Import

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
from sklearn.datasets import load_iris
%matplotlib inline
```

In [5]:
```
df = pd.read_csv("loan_data_set.csv")
```

In [6]:
```
df
```

Out[6]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coap
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
...	...	...	...	...	...	...	...	
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

614 rows × 13 columns

In [7]:
```
df.head()
```

Out[7]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coappl
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	

In [8]:
```
df.head(10)
```

Out[8]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coappl
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
5	LP001011	Male	Yes	2	Graduate	Yes	5417	
6	LP001013	Male	Yes	0	Not Graduate	No	2333	
7	LP001014	Male	Yes	3+	Graduate	No	3036	
8	LP001018	Male	Yes	2	Graduate	No	4006	
9	LP001020	Male	Yes	1	Graduate	No	12841	

In [9]:
```
df.tail()
```

Out[9]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coap
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

In [10]:
```
df.tail(10)
```

Out[10]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coap
604	LP002959	Female	Yes	1	Graduate	No	12000	
605	LP002960	Male	Yes	0	Not Graduate	No	2400	
606	LP002961	Male	Yes	1	Graduate	No	3400	
607	LP002964	Male	Yes	2	Not Graduate	No	3987	
608	LP002974	Male	Yes	0	Graduate	No	3232	
609	LP002978	Female	No	0	Graduate	No	2900	
610	LP002979	Male	Yes	3+	Graduate	No	4106	
611	LP002983	Male	Yes	1	Graduate	No	8072	
612	LP002984	Male	Yes	2	Graduate	No	7583	
613	LP002990	Female	No	0	Graduate	Yes	4583	

Data PreProcessing

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             601 non-null    object 
 2   Married            611 non-null    object 
 3   Dependents         599 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      582 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

In [13]:
```
df.columns
```

Out[13]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
      dtype='object')
```

In [14]:
```
df.shape
```

Out[14]:
```
(614, 13)
```

In [15]:
```
df.index
```

Out[15]:
```
RangeIndex(start=0, stop=614, step=1)
```

In [16]:
```
df.describe()
```

Out[16]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

In [17]:
```
df.isnull()
```

Out[17]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coapp
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
609	False	False	False	False	False	False	False	
610	False	False	False	False	False	False	False	
611	False	False	False	False	False	False	False	
612	False	False	False	False	False	False	False	
613	False	False	False	False	False	False	False	

614 rows × 13 columns

In [18]:
```
df.isnull().any()
```

Out[18]:
```
Loan_ID              False
Gender                True
Married               True
Dependents            True
Education            False
Self_Employed         True
ApplicantIncome      False
CoapplicantIncome    False
LoanAmount            True
Loan_Amount_Term      True
Credit_History        True
Property_Area        False
Loan_Status          False
dtype: bool
```

In [19]:
```
df.isnull().sum()
```

Out[19]:
```
Loan_ID               0
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```

In [22]:
```
df['Gender'].fillna('Not given', inplace = True)
```

In [24]:
```
df['Married'].fillna('Not given', inplace = True)
```

In [25]:
```
df['Dependents'].dropna()
```

Out[25]:
```
0       0
1       1
2       0
3       0
4       0
       ..
609     0
610    3+
611     1
612     2
613     0
Name: Dependents, Length: 599, dtype: object
```

In [27]:
```
df.fillna(0,inplace = True)
```

In [28]:
```
df.isnull().sum()
```

Out[28]:
```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

In [30]:
```
df.head(10)
```

Out[30]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Coappl
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	
5	LP001011	Male	Yes	2	Graduate	Yes	5417	
6	LP001013	Male	Yes	0	Not Graduate	No	2333	
7	LP001014	Male	Yes	3+	Graduate	No	3036	
8	LP001018	Male	Yes	2	Graduate	No	4006	
9	LP001020	Male	Yes	1	Graduate	No	12841	

Let us group the quantitative variables 'ApplicantIncome', 'Coapplicant Income', 'LoanAmount', 'Loan_Amount_Term', 'Credit_History' by 'Property_Area' categorical variable

In [31]:
```
df.describe().mean()
```

Out[31]:
```
ApplicantIncome      13220.187620
CoapplicantIncome     6289.280521
LoanAmount             241.407094
Loan_Amount_Term       323.798230
Credit_History          77.399056
dtype: float64
```

In [32]:
```
df.describe().min()
```

Out[32]:
```
ApplicantIncome      150.0
CoapplicantIncome      0.0
LoanAmount             0.0
Loan_Amount_Term       0.0
Credit_History         0.0
dtype: float64
```

In [33]:
```
df.describe().max()
```

Out[33]:
```
ApplicantIncome      81000.0
CoapplicantIncome    41667.0
LoanAmount             700.0
Loan_Amount_Term       614.0
Credit_History         614.0
dtype: float64
```

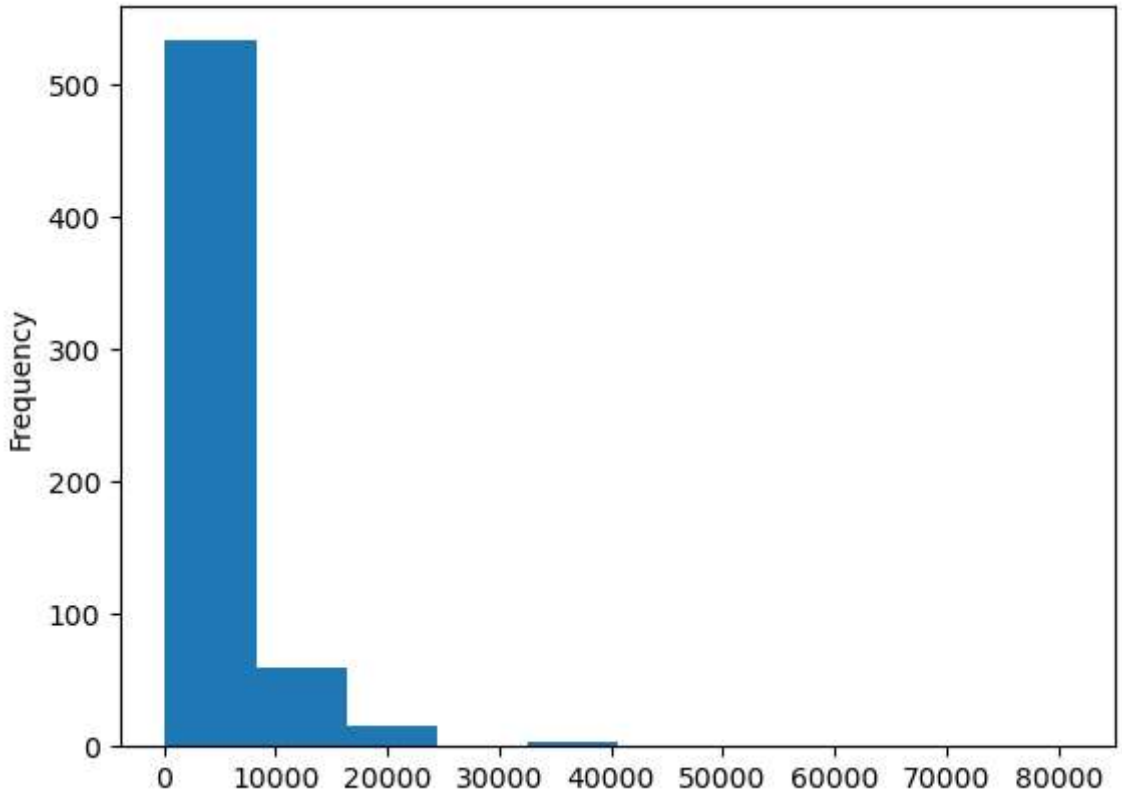
In [34]:
```
df.describe().median()
```

Out[34]:
```
ApplicantIncome      4607.979642
CoapplicantIncome    1404.872899
LoanAmount            133.083062
Loan_Amount_Term      360.000000
Credit_History          1.000000
dtype: float64
```

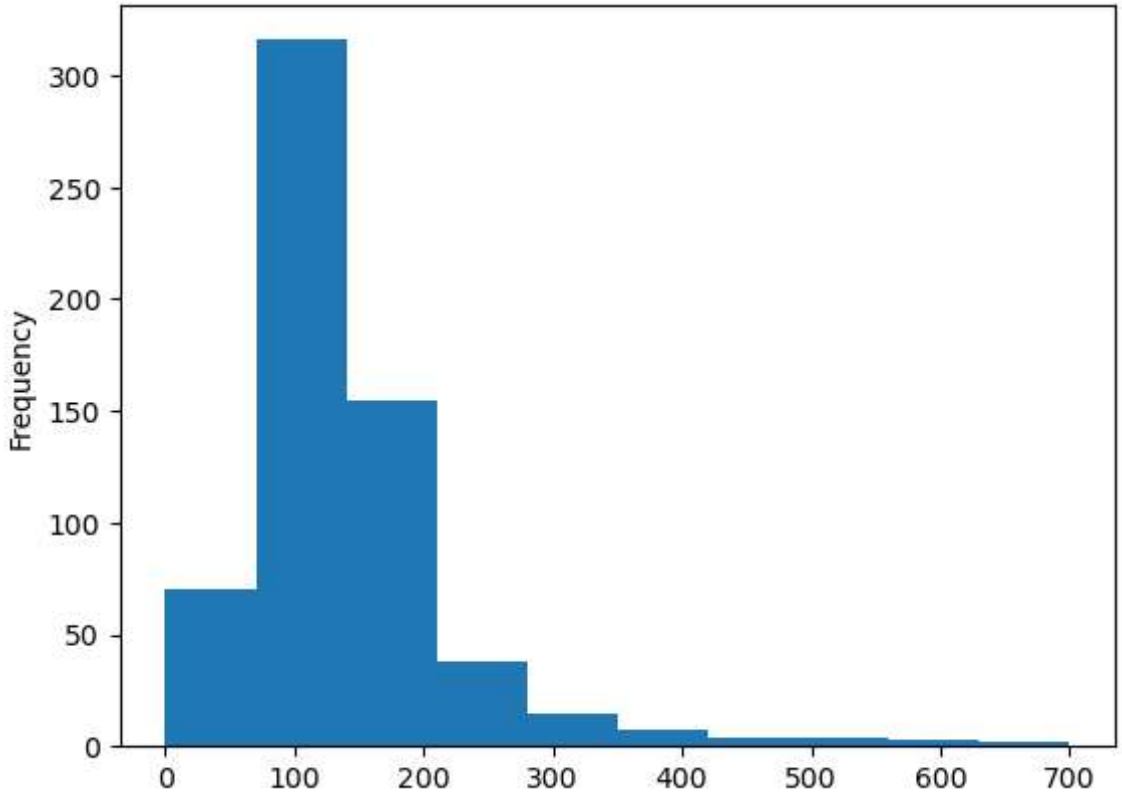
In [35]:
```
df.describe().std()
```

Out[35]:
```
ApplicantIncome      27480.194323
CoapplicantIncome    14332.564054
LoanAmount             262.101513
Loan_Amount_Term       198.522682
Credit_History         216.819827
dtype: float64
```

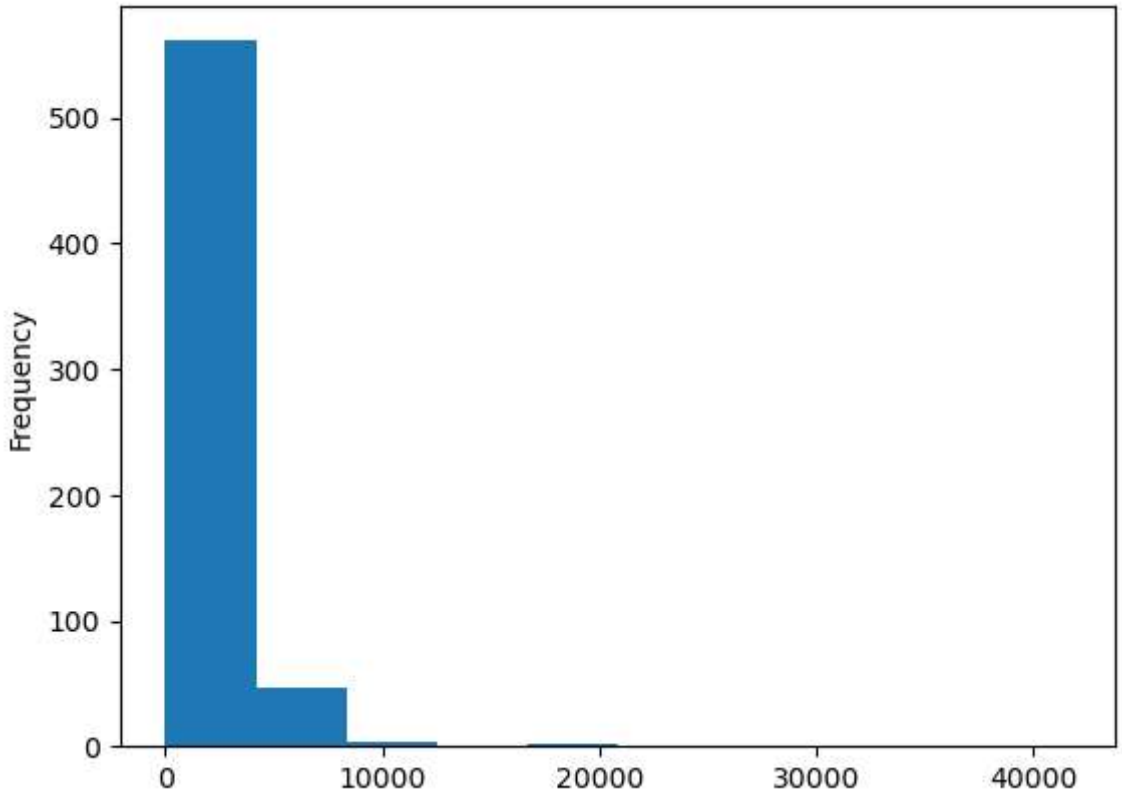
In [51]:
```
df["ApplicantIncome"].plot(kind="hist")
plt.show()
```



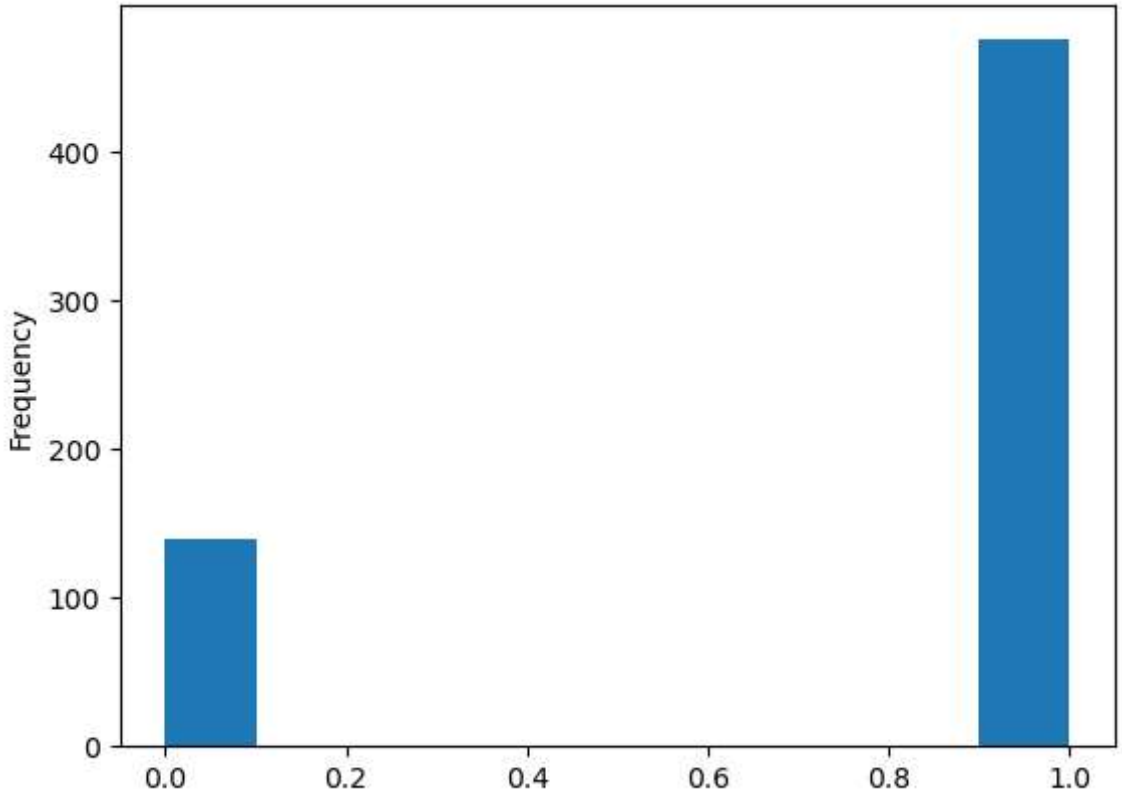
In [52]:
```
df["LoanAmount"].plot(kind="hist")
plt.show()
```



In [55]:
```
df["CoapplicantIncome"].plot(kind="hist")
plt.show()
```



In [56]:
```
df["Credit_History"].plot(kind="hist")
plt.show()
```

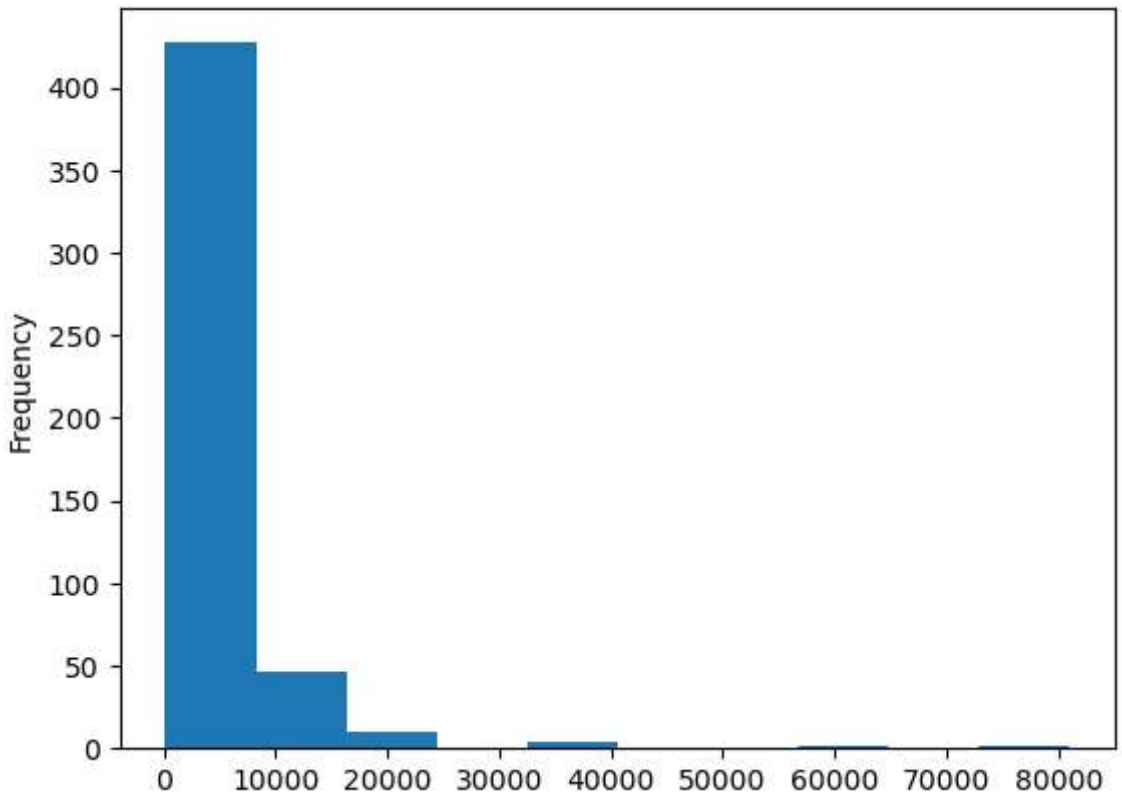


In [37]:
```
df[df.Gender == 'Male'].describe()
```

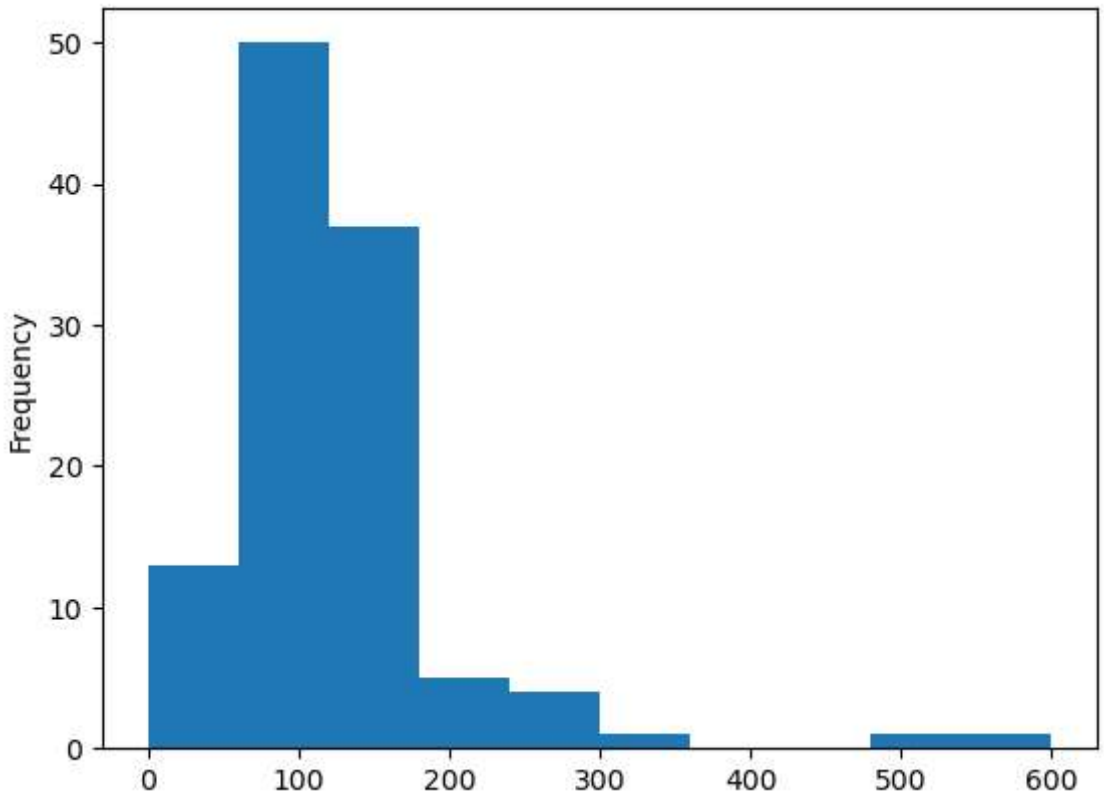
Out[37]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	489.000000	489.000000	489.000000	489.000000	489.000000
mean	5446.460123	1742.932352	143.466258	332.024540	0.781186
std	6185.789262	2606.507054	86.164988	83.316301	0.413866
min	150.000000	0.000000	0.000000	0.000000	0.000000
25%	2917.000000	0.000000	100.000000	360.000000	1.000000
50%	3865.000000	1430.000000	128.000000	360.000000	1.000000
75%	5923.000000	2436.000000	172.000000	360.000000	1.000000
max	81000.000000	33837.000000	650.000000	480.000000	1.000000

In [53]:
```
df.ApplicantIncome[df.Gender=="Male"].plot(kind="hist")
plt.show()
```



In [54]:
```
df.LoanAmount[df.Gender=="Female"].plot(kind="hist")
plt.show()
```



In [40]:
```
df1 = df.groupby('Property_Area').describe()
```

In [41]:
```
df1
```

Out[41]:

	ApplicantIncome								Coapp
	count	mean	std	min	25%	50%	75%	max	count
Property_Area									
Rural	179.0	5554.083799	6782.658637	150.0	2918.5	3975.0	6022.50	81000.0	179.0
Semiurban	233.0	5292.261803	5279.629359	210.0	2927.0	3859.0	5285.00	39999.0	233.0
Urban	202.0	5398.247525	6392.928779	416.0	2650.5	3505.0	5810.75	63337.0	202.0

3 rows × 40 columns

In [44]:
```
stats = df.groupby(df.Loan_Status).describe()
```

In [45]:
```
stats.LoanAmount
```

Out[45]:

Loan_Status	count	mean	std	min	25%	50%	75%	max
N	192.0	142.557292	90.495129	0.0	95.0	126.5	173.0	570.0
Y	422.0	140.533175	87.444357	0.0	99.0	125.0	160.0	700.0

In [46]:
```
df.dtypes
```

Out[46]:
```
Loan_ID               object
Gender                object
Married               object
Dependents            object
Education             object
Self_Employed         object
ApplicantIncome        int64
CoapplicantIncome    float64
LoanAmount           float64
Loan_Amount_Term     float64
Credit_History       float64
Property_Area         object
Loan_Status           object
dtype: object
```

In [47]:
```
stats.CoapplicantIncome
```

Out[47]:

Loan_Status	count	mean	std	min	25%	50%	75%	max
N	192.0	1877.807292	4384.060103	0.0	0.0	268.0	2273.75	41667.0
Y	422.0	1504.516398	1924.754855	0.0	0.0	1239.5	2297.25	20000.0

In [48]:
```
stats.Loan_Amount_Term
```

Out[48]:

Loan_Status	count	mean	std	min	25%	50%	75%	max
N	192.0	333.312500	90.807352	0.0	360.0	360.0	360.0	480.0
Y	422.0	334.606635	78.057119	0.0	360.0	360.0	360.0	480.0

In [49]:
```
stats.Credit_History
```

Out[49]:

Loan_Status	count	mean	std	min	25%	50%	75%	max
N	192.0	0.505208	0.501280	0.0	0.0	1.0	1.0	1.0
Y	422.0	0.895735	0.305967	0.0	1.0	1.0	1.0	1.0

In [50]:
```
stats.ApplicantIncome
```

Out[50]:

Loan_Status	count	mean	std	min	25%	50%	75%	max
N	192.0	5446.078125	6819.558528	150.0	2885.0	3833.5	5861.25	81000.0
Y	422.0	5384.068720	5765.441615	210.0	2877.5	3812.5	5771.50	63337.0

Iris Dataset

In [58]:
```
data=pd.read_csv('Iris.csv')
```

In [59]:
```
data
```

Out[59]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [60]:
```
data.head(10)
```

Out[60]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

In [65]:
```
data.tail(10)
```

Out[65]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
140	141	6.7	3.1	5.6	2.4	Iris-virginica
141	142	6.9	3.1	5.1	2.3	Iris-virginica
142	143	5.8	2.7	5.1	1.9	Iris-virginica
143	144	6.8	3.2	5.9	2.3	Iris-virginica
144	145	6.7	3.3	5.7	2.5	Iris-virginica
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [66]:
```
data.columns
```

Out[66]:
```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthC
m',
       'Species'],
      dtype='object')
```

In [67]:
```
data.shape
```

Out[67]:
```
(150, 6)
```

In [76]:
```
data.describe(include='all')
```

Out[76]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
count	150.000000	150.000000	150.000000	150.000000	150.000000	150
unique	NaN	NaN	NaN	NaN	NaN	3
top	NaN	NaN	NaN	NaN	NaN	Iris-setosa
freq	NaN	NaN	NaN	NaN	NaN	50
mean	75.500000	5.843333	3.054000	3.758667	1.198667	NaN
std	43.445368	0.828066	0.433594	1.764420	0.763161	NaN
min	1.000000	4.300000	2.000000	1.000000	0.100000	NaN
25%	38.250000	5.100000	2.800000	1.600000	0.300000	NaN
50%	75.500000	5.800000	3.000000	4.350000	1.300000	NaN
75%	112.750000	6.400000	3.300000	5.100000	1.800000	NaN
max	150.000000	7.900000	4.400000	6.900000	2.500000	NaN

In [77]:
```
Iris = data.groupby(data.Species).describe(include='all')
```

In [78]:
```
Iris.SepalWidthCm
```

Out[78]:

Species	count	mean	std	min	25%	50%	75%	max
Iris-setosa	50.0	3.418	0.381024	2.3	3.125	3.4	3.675	4.4
Iris-versicolor	50.0	2.770	0.313798	2.0	2.525	2.8	3.000	3.4
Iris-virginica	50.0	2.974	0.322497	2.2	2.800	3.0	3.175	3.8

In [79]:
```
Iris.SepalLengthCm
```

Out[79]:

Species	count	mean	std	min	25%	50%	75%	max
Iris-setosa	50.0	5.006	0.352490	4.3	4.800	5.0	5.2	5.8
Iris-versicolor	50.0	5.936	0.516171	4.9	5.600	5.9	6.3	7.0
Iris-virginica	50.0	6.588	0.635880	4.9	6.225	6.5	6.9	7.9

In [80]:
```
Iris.PetalLengthCm
```

Out[80]:

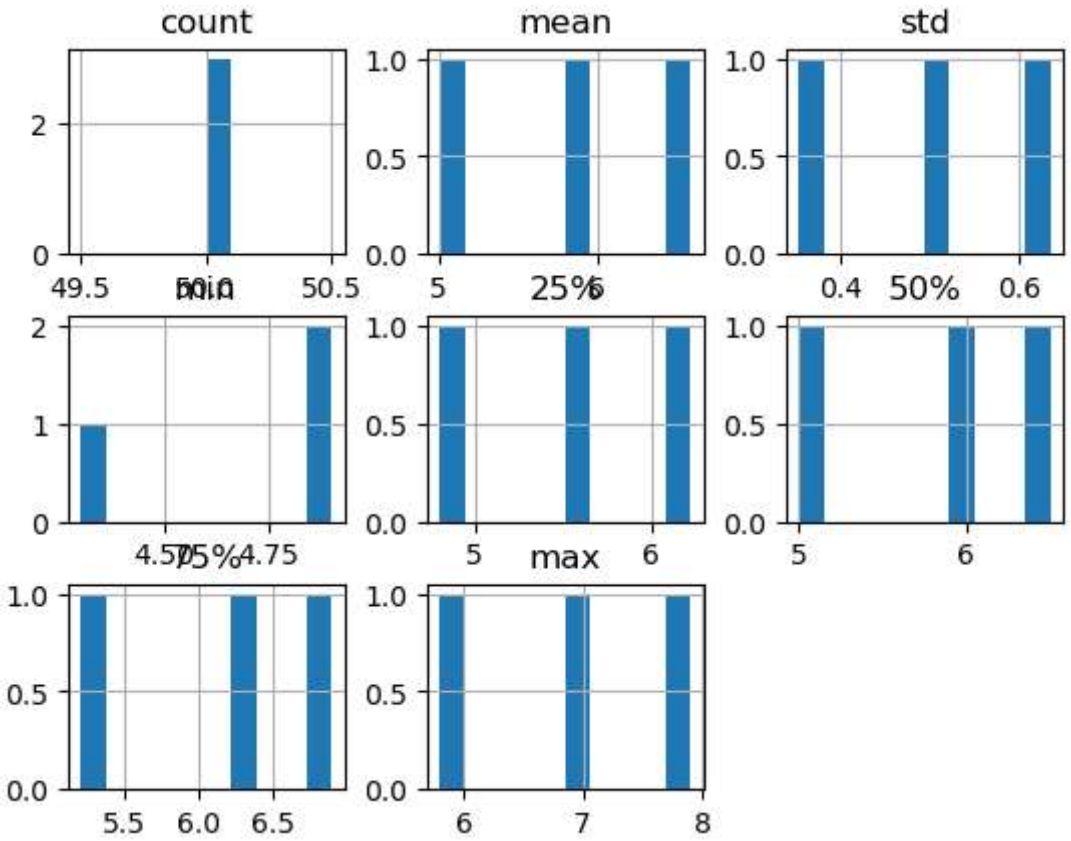
Species	count	mean	std	min	25%	50%	75%	max
Iris-setosa	50.0	1.464	0.173511	1.0	1.4	1.50	1.575	1.9
Iris-versicolor	50.0	4.260	0.469911	3.0	4.0	4.35	4.600	5.1
Iris-virginica	50.0	5.552	0.551895	4.5	5.1	5.55	5.875	6.9

In [81]:
```
Iris.PetalWidthCm
```

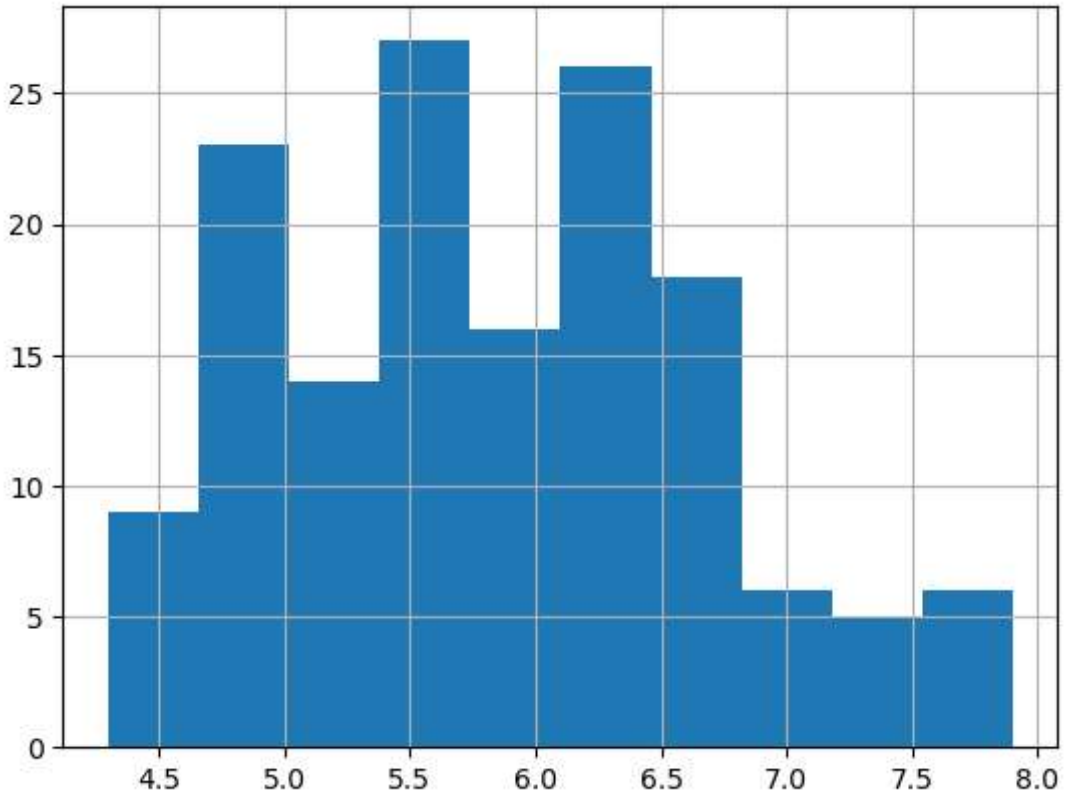
Out[81]:

Species	count	mean	std	min	25%	50%	75%	max
Iris-setosa	50.0	0.244	0.107210	0.1	0.2	0.2	0.3	0.6
Iris-versicolor	50.0	1.326	0.197753	1.0	1.2	1.3	1.5	1.8
Iris-virginica	50.0	2.026	0.274650	1.4	1.8	2.0	2.3	2.5

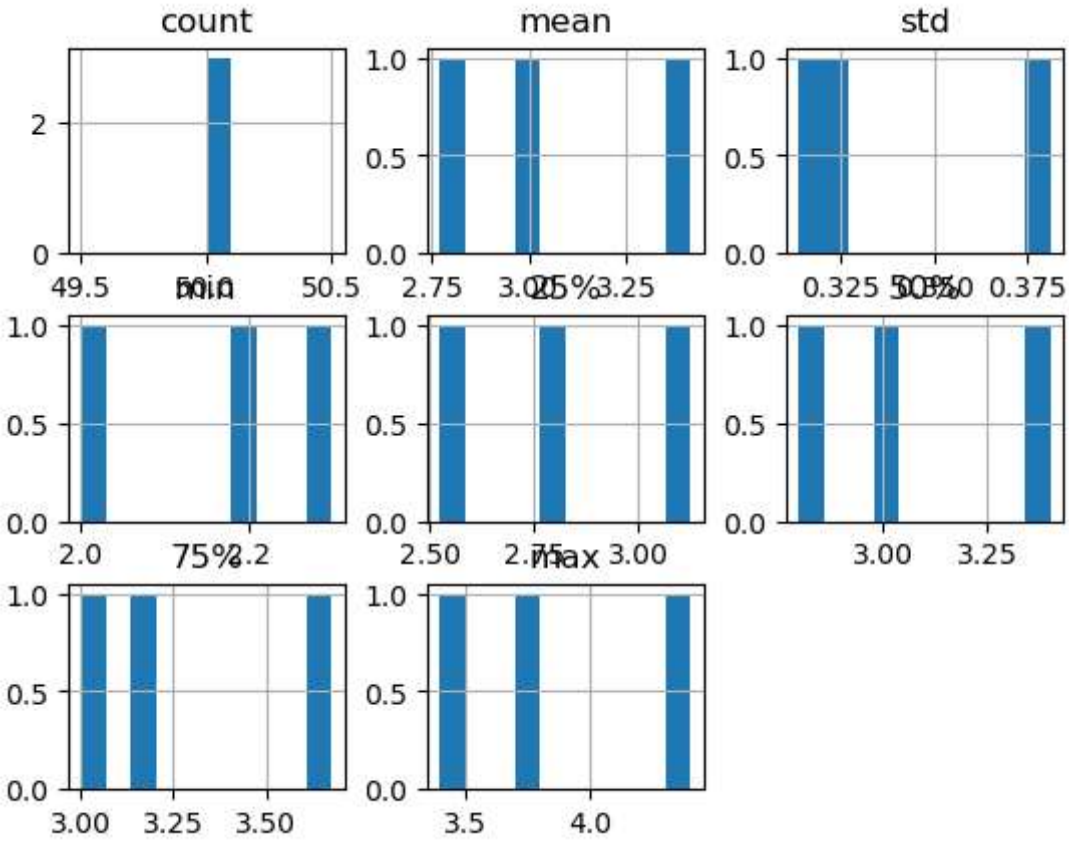
In [83]:
```
Iris['SepalLengthCm'].hist()
plt.show()
```



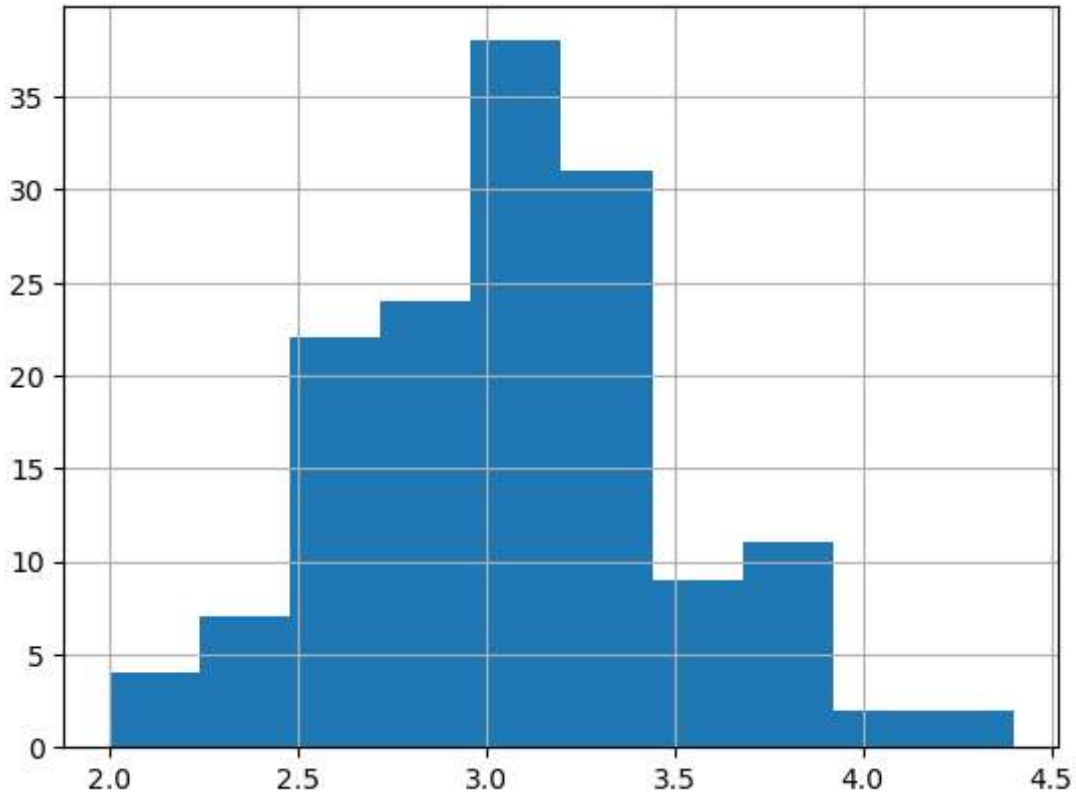
In [84]:
```
data['SepalLengthCm'].hist()
plt.show()
```



In [85]:
```
Iris['SepalWidthCm'].hist()
plt.show()
```



In [87]:
```
data['SepalWidthCm'].hist()
plt.show()
```



In [ ]: