Confidence interval

Two numerical columns

In [17]:
```
import pandas as pd
import scipy.stats as st
```

In [18]:
```
data = pd.read_csv('star_dataset.csv')
```

In [19]:
```
data.head()
```

Out[19]:

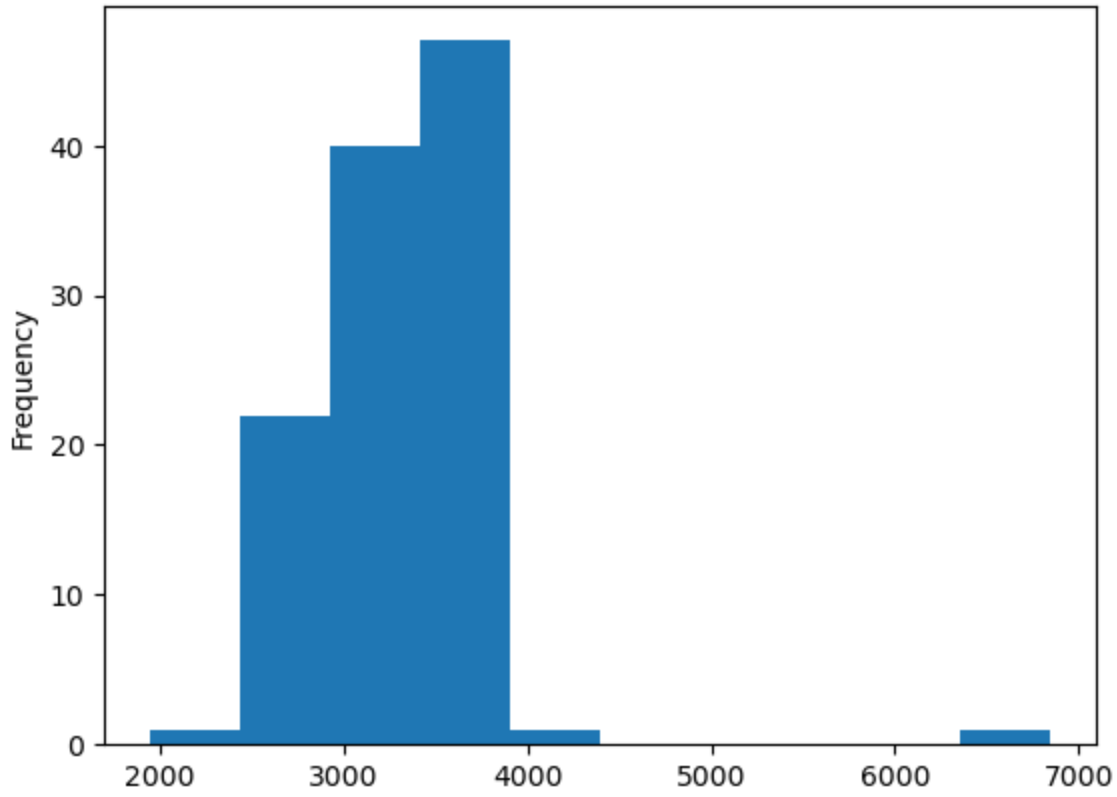
	Temperature (K)	Luminosity(L/Lo)	Radius(R/Ro)	Absolute magnitude(Mv)	Star type	Star color	Spectral Class
0	3068	0.002400	0.1700	16.12	Red Dwarf	Red	M
1	3042	0.000500	0.1542	16.60	Red Dwarf	Red	M
2	2600	0.000300	0.1020	18.70	Red Dwarf	Red	M
3	2800	0.000200	0.1600	16.65	Red Dwarf	Red	M
4	1939	0.000138	0.1030	20.06	Red Dwarf	Red	M

Er is niet echt een normal distribution te vinden in ongefilterde kolommen, dus ga ik kijken naar waarden gefilterd op hun type of hun kleur (zoals ook in assignment 07 te zien is).

Temperatuur voor rode sterren

In [38]:
```
filtered_T_data = data[data['Star color'] == 'Red']['Temperature (K)']
filtered_T_data.plot(kind='hist')
```

Out[38]:
```
<AxesSubplot:ylabel='Frequency'>
```



In [36]:
```
print('Mean: ' + str(filtered_T_data.mean()))
print('90% confidence: ' + str(st.t.interval(0.9, len(filtered_T_data)-1, loc=filtered_T
print('95% confidence: ' + str(st.t.interval(0.95, len(filtered_T_data)-1, loc=filtered_
print('99% confidence: ' + str(st.t.interval(0.99, len(filtered_T_data)-1, loc=filtered_
print('99.99% confidence: ' + str(st.t.interval(0.9999, len(filtered_T_data)-1, loc=filt
```

```
Mean: 3291.785714285714
90% confidence: (3214.86458578107, 3368.706842790358)
95% confidence: (3199.8916994243323, 3383.679729147096)
99% confidence: (3170.245375949639, 3413.3260526217896)
99.99% confidence: (3104.5870510336217, 3478.9843775378067)
```

Het vergroten van de confidence maakt het interval natuurlijk ook groter, aangezien je zekerder wilt zijn dat een waarde die je tegenkomt binnen dit interval ligt.

In [48]:
```
smaller_set = filtered_T_data[:int((len(filtered_T_data)/10))]
print('Count: ' + str(len(smaller_set)))
print('Mean smaller set: ' + str(smaller_set.mean()))
print('95% confidence on a smaller set: ' + str(st.t.interval(0.95, len(smaller_set)-1,
```

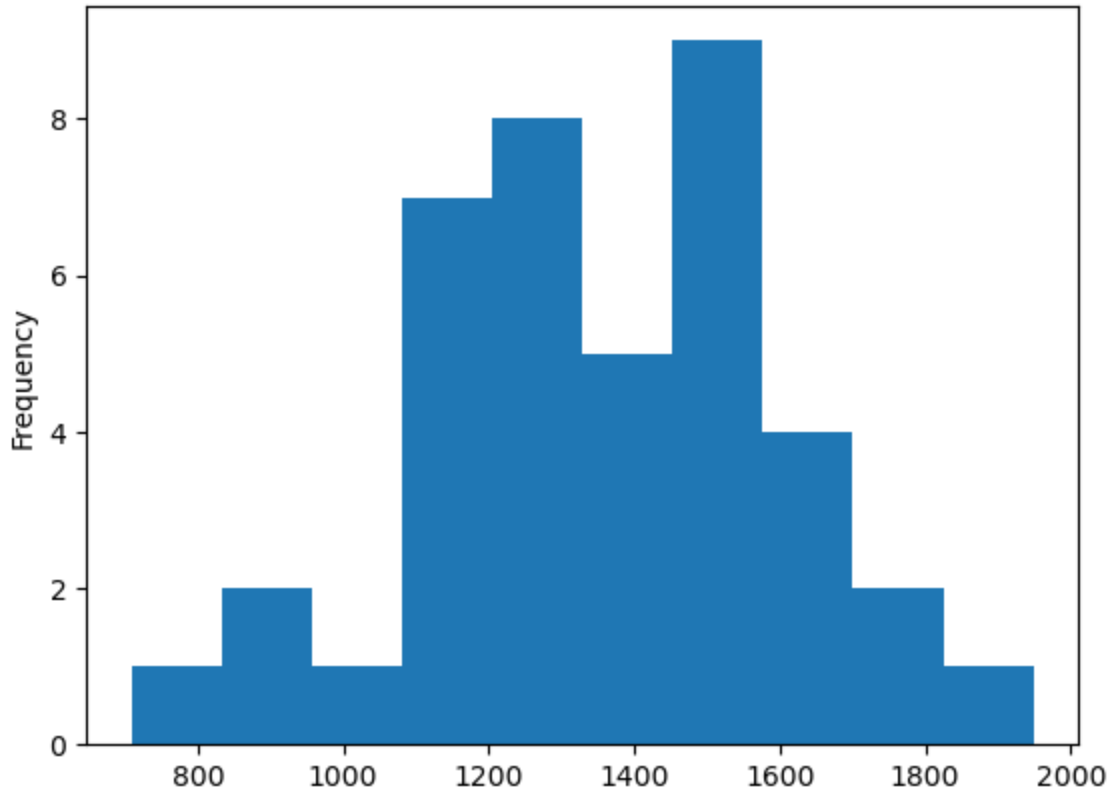
```
Count: 11
Mean smaller set: 2770.5454545454545
95% confidence on a smaller set: (2498.2484610328884, 3042.8424480580206)
```

Naast dat het gemiddelde veranderd is door de kleinere set, is ook de confidence interval groter geworden. Dit is logisch omdat minder waarden minder zekerheid geven dat een waarde binnen dat interval valt.

Radius voor hyper giants

In [46]:
```
filtered_R_data = data[data['Star type'] == 'Hyper Giant']['Radius(R/Ro)']
filtered_R_data.plot(kind='hist')
```

Out[46]:
```
<AxesSubplot:ylabel='Frequency'>
```



In [40]:
```
print('Mean: ' + str(filtered_R_data.mean()))
print('90% confidence: ' + str(st.t.interval(0.9, len(filtered_R_data)-1, loc=filtered_R
print('95% confidence: ' + str(st.t.interval(0.95, len(filtered_R_data)-1, loc=filtered_
print('99% confidence: ' + str(st.t.interval(0.99, len(filtered_R_data)-1, loc=filtered_
print('99.99% confidence: ' + str(st.t.interval(0.9999, len(filtered_R_data)-1, loc=filt
```

```
Mean: 1366.8975
90% confidence: (1298.8154055460816, 1434.9795944539185)
95% confidence: (1285.1650130576531, 1448.629986942347)
99% confidence: (1257.4766895198413, 1476.3183104801587)
99.99% confidence: (1191.8189402201092, 1541.9760597798909)
```

In [49]:
```
smaller_set = filtered_R_data[:int((len(filtered_R_data)/10))]
print('Count: ' + str(len(smaller_set)))
print('Mean smaller set: ' + str(smaller_set.mean()))
print('95% confidence on a smaller set: ' + str(st.t.interval(0.95, len(smaller_set)-1,
```

```
Count: 4
Mean smaller set: 1402.75
95% confidence on a smaller set: (1055.8746058225681, 1749.6253941774319)
```