Load the data from wine dataset. Check whether all attributes are standardized or not (mean is 0 and standard deviation is 1). If not, standardize the attributes. Do the same with Iris dataset.

Wine Dataset

In [56]:

```
import pandas as pd
import matplotlib.pyplot as plt
wine=pd.read_csv('https://gist.githubusercontent.com/tijptjik/9408623/raw/b237fa5848349
a14a14e5d4107dc7897c21951f5/wine.csv')
wine
```

Out[56]:

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nonflavanoid.phenols
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39
...	...	...	...	...	...	...	...	...	...
173	3	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52
174	3	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43
175	3	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43
176	3	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53
177	3	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56

178 rows × 14 columns

Checking standardization

In [57]:

```
pd.set_option("display.max_rows", None, "display.max_columns", None)
wine.groupby('Wine').describe().transpose()
```

Out[57]:

	Wine	1	2	3
Alcohol	**count**	59.000000	71.000000	48.000000
	mean	13.744746	12.278732	13.153750
	std	0.462125	0.537964	0.530241
	min	12.850000	11.030000	12.200000
	25%	13.400000	11.915000	12.805000
	50%	13.750000	12.290000	13.165000
	75%	14.100000	12.515000	13.505000
	max	14.830000	13.860000	14.340000
Malic.acid	**count**	59.000000	71.000000	48.000000
	mean	2.010678	1.932676	3.333750
	std	0.688549	1.015569	1.087906
	min	1.350000	0.740000	1.240000
	25%	1.665000	1.270000	2.587500
	50%	1.770000	1.610000	3.265000
	75%	1.935000	2.145000	3.957500
	max	4.040000	5.800000	5.650000
Ash	**count**	59.000000	71.000000	48.000000
	mean	2.455593	2.244789	2.437083
	std	0.227166	0.315467	0.184690
	min	2.040000	1.360000	2.100000
	25%	2.295000	2.000000	2.300000
	50%	2.440000	2.240000	2.380000
	75%	2.615000	2.420000	2.602500
	max	3.220000	3.230000	2.860000
Acl	**count**	59.000000	71.000000	48.000000
	mean	17.037288	20.238028	21.416667
	std	2.546322	3.349770	2.258161
	min	11.200000	10.600000	17.500000
	25%	16.000000	18.000000	20.000000
	50%	16.800000	20.000000	21.000000
	75%	18.700000	22.000000	23.000000
	max	25.000000	30.000000	27.000000
Mg	**count**	59.000000	71.000000	48.000000
	mean	106.338983	94.549296	99.312500
	std	10.498949	16.753497	10.890473
	min	89.000000	70.000000	80.000000
	25%	98.000000	85.500000	89.750000

	Wine	1	2	3
	50%	104.000000	88.000000	97.000000
	75%	114.000000	99.500000	106.000000
	max	132.000000	162.000000	123.000000
Phenols	count	59.000000	71.000000	48.000000
	mean	2.840169	2.258873	1.678750
	std	0.338961	0.545361	0.356971
	min	2.200000	1.100000	0.980000
	25%	2.600000	1.895000	1.407500
	50%	2.800000	2.200000	1.635000
	75%	3.000000	2.560000	1.807500
	max	3.880000	3.520000	2.800000
Flavanoids	count	59.000000	71.000000	48.000000
	mean	2.982373	2.080845	0.781458
	std	0.397494	0.705701	0.293504
	min	2.190000	0.570000	0.340000
	25%	2.680000	1.605000	0.580000
	50%	2.980000	2.030000	0.685000
	75%	3.245000	2.475000	0.920000
	max	3.930000	5.080000	1.570000
Nonflavanoid.phenols	count	59.000000	71.000000	48.000000
	mean	0.290000	0.363662	0.447500
	std	0.070049	0.123961	0.124140
	min	0.170000	0.130000	0.170000
	25%	0.255000	0.270000	0.397500
	50%	0.290000	0.370000	0.470000
	75%	0.320000	0.430000	0.530000
	max	0.500000	0.660000	0.630000
Proanth	count	59.000000	71.000000	48.000000
	mean	1.899322	1.630282	1.153542
	std	0.412109	0.602068	0.408836
	min	1.250000	0.410000	0.550000
	25%	1.640000	1.350000	0.855000
	50%	1.870000	1.610000	1.105000
	75%	2.090000	1.885000	1.350000
	max	2.960000	3.580000	2.700000
Color.int	count	59.000000	71.000000	48.000000
	mean	5.528305	3.086620	7.396250
	std	1.238573	0.924929	2.310942
	min	3.520000	1.280000	3.850000

	Wine	1	2	3
	25%	4.550000	2.535000	5.437500
	50%	5.400000	2.900000	7.550000
	75%	6.225000	3.400000	9.225000
	max	8.900000	6.000000	13.000000
Hue	**count**	59.000000	71.000000	48.000000
	mean	1.062034	1.056282	0.682708
	std	0.116483	0.202937	0.114441
	min	0.820000	0.690000	0.480000
	25%	0.995000	0.925000	0.587500
	50%	1.070000	1.040000	0.665000
	75%	1.130000	1.205000	0.752500
	max	1.280000	1.710000	0.960000
OD	**count**	59.000000	71.000000	48.000000
	mean	3.157797	2.785352	1.683542
	std	0.357077	0.496573	0.272111
	min	2.510000	1.590000	1.270000
	25%	2.870000	2.440000	1.510000
	50%	3.170000	2.830000	1.660000
	75%	3.420000	3.160000	1.820000
	max	4.000000	3.690000	2.470000
Proline	**count**	59.000000	71.000000	48.000000
	mean	1115.711864	519.507042	629.895833
	std	221.520767	157.211220	115.097043
	min	680.000000	278.000000	415.000000
	25%	987.500000	406.500000	545.000000
	50%	1095.000000	495.000000	627.500000
	75%	1280.000000	625.000000	695.000000
	max	1680.000000	985.000000	880.000000

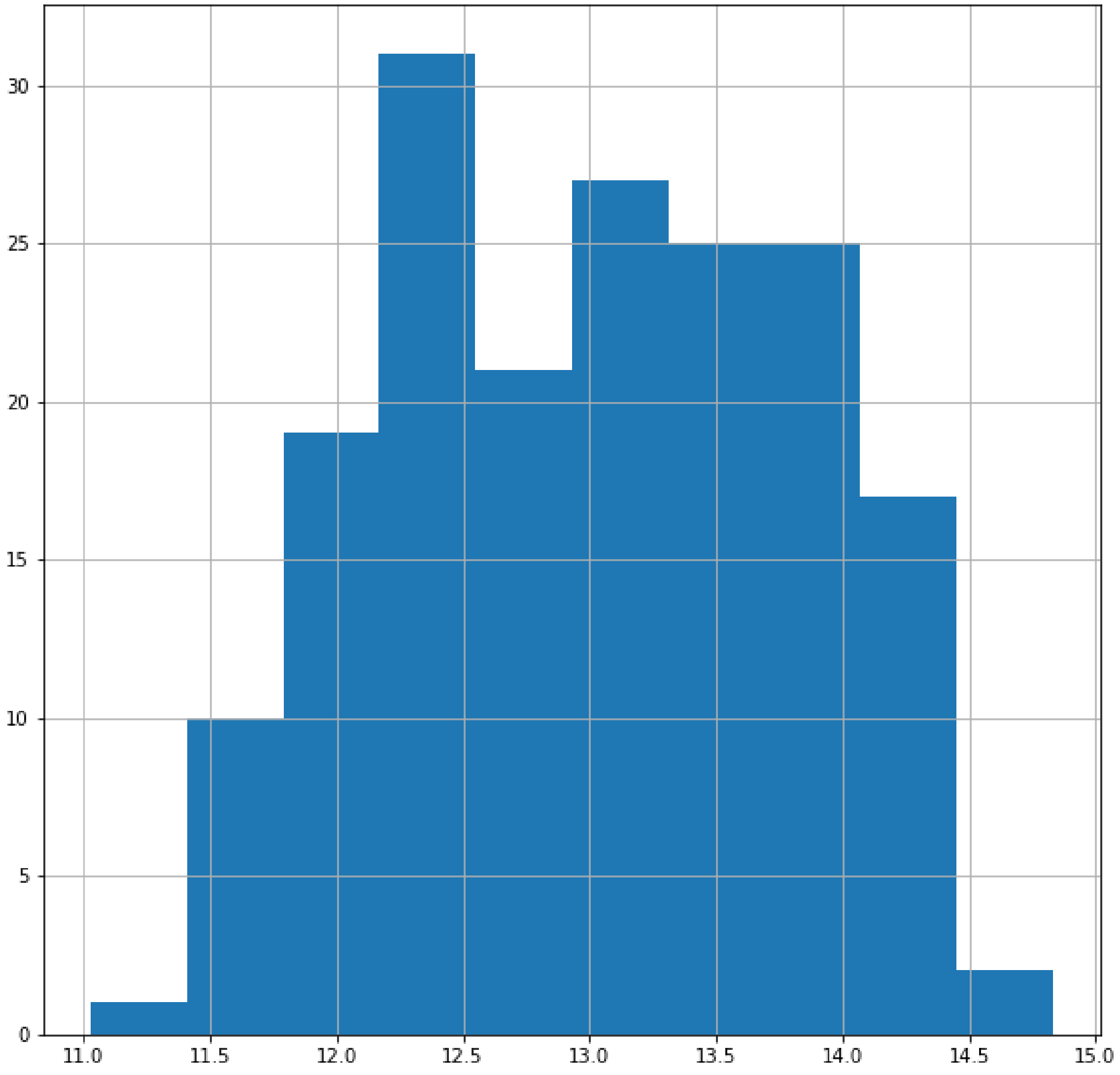
Visualization

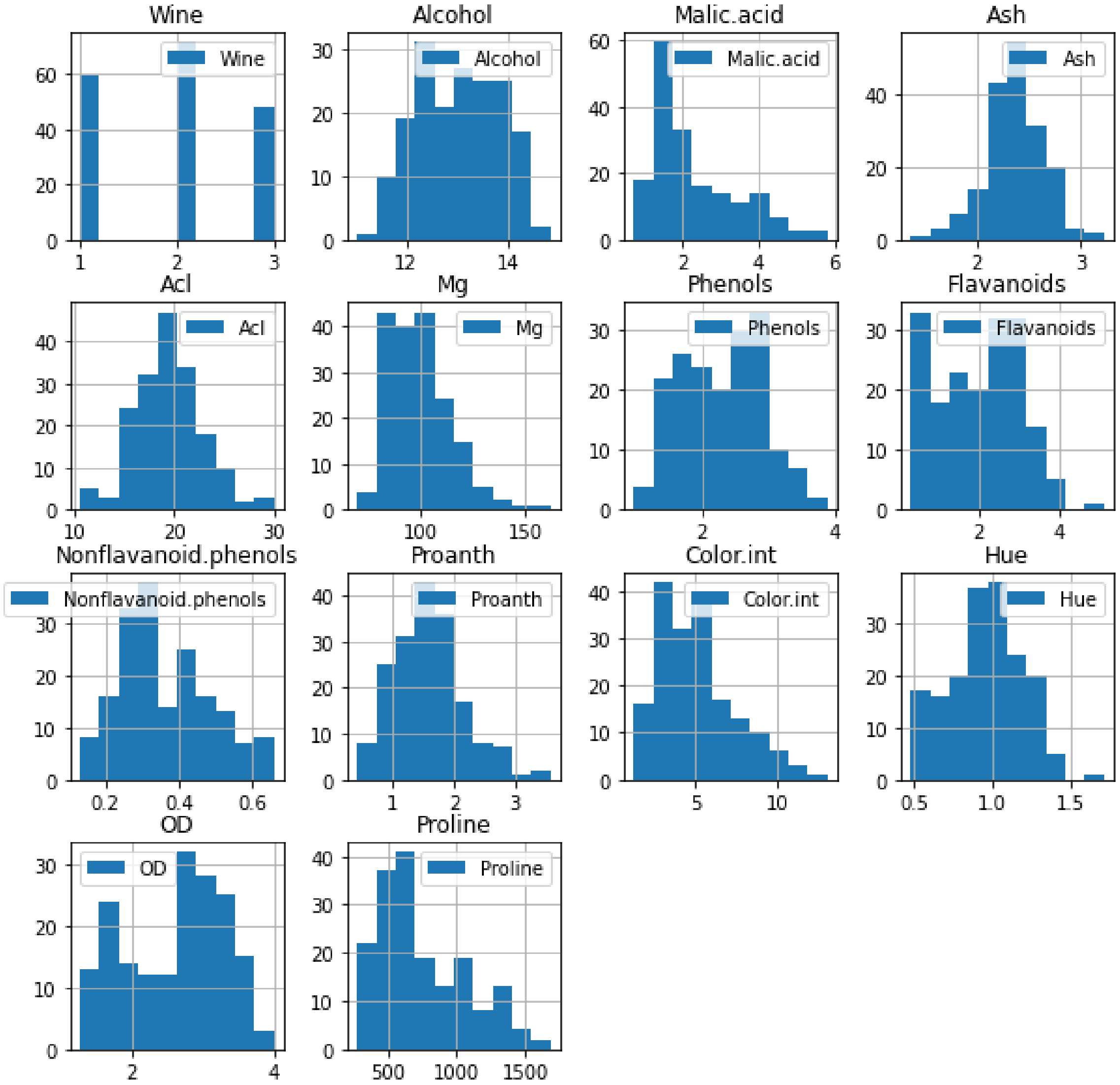
Before Standardisation

In [58]:

```
wine['Alcohol'].hist(figsize=(10,10))
wine.hist(figsize=(10,10),legend=True)
for i in range(1,wine.shape[1]):
  column=wine.columns[i]
  print("Mean",column,round(wine[column].mean(),4))
```

```
Mean Alcohol 13.0006
Mean Malic.acid 2.3363
Mean Ash 2.3665
Mean Acl 19.4949
Mean Mg 99.7416
Mean Phenols 2.2951
Mean Flavanoids 2.0293
Mean Nonflavanoid.phenols 0.3619
Mean Proanth 1.5909
Mean Color.int 5.0581
Mean Hue 0.9574
Mean OD 2.6117
Mean Proline 746.8933
```





Scaling/Standardising Data

In [59]:

```
pd.set_option("display.max_rows", 10, "display.max_columns", None)
from sklearn.preprocessing import scale

l = []
for i in wine.columns:
  if wine[i].mean() != 0 and wine[i].std() != 1 and i != 'Wine':
    l.append(True)
  else:
    l.append(False)

std_wine = wine[wine.columns[l]]
std_wine
```

Out[59]:

	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Nonflavanoid.phenols	Proant
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.2
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.2
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.8
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.1
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.8
...	...	...	...	...	...	...	...	...	.
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.0
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.4
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.3
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.4
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.3

178 rows × 13 columns

In [60]:

```
pd.set_option("display.max_rows", 10, "display.max_columns", None)
std_wine = scale(std_wine)
wine[wine.columns[1]] = pd.DataFrame(std_wine,columns=wine.columns[1])
wine
```

Out[60]:

	Wine	Alcohol	Malic.acid	Ash	Acl	Mg	Phenols	Flavanoids	Non
0	1	1.518613	-0.562250	0.232053	-1.169593	1.913905	0.808997	1.034819	
1	1	0.246290	-0.499413	-0.827996	-2.490847	0.018145	0.568648	0.733629	
2	1	0.196879	0.021231	1.109334	-0.268738	0.088358	0.808997	1.215533	
3	1	1.691550	-0.346811	0.487926	-0.809251	0.930918	2.491446	1.466525	
4	1	0.295700	0.227694	1.840403	0.451946	1.281985	0.808997	0.663351	
...	...	...	...	...	...	...	...	...	
173	3	0.876275	2.974543	0.305159	0.301803	-0.332922	-0.985614	-1.424900	
174	3	0.493343	1.412609	0.414820	1.052516	0.158572	-0.793334	-1.284344	
175	3	0.332758	1.744744	-0.389355	0.151661	1.422412	-1.129824	-1.344582	
176	3	0.209232	0.227694	0.012732	0.151661	1.422412	-1.033684	-1.354622	
177	3	1.395086	1.583165	1.365208	1.502943	-0.262708	-0.392751	-1.274305	

178 rows × 14 columns

In [61]:

```
pd.set_option("display.max_rows", None, "display.max_columns", None)
wine.groupby('Wine').describe().transpose()
```

Out[61]:

	Wine	1	2	3
Alcohol	**count**	59.000000	71.000000	48.000000
	mean	0.919195	-0.891720	0.189159
	std	0.570847	0.664528	0.654989
	min	-0.186053	-2.434235	-0.988975
	25%	0.493343	-1.341026	-0.241640
	50%	0.925685	-0.877801	0.203055
	75%	1.358028	-0.599867	0.623045
	max	2.259772	1.061565	1.654492
Malic.acid	**count**	59.000000	71.000000	48.000000
	mean	-0.292342	-0.362362	0.895331
	std	0.618085	0.911639	0.976573
	min	-0.885409	-1.432983	-0.984151
	25%	-0.602645	-0.957222	0.225450
	50%	-0.508390	-0.652016	0.833616
	75%	-0.360276	-0.171766	1.455248
	max	1.529305	3.109192	2.974543
Ash	**count**	59.000000	71.000000	48.000000
	mean	0.325604	-0.444958	0.257945
	std	0.830369	1.153140	0.675106
	min	-1.193530	-3.679162	-0.974210
	25%	-0.261418	-1.339744	-0.243142
	50%	0.268606	-0.462462	0.049285
	75%	0.908291	0.195499	0.862599
	max	3.119772	3.156325	1.803849
Acl	**count**	59.000000	71.000000	48.000000
	mean	-0.737997	0.223137	0.577065
	std	0.764622	1.005886	0.678092
	min	-2.490847	-2.671018	-0.599052
	25%	-1.049479	-0.448909	0.151661
	50%	-0.809251	0.151661	0.451946
	75%	-0.238710	0.752231	1.052516
	max	1.653086	3.154511	2.253656
Mg	**count**	59.000000	71.000000	48.000000
	mean	0.463226	-0.364567	-0.030127
	std	0.737166	1.176319	0.764656
	min	-0.754202	-2.088255	-1.386122
	25%	-0.122282	-0.999948	-0.701542

	Wine	1	2	3
	50%	0.298998	-0.824415	-0.192495
	75%	1.001132	-0.016962	0.439425
	max	2.264972	4.371372	1.633052
Phenols	count	59.000000	71.000000	48.000000
	mean	0.873362	-0.058067	-0.987617
	std	0.543128	0.873849	0.571986
	min	-0.152402	-1.914966	-2.107246
	25%	0.488531	-0.641113	-1.422249
	50%	0.808997	-0.152402	-1.057719
	75%	1.129464	0.424438	-0.781317
	max	2.539515	1.962676	0.808997
Flavanoids	count	59.000000	71.000000	48.000000
	mean	0.956884	0.051780	-1.252761
	std	0.399070	0.708500	0.294668
	min	0.161368	-1.465058	-1.695971
	25%	0.653312	-0.425953	-1.455019
	50%	0.954502	0.000733	-1.349602
	75%	1.220553	0.447498	-1.113670
	max	1.908270	3.062832	-0.461091
Nonflavanoid.phenols	count	59.000000	71.000000	48.000000
	mean	-0.578985	0.014569	0.690119
	std	0.564443	0.998856	1.000293
	min	-1.545922	-1.868234	-1.545922
	25%	-0.861008	-0.740141	0.287229
	50%	-0.578985	0.065639	0.871420
	75%	-0.337251	0.549108	1.354888
	max	1.113154	2.402403	2.160669
Proanth	count	59.000000	71.000000	48.000000
	mean	0.540383	0.069002	-0.766287
	std	0.722050	1.054873	0.716315
	min	-0.597284	-2.069034	-1.823742
	25%	0.086029	-0.422075	-1.289357
	50%	0.489009	0.033467	-0.851336
	75%	0.874467	0.515290	-0.422075
	max	2.398780	3.485073	1.943238
Color.int	count	59.000000	71.000000	48.000000
	mean	0.203401	-0.852799	1.011418
	std	0.535769	0.400097	0.999644
	min	-0.665332	-1.634288	-0.522583

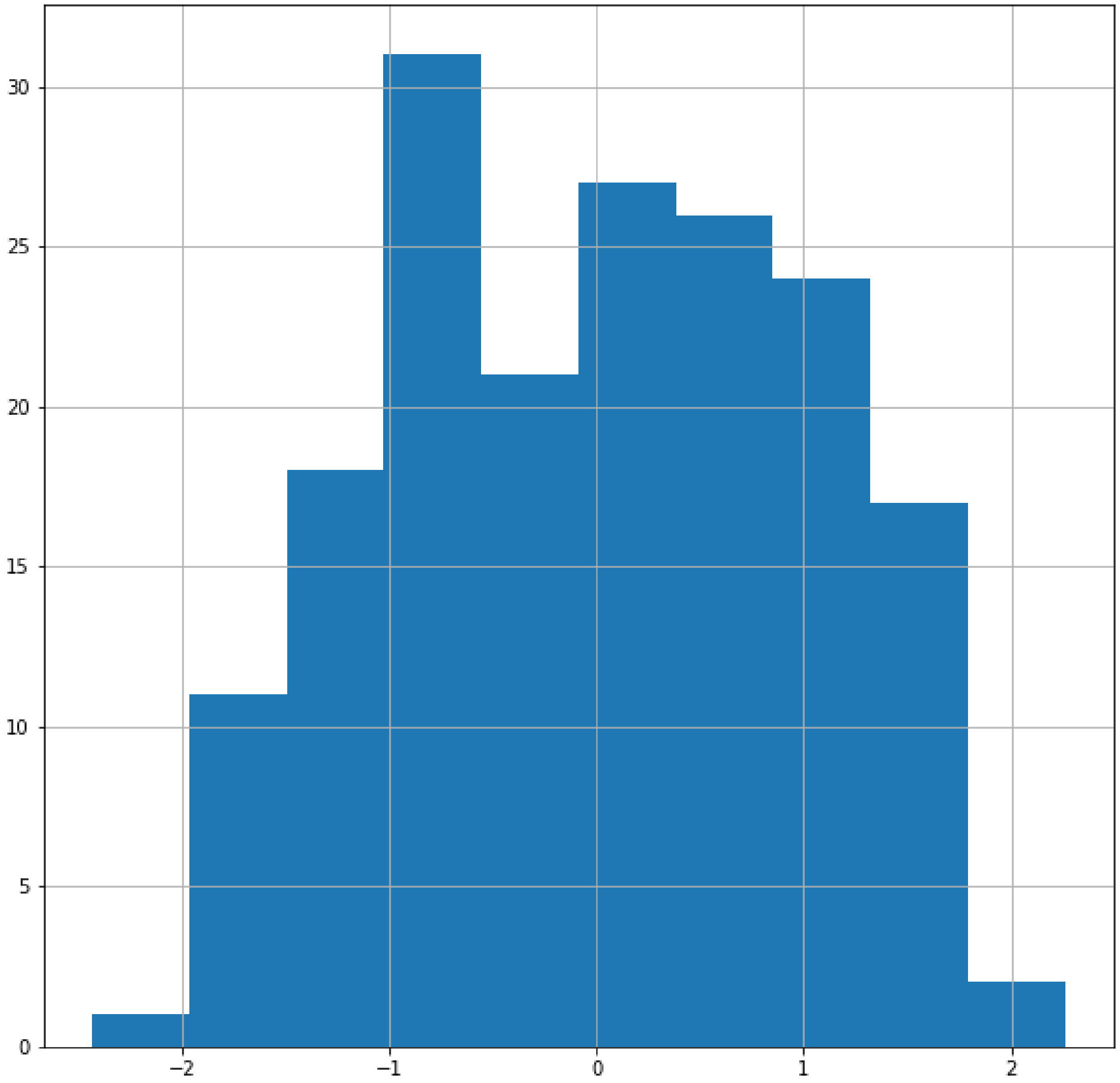
	Wine	1	2	3
	25%	-0.219784	-1.091413	0.164121
	50%	0.147900	-0.933525	1.077926
	75%	0.504770	-0.717240	1.802480
	max	1.661895	0.407442	3.435432
Hue	**count**	59.000000	71.000000	48.000000
	mean	0.458847	0.433611	-1.205382
	std	0.511049	0.890353	0.502092
	min	-0.603037	-1.173391	-2.094732
	25%	0.164747	-0.142367	-1.623093
	50%	0.493797	0.362177	-1.283075
	75%	0.757038	1.086088	-0.899183
	max	1.415139	3.301694	0.011190
OD	**count**	59.000000	71.000000	48.000000
	mean	0.771351	0.245294	-1.310950
	std	0.504350	0.701382	0.384342
	min	-0.143625	-1.443072	-1.895054
	25%	0.364855	-0.242496	-1.556068
	50%	0.788587	0.308357	-1.344201
	75%	1.141698	0.774463	-1.118210
	max	1.960915	1.523058	-0.200123
Proline	**count**	59.000000	71.000000	48.000000
	mean	1.174501	-0.724110	-0.372578
	std	0.705431	0.500638	0.366526
	min	-0.213021	-1.493188	-1.056912
	25%	0.766211	-1.083980	-0.642928
	50%	1.108544	-0.802153	-0.380207
	75%	1.697675	-0.388168	-0.165254
	max	2.971473	0.758249	0.423878

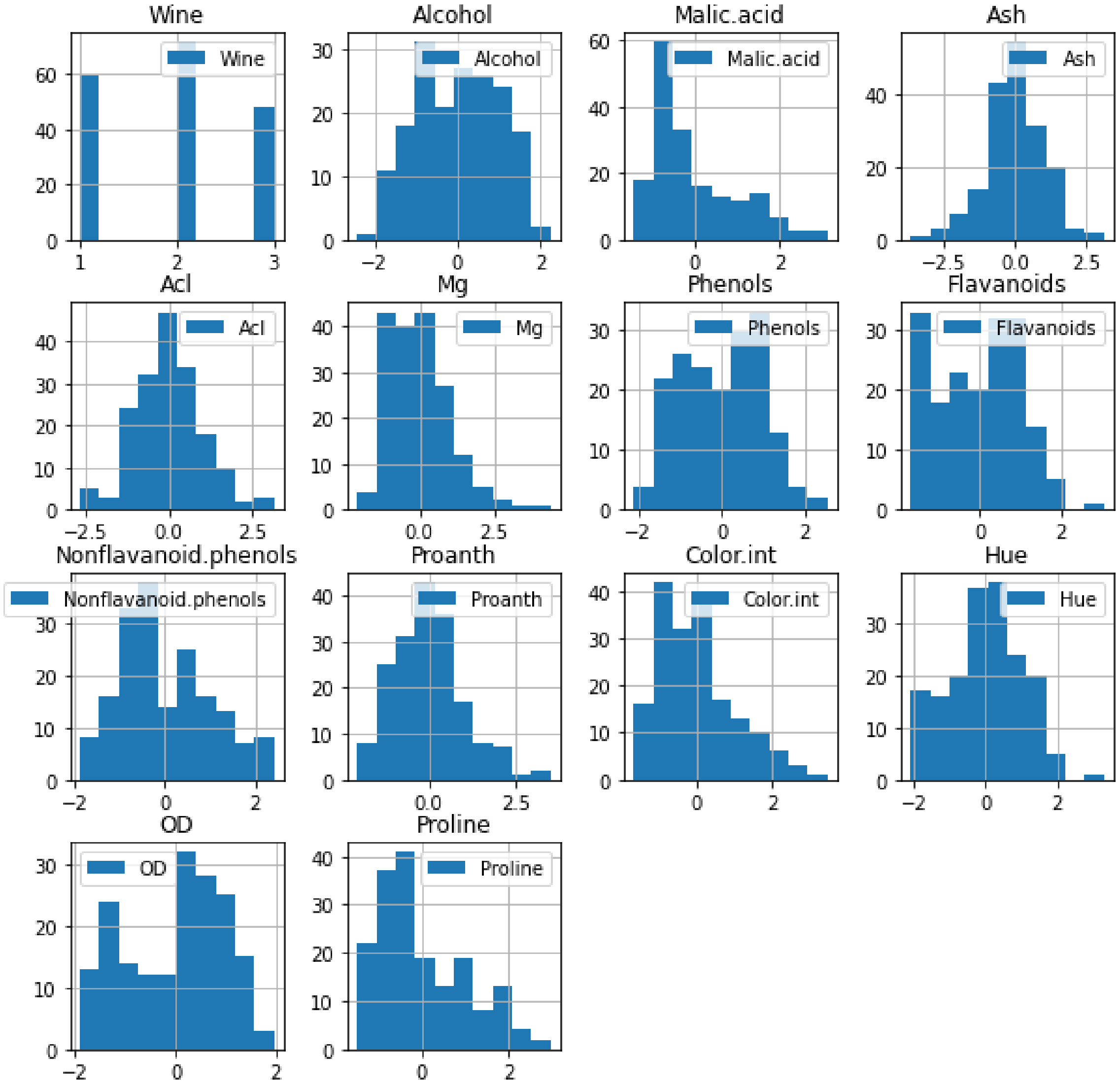
In [62]:

```
wine['Alcohol'].hist(figsize=(10,10))
wine.hist(figsize=(10,10),legend=True)
```

Out[62]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fdd51a70350>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd51198a50>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd51d46d10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd52a509d0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fdd5465ed10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd5180fe10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd51d23590>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd51925090>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fdd51925350>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd51fed650>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd5458b510>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd510f6290>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fdd52c06ed0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd51853550>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd546241d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fdd52e6f550
>]],
      dtype=object)
```




Wine
Wine
0
20
40
60
1
2
3
Alcohol
Alcohol
0
10
20
30
−2
0
2
Malic.acid
Malic.acid
0
20
40
60
0
2
Ash
Ash
0
20
40
−2.5
0.0
2.5
Acl
Acl
0
10
20
30
40
−2.5
0.0
2.5
Mg
Mg
0
10
20
30
40
0.0
2.5
Phenols
Phenols
0
10
20
30
−2
0
2
Flavanoids
Flavanoids
0
10
20
30
0
2
Nonflavanoid.phenols
Nonflavanoid.phenols
0
10
20
30
−2
0
2
Proanth
Proanth
0
10
20
30
40
0.0
2.5
Color.int
Color.int
0
10
20
30
40
0
2
Hue
Hue
0
10
20
30
−2
0
2
OD
OD
0
10
20
30
−2
0
2
Proline
Proline
0
10
20
30
40
0
2

In [63]:

```
M=[]
SD=[]
for i in wine:
  if i != 'Wine':
    M.append(round(wine[i].mean()))
    SD.append(round(wine[i].std()))
Scaled=pd.DataFrame(data=(M,SD),columns=wine.columns[1:],index=('Mean','Standard Deviation'))
Scaled.transpose()
```

Out[63]:

	Mean	Standard Deviation
Alcohol	0	1
Malic.acid	0	1
Ash	0	1
Acl	0	1
Mg	0	1
Phenols	0	1
Flavanoids	0	1
Nonflavanoid.phenols	0	1
Proanth	0	1
Color.int	0	1
Hue	0	1
OD	0	1
Proline	0	1

Iris dataset

In [64]:

```
pd.set_option("display.max_rows", 10, "display.max_columns", None)
iris = pd.read_csv('https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/
0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv')
iris
```

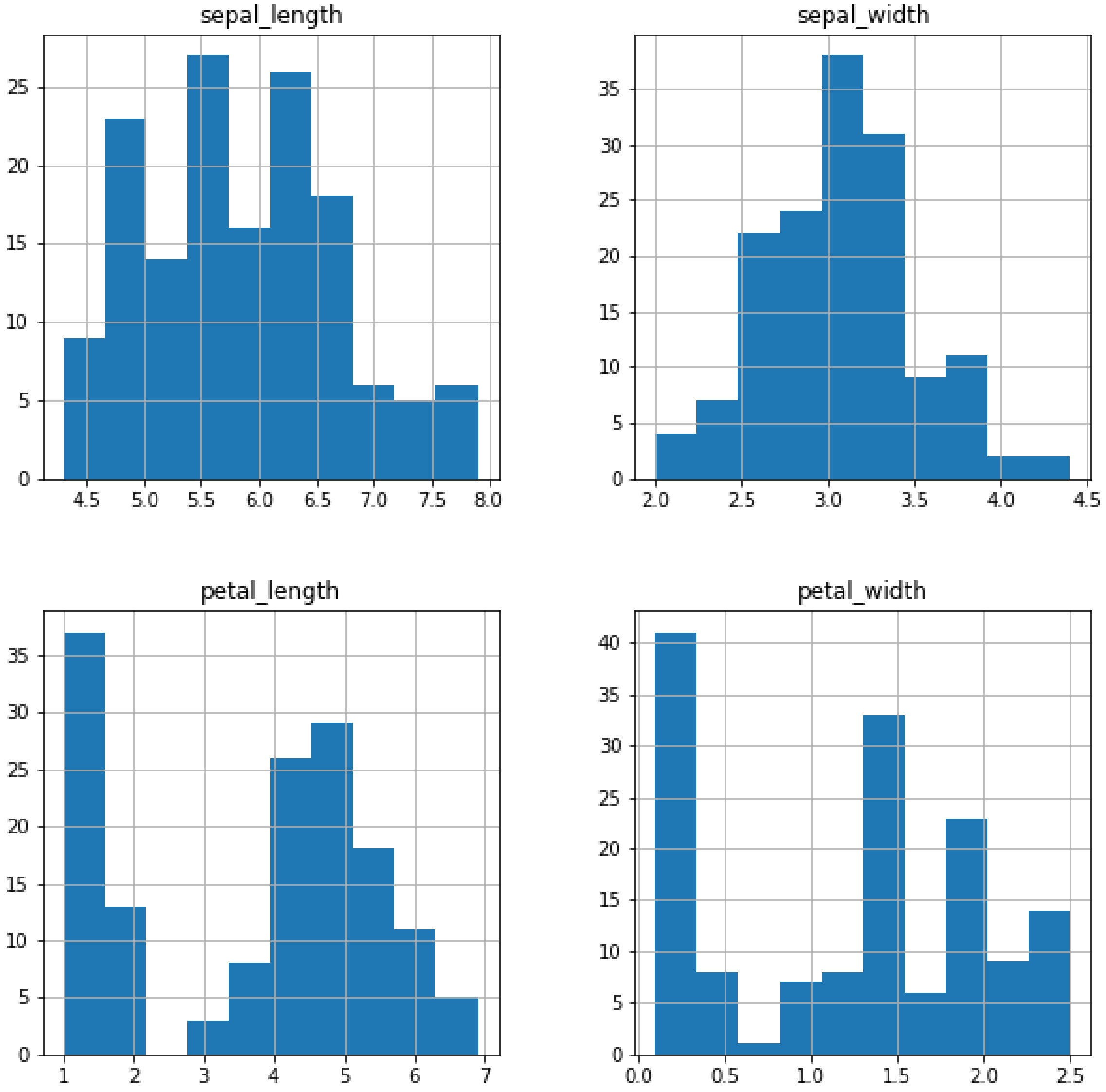
Out[64]:

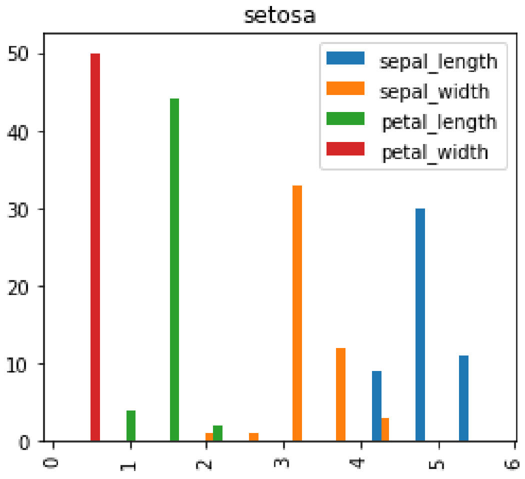
	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

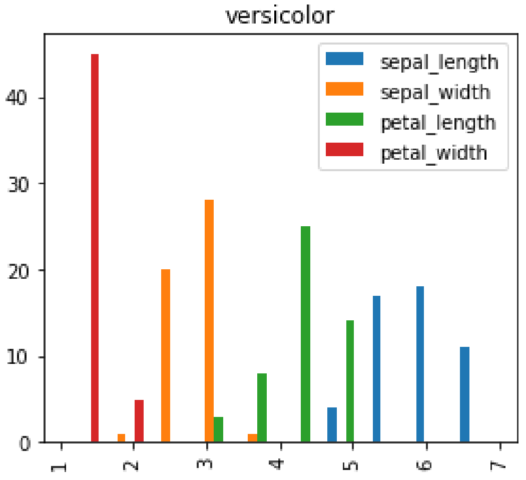
150 rows × 5 columns

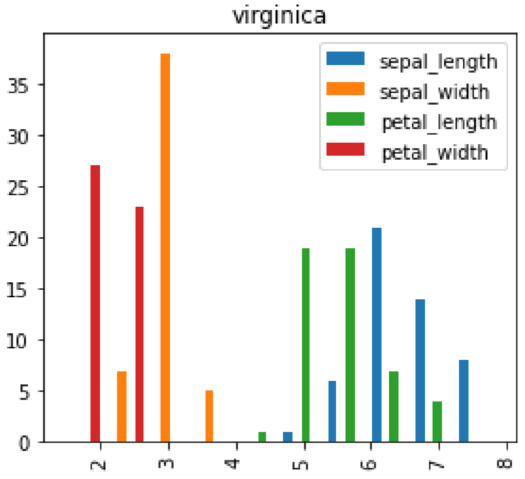
In [65]:

```
iris.hist(figsize=(10,10))
iris[:].hist(by=iris['species'],figsize=(10,10),legend=True)
plt.show()
```


sepal_length
0
5
10
15
20
25
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
sepal_width
0
5
10
15
20
25
30
35
2.0
2.5
3.0
3.5
4.0
4.5
petal_length
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7
petal_width
0
5
10
15
20
25
30
35
40
0.0
0.5
1.0
1.5
2.0
2.5


setosa
sepal_length
sepal_width
petal_length
petal_width
0
10
20
30
40
50
0
1
2
3
4
5
6


versicolor
sepal_length
sepal_width
petal_length
petal_width
0
10
20
30
40
1
2
3
4
5
6
7


virginica
sepal_length
sepal_width
petal_length
petal_width
0
5
10
15
20
25
30
35
2
3
4
5
6
7
8

In [66]:

```
iris.groupby('species').describe().transpose()
```

Out[66]:

	species	setosa	versicolor	virginica
sepal_length	**count**	50.00000	50.000000	50.00000
	mean	5.00600	5.936000	6.58800
	std	0.35249	0.516171	0.63588
	min	4.30000	4.900000	4.90000
	25%	4.80000	5.600000	6.22500
...	**...**	...	...	...
petal_width	**min**	0.10000	1.000000	1.40000
	25%	0.20000	1.200000	1.80000
	50%	0.20000	1.300000	2.00000
	75%	0.30000	1.500000	2.30000
	max	0.60000	1.800000	2.50000

32 rows × 3 columns

In [67]:

```
l = []
for i in iris.columns:
  if i != 'species' and iris[i].mean() != 0 and iris[i].std() != 1 :
    l.append(True)
  else:
    l.append(False)

scd = iris[iris.columns[l]]
```

In [68]:

```
scaled = scale(scd)
iris[iris.columns[l]] = pd.DataFrame(scaled,columns=iris.columns[l])
pd.set_option("display.max_rows", 10, "display.max_columns", None)
iris
```

Out[68]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	-0.900681	1.032057	-1.341272	-1.312977	setosa
1	-1.143017	-0.124958	-1.341272	-1.312977	setosa
2	-1.385353	0.337848	-1.398138	-1.312977	setosa
3	-1.506521	0.106445	-1.284407	-1.312977	setosa
4	-1.021849	1.263460	-1.341272	-1.312977	setosa
...	...	...	...	...	...
145	1.038005	-0.124958	0.819624	1.447956	virginica
146	0.553333	-1.281972	0.705893	0.922064	virginica
147	0.795669	-0.124958	0.819624	1.053537	virginica
148	0.432165	0.800654	0.933356	1.447956	virginica
149	0.068662	-0.124958	0.762759	0.790591	virginica

150 rows × 5 columns

Standardising Dataset

In [69]:

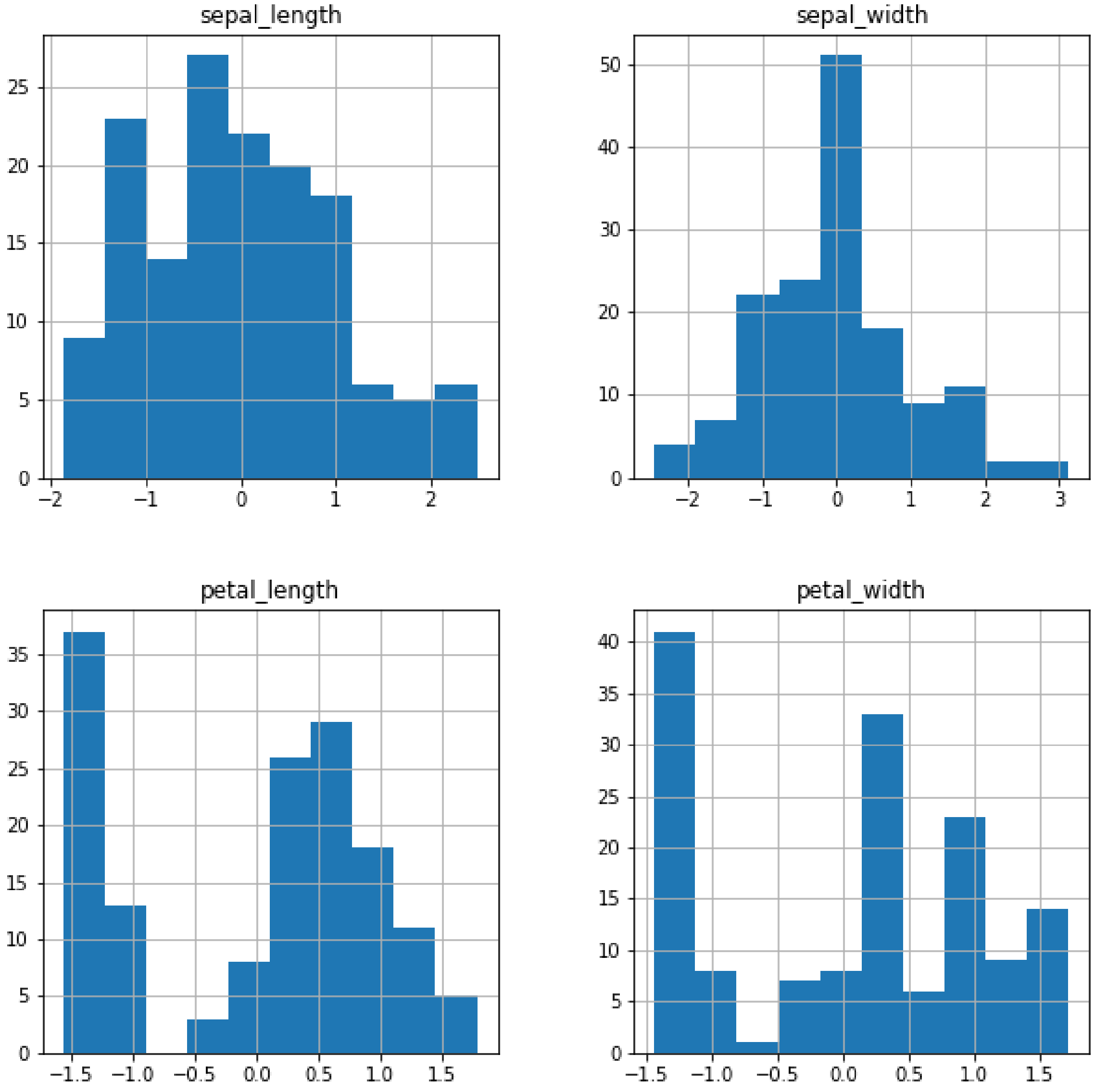
```
M=[]
SD=[]
for i in iris:
  if i != 'species':
    M.append(round(iris[i].mean()))
    SD.append(round(iris[i].std()))
Scaled=pd.DataFrame(data=(M,SD),columns=iris.columns[:-1],index=('Mean','Standard Deviation'))
Scaled.transpose()
```

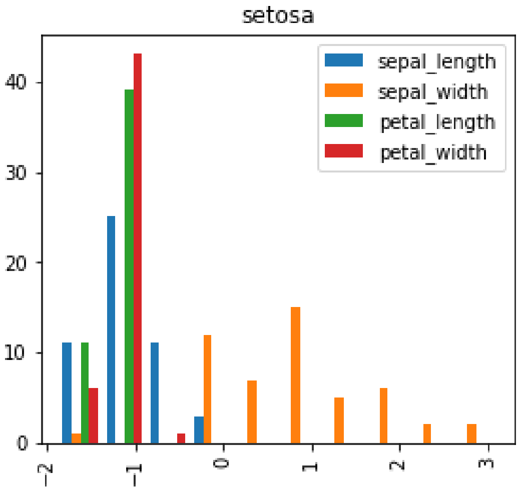
Out[69]:

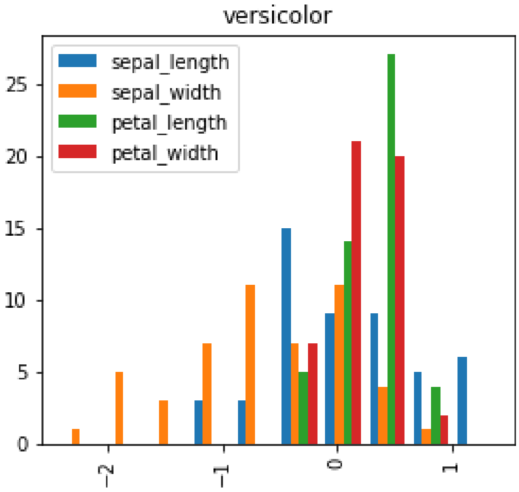
	Mean	Standard Deviation
sepal_length	0	1
sepal_width	0	1
petal_length	0	1
petal_width	0	1

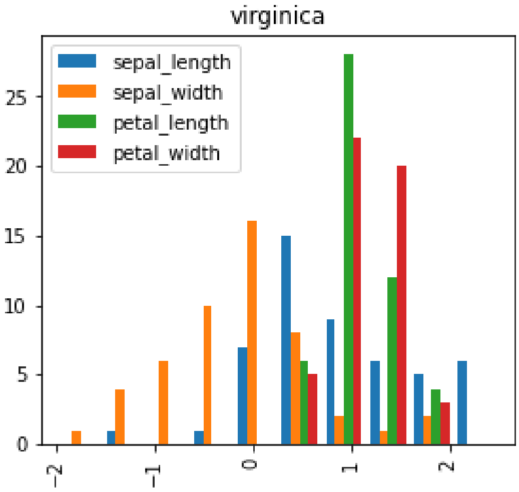
In [70]:

```
ax1=iris[:].hist(figsize=(10,10))
ax2=iris[:].hist(by=iris['species'],figsize=(10,10),legend=True)
```


sepal_length
0
5
10
15
20
25
−2
−1
0
1
2
sepal_width
0
10
20
30
40
50
−2
−1
0
1
2
3
petal_length
0
5
10
15
20
25
30
35
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
petal_width
0
5
10
15
20
25
30
35
40
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


setosa
sepal_length
sepal_width
petal_length
petal_width
40
30
20
10
0
−2
−1
0
1
2
3


versicolor
sepal_length
sepal_width
petal_length
petal_width
25
20
15
10
5
0
−2
−1
0
1


virginica
sepal_length
sepal_width
petal_length
petal_width
25
20
15
10
5
0
−2
−1
0
1
2

In [76]:

```
!cp  '/content/drive/MyDrive/Colab Notebooks/Practical 3.ipynb' ./
!jupyter nbconvert --to html "Practical 3"
```

```
[NbConvertApp] Converting notebook Practical 3.ipynb to html
[NbConvertApp] Writing 561465 bytes to Practical 3.html
```