In [1]:
```
#22rd Aug 2022
#BU MSBA2022 BootCamp Group Project part1
#Group 10: Jiadai Yu, Yuesen Zhang, Xin Su
#Credit to Prof.Hemant Sangwan
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

filename = r'D:\0-Summer Pre\boot camp\Ford-Data-GroupProject.xlsx'
Ford = pd.read_excel(filename , header=0)
```

In [3]:
```
#--------------------------------
#Basic Information about the data

print(Ford.head())
print(Ford.shape)    #shape of data (rows, columns)
print(Ford.dtypes)  # variable types (integer, object, float)
print(Ford.info())  # info about data columns, variable names, etc.
print(Ford.count())  # non-missing observationcounts
```

```
     model  year transmission fuelType  engineSize  price  mileage  tax   mpg
0   Fiesta  2017    Automatic   Petrol         1.0  12000    15944  150  57.7
1    Focus  2018       Manual   Petrol         1.0  14000     9083  150  57.7
2    Focus  2017       Manual   Petrol         1.0  13000    12456  150  57.7
3   Fiesta  2019       Manual   Petrol         1.5  17500    10460  145  40.3
4   Fiesta  2019    Automatic   Petrol         1.0  16500     1482  145  48.7
(17966, 9)
model            object
year              int64
transmission     object
fuelType         object
engineSize      float64
price             int64
mileage           int64
tax               int64
mpg             float64
dtype: object
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17966 entries, 0 to 17965
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   model         17966 non-null  object 
 1   year          17966 non-null  int64  
 2   transmission  17966 non-null  object 
 3   fuelType      17966 non-null  object 
 4   engineSize    17966 non-null  float64
 5   price         17966 non-null  int64  
 6   mileage       17966 non-null  int64  
 7   tax           17966 non-null  int64  
 8   mpg           17966 non-null  float64
dtypes: float64(2), int64(4), object(3)
memory usage: 1.2+ MB
None
model           17966
year            17966
transmission    17966
fuelType        17966
engineSize      17966
price           17966
mileage         17966
tax             17966
```

```
mpg             17966
dtype: int64
```

In [4]:

```
#--------------------------------
#Q1: summary statistics

#quick look through all attributes
for i in ['model','year','transmission','fuelType','engineSize']:
    print('prices vary across '+ i +' :')
    print(pd.DataFrame(Ford.groupby(i)['price'].mean()))
    print()
```

```
prices vary across model :
                              price
model
 B-MAX                    8287.526761
 C-MAX                    9914.567219
 EcoSport                12499.268591
 Edge                    22810.500000
 Escort                   3000.000000
 Fiesta                  10196.298002
 Focus                   13185.882956
 Fusion                   2555.812500
 Galaxy                  17841.872807
 Grand C-MAX             10881.574899
 Grand Tourneo Connect   14874.915254
 KA                       5186.125628
 Ka+                      8707.856874
 Kuga                    15823.472360
 Mondeo                  12305.709125
 Mustang                 34631.263158
 Puma                    21447.250000
 Ranger                  14495.000000
 S-MAX                   17720.226351
 Streetka                 1924.500000
 Tourneo Connect         13805.818182
 Tourneo Custom          21165.985507
 Transit Tourneo         12450.000000
Focus                     8299.000000

prices vary across year :
             price
year
1996   3000.000000
1998   2699.000000
2000   1995.000000
2002   1928.333333
2003   2063.000000
2004   1436.000000
2005   1593.000000
2006   2202.615385
2007   2603.562500
2008   2598.894737
2009   3719.725275
```

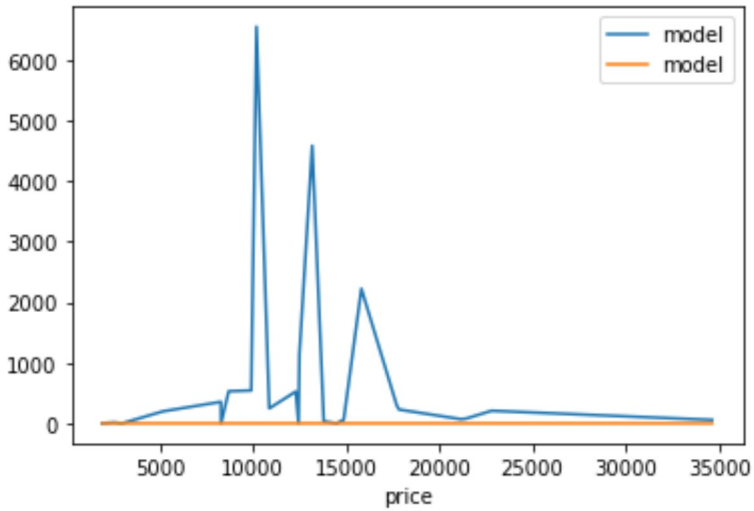
```
2010   4058.253731
2011   5022.478723
2012   5680.930435
2013   6703.940887
2014   7541.874534
2015   8777.817982
2016  10665.668812
2017  11965.689853
2018  13157.894619
2019  17176.449280
2020  20819.872093
2060   6495.000000

prices vary across transmission :
                     price
transmission
Automatic     15727.234386
Manual        11792.264918
Semi-Auto     14919.034039

prices vary across fuelType :
                 price
fuelType
Diesel    13659.173724
Electric  15737.500000
Hybrid    22149.090909
Other     13800.000000
Petrol    11608.293702

prices vary across engineSize :
                   price
engineSize
0.0         11776.764706
1.0         11892.877785
1.1         10245.579606
1.2          7436.312423
1.3          2439.923077
1.4          5237.955357
1.5         12863.070802
1.6          7717.875406
1.7          2195.000000
1.8          2961.171429
```

```
2.0         16287.645726
2.2         10400.923077
2.3         26791.887500
2.5         18292.615385
3.2         14495.000000
5.0         34967.800000
```

In [5]:
```
#drop the unrealistic year = 2060 as an input error
Ford.drop(Ford.index[Ford['year'] == 2060], inplace=True)
```

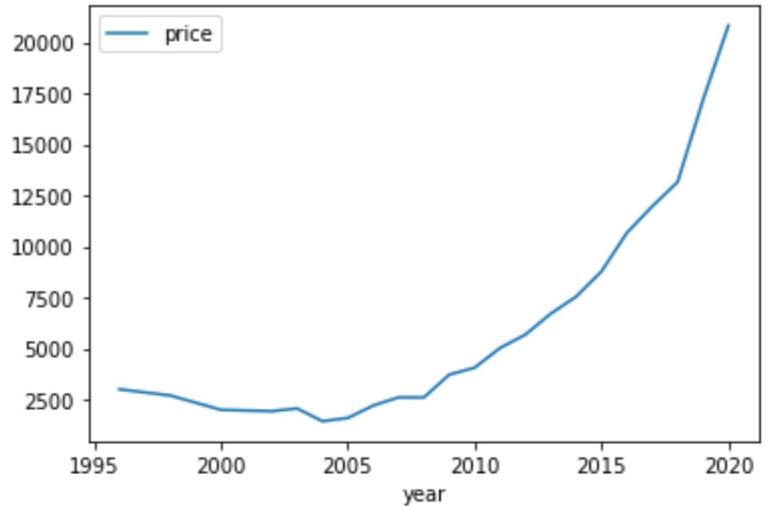
In [6]:
```
#'model'
price_by_model = pd.DataFrame(Ford.groupby('model')['price'].mean())
model_count = Ford['model'].value_counts()
model_count_props = Ford['model'].value_counts(normalize = True)
price_by_model_count_srt = pd.concat([price_by_model,model_count,model_count_props] , axis=1).sort_values('price')

print(price_by_model_count_srt)
price_by_model_count_srt.plot(x='price' , y='model' )
plt.show()
```

	price	model	model
Streetka	1924.500000	2	0.000111
Fusion	2555.812500	16	0.000891
Escort	3000.000000	1	0.000056
KA	5186.125628	199	0.011077
B-MAX	8287.526761	355	0.019761
Focus	8299.000000	1	0.000056
Ka+	8707.856874	531	0.029557
C-MAX	9914.567219	543	0.030225
Fiesta	10196.862569	6556	0.364932
Grand C-MAX	10881.574899	247	0.013749
Mondeo	12305.709125	526	0.029279
Transit Tourneo	12450.000000	1	0.000056
EcoSport	12499.268591	1143	0.063624
Focus	13185.882956	4588	0.255385
Tourneo Connect	13805.818182	33	0.001837
Ranger	14495.000000	1	0.000056
Grand Tourneo Connect	14874.915254	59	0.003284
Kuga	15823.472360	2225	0.123852
S-MAX	17720.226351	296	0.016476
Galaxy	17841.872807	228	0.012691
Tourneo Custom	21165.985507	69	0.003841
Puma	21447.250000	80	0.004453
Edge	22810.500000	208	0.011578
Mustang	34631.263158	57	0.003173



In [7]:
```
#'year'
price_by_year = pd.DataFrame(Ford.groupby('year')['price'].mean())

print(price_by_year.sort_values('price'))
price_by_year.plot()
plt.show()
```

```
             price
year
2004   1436.000000
2005   1593.000000
2002   1928.333333
2000   1995.000000
2003   2063.000000
2006   2202.615385
2008   2598.894737
2007   2603.562500
1998   2699.000000
1996   3000.000000
2009   3719.725275
2010   4058.253731
2011   5022.478723
2012   5680.930435
2013   6703.940887
2014   7541.874534
2015   8777.817982
2016  10665.668812
2017  11965.689853
2018  13157.894619
2019  17176.449280
2020  20819.872093
```



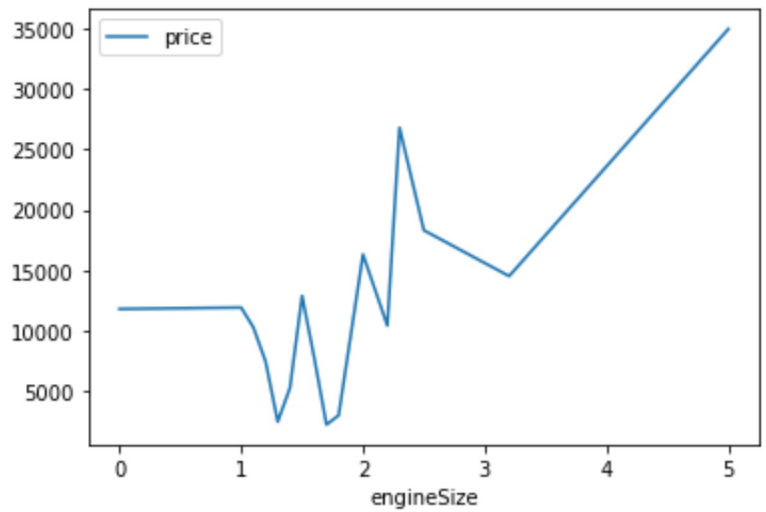
In [8]:
```
#'transmission'
price_by_trans = pd.DataFrame(Ford.groupby('transmission')['price'].mean())
print(price_by_trans.sort_values('price'))
```

```
                    price
transmission
Manual        11792.264918
Semi-Auto     14919.034039
Automatic     15734.022794
```

In [9]:
```
#'fuelType'
price_by_fuel = pd.DataFrame(Ford.groupby('fuelType')['price'].mean())
fuelType_count = Ford['fuelType'].value_counts()
print(price_by_fuel.sort_values('price'))
print(fuelType_count)
```

```
                price
fuelType
Petrol    11608.713582
Diesel    13659.173724
Other     13800.000000
Electric  15737.500000
Hybrid    22149.090909
Petrol      12178
Diesel       5762
Hybrid         22
Electric        2
Other           1
Name: fuelType, dtype: int64
```

In [10]:
```
#'engineSize'
price_by_engineSize = pd.DataFrame(Ford.groupby('engineSize')['price'].mean())
price_by_engineSize.plot()
plt.show()
```



In [11]:
```
#--------------------------------
#Q2 identify outlier of price, mileage, tax, and mpg

#import zscore
from scipy.stats import zscore

Ford_outliers = []
for i in ['price', 'mileage', 'tax', 'mpg']:
    standardized = pd.DataFrame(Ford[i].transform(zscore))
    outliers = ((standardized[i] < -3) | (standardized[i] > 3))
    Ford_outliers = Ford[outliers]
    print(i)
    print(Ford[i].agg([np.min,np.max,np.mean,np.median]))
    print("outlier_mean\n",Ford_outliers[i].agg([np.min,np.max,np.mean,np.median]))
print(Ford_outliers.head()) ##Outlier mean

#credit to slides-S4 Page18
```

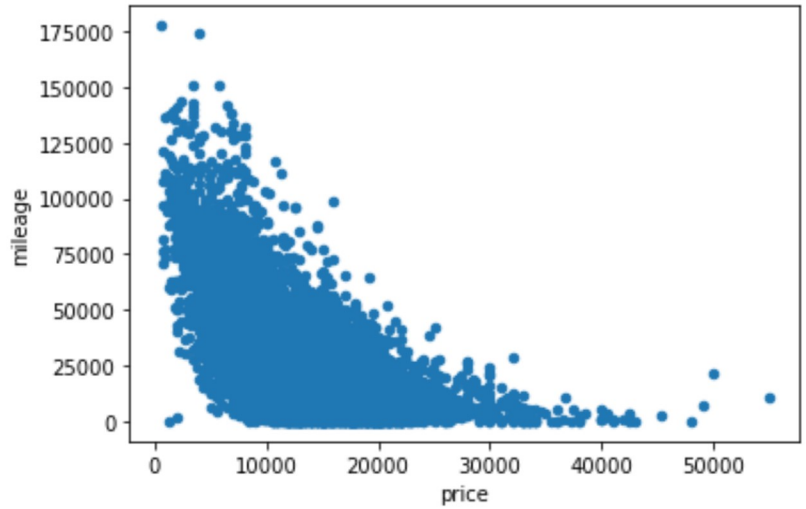
```
price
amin        495.000000
amax      54995.000000
mean      12279.856833
median    11291.000000
Name: price, dtype: float64
outlier_mean
 amin      26698.000000
amax      54995.000000
mean      31887.508772
median    30000.000000
Name: price, dtype: float64
mileage
amin            1.000000
amax       177644.000000
mean        23360.858447
median      18242.000000
Name: mileage, dtype: float64
outlier_mean
 amin       81899.000000
amax      177644.000000
mean      100954.810976
median     95511.000000
Name: mileage, dtype: float64
tax
amin        0.000000
amax      580.000000
mean      113.324353
median    145.000000
Name: tax, dtype: float64
outlier_mean
 amin      300.000000
amax      580.000000
mean      403.636364
median    325.000000
Name: tax, dtype: float64
mpg
amin       20.800000
amax      201.800000
mean       57.907821
median     58.900000
Name: mpg, dtype: float64
```

```
outlier_mean
 amin       20.800000
amax      201.800000
mean       69.464234
median     88.300000
Name: mpg, dtype: float64
        model  year transmission fuelType  engineSize  price  mileage  tax  \
302   Mustang  2020    Automatic   Petrol         5.0  42489     3500  145
354   Mustang  2018    Automatic   Petrol         5.0  31498     6250  145
364   Mustang  2016    Semi-Auto   Petrol         5.0  24999    42086  570
387   Mustang  2019    Automatic   Petrol         5.0  39998     5000  145
741   Mustang  2018       Manual   Petrol         5.0  29998    20847  145

      mpg
302  22.1
354  23.5
364  23.5
387  22.6
741  20.9
```

In [12]:
```
#price

#mileage

#tax
##outliers: tax=0

#mpg mile per gallon
##outliers: mpg=201.8
```

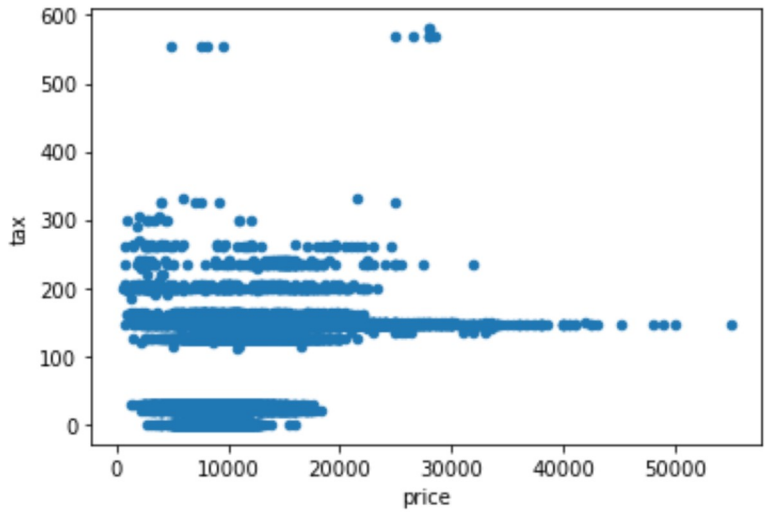
In [13]:
```
#--------------------------------
#Q3 (price, mileage, tax, and mpg) correlation matrix

#price mileage
print(np.corrcoef(Ford['price'],Ford['mileage']))
Ford.plot.scatter(x='price' , y='mileage' )
plt.show()
```

```
[[ 1.         -0.53060994]
 [-0.53060994  1.        ]]
```



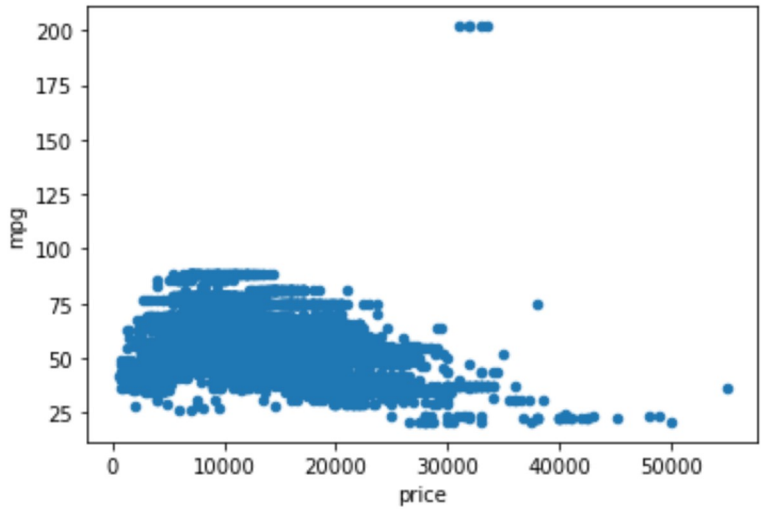
In [14]:
```
#price tax
print(np.corrcoef(Ford['price'],Ford['tax']))
Ford.plot.scatter(x='price' , y='tax' )
plt.show()
```

```
[[1.         0.40699933]
 [0.40699933 1.        ]]
```



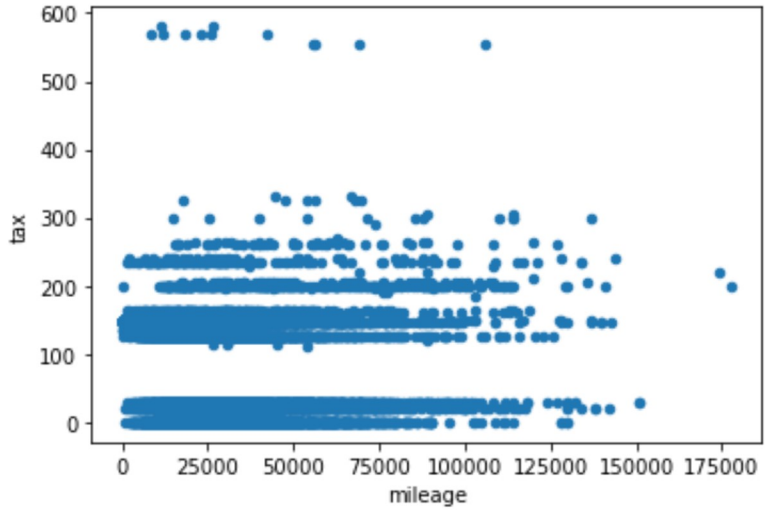
In [15]:
```
#price mpg
print(np.corrcoef(Ford['price'],Ford['mpg']))
Ford.plot.scatter(x='price' , y='mpg' )
plt.show()
```

```
[[ 1.         -0.34655662]
 [-0.34655662  1.        ]]
```



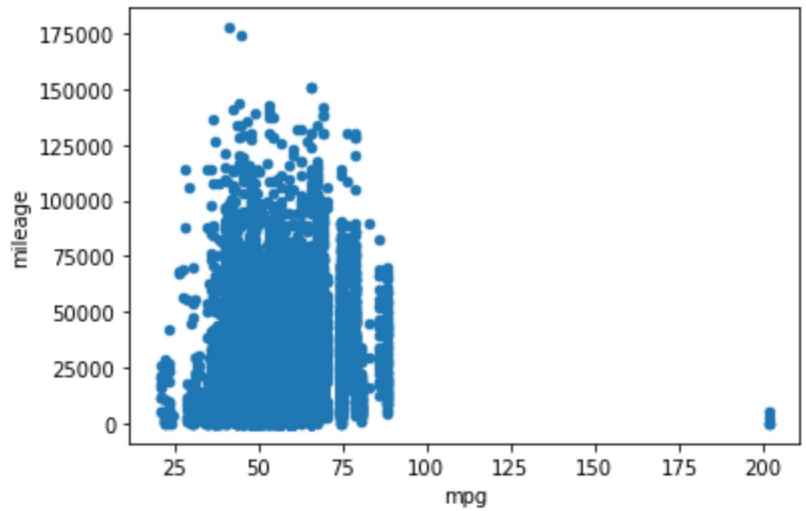
In [16]:
```
#mileage tax
print(np.corrcoef(Ford['mileage'],Ford['tax']))
Ford.plot.scatter(x='mileage' , y='tax' )
plt.show()
```

```
[[ 1.         -0.26061865]
 [-0.26061865  1.        ]]
```



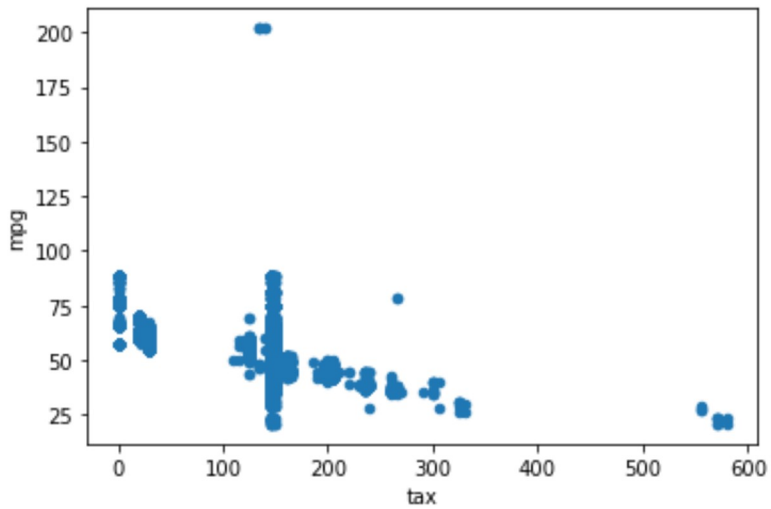
In [17]:
```
#mileage mpg
print(np.corrcoef(Ford['mileage'],Ford['mpg']))
Ford.plot.scatter(x='mpg' , y='mileage' )
plt.show()
```

```
[[1.         0.12022522]
 [0.12022522 1.        ]]
```



In [18]:
```
#tax mpg
print(np.corrcoef(Ford['tax'],Ford['mpg']))
Ford.plot.scatter(x='tax' , y='mpg' )
plt.show()
```

```
[[ 1.         -0.50291947]
 [-0.50291947  1.        ]]
```



In [ ]: