In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('seaborn-dark')
%config InlineBackend.figure_format = 'retina'
pd.options.display.max_rows = 20
pd.options.display.max_columns = 20

plt.rcParams["figure.figsize"] = (14,4)
plt.rcParams['lines.linewidth'] = 2
plt.rcParams['lines.color'] = 'r'
plt.rcParams['axes.grid'] = True
```

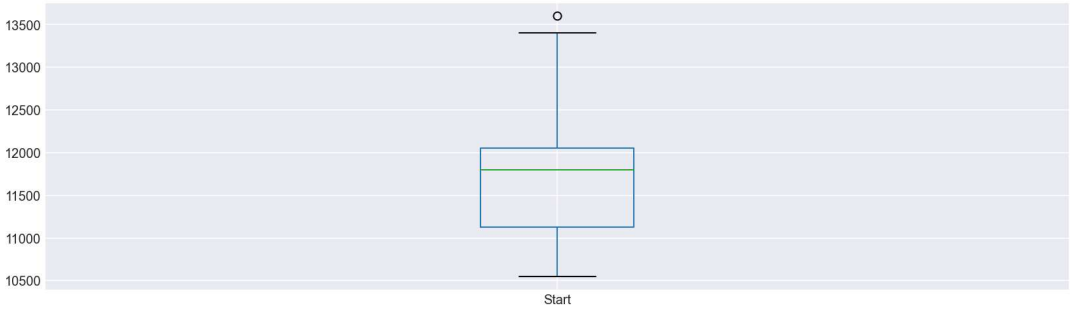
과제 1

Start column의 분포 및 이상치 여부를 탐색하는 Box Plot 및 이상치를 구하세요

In [2]:
```
df = pd.read_csv('../Data/주가데이터.csv')
df['NDate'] = pd.to_datetime(df.Date)
df1 = df.set_index('NDate')
df1.drop(['Date','Volume'],axis=1,inplace=True)
```

In [3]:
```
df1['Start'].plot(kind='box')
df1['Start'].describe()
```

```
count       20.000000
mean     11755.000000
std        865.250192
min      10550.000000
25%      11125.000000
50%      11800.000000
75%      12050.000000
max      13600.000000
Name: Start, dtype: float64
```



In [4]:
```
Q1 = df1['Start'].describe()['25%']
Q3 = df1['Start'].describe()['75%']
IQR = Q3 - Q1
outlier = df[(df['Start'] <= (Q1-IQR*1.5)) | ((df['Start'] >= (Q3+

outlier_2 = []
for i in df['Start']:
    if not Q1 - IQR * 1.5 < i < Q3 + IQR * 1.5:
        outlier_2.append(i)
display(outlier)
print(outlier_2)
```

	Date	Close	Start	High	Low	Volume	NDate
11	2018-06-15	13400	13600	13600	12900	201376	2018-06-15

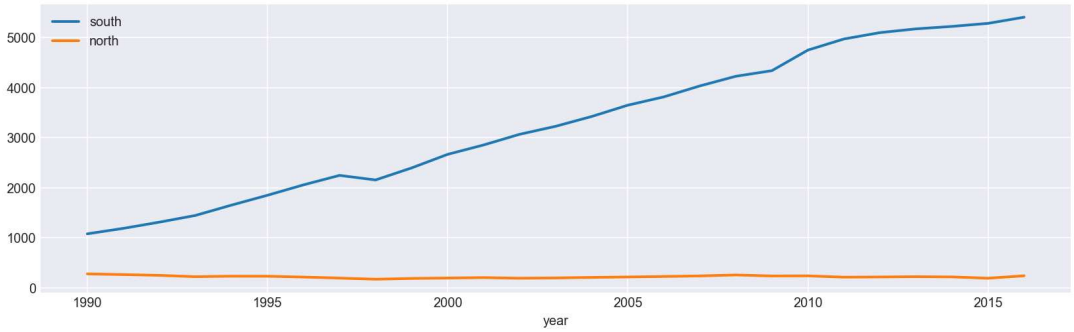
```
[13600]
```

과제2

남북한 발전량 데이터를 시각적으로 탐색하고 그 특징을 요약 기술하세요.

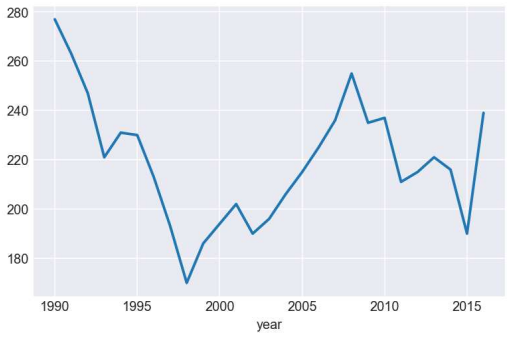
In [5]:
```
df_ep = pd.read_excel('../Data/df_ep.xlsx')
df_ep.set_index('year',inplace=True)
df_ep.plot()
```

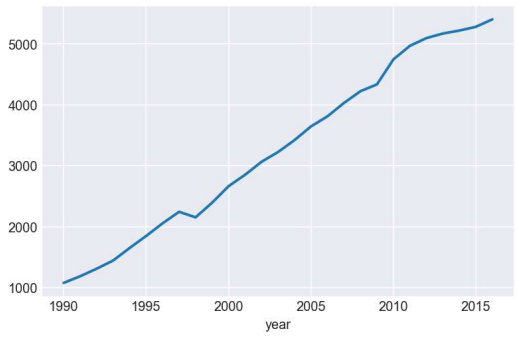
```
<AxesSubplot:xlabel='year'>
```



In [6]:
```
plt.subplot(1,2,1)
df_ep['north'].plot()
plt.subplot(1,2,2)
df_ep['south'].plot()
```

```
<AxesSubplot:xlabel='year'>
```





In [7]:
```
df_ep.describe()
```

	south	north
count	27.000000	27.000000
mean	3278.629630	219.037037
std	1435.906927	25.236545
min	1077.000000	170.000000
25%	2104.000000	199.000000
50%	3225.000000	216.000000
75%	4541.500000	235.500000
max	5404.000000	277.000000

1. 남한과 북한의 전력난은 데이터의 시작, 1990년대부터 크게 차이가 나고 있었다. (설비시설의 차이)
2. 1995~2000년 사이에 북/남측 둘다 전력난이 감소한 상황이 생겼다.
3. 이후 북한의 발전량과 남한의 발전량은 비교적 상승선을 보임.
4. 하지만 2005~2010년 사이 북한의 발전량은 하락함을 보이고
5. 2015년에 또다시 저점에 도달함.

과제

df_auto의 각 컬럼을 시각화해서 탐색한 후 인사이트를 기술하세요

In [8]:
```
import pandas as pd
df_auto = pd.read_excel('../Data/auto-mpg.xlsx')
df_auto.head()
df_auto.sort_values('acceleration',ascending=True)
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin
11	14.0	8	340.0	160	3609	8.0	70	1
9	15.0	8	390.0	190	3850	8.5	70	1
7	14.0	8	440.0	215	4312	8.5	70	1
6	14.0	8	454.0	220	4354	9.0	70	1
116	16.0	8	400.0	230	4278	9.5	73	1
...	...	...	...	...	...	...	...	...
300	23.9	8	260.0	90	3420	22.2	79	1
59	23.0	4	97.0	54	2254	23.5	72	2
326	43.4	4	90.0	48	2335	23.7	80	2
394	44.0	4	97.0	52	2130	24.6	82	2
299	27.2	4	141.0	71	3190	24.8	79	2

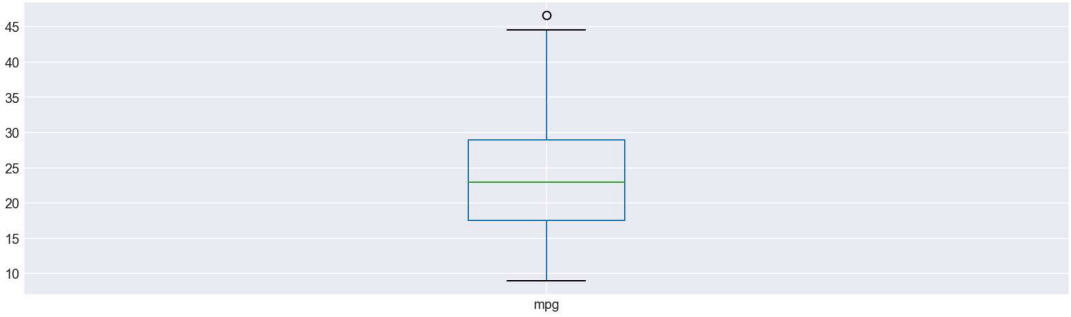
398 rows × 9 columns

In [9]:
```
df_auto.corr()
```

	mpg	cylinders	displacement	weight	acceleration	model year	o
mpg	1.000000	-0.775396	-0.804203	-0.831741	0.420289	0.579267	0.563
cylinders	-0.775396	1.000000	0.950721	0.896017	-0.505419	-0.348746	-0.56
displacement	-0.804203	0.950721	1.000000	0.932824	-0.543684	-0.370164	-0.60
weight	-0.831741	0.896017	0.932824	1.000000	-0.417457	-0.306564	-0.58
acceleration	0.420289	-0.505419	-0.543684	-0.417457	1.000000	0.288137	0.205
model year	0.579267	-0.348746	-0.370164	-0.306564	0.288137	1.000000	0.180
origin	0.563450	-0.562543	-0.609409	-0.581024	0.205873	0.180662	1.000

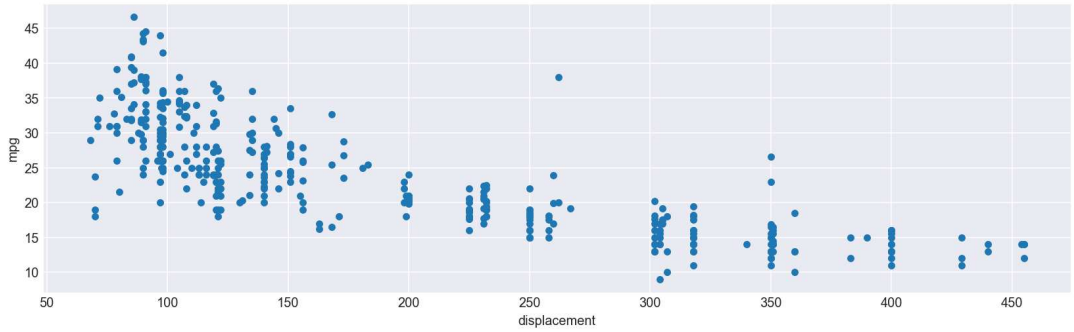
In [10]:
```
df_auto['mpg'].plot(kind='box') #이상치 하나 존재. -이상치 파악
df_auto['mpg'].describe()
```

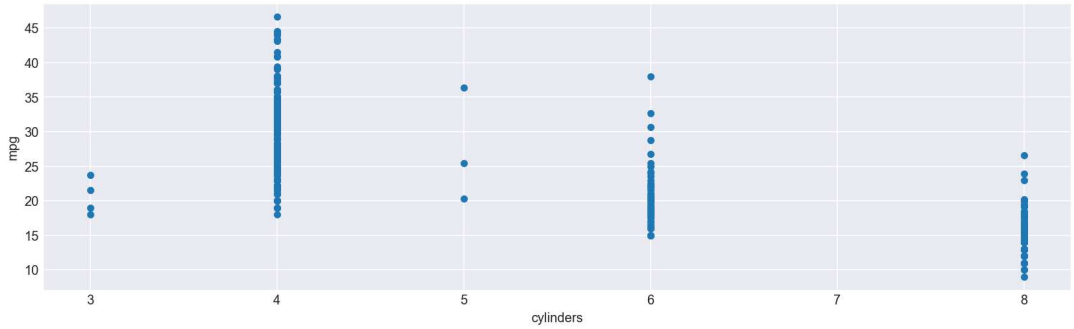
```
count    398.000000
mean      23.514573
std        7.815984
min        9.000000
25%       17.500000
50%       23.000000
75%       29.000000
max       46.600000
Name: mpg, dtype: float64
```



In [11]:
```
df_auto.plot(x='displacement',y='mpg',kind='scatter')
df_auto.plot(x='cylinders',y='mpg',kind='scatter')
```

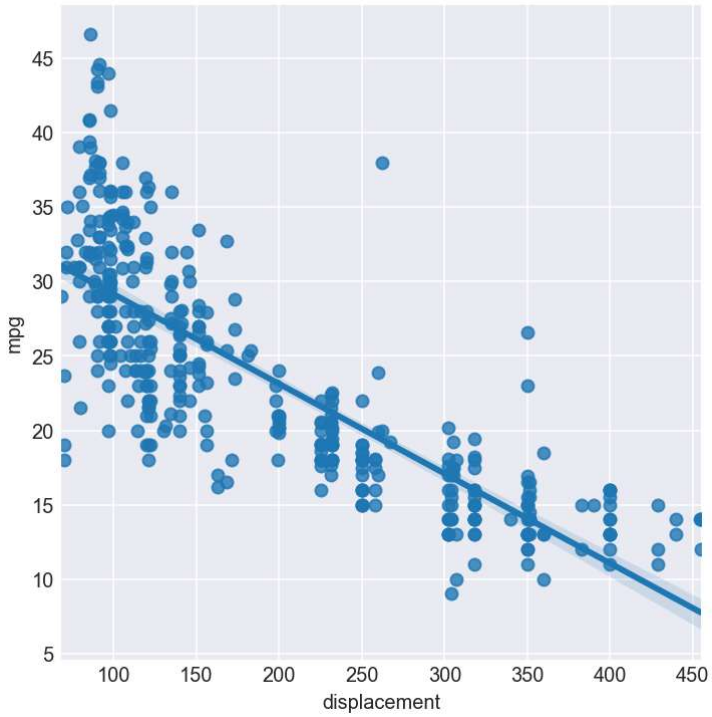
```
<AxesSubplot:xlabel='cylinders', ylabel='mpg'>
```

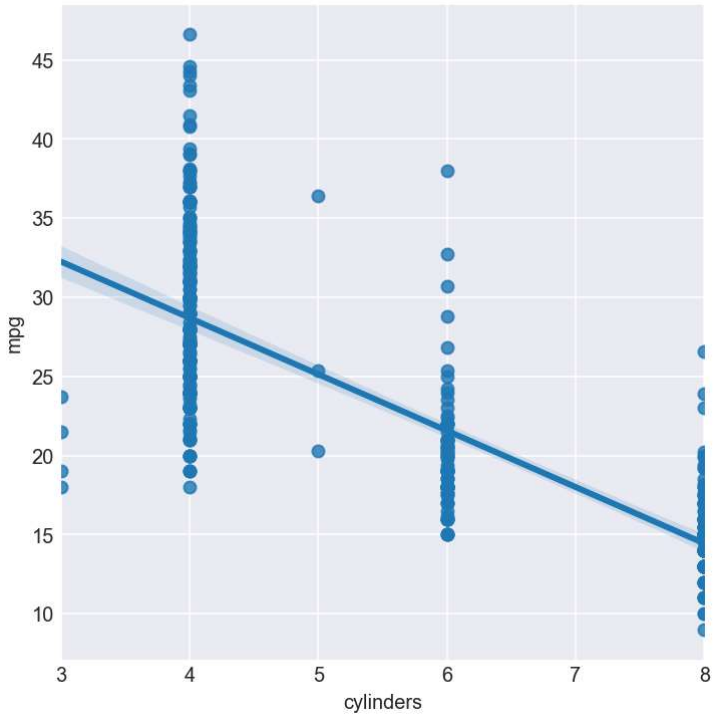




In [12]:
```
sns.lmplot(data=df_auto,x='displacement',y='mpg')
# displacement(배기량)이 높을수록 mpg는 감소함.
sns.lmplot(data=df_auto,x='cylinders',y='mpg')
#displacement(배기량)과 마찬가지로 cylinders(기통수)가 높을수록 연비는
```

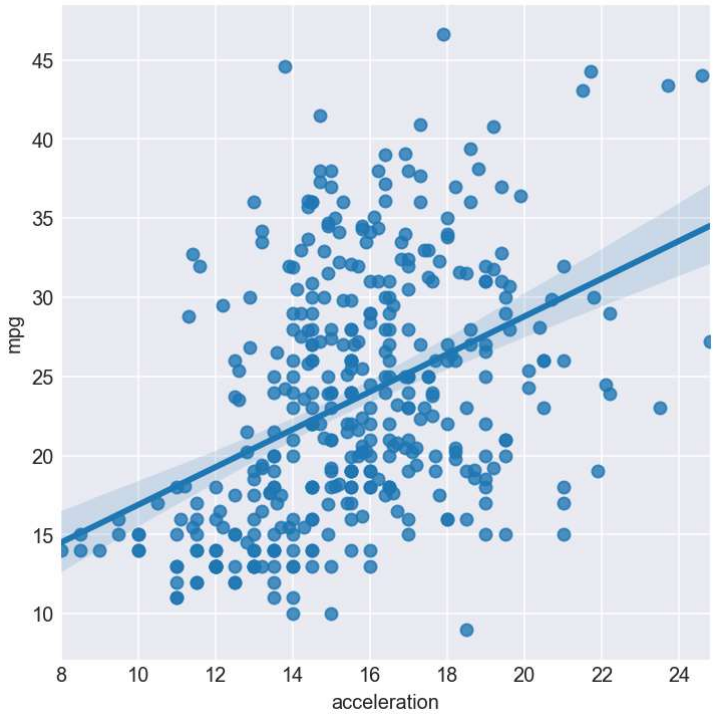
```
<seaborn.axisgrid.FacetGrid at 0x1ab64470be0>
```





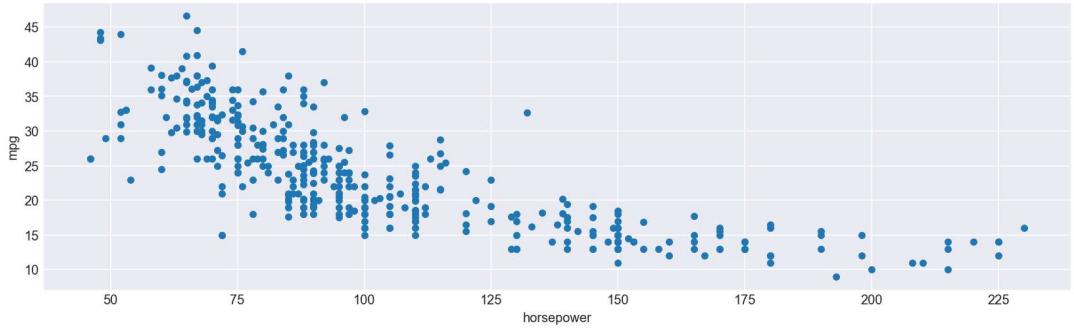
In [13]:
```
sns.lmplot(data=df_auto,x='acceleration',y='mpg')
#연비와 제로백의 시간은 정비례관계를 보인다.
```

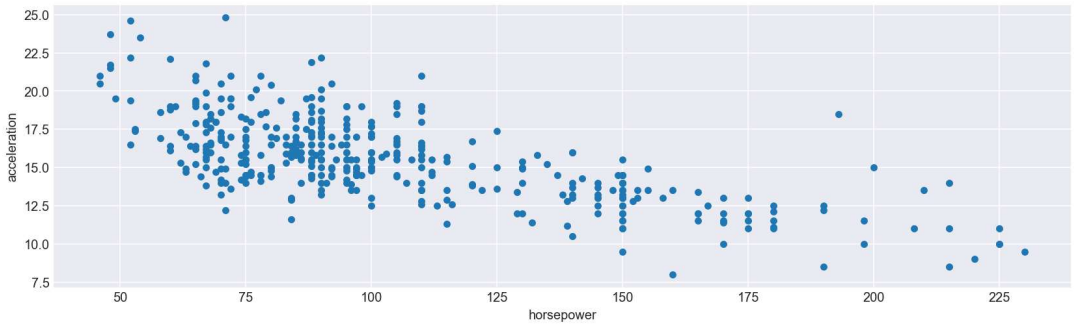
```
<seaborn.axisgrid.FacetGrid at 0x1ab66aa5460>
```

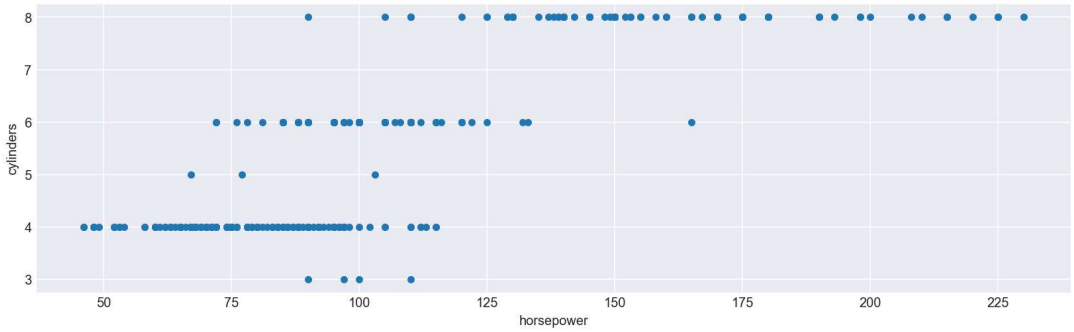


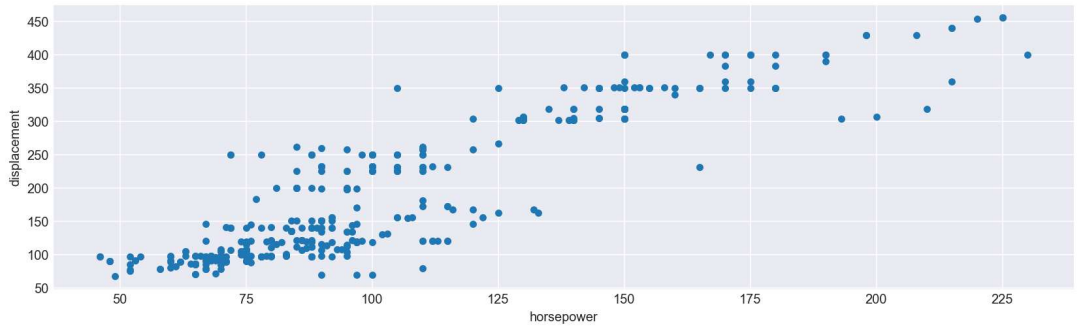
In [14]:
```
# df_auto.plot(x='horsepower',y='mpg',kind='scatter') #error
df_auto['horsepower'] = df_auto['horsepower'].replace('?',np.nan)
df_auto['horsepower'] = df_auto['horsepower'].fillna(method='ffill
df_auto.plot(x='horsepower',y='mpg',kind='scatter')
df_auto.plot(x='horsepower',y='acceleration',kind='scatter')
df_auto.plot(x='horsepower',y='cylinders',kind='scatter')
df_auto.plot(x='horsepower',y='displacement',kind='scatter')
```

```
<AxesSubplot:xlabel='horsepower', ylabel='displacement'>
```



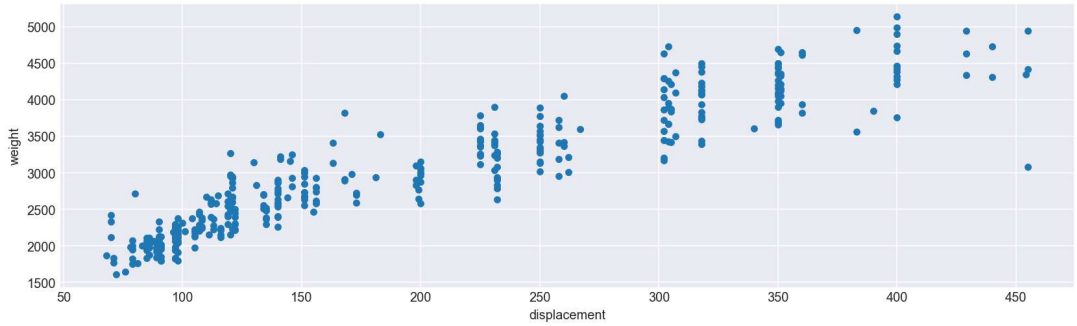


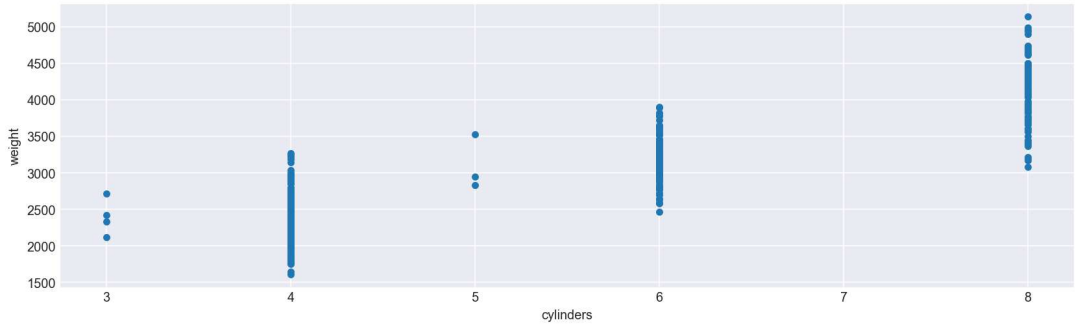


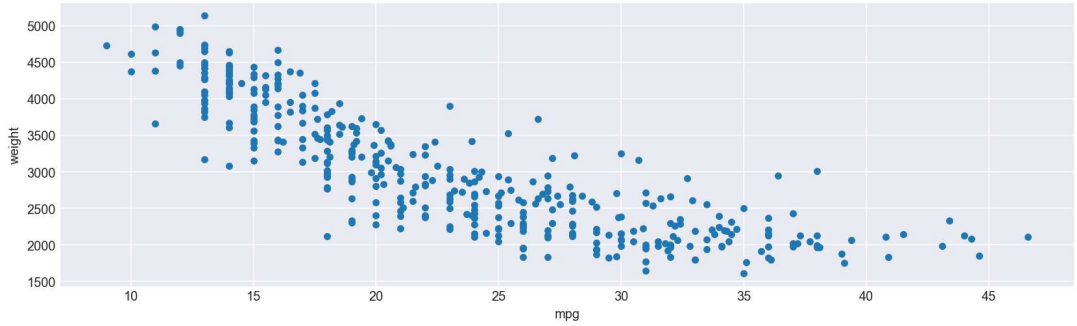

displacement
horsepower
450
400
350
300
250
200
150
100
50
50
75
100
125
150
175
200
225

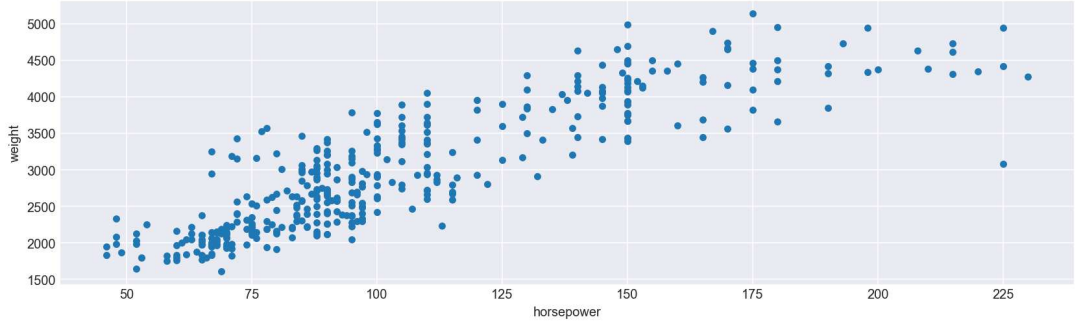
In [15]:
```
df_auto.plot(x='displacement',y='weight',kind='scatter')
df_auto.plot(x='cylinders',y='weight',kind='scatter')
df_auto.plot(x='mpg',y='weight',kind='scatter')
df_auto.plot(x='horsepower',y='weight',kind='scatter')
```

```
<AxesSubplot:xlabel='horsepower', ylabel='weight'>
```







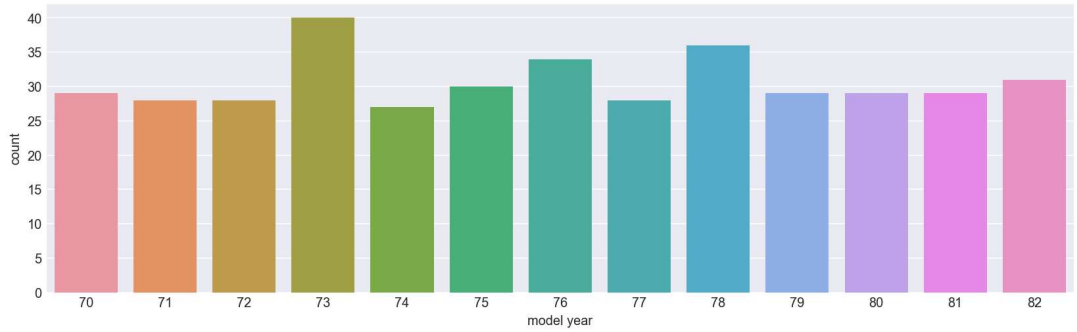


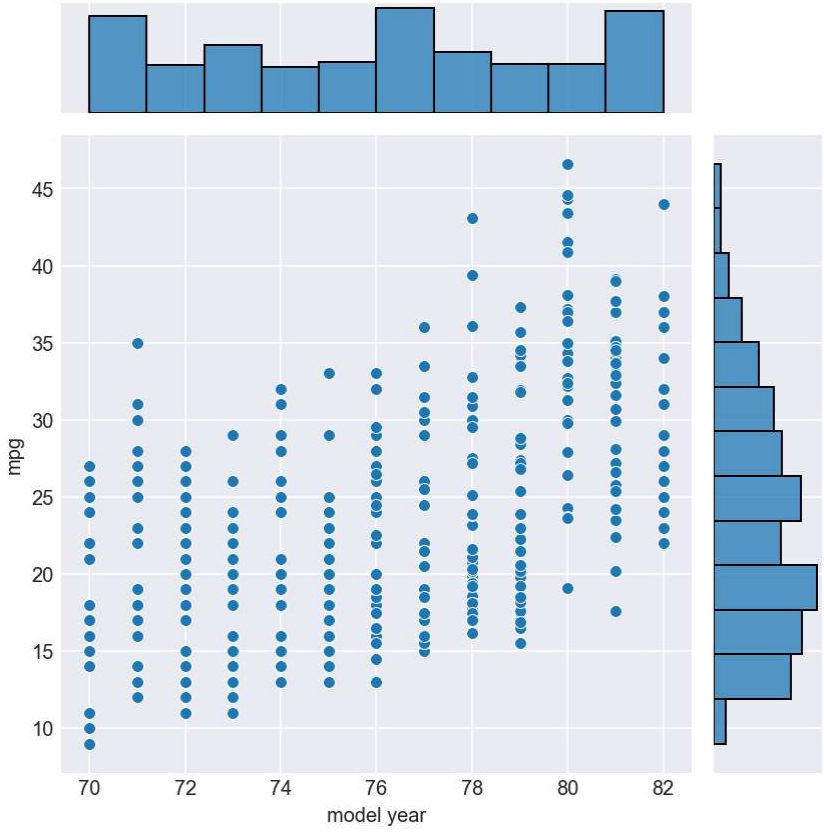
In [16]:
```
df_auto.columns
```

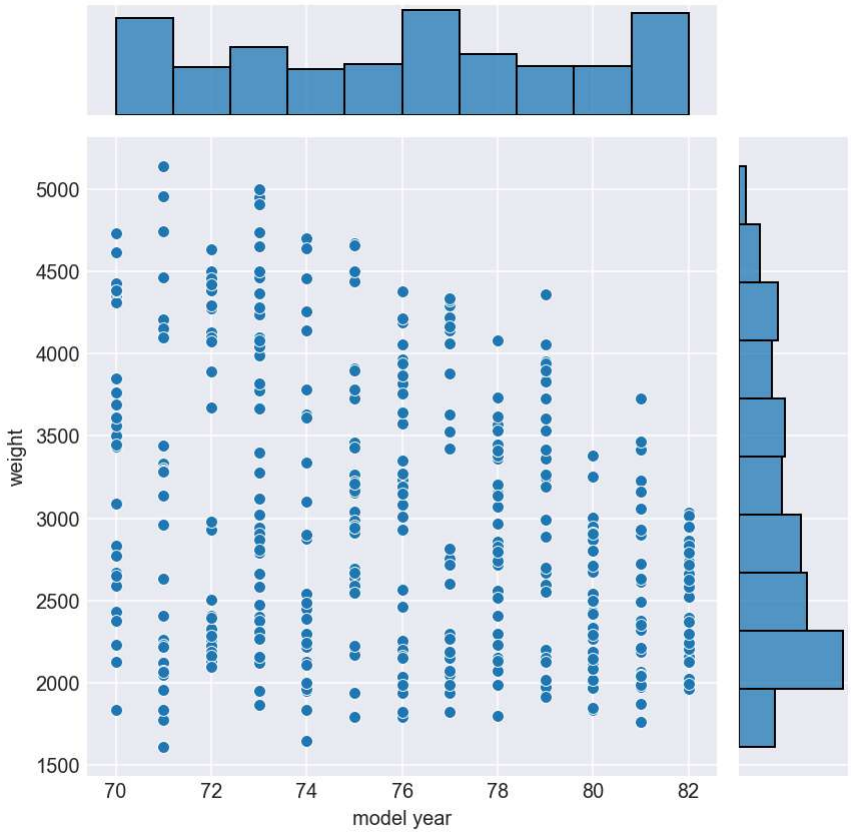
```
Index(['mpg', 'cylinders', 'displacement', 'horsepower', 'weight',
       'acceleration', 'model year', 'origin', 'car name'],
      dtype='object')
```

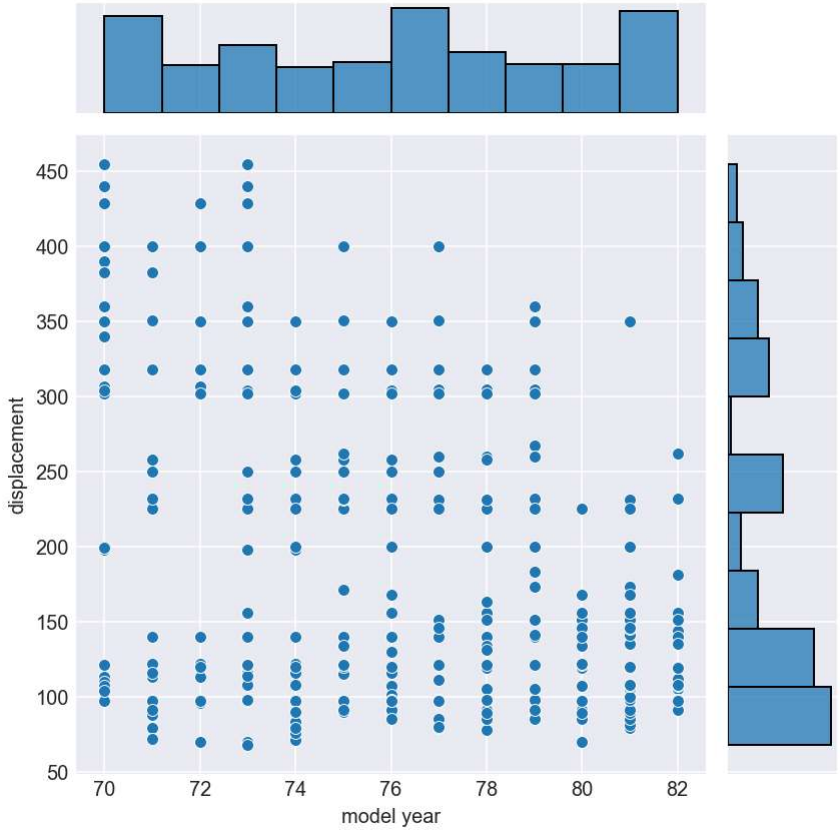
In [17]:
```
sns.countplot(x='model year',data=df_auto)
sns.jointplot(x='model year',y='mpg',data=df_auto)
sns.jointplot(x='model year',y='weight',data=df_auto)
sns.jointplot(x='model year',y='displacement',data=df_auto)
```

```
<seaborn.axisgrid.JointGrid at 0x1ab67c417c0>
```





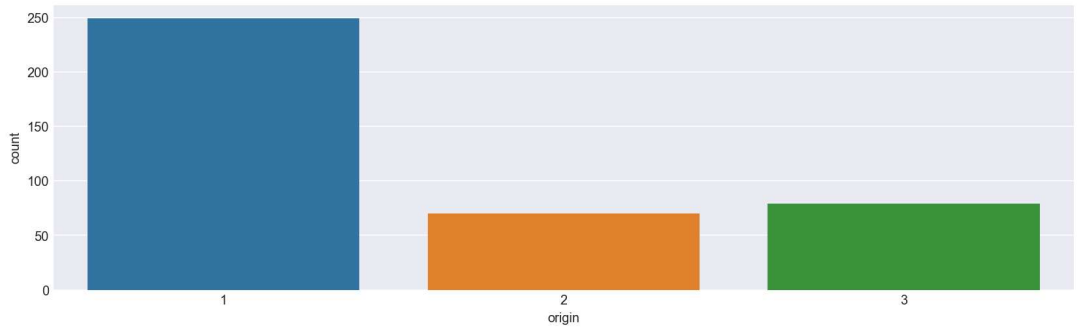

weight
model year
5000
4500
4000
3500
3000
2500
2000
1500
70
72
74
76
78
80
82

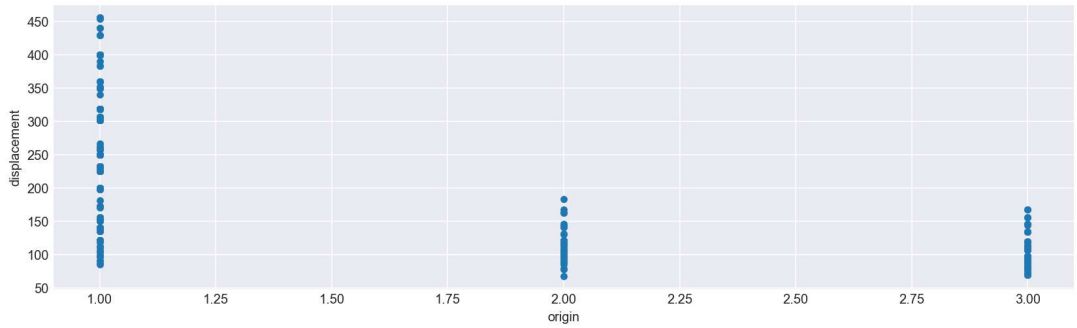

displacement
model year
450
400
350
300
250
200
150
100
50
70
72
74
76
78
80
82

In [18]:
```
sns.countplot(df_auto['origin'])
df_auto.plot(x='origin',y='displacement',kind='scatter')
```

```
C:\Users\kikir\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='origin', ylabel='displacement'>
```






```
# domain
displacement = 배기량 ,CC
cylinders = 기통 (몇 기통)
acceleration = zero-100

# 인사이트
1. displacement 와 cylinder 의 관계는 높다. 결국 mpg와의 관계가 깊음. (연비)
```

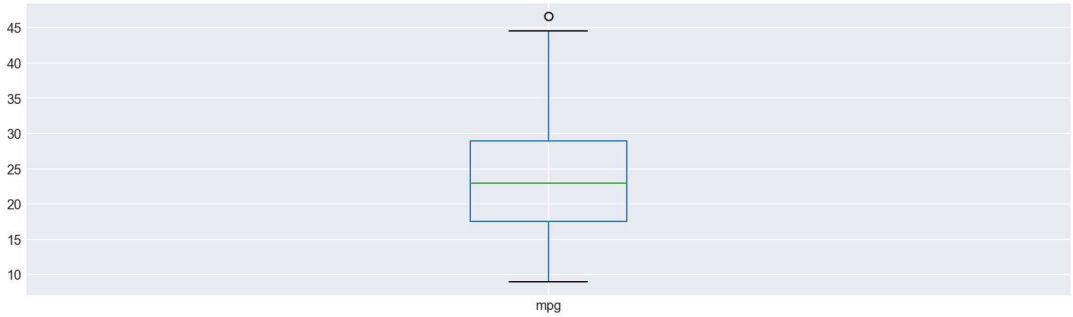
따라서 배기량이 높고, 기통 수가 높을 수록 연비가 낮고, 연비가 낮으면 제로백의 시간은 짧아진다.
2. horsepower, 마력이 높다는 것은 배기량이 높고, 기통 수가 높다는 것을 의미한다. 1과 마찬가지로 마력이 높다는 것은
연비, mpg가 낮을 가능성이 높다.
3. 이와 반대로, displacement와 cylinder의 값이 높을수록 horsepower의 값은 증가하지만 이와 반대로 weight가 증가하는 것을 보인다.
이는 고출력을 내야하는 엔진은 높은 배기량과 6기통 이상으로 높은 마력을 출력하지만 차체 무게가 무거워져 결국 연비를
저하시키는 요인이 된다는 것 을 알 수 있다.
4. 또한 연비, mpg 는 차량의 생산년도에 따라서 점차 좋아지는것을 확인 할 수 있다. 이는 엔진의 무게를 줄였다는것으로 의미하는데,
70년대에 비해 82년대에 고출력 엔진보다 상대적으로 낮은 엔진을 생산했기 때문이라고 생각된다.
5. origin의 경우에는 생산지를 의미하고, 생산지의 1은 미국, 2는 유럽, 3은 일본을 의미한다. 그래프를 보면, 1 에 고출력 자동차를 많이 생산되는 것을 확인할 수 있다. 이를 미루어
보면, 고출력 자동차의 엔진은 엔진 기술의 정점에 위치하여 있고, 기술에 대해서 고부가가치 산업을 중요시한 미국이 이를 타국 공장이 아닌 자국 생산만 가능하게 하여 기술 유출방지를
했다고 생각된다. 미국의 자동차 변천사에 비해 유럽과 일본은

과제

mpg컬럼의 이상치를 구하세요

In [19]:
```
df_auto['mpg'].plot(kind='box') #이상치 하나 존재. -이상치 파악
df_auto['mpg'].describe()
```

```
count    398.000000
mean      23.514573
std        7.815984
min        9.000000
25%       17.500000
50%       23.000000
75%       29.000000
max       46.600000
Name: mpg, dtype: float64
```



In [26]:

	mpg
0	18.0
1	15.0
2	18.0
3	16.0
4	17.0
...	...
393	27.0
394	44.0
395	32.0
396	28.0
397	31.0

398 rows × 1 columns

In [63]:
```
Q1 = df_auto['mpg'].describe()['25%']
Q3 = df_auto['mpg'].describe()['75%']
IQR = Q3 - Q1
outlier = float(df_auto[(df_auto['mpg'] <= (Q1-IQR*1.5)) | ((df_au
outlier

lower_whisker = df_auto[df_auto['mpg'] > (Q1-IQR*1.5)]['mpg'].min(
upper_whisker = df_auto[df_auto['mpg'] < (Q3+IQR*1.5)]['mpg'].max(

print('outlier >',outlier)
print('upper_whisker > ',upper_whisker)
print('IQR > ',IQR)
print('lower_whisker > ',lower_whisker)
```

```
outlier > 46.6
upper_whisker >  44.6
IQR >  11.5
lower_whisker >  9.0
```

과제

통계학의 기초 개념을 설명하세요

[과제] 통계학의 기초 개념을 설명하세요.

모수의 개념 및 사례
통계량의 개념 및 사례
확률변수, 확률, 확률분포
도수, 도수분포, 상대도수
평균값, 기대값, 분산, 표준편차
확률질량함수, 확률밀도함수
정규분포, 이항분포, 포아송분포
표본분산(n), 불편분산(n-1)
기술통계, 추측통계
가설과 검정

모수의 개념 및 사례

모수 : 관심을 갖고 있는 모집단 관측치의 대표값. 즉, 집단을 대표하는 값들을 의미한다. 이는 평균/산포도 등으로 표시.
모수 사례 : 국가 나이 분포, 학과 성적 분포 등등.

통계량의 개념 및 사례

통계량 : 표본의 대푯값. (모수와 다른점). 마찬가지로 평균 분산 등등으로 표시함.
통계량 사례 : 표본들의 최댓값 등등

확률변수, 확률, 확률분포

확률 : 사건 A가 일어날 가능성.
확률변수 : 특정 확률로 발생하는 각각의 결과를 의미함.
확률분포 : 확률 변수가 취할수 있는 모든 값이 나타날 확률. (합계 = 1.0)

도수, 도수분포, 상대도수

도수 : 각 계급에 속하는 자료의 개수 주사위: 1,2,3,4,5,6
도수분포 : 측정값을 몇 개의 계급으로 나누고 각 계급에 속하는 수치의 빈도.
상대도수 : 총 도수에 대한 도수의 비율.

평균값, 기대값, 분산, 표준편차

평균값 : 데이터의 중심을 나타냄
기대값 : 각 사건이 벌어졌을때의 이득과 그 사건이 벌어진 확률을 곱한 것을 전체 사건에 대해 합한 값.
확률적 사건에 대한 평균
분산 : 평균에 대한 편차 제곱의 평균 . 데이터가 얼마나 넓게 퍼져있는지 나타내는 값. 산포도.
표준편차 : 분산의 크기를 일정하게 만들어 줘서 비교가 가능하게 함.

확률질량함수, 확률밀도함수

확률질량함수 : 확률변수가 취할 수 있는 값이 유한개 혹은 자연수와 같이 셀 수 있을때, 불연속한 값에 대한 확률. (아상형 확률변수)
--> 확률변수에 속한 변량들이 서로 떨어져 분리 된 것.

확률밀도함수 : 확률 변수의 분포를 나타내는 함수.(연속확률변수)
--> 확률 변수에 속한 변량들이 서로 끊어지지 않고 연결되어 있는것.
변량이 무수히 많아 셀수 없다는 특징이 있음.

> 정규분포, 이항분포, 포아송분포

정규분포 : 평균과 분산에 의해ㅐ 분포가 확정. 대표적인 연속 확률분포
평균이 확률밀도함수의 최빈값과 일치. 평균을 중심으로 좌우대칭을 이뤄
평균과 중앙값이 일치.
확률 밀도 함수가 모든 실수에 대해 0보다 크지만, 평균으로 부터 멀어
지면서 그 값이 급격하게 작아짐.
이항분포, 연속된 n번의 독립적 시행에서 각 시행이 확률 p를 가질 떄에

In [ ]:

In [ ]:

민아님 등판

기술통계 : 수집한 표본의 통계량을 기술, 설명하는 통계기법.
추측통계 : 표본에서 추출한 표본 통계량을 기반으로 모집단의 모수를
예측하는 통계기법.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: