In [2]:
```
import pandas as pd
df1 = pd.read_csv('Medical_Image_Data_01.csv', encoding='cp949')
df2 = pd.read_csv('Patient_Diagnosis_Data.csv')
df3 = pd.read_csv('Patient_Surgery_Data.csv')
df1.isnull().sum()
```

Out[2]:
```
환자ID                0
전방디스크높이(mm)          0
후방디스크높이(mm)          0
지방축적도               3
Instability         0
MF + ES             0
Modic change        0
PI                  4
PT                  4
Seg Angle(raw)      1
Vaccum disc         0
골밀도               896
디스크단면적               1
디스크위치               0
척추이동척도               0
척추전방위증               0
dtype: int64
```

In [3]:
```
df2.isnull().sum()
```

Out[3]:
```
환자ID                      0
Large Lymphocyte            0
Location of herniation      0
ODI                      1432
가족력                        51
간질성폐질환                     0
고혈압여부                      0
과거수술횟수                     0
당뇨여부                       0
말초동맥질환여부                   0
빈혈여부                       0
성별                         0
스테로이드치료                    0
신부전여부                      0
신장                         0
심혈관질환                      0
암발병여부                      0
연령                         0
우울증여부                      0
입원기간                       0
입원일자                       0
종양진행여부                     0
직업                       415
체중                         0
퇴원일자                       0
헤모글로빈수치                    1
혈전합병증여부                    0
```

```
환자통증정도                  0
흡연여부                    0
통증기간(월)                  4
dtype: int64
```

In [4]:
```
df3.isnull().sum()
```

Out[4]:
```
환자ID          0
수술기법         81
수술시간         54
수술실패여부        0
수술일자          0
신장            0
연령            0
입원일자          0
재발여부          0
체중            0
퇴원일자          0
헤모글로빈수치       1
환자통증정도        0
통증기간(월)       4
혈액형           0
dtype: int64
```

In [5]:
```
import matplotlib.pyplot as plt
import matplotlib

plt.rc('font', family='NanumBarunGothic')
matplotlib.rc('axes',unicode_minus=False)
```

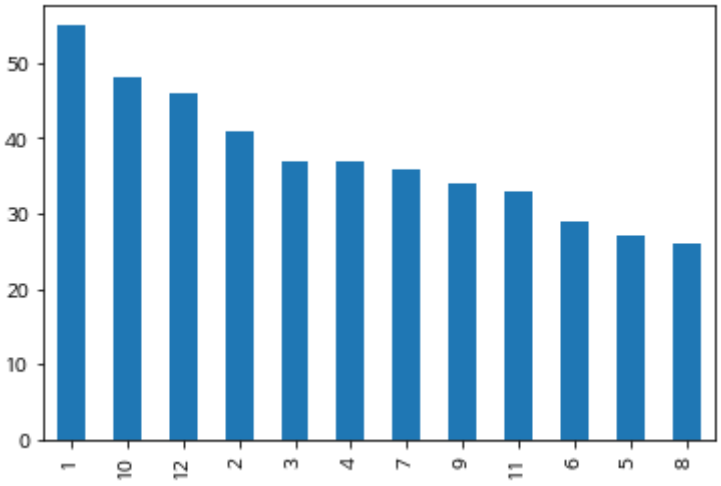
In [6]:
```
merge1 = pd.merge(df1, df2, on='환자ID', how='inner')
final = pd.merge(merge1, df3, on=['환자ID','연령', '입원일자', '신장', '체중', '퇴원일자', '헤모
final.columns
```

Out[6]:
```
Index(['환자ID', '전방디스크높이(mm)', '후방디스크높이(mm)', '지방축적도', 'Instability', 'MF + ES',
       'Modic change', 'PI', 'PT', 'Seg Angle(raw)', 'Vaccum disc', '골밀도',
       '디스크단면적', '디스크위치', '척추이동척도', '척추전방위증', 'Large Lymphocyte',
       'Location of herniation', 'ODI', '가족력', '간질성폐질환', '고혈압여부', '과거수술횟수',
       '당뇨여부', '말초동맥질환여부', '빈혈여부', '성별', '스테로이드치료', '신부전여부', '신장', '심혈관질환',
       '암발병여부', '연령', '우울증여부', '입원기간', '입원일자', '종양진행여부', '직업', '체중', '퇴원일자',
       '헤모글로빈수치', '혈전합병증여부', '환자통증정도', '흡연여부', '통증기간(월)', '수술기법', '수술시간',
       '수술실패여부', '수술일자', '재발여부', '혈액형'],
      dtype='object')
```

In [68]:
```
from datetime import datetime, timedelta
def 요일(date_time):
    s = str(date_time)
    days = ['월','화','수','목','금','토','일']
    date = int(s[4:6])#year=int(s[0:4]), month=int(s[4:6]), day=int(s[6:8]))
    return date#days[date.weekday()]
```

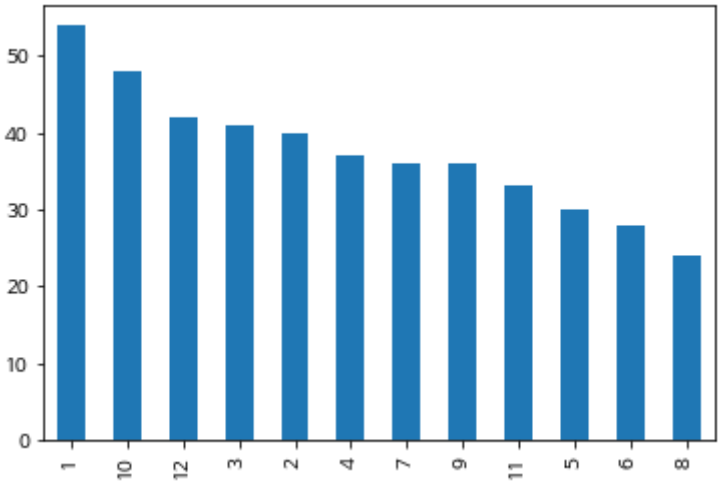
In [69]:
```
# final['입원일자'].apply(요일)
pd.value_counts(final[final['직업']=='사무직']['입원일자'].apply(요일).values).plot.bar()
```

Out[69]: <AxesSubplot:>



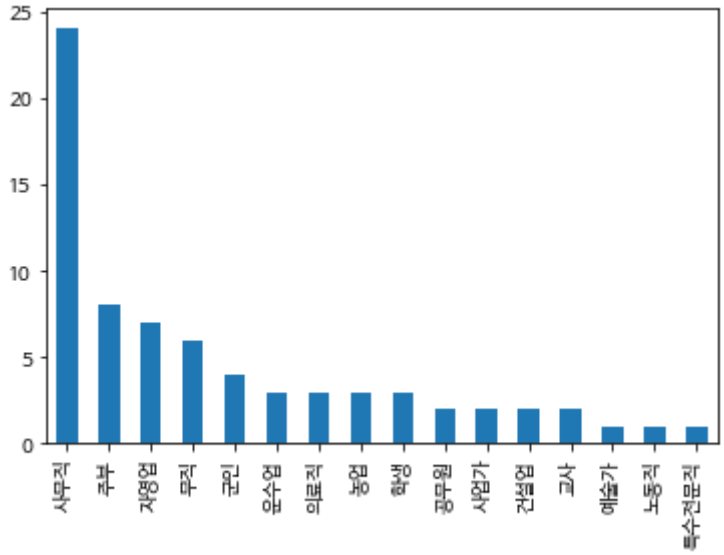
In [70]:
```
# final['입원일자'].apply(요일)
pd.value_counts(final[final['직업']=='사무직']['수술일자'].apply(요일).values).plot.bar()
```

Out[70]: <AxesSubplot:>



In [71]:
```
pd.value_counts(final[final['수술일자'].apply(요일)==8]['직업']).plot.bar()
```

Out[71]: <AxesSubplot:>



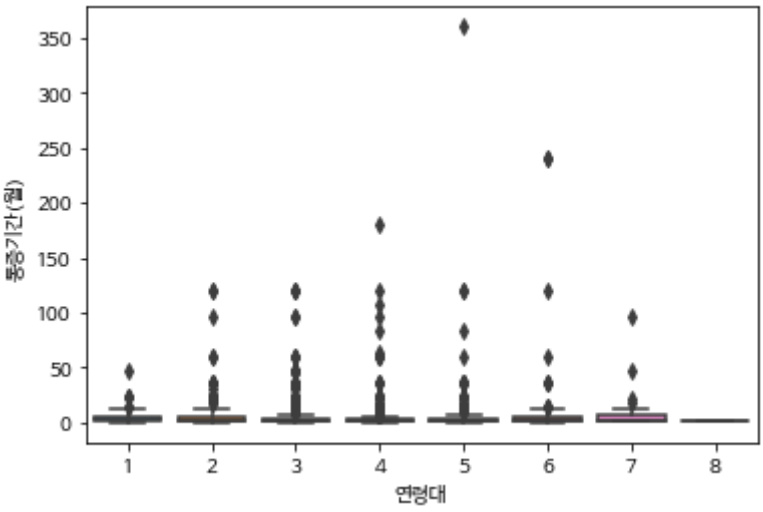
In [72]:
```
type(final['입원일자'][0])
```

Out[72]: numpy.int64

In [73]:
```
# 연령별 수술시간 비교 -> 효과적인 수술 시간을 배치를 해여 의사의 피로도를 줄여 의료서비스 개선? 인
# 재발여부 직업군에 따라 / 기타 특징들
```

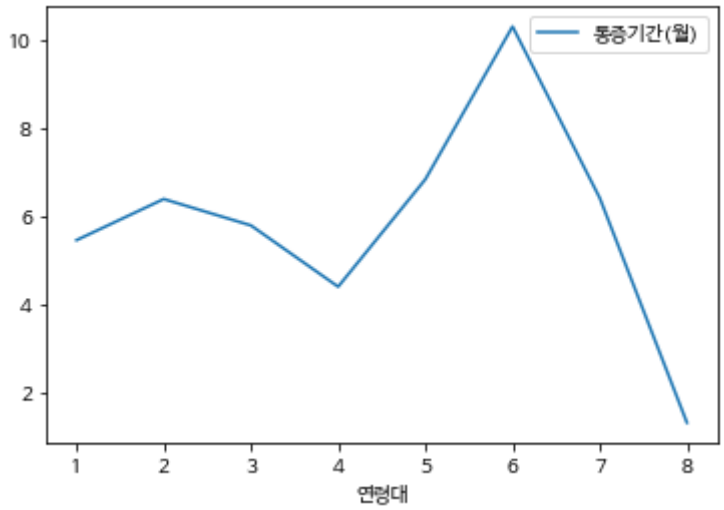
In [74]:
```
import seaborn as sns

# 연령대 통증기간 긴지? -> 참아온것 이를 개선할 수 있는가?
def 연령(age):
  return age//10

final['연령대'] = final['연령'].apply(연령)

sns.boxplot(x='연령대', y='통증기간(월)', data=final)
```

Out[74]: <AxesSubplot:xlabel='연령대', ylabel='통증기간(월)'>



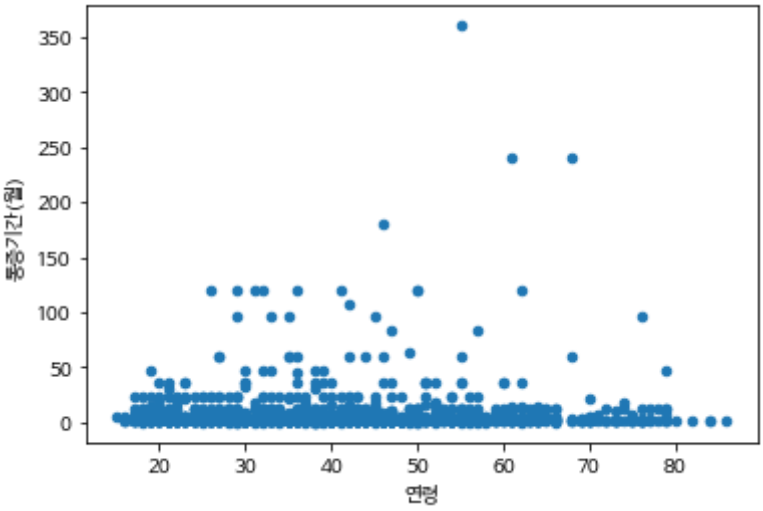
In [75]:
```
final[['통증기간(월)','연령대']].groupby('연령대').mean().plot.line()
```

Out[75]: <AxesSubplot:xlabel='연령대'>



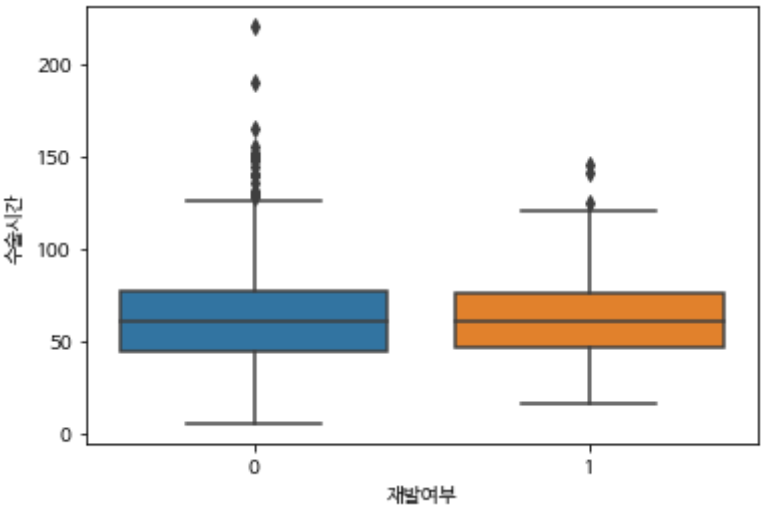
In [76]:
```
final.plot.scatter(x='연령',y='통증기간(월)')
```

Out[76]: <AxesSubplot:xlabel='연령', ylabel='통증기간(월)'>



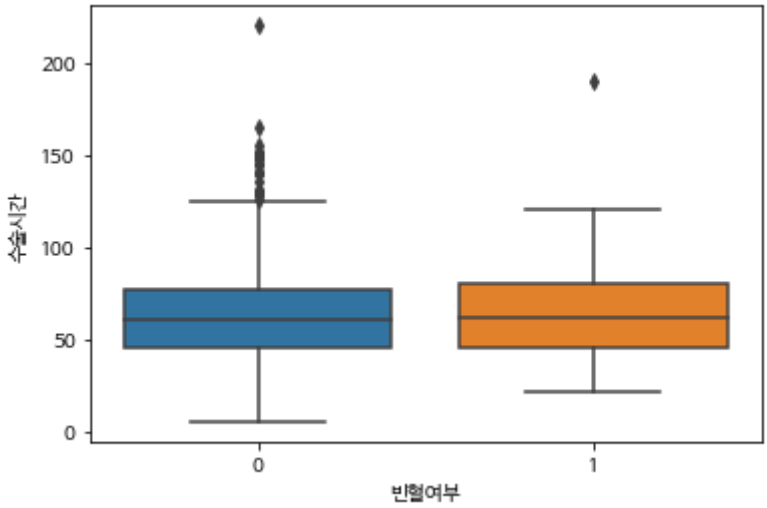
In [77]:
```
sns.boxplot(x='재발여부', y='수술시간', data=final)
```

Out[77]: <AxesSubplot:xlabel='재발여부', ylabel='수술시간'>



In [78]:
```
sns.boxplot(x='빈혈여부', y='수술시간', data=final)
```

Out[78]: <AxesSubplot:xlabel='빈혈여부', ylabel='수술시간'>



In [79]:
```
final.columns
```

Out[79]:
```
Index(['환자ID', '전방디스크높이(mm)', '후방디스크높이(mm)', '지방축적도', 'Instability', 'MF + ES',
       'Modic change', 'PI', 'PT', 'Seg Angle(raw)', 'Vaccum disc', '골밀도',
       '디스크단면적', '디스크위치', '척추이동척도', '척추전방위증', 'Large Lymphocyte',
       'Location of herniation', 'ODI', '가족력', '간질성폐질환', '고혈압여부', '과거수술횟수',
       '당뇨여부', '말초동맥질환여부', '빈혈여부', '성별', '스테로이드치료', '신부전여부', '신장', '심혈관질환',
       '암발병여부', '연령', '우울증여부', '입원기간', '입원일자', '종양진행여부', '직업', '체중', '퇴원일자',
       '헤모글로빈수치', '혈전합병증여부', '환자통증정도', '흡연여부', '통증기간(월)', '수술기법', '수술시간',
       '수술실패여부', '수술일자', '재발여부', '혈액형', 'SS', '연령대'],
      dtype='object')
```

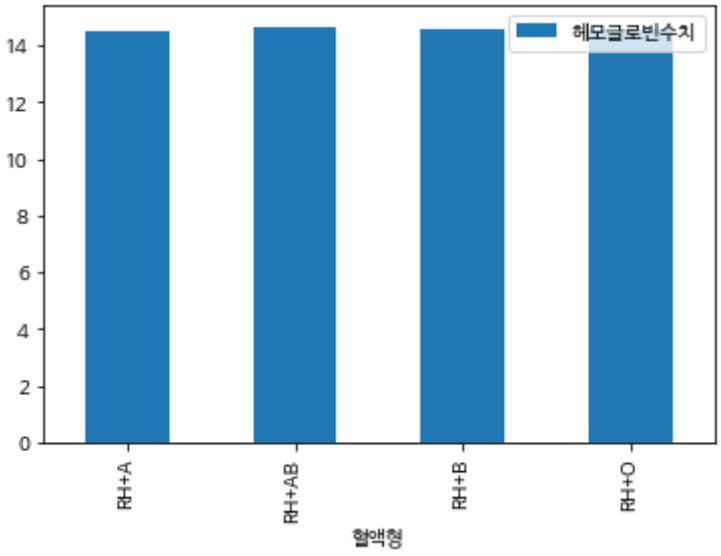
In [80]:
```
from datetime import datetime, timedelta
def 연도(date_time):
    s = str(date_time)
    days = ['월','화','수','목','금','토','일']
    date = int(s[:4])#year=int(s[0:4]), month=int(s[4:6]), day=int(s[6:8]))
    return date#days[date.weekday()]
```

In [81]:
```
# 헤모글로빈 수치 & 혈액팩 -> 수혈팩 재고 관리?
final['연도'] = final['수술일자'].apply(연도)
final[['연도','헤모글로빈수치']].groupby('연도').mean().plot.line()
```

Out[81]: <AxesSubplot:xlabel='연도'>

In [82]:
```
final[['헤모글로빈수치','혈액형']].groupby('혈액형').mean().plot.bar()
```

Out[82]: <AxesSubplot:xlabel='혈액형'>



In [83]:
```
df = pd.DataFrame()
df['전체'] = final[['직업','재발여부']].groupby('직업').count()['재발여부']
df['재발'] = final[final['재발여부']==1][['직업','재발여부']].groupby('직업').count()['재발여부
```

In [84]:
```
df = df.fillna(0)
pd.DataFrame(df['재발']/df['전체']).sort_values(by=0,ascending=False)
```

Out[84]:

직업	0
건설업	0.235294
운동선수	0.214286
교사	0.200000
공무원	0.176471
의료직	0.175000
자영업	0.169591

직업	0
사무직	0.135857
사업가	0.128205
노동직	0.119048
무직	0.115854
농업	0.100000
운수업	0.080000
주부	0.078947
학생	0.051852
군인	0.044444
특수전문직	0.043478
예술가	0.000000

In [85]:
```
# 병의 유무에 따라 통증정도
['환자ID', '전방디스크높이(mm)', '후방디스크높이(mm)', '지방축적도', 'Instability', 'MF + ES',
       'Modic change', 'PI', 'PT', 'Seg Angle(raw)', 'Vaccum disc', '골밀도',
       '디스크단면적', '디스크위치', '척추이동척도', '척추전방위증', 'Large Lymphocyte',
       'Location of herniation', 'ODI', '가족력', '간질성폐질환', '고혈압여부', '과거수술횟수',
       '당뇨여부', '말초동맥질환여부', '빈혈여부', '성별', '스테로이드치료', '신부전여부', '신장'
       '암발병여부', '연령', '우울증여부', '입원기간', '입원일자', '종양진행여부', '직업', '체중'
       '헤모글로빈수치', '혈전합병증여부', '환자통증정도', '흡연여부', '통증기간(월)', '수술기법'
       '수술실패여부', '수술일자', '재발여부', '혈액형', 'SS', '연령대']
```

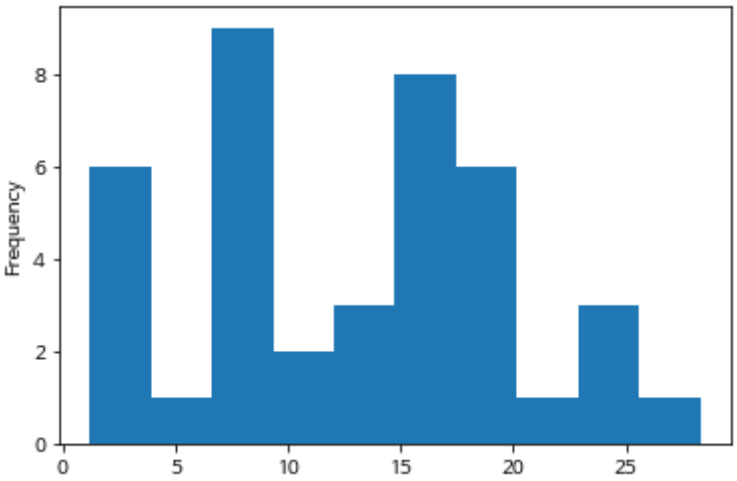
Out[85]:
```
['환자ID',
 '전방디스크높이(mm)',
 '후방디스크높이(mm)',
 '지방축적도',
 'Instability',
 'MF + ES',
 'Modic change',
 'PI',
 'PT',
 'Seg Angle(raw)',
 'Vaccum disc',
 '골밀도',
 '디스크단면적',
 '디스크위치',
 '척추이동척도',
 '척추전방위증',
 'Large Lymphocyte',
 'Location of herniation',
 'ODI',
 '가족력',
 '간질성폐질환',
 '고혈압여부',
 '과거수술횟수',
 '당뇨여부',
```

```
'말초동맥질환여부',
'빈혈여부',
'성별',
'스테로이드치료',
'신부전여부',
'신장',
'심혈관질환',
'암발병여부',
'연령',
'우울증여부',
'입원기간',
'입원일자',
'종양진행여부',
'직업',
'체중',
'퇴원일자',
'헤모글로빈수치',
'혈전합병증여부',
'환자통증정도',
'흡연여부',
'통증기간(월)',
'수술기법',
'수술시간',
'수술실패여부',
'수술일자',
'재발여부',
'혈액형',
'SS',
'연령대']
```

In [86]:
```
dis = ['척추전방위증','간질성폐질환','고혈압여부','당뇨여부','말초동맥질환여부','빈혈여부','신부전
```

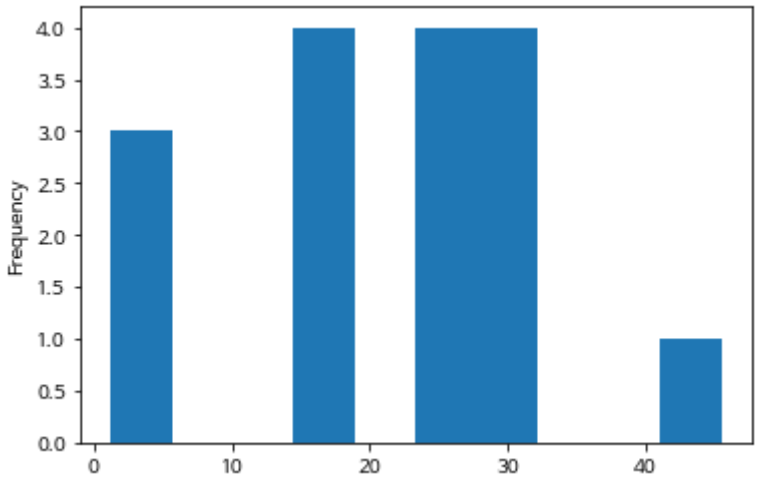
In [87]:
```
# pi, 척추이동척도 extremely up, down
final[(final['척추이동척도']=='Extremely down')]['PT'].plot.hist()
```

Out[87]: <AxesSubplot:ylabel='Frequency'>



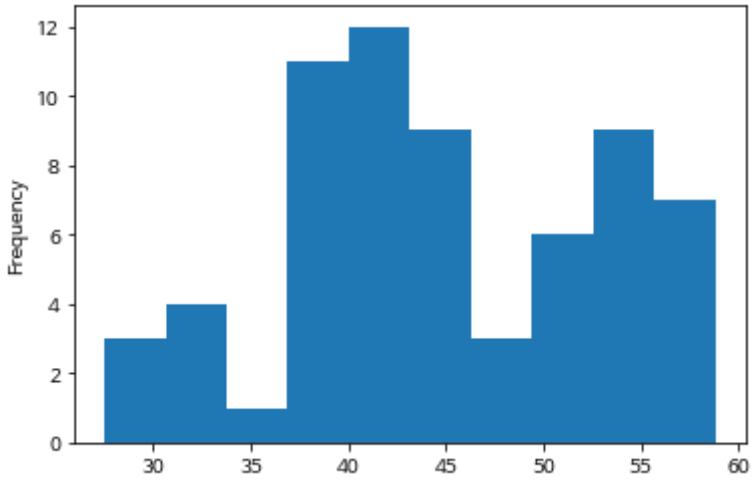
In [88]:
```
final[(final['척추이동척도']=='Extremely up')]['PT'].plot.hist()
```

Out[88]: <AxesSubplot:ylabel='Frequency'>



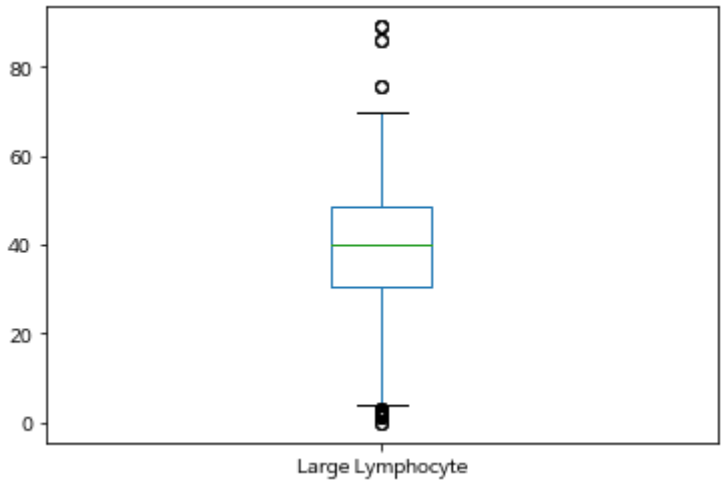
In [89]:
```
final[(final['척추이동척도']=='Up')]['PI'].plot.hist()
```

Out[89]: <AxesSubplot:ylabel='Frequency'>



In [90]:
```
df2['Large Lymphocyte'].plot.box()
```

Out[90]: <AxesSubplot:>



In [91]:
```
df2['Large Lymphocyte'].describe() # 0인 값? & 89?
```

Out[91]:
```
count    1894.000000
mean       39.270750
```

```
std        13.675874
min         0.000000
25%        30.700000
50%        40.200000
75%        48.600000
max        89.000000
Name: Large Lymphocyte, dtype: float64
```

In [92]:
```
df2[df2['ODI']>70]
```

Out[92]:

환자ID	Large Lymphocyte	Location of herniation	ODI	가족력	간질성폐질환	고혈압여부	과거수술횟수	당뇨여부	말초동맥질환여부	...	입원일자	종양진행여부	직업	체중	퇴원일자	헤모글로빈수치	혈전합병증여부	환자통증정도	흡연여부

0 rows × 30 columns

In [93]:
```
df1['환자ID']
```

Out[93]:
```
0          1PT
1          2PT
2          3PT
3          4PT
4          5PT
         ...
1889    1890PT
1890    1891PT
1891    1892PT
1892    1893PT
1893    1894PT
Name: 환자ID, Length: 1894, dtype: object
```

In [94]:
```
# final.to_csv('total.csv',index=False)
```

데이터 분석 1

연령 별

퇴원일자 - 입원일자 => '입원기간' 이게 제일 정확하다

In [96]:
```
final.info(0)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1894 entries, 0 to 1893
Data columns (total 54 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   환자ID                    1894 non-null   object
```

```
 1   전방디스크높이(mm)             1894 non-null   float64
 2   후방디스크높이(mm)             1894 non-null   float64
 3   지방축적도                    1891 non-null   float64
 4   Instability              1894 non-null   int64
 5   MF + ES                  1894 non-null   float64
 6   Modic change             1894 non-null   int64
 7   PI                       1890 non-null   float64
 8   PT                       1890 non-null   float64
 9   Seg Angle(raw)           1893 non-null   float64
 10  Vaccum disc              1894 non-null   int64
 11  골밀도                      998 non-null    float64
 12  디스크단면적                   1893 non-null   float64
 13  디스크위치                    1894 non-null   int64
 14  척추이동척도                   1894 non-null   object
 15  척추전방위증                   1894 non-null   int64
 16  Large Lymphocyte         1894 non-null   float64
 17  Location of herniation   1894 non-null   int64
 18  ODI                      462 non-null    float64
 19  가족력                      1843 non-null   float64
 20  간질성폐질환                   1894 non-null   int64
 21  고혈압여부                    1894 non-null   int64
 22  과거수술횟수                   1894 non-null   int64
 23  당뇨여부                     1894 non-null   int64
 24  말초동맥질환여부                 1894 non-null   int64
 25  빈혈여부                     1894 non-null   int64
 26  성별                       1894 non-null   int64
 27  스테로이드치료                   1894 non-null   int64
 28  신부전여부                    1894 non-null   int64
 29  신장                       1894 non-null   int64
 30  심혈관질환                    1894 non-null   int64
 31  암발병여부                    1894 non-null   int64
 32  연령                       1894 non-null   int64
 33  우울증여부                    1894 non-null   int64
 34  입원기간                     1894 non-null   int64
 35  입원일자                     1894 non-null   int64
 36  종양진행여부                   1894 non-null   int64
 37  직업                       1479 non-null   object
 38  체중                       1894 non-null   float64
 39  퇴원일자                     1894 non-null   int64
 40  헤모글로빈수치                  1893 non-null   float64
 41  혈전합병증여부                  1894 non-null   int64
 42  환자통증정도                   1894 non-null   int64
 43  흡연여부                     1894 non-null   int64
 44  통증기간(월)                  1890 non-null   float64
 45  수술기법                     1813 non-null   object
 46  수술시간                     1840 non-null   float64
 47  수술실패여부                   1894 non-null   int64
 48  수술일자                     1894 non-null   int64
 49  재발여부                     1894 non-null   int64
 50  혈액형                      1894 non-null   object
 51  SS                       1890 non-null   float64
 52  연령대                      1894 non-null   int64
 53  연도                       1894 non-null   int64
dtypes: float64(17), int64(32), object(5)
memory usage: 878.4+ KB
```

In [97]:
```
final['입원일자']
```

Out[97]:
```
0       20190713
1       20190715
2       20190729
3       20190731
4       20190903
          ...
1889    20170407
1890    20170426
1891    20170410
1892    20170408
1893    20170412
Name: 입원일자, Length: 1894, dtype: int64
```

In [98]:
```
# final['퇴원일자_1'] = pd.to_datetime(final['퇴원일자'], format = '%y/%m/%d')
```

In [99]:
```
final['입원일자(date)'] = pd.to_datetime(final['입원일자'], format='%Y%m%d')
final['퇴원일자(date)'] = pd.to_datetime(final['퇴원일자'], format='%Y%m%d')
```

In [100]:
```
final['입원기간']= final['퇴원일자(date)'] - final['입원일자(date)']
```

In [101]:
```
def func1(row):
    return int(str(row).split(' ')[0])

final['입원기간(int)'] = final['입원기간'].apply(func1)
```

In [102]:
```
cond1 = (final['입원기간(int)']>0)
final_1 = final.loc[cond1]
```

In [103]:
```
final_1.pivot_table(index='연령대',values='입원기간(int)'
                    ,aggfunc=['mean','min','max']).round(1)
```

Out[103]:

	mean	min	max
	입원기간(int)	입원기간(int)	입원기간(int)
연령대			
1	2.3	1	17
2	9.7	1	1125
3	6.5	1	674
4	3.0	1	217
5	4.1	1	177
6	7.0	1	318
7	10.1	1	173
8	8.6	2	27

a : 1,3,2,4,5 -> 평균 :3 b : 1,3,2,4,1000 -> 평균 :202

In [128]:
```
from scipy import stats
from scipy.stats import shapiro , normaltest , anderson , kstest
# 집단간 비교 (X:범주 / Y:연속)
#1. 연속형이 정규분포를 띄는가
# 귀무 : 해당 분포는 정규분포이다.
# 대립 : 해당 분포는 정규분포가 아니다.
stats.normaltest(final_1['입원기간(int)'])
# P.value < 0.05 / 대립가설 참 / 해당 분포는 정규분포가 아니다.
```

Out[128]:
```
NormaltestResult(statistic=4068.0756431592235, pvalue=0.0)
```

In [129]:
```
# 2. 비모수 검정
# 귀무 : 연령대 별 입원기간의 차이가 없다.
# 대립 : 연령대 별 입원기간의 차이가 있다.
cond1 = (final_1['연령대']==2)

df1_20 = final_1.loc[cond1]
df1_non_20 = final_1.loc[~cond1]
```

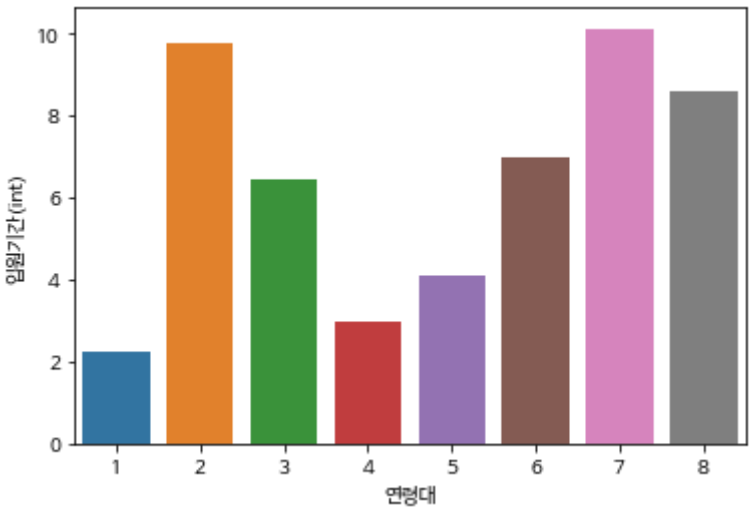
In [130]:
```
stats.ranksums(df1_20['입원기간(int)'], df1_non_20['입원기간(int)'])
```

Out[130]:
```
RanksumsResult(statistic=-0.39062368311301005, pvalue=0.6960754172159893)
```

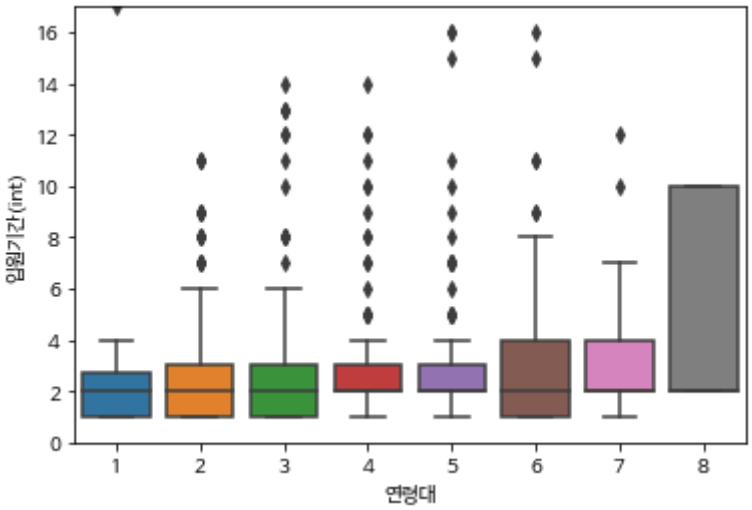
In [131]:
```
# 귀무 : 연령대 별 입원기간의 차이가 없다.
```

In [132]:
```
sns.barplot(data=final_1, x='연령대',y='입원기간(int)',ci=None)
```

Out[132]:
```
<AxesSubplot:xlabel='연령대', ylabel='입원기간(int)'>
```



In [133]:
```
sns.boxplot(data=final_1, x='연령대',y='입원기간(int)')
plt.ylim(0,17)
plt.show()
```



In [134]:
```
cond1 = (final_1['연령대'] == 1 | 2 | 3)
```

In [135]:
```
df1_till30 = final_1.loc[cond1]
df1_non_till30 = final_1.loc[~cond1]
```

In [136]:
```
stats.ranksums(df1_till30['입원기간(int)'], df1_non_till30['입원기간(int)'])
```

Out[136]: RanksumsResult(statistic=-0.9838806825763513, pvalue=0.32517418279570576)

In [137]:
```
# 귀무 : 연령대 별 입원기간의 차이가 없다.
```

In [138]:
```
final_1.groupby('연령대').count()
```

Out[138]:

연령대	환자ID	전방디스크높이(mm)	후방디스크높이(mm)	지방축적도	Instability	MF+ES	Modic change	PI	PT	Seg Angle(raw)	...	수술시간	수술실패여부	수술일자	
1	66	66	66	66	66	66	66	66	66	66	...	63	66	66	6
2	317	317	317	316	317	317	317	315	315	317	...	302	317	317	31
3	450	450	450	450	450	450	450	449	449	450	...	440	450	450	45
4	560	560	560	559	560	560	560	559	559	560	...	550	560	560	56
5	258	258	258	258	258	258	258	258	258	258	...	248	258	258	25
6	108	108	108	108	108	108	108	108	108	108	...	105	108	108	10
7	57	57	57	56	57	57	57	57	57	57	...	57	57	57	5
8	5	5	5	5	5	5	5	5	5	5	...	3	5	5	

8 rows × 56 columns

**Insight: 연령별 입원일자를 살펴 보았는데, 의외로 20대의 입원기간이 길다.

- 평균 값이므로 절대 인원가 비교해보면 유의미한 연령대는 20대 (317명-전체의 17.4%)

In [139]:
```
66+317+450+560+258+108+57+5
```

Out[139]: 1821

In [140]:
```
317/1821 * 100
```

Out[140]: 17.40801757276222

데이터 분석 2

수술 기법이 없는 사람들에 대한 특징 조사

- 개복하면 아무 문제가 없어서 수술 안한 사람들일 수도 있어서

In [141]:
```
final.head()
```

Out[141]:

	환자 ID	전방 디스크높이 (mm)	후방 디스크높이 (mm)	지방 축적도	Instability	MF + ES	Modic change	PI	PT	Seg Angle(raw)	...	수술실패여부	수술일자
0	1PT	16.1	12.3	282.3	0	1824.6	3	51.6	36.6	14.4	...	0	20190715
1	2PT	13.7	6.4	177.3	0	1737.5	0	40.8	7.2	17.8	...	0	20190716
2	3PT	13.6	7.4	256.8	0	1188.5	0	67.5	27.3	10.2	...	0	20190731
3	4PT	10.6	7.3	250.1	0	2534.5	0	49.2	18.7	19.9	...	0	20190802
4	5PT	17.1	8.1	232.2	0	1840.6	0	58.8	14.7	5.2	...	0	20190906

5 rows × 57 columns

In [142]:
```
final['수술기법'].uniquecond1 = (final['입원기간(int)']>0)
final_1 = final.loc[cond1]
```

```
---------------------------------------------------------------------------
IndexingError                             Traceback (most recent call last)
<ipython-input-142-81d374263080> in <module>
      1 final['수술기법'].uniquecond1 = (final['입원기간(int)']>0)
----> 2 final_1 = final.loc[cond1]

~/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in __getitem__(self, key)
    893
    894             maybe_callable = com.apply_if_callable(key, self.obj)
```

```
--> 895             return self._getitem_axis(maybe_callable, axis=axis)
    896
    897     def _is_scalar_access(self, key: Tuple):

~/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in _getitem_axis(self, key, a
xis)
   1102             return self._get_slice_axis(key, axis=axis)
   1103         elif com.is_bool_indexer(key):
-> 1104             return self._getbool_axis(key, axis=axis)
   1105         elif is_list_like_indexer(key):
   1106

~/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in _getbool_axis(self, key, a
xis)
    910         # caller is responsible for ensuring non-None axis
    911         labels = self.obj._get_axis(axis)
--> 912         key = check_bool_indexer(labels, key)
    913         inds = key.nonzero()[0]
    914         return self.obj._take_with_is_copy(inds, axis=axis)

~/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py in check_bool_indexer(index,
 key)
   2267         mask = isna(result._values)
   2268         if mask.any():
-> 2269             raise IndexingError(
   2270                 "Unalignable boolean Series provided as "
   2271                 "indexer (index of the boolean Series and of "

IndexingError: Unalignable boolean Series provided as indexer (index of the boolean Series an
d of the indexed object do not match).
```

In [163]:
```
surgery_o = final[final['수술기법'].notna()]
```

In [164]:
```
surgery_o
```

Out[164]:

	환자ID	전방 디스크높이 (mm)	후방 디스크높이 (mm)	지방 축적도	Instability	MF + ES	Modic change	PI	PT	Seg Angle(raw)	...	수술실패여부	수
0	1PT	16.1	12.3	282.3	0	1824.6	3	51.6	36.6	14.4	...	0	201
1	2PT	13.7	6.4	177.3	0	1737.5	0	40.8	7.2	17.8	...	0	201
2	3PT	13.6	7.4	256.8	0	1188.5	0	67.5	27.3	10.2	...	0	201
3	4PT	10.6	7.3	250.1	0	2534.5	0	49.2	18.7	19.9	...	0	201
4	5PT	17.1	8.1	232.2	0	1840.6	0	58.8	14.7	5.2	...	0	201
...	...	...	...	...	...	...	...	...	...	...	...	...	...

	환자ID	전방 디스크높이 (mm)	후방 디스크높이 (mm)	지방 축적도	Instability	MF + ES	Modic change	PI	PT	Seg Angle(raw)	...	수술실패여부	수
1870	1871PT	8.5	9.0	182.5	0	1919.5	0	31.7	14.9	9.6	...	0	201
1872	1873PT	11.6	7.2	94.2	0	2398.9	0	39.4	8.0	19.5	...	0	201
1874	1875PT	11.1	7.6	126.1	1	1970.3	2	43.6	17.7	9.1	...	0	201
1879	1880PT	12.7	8.7	207.4	0	2220.1	0	34.0	19.0	6.0	...	0	201
1891	1892PT	13.5	5.5	148.5	0	3864.1	0	44.6	15.0	17.4	...	0	201

1813 rows × 57 columns

In [165]:
```
final['수술기법1'] = final['수술기법'].fillna(value = 0)
```

In [166]:
```
final
```

Out[166]:

	환자ID	전방 디스크높이 (mm)	후방 디스크높이 (mm)	지방 축적도	Instability	MF + ES	Modic change	PI	PT	Seg Angle(raw)	...	수술일
0	1PT	16.1	12.3	282.3	0	1824.6	3	51.6	36.6	14.4	...	201907
1	2PT	13.7	6.4	177.3	0	1737.5	0	40.8	7.2	17.8	...	201907
2	3PT	13.6	7.4	256.8	0	1188.5	0	67.5	27.3	10.2	...	201907
3	4PT	10.6	7.3	250.1	0	2534.5	0	49.2	18.7	19.9	...	201908
4	5PT	17.1	8.1	232.2	0	1840.6	0	58.8	14.7	5.2	...	201909
...	...	...	...	...	...	...	...	...	...	...	...	
1889	1890PT	17.0	10.7	237.5	0	2795.7	2	59.5	23.0	21.8	...	201704
1890	1891PT	9.4	8.2	288.0	0	1473.0	0	47.7	20.2	5.0	...	201704
1891	1892PT	13.5	5.5	148.5	0	3864.1	0	44.6	15.0	17.4	...	201704
1892	1893PT	14.0	10.0	89.0	0	2481.8	2	32.2	11.1	17.7	...	201704
1893	1894PT	16.1	9.5	251.4	0	1796.1	0	38.9	6.8	27.8	...	201704

1894 rows × 58 columns

- 수술기법이 0일때의 PI(평균),골밀도,지방축적도,스테로이드치료여부(0,1)(범주형), segangle
- 수술기법이 0이 아닐때의 PI, 골밀도, 스테로이드치료여부(0,1)

In [167]:
```
cond1 = (final['수술기법1']==0)
final_1 = final.loc[cond1]
```

In [168]:
```
final_1.pivot_table(index='수술기법1',values='PI'
                    ,aggfunc=['mean','min','max'])
```

Out[168]:

	mean	min	max
	PI	PI	PI
수술기법1			
0	45.045679	23.0	59.9

In [169]:
```
cond2 = (final['수술기법1']!=0)
final_2 = final.loc[cond2]
```

In [170]:
```
final_2.pivot_table(index='수술기법1',values='PI'
                    ,aggfunc=['mean','min','max'])
```

Out[170]:

	mean	min	max
	PI	PI	PI
수술기법1			
IELD	42.492086	14.0	78.0
TELD	47.010120	11.9	559.0

In [171]:
```
final_1.pivot_table(index='수술기법1',values='골밀도'
                    ,aggfunc=['mean','min','max'])
```

Out[171]:

	mean	min	max
	골밀도	골밀도	골밀도
수술기법1			
0	-1.5505	-2.46	-0.6

In [172]:
```
final_2.pivot_table(index='수술기법1',values='골밀도'
                    ,aggfunc=['mean','min','max'])
```

Out[172]:

	mean	min	max
	골밀도	골밀도	골밀도
수술기법1			
IELD	-1.473235	-2.46	1.14

	mean 골밀도	min 골밀도	max 골밀도
수술기법1			
TELD	-1.507989	-2.84	1.70

In [173]:
```
final_1.sum()
```

Out[173]:
```
환자ID                    152PT341PT451PT452PT453PT454PT455PT456PT457PT4...
전방디스크높이(mm)                                                  960.5
후방디스크높이(mm)                                                 675.93
지방축적도                                                      16610.88
Instability                                                        4
MF + ES                                                       182141.95
Modic change                                                      42
PI                                                              3648.7
PT                                                              1232.5
Seg Angle(raw)                                                  1396.9
Vaccum disc                                                        8
골밀도                                                           -62.02
디스크단면적                                                    160484.99
디스크위치                                                        327
척추이동척도              MiddleMiddleDownDownMiddleMiddleMiddleMiddleMi...
척추전방위증                                                       7
Large Lymphocyte                                                3136.5
Location of herniation                                             156
ODI                                                               49.0
가족력                                                             1.0
간질성폐질환                                                         0
고혈압여부                                                          9
과거수술횟수                                                        14
당뇨여부                                                            8
말초동맥질환여부                                                     1
빈혈여부                                                            2
성별                                                              118
스테로이드치료                                                      53
신부전여부                                                          2
신장                                                            13609
심혈관질환                                                          4
암발병여부                                                          1
연령                                                             3296
우울증여부                                                          0
입원기간                                          276 days 00:00:00
입원일자                                                   1634515450
종양진행여부                                                        1
체중                                                           5483.0
퇴원일자                                                   1634525156
헤모글로빈수치                                                 1181.24
혈전합병증여부                                                       0
환자통증정도                                                      567
흡연여부                                                           21
통증기간(월)                                                    532.0
수술기법                                                           0
```

```
수술시간                                                    4688.0
수술실패여부                                                     3
수술일자                                                1634524564
재발여부                                                      16
혈액형                    RH+BRH+ARH+BRH+ORH+ARH+ABRH+ORH+BRH+ARH+ARH+BR...
SS                                                      2416.2
연령대                                                       288
연도                                                     163449
입원기간(int)                                                276
수술기법1                                                      0
dtype: object
```

SEG ANGLE - 유의미해보임 (근거 더 필요)

In [174]:
```
final_1.pivot_table(index='수술기법1',values='Seg Angle(raw)'
                    ,aggfunc=['mean','min','max'])
```

Out[174]:

	mean	min	max
	Seg Angle(raw)	Seg Angle(raw)	Seg Angle(raw)
수술기법1			
0	17.245679	0.3	36.3

(수술일자는 있음) 수술기법이 공백인 사람들 70명

In [175]:
```
final_2.pivot_table(index='수술기법1',values='Seg Angle(raw)'
                    ,aggfunc=['mean','min','max'])
```

Out[175]:

	mean	min	max
	Seg Angle(raw)	Seg Angle(raw)	Seg Angle(raw)
수술기법1			
IELD	18.079286	0.4	36.8
TELD	14.959366	-27.4	165.0

- IELD :
- TELD :

- 연속형(Y): Seg Angle
- 범주형(X): 수술기법1

연속성 변수인 Y의 정규성 검정

정규성 검정하기

In [176]:
```
df1 = final_2[final_2['수술기법1'] == 'TELD']
```

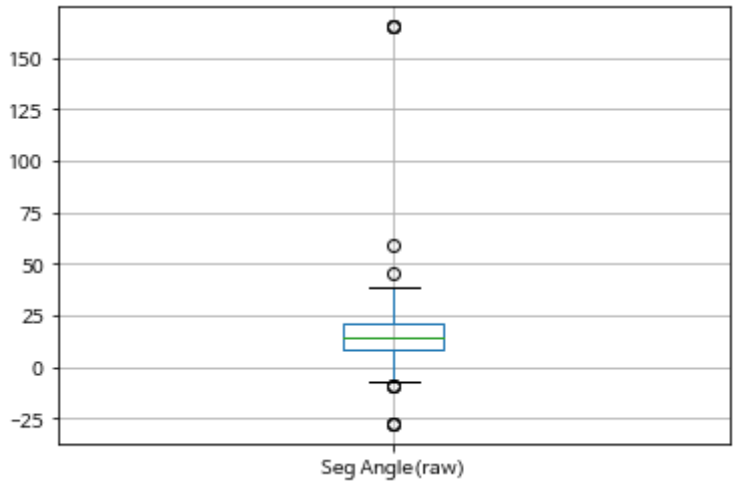
In [177]:
```
df1_ = pd.DataFrame(df1['Seg Angle(raw)'])
```

In [178]:
```
df1_.isnull().sum()
```

Out[178]:
```
Seg Angle(raw)    1
dtype: int64
```

In [179]:
```
df1_.boxplot()
```

Out[179]:
```
<AxesSubplot:>
```



In [180]:
```
df1_[df1_['Seg Angle(raw)']<0]
```

Out[180]:

	Seg Angle(raw)
55	-1.8
206	-9.4
397	-4.9
408	-27.4
525	-1.8
676	-9.4
867	-4.9
878	-27.4
1034	-5.0
1066	-1.8
1378	-9.4
1393	-3.4
1395	-6.9
1748	-4.9
1763	-27.4

In [181]:
```
df1_.dropna(inplace=True)
```

In [182]:
```
df1_.isnull().sum()
```

Out[182]:
```
Seg Angle(raw)    0
dtype: int64
```

In [183]:
```
df2 = final_2[final_2['수술기법1'] == 'IELD']
```

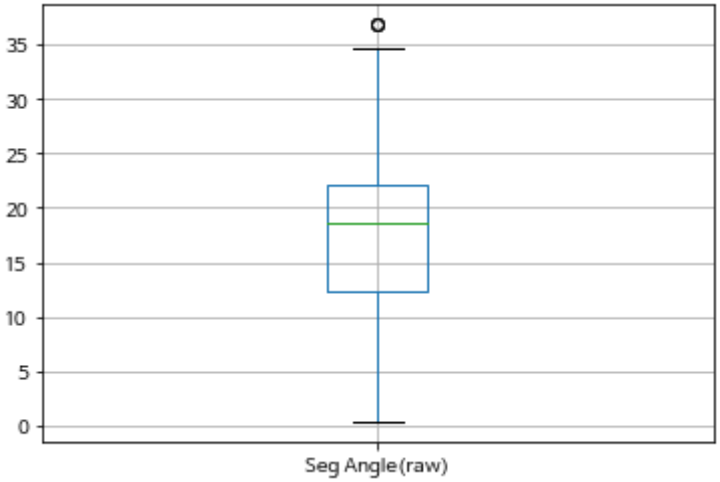
In [184]:
```
df2_ = pd.DataFrame(df2['Seg Angle(raw)'])
```

In [185]:
```
df2_.count()
```

Out[185]:
```
Seg Angle(raw)    140
dtype: int64
```

In [186]:
```
df2_.boxplot()
```

Out[186]:
```
<AxesSubplot:>
```



In [187]:
```
df2_.isnull().sum()
```

Out[187]:
```
Seg Angle(raw)    0
dtype: int64
```

In [188]:
```
from scipy import stats
```

In [189]:
```
t_result = stats.ttest_ind(df1_, df2_)
```

In [190]:
```
t, p = t_result.statistic.round(3), t_result.pvalue.round(3)

print("2_sample t-test")
print("t:{}".format(t))
print("p:{}".format(p))
```

```
2_sample t-test
t:[-3.399]
p:[0.001]
```

스테로이드는 count하자

스테로이드 - 유의미해보임 (근거 더 필요)

In [191]:
```
final_1.sum()
```

Out[191]:

환자ID	152PT341PT451PT452PT453PT454PT455PT456PT457PT4...
전방디스크높이(mm)	960.5
후방디스크높이(mm)	675.93
지방축적도	16610.88
Instability	4
MF + ES	182141.95
Modic change	42
PI	3648.7
PT	1232.5
Seg Angle(raw)	1396.9
Vaccum disc	8
골밀도	-62.02
디스크단면적	160484.99
디스크위치	327
척추이동척도	MiddleMiddleDownDownMiddleMiddleMiddleMiddleMi...
척추전방위증	7
Large Lymphocyte	3136.5
Location of herniation	156
ODI	49.0
가족력	1.0
간질성폐질환	0
고혈압여부	9
과거수술횟수	14
당뇨여부	8
말초동맥질환여부	1
빈혈여부	2
성별	118
스테로이드치료	53
신부전여부	2
신장	13609
심혈관질환	4
암발병여부	1
연령	3296
우울증여부	0
입원기간	276 days 00:00:00
입원일자	1634515450
종양진행여부	1
체중	5483.0
퇴원일자	1634525156
헤모글로빈수치	1181.24
혈전합병증여부	0
환자통증정도	567
흡연여부	21
통증기간(월)	532.0
수술기법	0
수술시간	4688.0
수술실패여부	3
수술일자	1634524564
재발여부	16
혈액형	RH+BRH+ARH+BRH+ORH+ARH+ABRH+ORH+BRH+ARH+ARH+BR...
SS	2416.2
연령대	288
연도	163449
입원기간(int)	276

```
수술기법1                                                    0
dtype: object
```

In [192]:
```
freq = final['수술기법'].value_counts()
freq
```

Out[192]:
```
TELD    1673
IELD     140
Name: 수술기법, dtype: int64
```

In [193]:
```
freq = final_1['수술기법'].value_counts()
freq
```

Out[193]:
```
Series([], Name: 수술기법, dtype: int64)
```

In [194]:
```
final_2
```

Out[194]:

	환자ID	전방 디스크높이 (mm)	후방 디스크높이 (mm)	지방 축적도	Instability	MF + ES	Modic change	PI	PT	Seg Angle(raw)	...	수술일
0	1PT	16.1	12.3	282.3	0	1824.6	3	51.6	36.6	14.4	...	201907
1	2PT	13.7	6.4	177.3	0	1737.5	0	40.8	7.2	17.8	...	201907
2	3PT	13.6	7.4	256.8	0	1188.5	0	67.5	27.3	10.2	...	201907
3	4PT	10.6	7.3	250.1	0	2534.5	0	49.2	18.7	19.9	...	201908
4	5PT	17.1	8.1	232.2	0	1840.6	0	58.8	14.7	5.2	...	201909
...	...	...	...	...	...	...	...	...	...	...	...	
1870	1871PT	8.5	9.0	182.5	0	1919.5	0	31.7	14.9	9.6	...	201703
1872	1873PT	11.6	7.2	94.2	0	2398.9	0	39.4	8.0	19.5	...	201703
1874	1875PT	11.1	7.6	126.1	1	1970.3	2	43.6	17.7	9.1	...	201709
1879	1880PT	12.7	8.7	207.4	0	2220.1	0	34.0	19.0	6.0	...	201706
1891	1892PT	13.5	5.5	148.5	0	3864.1	0	44.6	15.0	17.4	...	201704

1813 rows × 58 columns

데이터 분석 3

In [195]:
```
from datetime import datetime, timedelta
def 요일(date_time):
    s = str(date_time)
```

```
    days = ['월','화','수','목','금','토','일']
    date = int(s[4:6])#datetime(year=int(s[0:4]), month=int(s[4:6]), day=int(s[6:8]))
    return date#days[date.weekday()]
```

In [196]:
```
pd.value_counts(final[final['직업']=='사무직'].apply(요일).values()
```

```
  File "<ipython-input-196-423e610939e0>", line 1
    pd.value_counts(final[final['직업']=='사무직'].apply(요일).values()
                                                                      ^
SyntaxError: unexpected EOF while parsing
```

In [198]:
```
final['수술일자(date)'] = pd.to_datetime(final['수술일자'], format='%Y%m%d')
```

In [199]:
```
final['수술일자(date)']
```

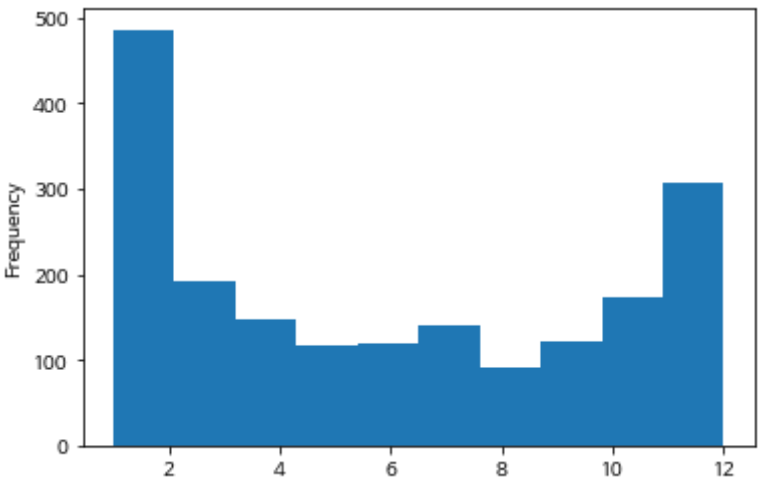
Out[199]:
```
0      2019-07-15
1      2019-07-16
2      2019-07-31
3      2019-08-02
4      2019-09-06
          ...
1889   2017-04-07
1890   2017-04-27
1891   2017-04-11
1892   2017-04-10
1893   2017-04-12
Name: 수술일자(date), Length: 1894, dtype: datetime64[ns]
```

In [200]:
```
final['수술일자(month)'] = final['수술일자(date)'].dt.month
```

In [201]:
```
# def func1(row):
#     return int(str(row).split(' ')[0])

# final['입원기간(int)'] = final['입원기간'].apply(func1)
```

In [202]:
```
final['수술일자(month)'].plot.hist()
```

Out[202]:
```
<AxesSubplot:ylabel='Frequency'>
```



In [203]:
```
final.groupby(by=['수술일자(month)'], as_index=False).count()
```

Out[203]:

	수술일자(month)	환자ID	전방디스크높이(mm)	후방디스크높이(mm)	지방축적도	Instability	MF+ES	Modic change	PI	PT	...	재발여부	혈액형	SS	연령대
0	1	285	285	285	285	285	285	285	285	285	...	285	285	285	285
1	2	201	201	201	201	201	201	201	200	200	...	201	201	200	201
2	3	192	192	192	192	192	192	192	191	191	...	192	192	191	192
3	4	148	148	148	148	148	148	148	148	148	...	148	148	148	148
4	5	116	116	116	115	116	116	116	116	116	...	116	116	116	116
5	6	119	119	119	119	119	119	119	119	119	...	119	119	119	119
6	7	140	140	140	140	140	140	140	139	139	...	140	140	139	140
7	8	91	91	91	91	91	91	91	91	91	...	91	91	91	91
8	9	121	121	121	120	121	121	121	120	120	...	121	121	120	121
9	10	173	173	173	173	173	173	173	173	173	...	173	173	173	173
10	11	131	131	131	131	131	131	131	131	131	...	131	131	131	131
11	12	177	177	177	176	177	177	177	177	177	...	177	177	177	177

12 rows × 60 columns

In [204]:
```
285+201+192+148+116+119+140+91+121+173+131+177
```

Out[204]: 1894

In [205]:
```
# final['입원일자'].apply(요일)
pd.value_counts(final[final['직업']=='사무직']['입원일자'].apply(요일).values)
```

Out[205]:
```
1     55
10    48
12    46
2     41
3     37
4     37
7     36
9     34
11    33
6     29
5     27
8     26
dtype: int64
```

In [206]:
```
55+48+46+41+37+37+36+34+33+29+27+26
```

Out[206]: 449

In [207]:
```
import seaborn as sns

# 연령대 통증기간 긴지? -> 참아온것 이를 개선할 수 있는가?
def 연령(age):
  return age//10

final['연령대'] = final['연령'].apply(연령)
```

In [215]:
```
final
```

Out[215]:

	환자ID	전방 디스크높이 (mm)	후방 디스크높이 (mm)	지방 축적도	Instability	MF + ES	Modic change	PI	PT	Seg Angle(raw)	...	혈액형
0	1PT	16.1	12.3	282.3	0	1824.6	3	51.6	36.6	14.4	...	RH+A
1	2PT	13.7	6.4	177.3	0	1737.5	0	40.8	7.2	17.8	...	RH+A
2	3PT	13.6	7.4	256.8	0	1188.5	0	67.5	27.3	10.2	...	RH+B
3	4PT	10.6	7.3	250.1	0	2534.5	0	49.2	18.7	19.9	...	RH+O
4	5PT	17.1	8.1	232.2	0	1840.6	0	58.8	14.7	5.2	...	RH+A
...	...	...	...	...	...	...	...	...	...	...	...	...
1889	1890PT	17.0	10.7	237.5	0	2795.7	2	59.5	23.0	21.8	...	RH+A
1890	1891PT	9.4	8.2	288.0	0	1473.0	0	47.7	20.2	5.0	...	RH+B
1891	1892PT	13.5	5.5	148.5	0	3864.1	0	44.6	15.0	17.4	...	RH+O
1892	1893PT	14.0	10.0	89.0	0	2481.8	2	32.2	11.1	17.7	...	RH+A
1893	1894PT	16.1	9.5	251.4	0	1796.1	0	38.9	6.8	27.8	...	RH+AB

1894 rows × 60 columns

년도 별 / 연령 별 방문자 수

In [211]:
```
final['연령대'].unique
```

Out[211]:
```
<bound method Series.unique of 0       6
1       4
2       3
3       4
4       4
       ..
1889    5
1890    4
1891    6
```

```
1892    2
1893    3
Name: 연령대, Length: 1894, dtype: int64>
```

In [213]:
```
df_1 = [final['연령대'] == '1']
df_2 = [final['연령대'] == '2']
df_3 = [final['연령대'] == '3']
df_4 = [final['연령대'] == '4']
df_5 = [final['연령대'] == '5']
df_6 = [final['연령대'] == '6']
df_7 = [final['연령대'] == '7']
df_8 = [final['연령대'] == '8']
```

In [217]:
```
#2020년 연령대
```

In [ ]:
```
#2019년 연령대
```

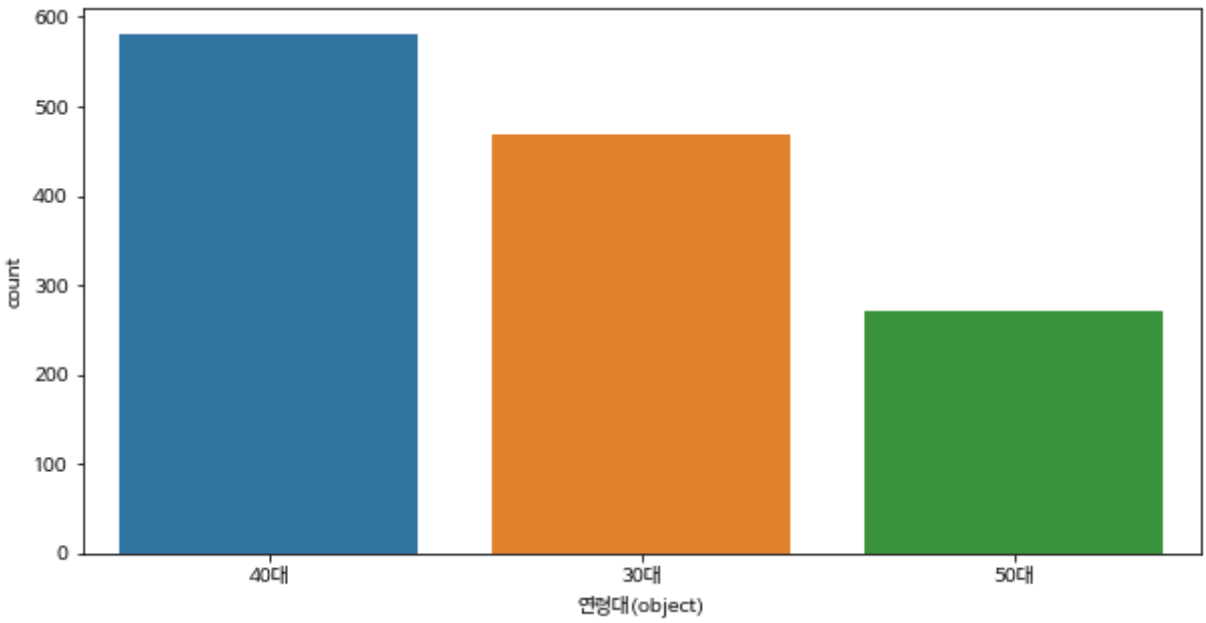
In [ ]:
```
#2018년 연령대
```

In [223]:
```
final.head()
final['count'] = 1
```

In [224]:
```
def func1(row):
    return str(row) +'0대'

final['연령대(object)'] = final['연령대'].apply(func1)
```

In [225]:
```
cond1 = (final['연령대(object)']=='30대')|(final['연령대(object)']=='40대')|(final['연령대(obj


plt.figure(figsize=[10,5])
sns.barplot(data=final.loc[cond1], x='연령대(object)',y='count',estimator=sum, ci=None)
```

Out[225]: <AxesSubplot:xlabel='연령대(object)', ylabel='count'>



In [226]:
```
plt.figure(figsize=[10,5])
sns.barplot(data=final, x='연령대(object)',y='count',estimator=sum, ci=None,
            order=['10대','20대','30대','40대','50대','60대','70대','80대'])
```

Out[226]: <AxesSubplot:xlabel='연령대(object)', ylabel='count'>