wine.csv 파일로부터 데이터 불러오기

info()함수로 기본 정보 불러오기

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#파일로부터 데이터 불러오기
file='../data/wine.csv'
wine_data=pd.read_csv(file)
wine_data.info()#info()함수로 기본정보 불러오기
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   type                  6497 non-null   object 
 1   fixed acidity         6497 non-null   float64
 2   volatile acidity      6497 non-null   float64
 3   citric acid           6497 non-null   float64
 4   residual sugar        6497 non-null   float64
 5   chlorides             6497 non-null   float64
 6   free sulfur dioxide   6497 non-null   float64
 7   total sulfur dioxide  6497 non-null   float64
 8   density               6497 non-null   float64
 9   pH                    6497 non-null   float64
 10  sulphates             6497 non-null   float64
 11  alcohol               6497 non-null   float64
 12  quality               6497 non-null   int64  
dtypes: float64(11), int64(1), object(1)
memory usage: 660.0+ KB
```

In [2]:

```
wine_data
```

Out[2]:

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulpha
0	red	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0
1	red	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0
2	red	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0
3	red	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0
4	red	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0
...	...	...	...	...	...	...	...	...	...	...	
6492	white	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3.27	0
6493	white	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3.15	0
6494	white	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2.99	0
6495	white	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3.34	0
6496	white	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3.26	0

6497 rows × 13 columns

함수를 사용해 수치통계(8개) 구하기

In [3]:

```
wine_des=wine_data.describe()
wine_des
```

Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total su dio:
count	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000000	6497.000
mean	7.215307	0.339666	0.318633	5.443235	0.056034	30.525319	115.744
std	1.296434	0.164636	0.145318	4.757804	0.035034	17.749400	56.521
min	3.800000	0.080000	0.000000	0.600000	0.009000	1.000000	6.000
25%	6.400000	0.230000	0.250000	1.800000	0.038000	17.000000	77.000
50%	7.000000	0.290000	0.310000	3.000000	0.047000	29.000000	118.000
75%	7.700000	0.400000	0.390000	8.100000	0.065000	41.000000	156.000
max	15.900000	1.580000	1.660000	65.800000	0.611000	289.000000	440.000

In [4]:

```
wine_des.agg(['mean','min','max'])
```

Out[4]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfu dioxid
mean	818.288968	812.510538	812.509244	823.31263	812.232633	865.28434	933.28330
min	1.296434	0.080000	0.000000	0.60000	0.009000	1.00000	6.00000
max	6497.000000	6497.000000	6497.000000	6497.00000	6497.000000	6497.00000	6497.00000

와인의 품질 등급 단계 알아보기

와인의 품질 등급 별 빈도수 구하기

In [5]:

```
wine_quality=wine_data.groupby('quality')
wine_quality.count()['type']
```

Out[5]:

```
quality
3      30
4     216
5    2138
6    2836
7    1079
8     193
9       5
Name: type, dtype: int64
```

가장 빈도수가 많은 품질 등급 구하기

In [6]:

```
wine_quality.count()['type'].idxmax()
```

Out[6]:

```
6
```

In [69]:

```
wdq=wine_data.groupby("type")["quality"]
wdq
```

Out[69]:

```
<pandas.core.groupby.generic.SeriesGroupBy object at 0x0000028676CB2C10>
```

In [70]:

```
result=wdq.agg(['mean','min','max'])
result
```

Out[70]:

	mean	min	max
type			
red	5.636023	3	8
white	5.877909	3	9

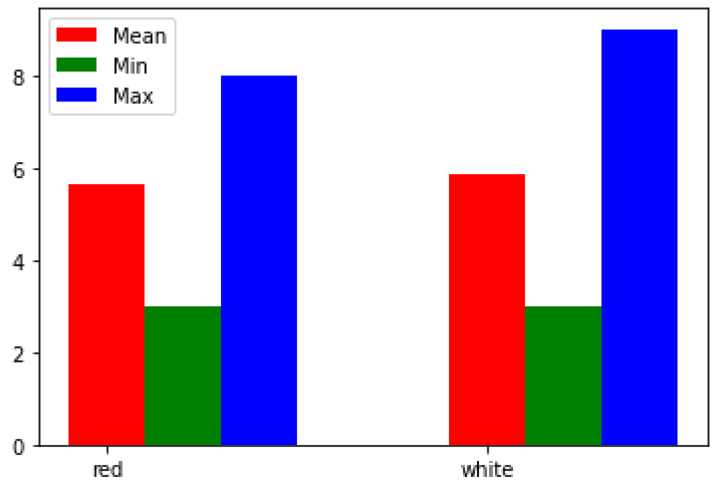
In [71]:

```
w_groups=len(result.index)
means=result['mean'].tolist()
mins=result['min'].tolist()
maxs=result['max'].tolist()

print(means)
print(mins)
print(maxs)
```

```
[5.6360225140712945, 5.87790935075541]
[3, 3]
[8, 9]
```

In [79]:

```
index=np.arange(w_groups)
bar_width=0.2
#평균값에 대한 그래프 생성
rects1=plt.bar(index,means,bar_width,color='r',label='Mean') #각 bar를 설정함.
rects2=plt.bar(index+bar_width,mins,bar_width,color='g',label='Min')
rects3=plt.bar(index+bar_width*2,maxs,bar_width,color='b',label='Max')

plt.xticks(index,result.index.tolist())
#plt.axis([0,2,0,10])#axis([xmin,xmax,ymin,ymax])
plt.legend() ##범례표시(범례 : 참고사항)
plt.show()
```



유형 별로 품질 등급별 수치 통계구하기

유형에 따른 품질 등급 시각화 하기.

In [49]:

```
red_wine=[] ##red_wine배열에 for문으로 돌면서 아래 조건을 반복하여 저장.

for i in range(3,10,1):
    red_wine.append(wine_data.loc[(wine_data['type']=='red')&(wine_data['quality']==i)])
                                        ##type이 red이고 quality가 같은 rows를 red_wine에 저장.

#r3=wine_data.loc[(wine_data['type']=='red')&(wine_data['quality']==3)]
#r4=wine_data.loc[(wine_data['type']=='red')&(wine_data['quality']==4)]
#r5=wine_data.loc[(wine_data['type']=='red')&(wine_data['quality']==5)]
#r6=wine_data.loc[(wine_data['type']=='red')&(wine_data['quality']==6)]
#r7=wine_data.loc[(wine_data['type']=='red')&(wine_data['quality']==7)]
#r8=wine_data.loc[(wine_data['type']=='red')&(wine_data['quality']==8)]
#r9=wine_data.loc[(wine_data['type']=='red')&(wine_data['quality']==9)]

red_wine[0]
```

Out[49]:

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphat
459	red	11.6	0.580	0.66	2.20	0.074	10.0	47.0	1.00080	3.25	0.
517	red	10.4	0.610	0.49	2.10	0.200	5.0	16.0	0.99940	3.16	0.
690	red	7.4	1.185	0.00	4.25	0.097	5.0	14.0	0.99660	3.63	0.
832	red	10.4	0.440	0.42	1.50	0.145	34.0	48.0	0.99832	3.38	0.
899	red	8.3	1.020	0.02	3.40	0.084	6.0	11.0	0.99892	3.48	0.
1299	red	7.6	1.580	0.00	2.10	0.137	5.0	9.0	0.99476	3.50	0.
1374	red	6.8	0.815	0.00	1.20	0.267	16.0	29.0	0.99471	3.32	0.
1469	red	7.3	0.980	0.05	2.10	0.061	20.0	49.0	0.99705	3.31	0.
1478	red	7.1	0.875	0.05	5.70	0.082	3.0	14.0	0.99808	3.40	0.
1505	red	6.7	0.760	0.02	1.80	0.078	6.0	12.0	0.99600	3.55	0.

In [52]:

```
white_wine=[] ##red_wine배열에 for문으로 돌면서 아래 조건을 반복하여 저장.
for i in range(3,10,1):
    white_wine.append(wine_data.loc[(wine_data['type']=='white')&(wine_data['quality']==i)])
                                                ##type이 white인 행에서 quality가 같은 rows를 red_wine에 저장.
white_wine[0]
```

Out[52]:

	type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulpha
1850	white	8.5	0.260	0.21	16.20	0.074	41.0	197.0	0.99800	3.02	C
1852	white	5.8	0.240	0.44	3.50	0.029	5.0	109.0	0.99130	3.53	C
1893	white	9.1	0.590	0.38	1.60	0.066	34.0	182.0	0.99680	3.23	C
2044	white	7.1	0.320	0.32	11.00	0.038	16.0	66.0	0.99370	3.24	C
2339	white	6.9	0.390	0.40	4.60	0.022	5.0	19.0	0.99150	3.31	C
2472	white	10.3	0.170	0.47	1.40	0.037	5.0	33.0	0.99390	2.89	C
2633	white	7.9	0.640	0.46	10.60	0.244	33.0	227.0	0.99830	2.87	C
2828	white	8.3	0.330	0.42	1.15	0.033	18.0	96.0	0.99110	3.20	C
3016	white	8.6	0.550	0.35	15.55	0.057	35.5	366.5	1.00010	3.04	C
3083	white	7.5	0.320	0.24	4.60	0.053	8.0	134.0	0.99580	3.14	C
3287	white	6.7	0.250	0.26	1.55	0.041	118.5	216.0	0.99490	3.55	C
3530	white	7.1	0.490	0.22	2.00	0.047	146.5	307.5	0.99240	3.24	C
3649	white	11.8	0.230	0.38	11.10	0.034	15.0	123.0	0.99970	2.93	C
3972	white	7.6	0.480	0.37	1.20	0.034	5.0	57.0	0.99256	3.05	C
4686	white	6.1	0.200	0.34	9.50	0.041	38.0	201.0	0.99500	3.14	C
4864	white	4.2	0.215	0.23	5.10	0.041	64.0	157.0	0.99688	3.42	C
4906	white	9.4	0.240	0.29	8.50	0.037	124.0	208.0	0.99395	2.90	C
5008	white	6.2	0.230	0.35	0.70	0.051	24.0	111.0	0.99160	3.37	C
5409	white	6.8	0.260	0.34	15.10	0.060	42.0	162.0	0.99705	3.24	C
6344	white	6.1	0.260	0.25	2.90	0.047	289.0	440.0	0.99314	3.44	C

In [86]:

```
print(red_wine[0].count()['type'])
print(white_wine[0].count()['type'])
```

```
10
20
```

In [ ]:

```
```