%, 50%, 75%)
- maximum value

```
[ ]: df.describe()
```

```
[ ]: df.describe (include=['object'])
```

Value Counts value_counts() works only Pandas series and not on dataframes

```
[ ]: df['drive-wheels'].value_counts()
```

```
[ ]: # convert series to dataframe
df['drive-wheels'].value_counts().to_frame()
```

```
[ ]: drive_wheels_counts = df['drive-wheels'].value_counts().to_frame()
     drive_wheels_counts.rename(columns={'drive-wheels': 'value_counts'},␣
      ↪inplace=True)
     drive_wheels_counts
```

```
[ ]: drive_wheels_counts.index.name = 'drive-wheels'
     drive_wheels_counts
```

```
[ ]: # repeat process with 'engine-location'
     engine_loc_counts = df['engine-location'].value_counts().to_frame()
     engine_loc_counts.rename(columns={'engine-location': 'value_counts'},␣
      ↪inplace=True)
     engine_loc_counts.index.name = 'engine-location'
     engine_loc_counts.head(10)
```

Grouping Data

```
[ ]: # how many unique values for 'drive-wheels'
     df['drive-wheels'].unique()
```

```
[ ]: # how many unique values for 'body-style'
     df['body-style'].unique()
```

```
[ ]: # let us create a group
     df_group_one = df[['drive-wheels','price']]
```

```
[ ]: # find the mean for each group
     df_group_one = df_group_one.groupby(['drive-wheels'],as_index=False).mean()
     df_group_one
```

```
[ ]: # let us create another group and find the mean
     df_group_two = df[['body-style','price']]
     df_group_two = df_group_two.groupby(['body-style'],as_index=False).mean()
     df_group_two
```

```
[ ]: # grouping results
     df_gptest = df[['drive-wheels','body-style','price']]
     grouped_test1 = df_gptest.groupby(['drive-wheels','body-style'],as_index=False).
      ↪mean()
     grouped_test1
```

Pivot Table = Excel Table

```
[ ]: grouped_pivot = grouped_test1.pivot(index='drive-wheels',columns='body-style')
     grouped_pivot
```

[ ]:
```python
grouped_pivot = grouped_pivot.fillna(0) #fill missing values with 0
grouped_pivot
```

Heat Map

[ ]:
```python
#use the grouped results
plt.pcolor(grouped_pivot, cmap='RdBu')
plt.colorbar()
plt.show()
```

[ ]:
```python
fig, ax = plt.subplots()
im = ax.pcolor(grouped_pivot, cmap='RdBu')

#label names
row_labels = grouped_pivot.columns.levels[1]
col_labels = grouped_pivot.index

#move ticks and labels to the center
ax.set_xticks(np.arange(grouped_pivot.shape[1]) + 0.5, minor=False)
ax.set_yticks(np.arange(grouped_pivot.shape[0]) + 0.5, minor=False)

#insert labels
ax.set_xticklabels(row_labels, minor=False)
ax.set_yticklabels(col_labels, minor=False)

#rotate label if too long
plt.xticks(rotation=90)

fig.colorbar(im)
plt.show()
```

Correlation and Causation

- Correlation: a measure of the extent of interdependence between variables
- Causation: the relationship between cause and effect between two variables

Pearson Correlation Compute the linear correlation between two variables
* 1: perfect positive correlation * 0: no correlation * -1: perfect negative correlation

[ ]:
```python
# find correlation
df.corr(numeric_only = True)
```

P-value P-value is the probability that the correlation between two variables is statistically significant. We choose a significance level of 0.05 which means that we are 95% confident that the correlation between two objects is significant.

- $p < 0.001$: strong evidence that the correlation is significant
- $p < 0.05$: moderate evidence that the correlation is significant

- p < 0.1: weak evidence that the correlation is significant
- p > 0.1: no evidence that the correlation is significant

[ ]:
```
# import the stats library
from scipy import stats
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'wheel-base' and 'price'
pearson_coef, p_value = stats.pearsonr(df['wheel-base'], df['price'])
print("Correlation Coefficient = ", pearson_coef, " P-value =", p_value)
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'horsepower' and 'price'
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'length' and 'price'
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'width' and 'price'
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'curb-weight' and 'price'
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'engine-size' and 'price'
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'bore' and 'price'
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'city-mpg' and 'price'
```

[ ]:
```
# compute the correlation coefficient and p-value between
# 'highway-mpg' and 'price'
```

Analysis of Variance (ANOVA) Analysis of Variance is a statistical method to test whether there are significant differences between the means of two or more groups. ANOVA returns two parameters:
* F-test score: ANOVA assumes the means of all groups are the same, calculates how much the actual means deviate from this assumption, and reports is as the F-test score. A larger score means there is a larger difference between the means. * P-value: tells us how statistically significant our calculated score is
If our price variable is strongly correlated with the variable that we are analyzing we expect ANOVA to return a large F-test score and a small P-value.

[ ]:
```
grouped_test2=df_gptest[['drive-wheels', 'price']].groupby(['drive-wheels'])
grouped_test2.head(2)
```

```
[ ]: df_gptest
```

```
[ ]: grouped_test2.get_group('4wd')['price']
```

```
[ ]: # ANOVA
f_val, p_val = stats.f_oneway(grouped_test2.get_group('fwd')['price'],␣
 ↪grouped_test2.get_group('rwd')['price'], grouped_test2.
 ↪get_group('4wd')['price'])

print( "ANOVA results: F=", f_val, ", P =", p_val)
```

fwd and rwd

```
[ ]: f_val, p_val = stats.f_oneway(grouped_test2.get_group('fwd')['price'],␣
 ↪grouped_test2.get_group('rwd')['price'])

print( "ANOVA results: F=", f_val, ", P =", p_val )
```

4wd and rwd

```
[ ]: f_val, p_val = stats.f_oneway(grouped_test2.get_group('4wd')['price'],␣
 ↪grouped_test2.get_group('rwd')['price'])

print( "ANOVA results: F=", f_val, ", P =", p_val)
```

4wd and fwd

```
[ ]: f_val, p_val = stats.f_oneway(grouped_test2.get_group('4wd')['price'],␣
 ↪grouped_test2.get_group('fwd')['price'])

print("ANOVA results: F=", f_val, ", P =", p_val)
```

Conclusion

These are the variables that are important in determining the price of a used car: * Length * Width * Curb-weight * Engine-size * Horsepower * City-mpg * Highway-mpg * Wheel-base * Bore * Drive-wheels (categorical)