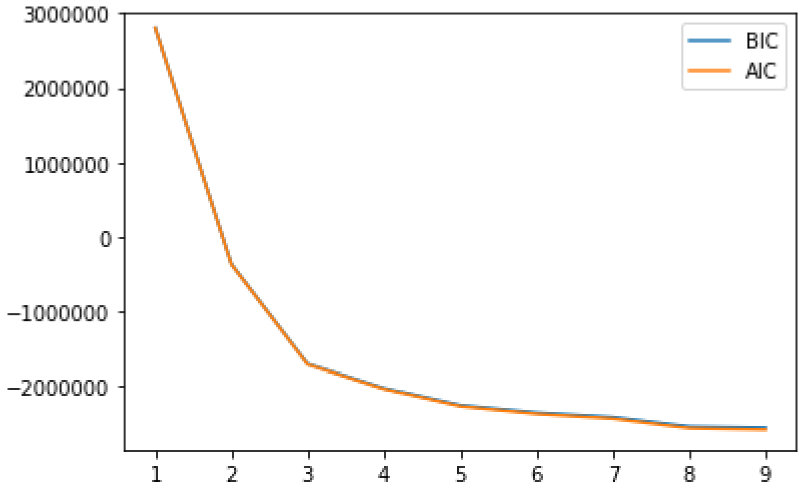
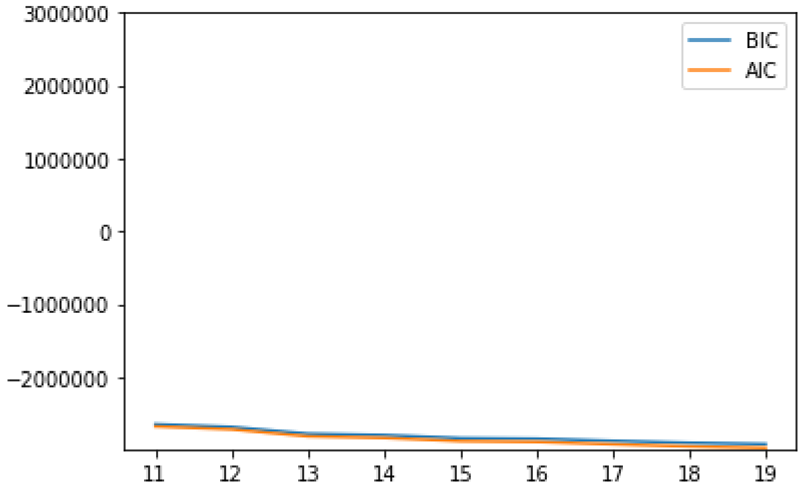
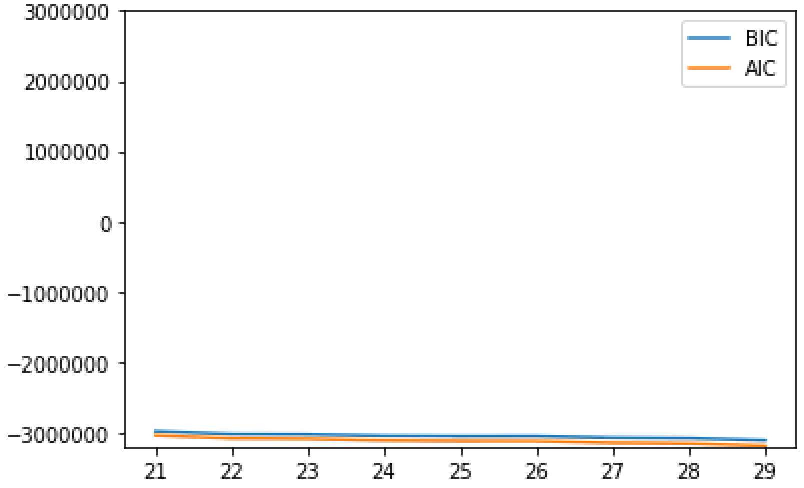
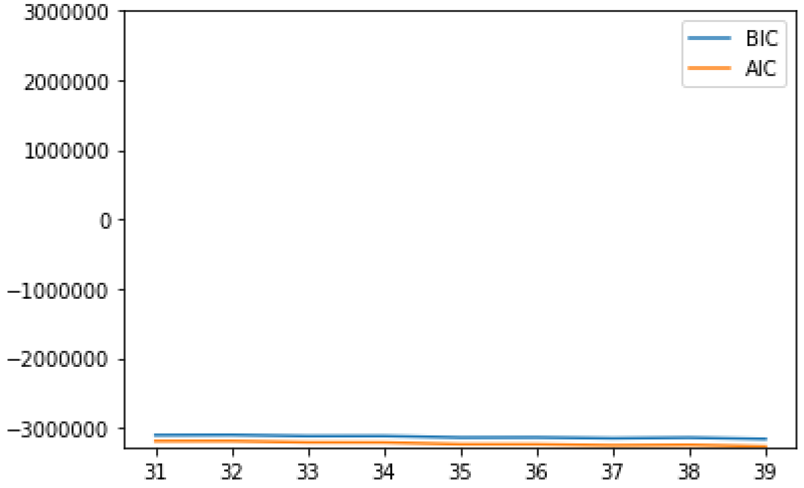
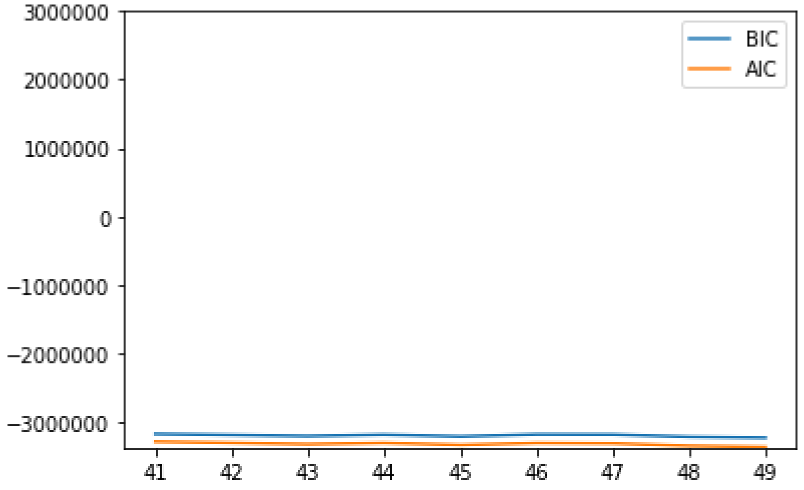
In [10]:
```
# GMM on high-dimensional datasets
badcols = ["TEFF", "LOGG", "M_H"]
abundance_pos_df = master_df.drop(columns=[col for col in master_df.columns

# print(abundance_pos_df)
n_clusters = 30

u = abundance_pos_df
n_estimators = np.arange(n_clusters)
clfs = [GaussianMixture(n_components=n, covariance_type='full').fit(abundan
bics = [clf.bic(u) for clf in clfs]
aics = [clf.aic(u) for clf in clfs]

plt.plot(n_estimators, bics, label='BIC')
plt.plot(n_estimators, aics, label='AIC')
plt.ylim(None, 3e6)
plt.legend()
plt.show()
```



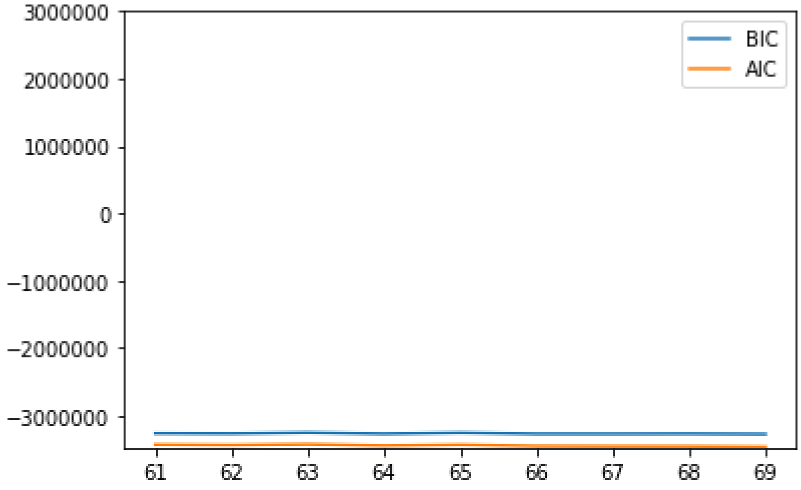


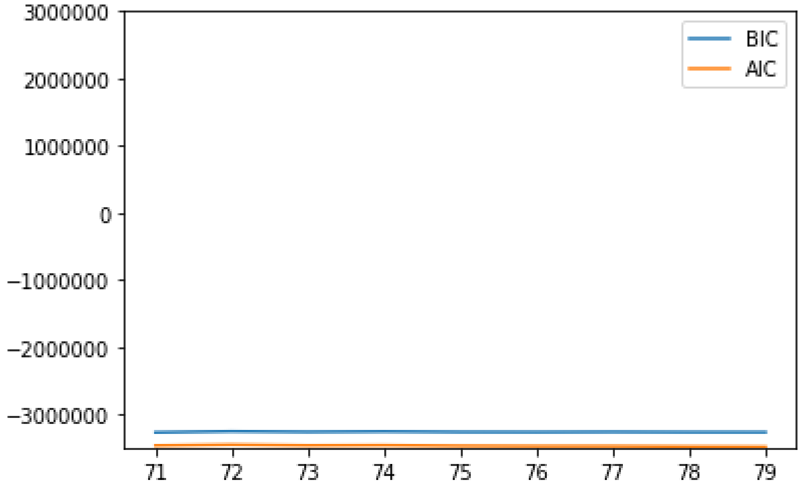

3000000
2000000
1000000
0
−1000000
−2000000
−3000000
21
22
23
24
25
26
27
28
29
BIC
AIC

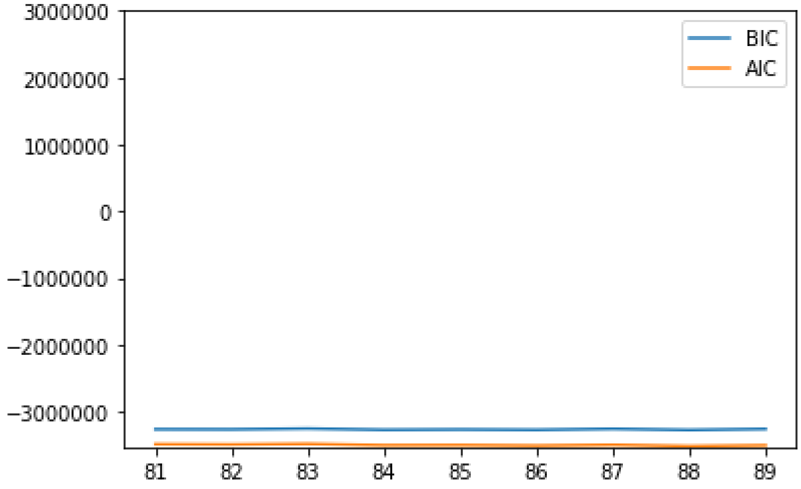

3000000
2000000
1000000
0
−1000000
−2000000
−3000000
31
32
33
34
35
36
37
38
39
BIC
AIC

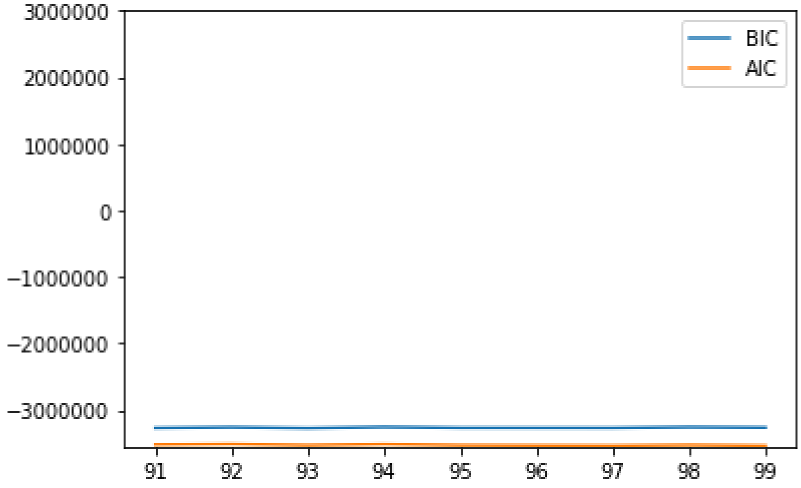

3000000
2000000
1000000
0
−1000000
−2000000
−3000000
41
42
43
44
45
46
47
48
49
BIC
AIC

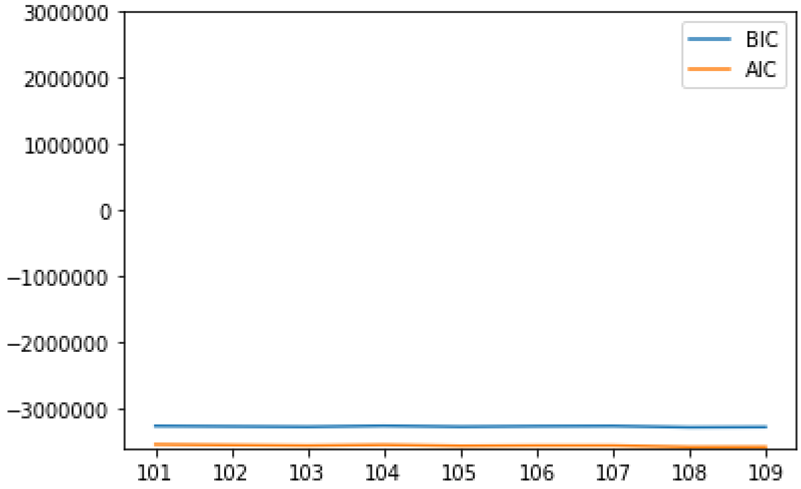
3000000
2000000
BIC
AIC


3000000
2000000
1000000
0
−1000000
−2000000
−3000000
61
62
63
64
65
66
67
68
69
BIC
AIC


3000000
2000000
1000000
0
−1000000
−2000000
−3000000
71
72
73
74
75
76
77
78
79
BIC
AIC


3000000
2000000
1000000
0
−1000000
−2000000
−3000000
81
82
83
84
85
86
87
88
89
BIC
AIC





```
---------------------------------------------------------------------------
--
KeyboardInterrupt                         Traceback (most recent call las
t)
<ipython-input-10-cb10eef13407> in <module>
      8 for i in range(20):
      9     n_estimators = np.arange(i*10 + 1, (i+1)*10)
---> 10     clfs = [GaussianMixture(n_components=n, covariance_type='ful
l').fit(abundance_pos_df) for n in n_estimators]
     11     bics = [clf.bic(u) for clf in clfs]
     12     aics = [clf.aic(u) for clf in clfs]

<ipython-input-10-cb10eef13407> in <listcomp>(.0)
      8 for i in range(20):
      9     n_estimators = np.arange(i*10 + 1, (i+1)*10)
---> 10     clfs = [GaussianMixture(n_components=n, covariance_type='ful
```