In [1]:

```python
import sklearn
import numpy as np
import pandas as pd
import seaborn as sns
```

In [2]:

```python
iris_df = pd.read_csv('iris.data', header=None, names=['sepal length', 'sepal width', 'petal length', 'petal width', 'class'])
glass_df = pd.read_csv('glass.data', index_col=0, header=None, names=['Id', 'RI', 'Na', 'Mg', 'Al', 'Si', 'K', 'Ca', 'Ba', 'Fe', 'Type of glass'])
wine_df = pd.read_csv('wine.data', header=None, names=['class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline'])
```

Iris

In [3]:

```python
print(iris_df)
```

```
     sepal length  sepal width  petal length  petal width           class
0             5.1          3.5           1.4          0.2     Iris-setosa
1             4.9          3.0           1.4          0.2     Iris-setosa
2             4.7          3.2           1.3          0.2     Iris-setosa
3             4.6          3.1           1.5          0.2     Iris-setosa
4             5.0          3.6           1.4          0.2     Iris-setosa
..            ...          ...           ...          ...             ...
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica

[150 rows x 5 columns]
```

3 klasy oznaczające gatunki irysów, po 50 instancji każdej

pierwsze 2 kolumny `sepal length` **i** `sepal width` **oznaczają długość i szerokość działki kielicha w cm**

`petal length` **i** `petal width` **oznaczają długość i szerokość płatków w cm**

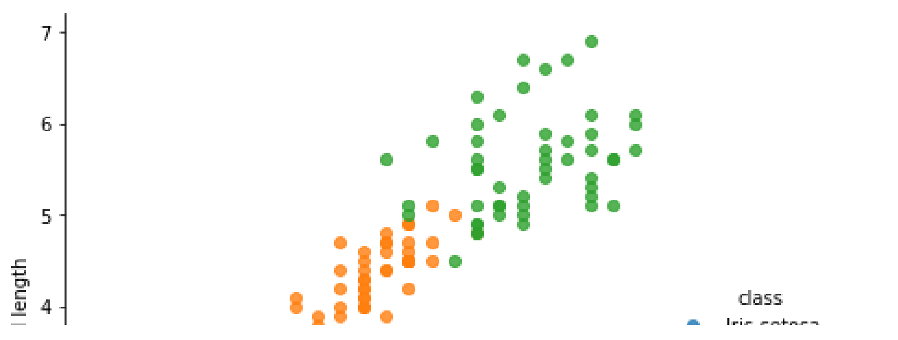
Iris-versicolor i Iris-virginica są do siebie mniej więcej podobne patrząc na dowolne atrybuty, podaczas gdy Iris-setosa wyróżnia się kiedy patrzymy na dowolny wymiar płatka lub oba wymiary działki kielicha

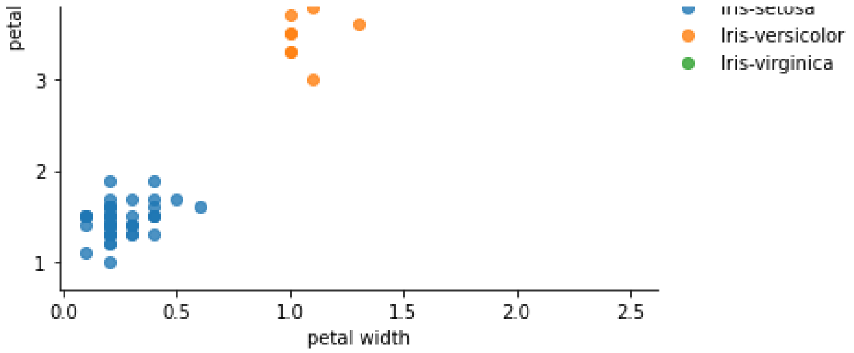
In [4]:

```python
sns.lmplot(x='petal width', y='petal length', hue='class', data=iris_df, fit_reg=False)
```

Out[4]:

```
<seaborn.axisgrid.FacetGrid at 0x14981d82580>
```



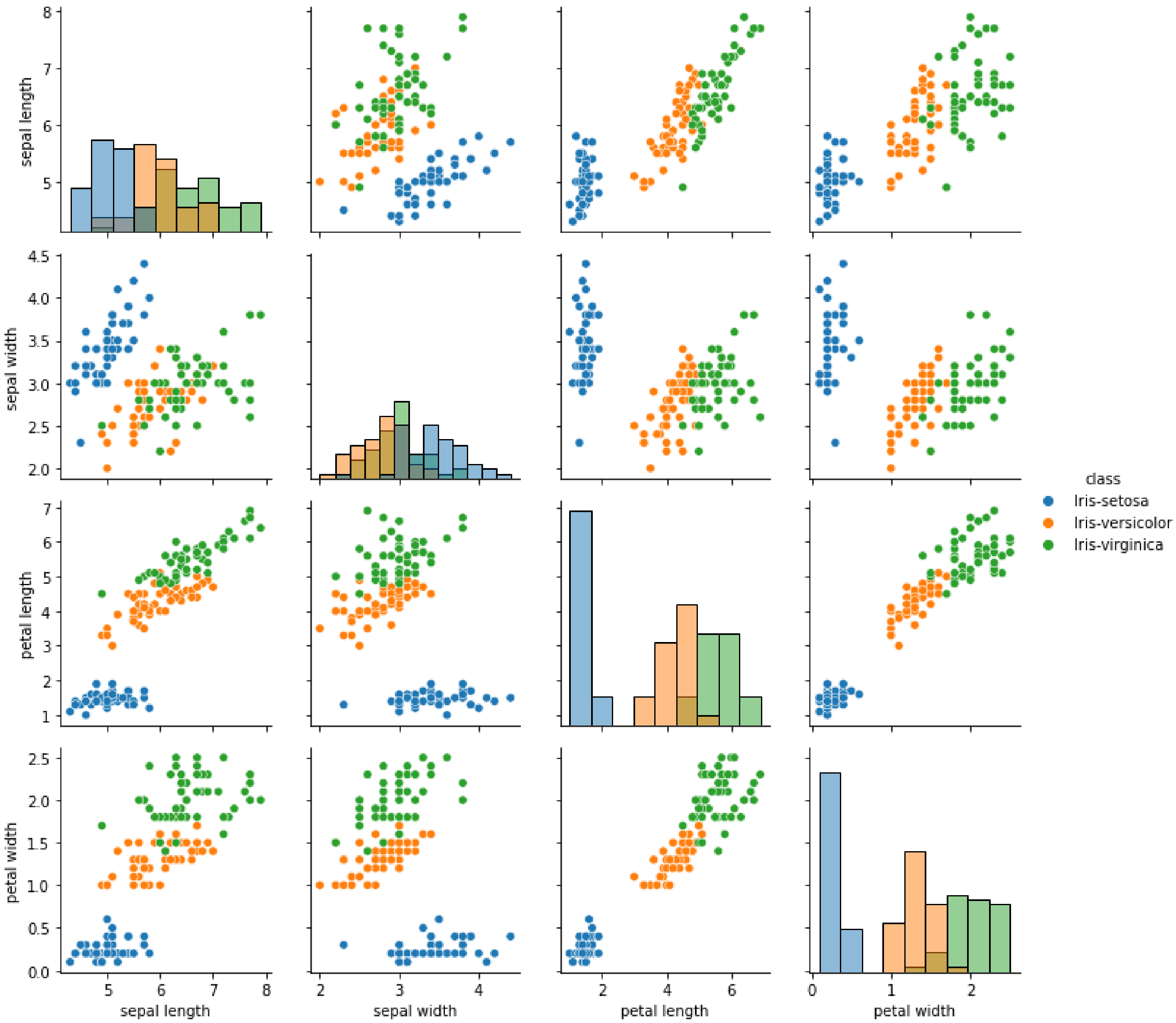


In [5]:

```
iris_grid = sns.PairGrid(iris_df, hue='class')
iris_grid.map_diag(sns.histplot)
iris_grid.map_offdiag(sns.scatterplot)
iris_grid.add_legend()
```

Out[5]:

```
<seaborn.axisgrid.PairGrid at 0x14984ef8670>
```



Glass

In [6]:

```
print(glass_df)
```

```
          RI     Na    Mg    Al     Si     K    Ca    Ba   Fe  Type of glass
Id
1    1.52101  13.64  4.49  1.10  71.78  0.06  8.75  0.00  0.0              1
2    1.51761  13.89  3.60  1.36  72.73  0.48  7.83  0.00  0.0              1
3    1.51618  13.53  3.55  1.54  72.99  0.39  7.78  0.00  0.0              1
4    1.51766  13.21  3.69  1.29  72.61  0.57  8.22  0.00  0.0              1
5    1.51742  13.27  3.62  1.24  73.08  0.55  8.07  0.00  0.0              1
..       ...    ...   ...   ...    ...   ...   ...   ...  ...            ...
210  1.51623  14.14  0.00  2.88  72.61  0.08  9.18  1.06  0.0              7
211  1.51685  14.92  0.00  1.99  73.06  0.00  8.40  1.59  0.0              7
212  1.52065  14.36  0.00  2.02  73.42  0.00  8.44  1.64  0.0              7
213  1.51651  14.38  0.00  1.94  73.61  0.00  8.48  1.57  0.0              7
214  1.51711  14.23  0.00  2.08  73.36  0.00  8.62  1.67  0.0              7

[214 rows x 10 columns]
```

kolumna `Id` zawiera indeksy próbek szkła

kolumna `RI` zawiera współczynniki załamania szkła

ostatnia kolumna `Type of glass` zawiera indeksy 1-7 różnych typów szkła

liczba instacji każdej klasy:

1. 70
2. 76
3. 17
4. 0
5. 13
6. 9
7. 29

pozostałe kolumny zawierają procentową zawartość odpowiadających pierwiastków

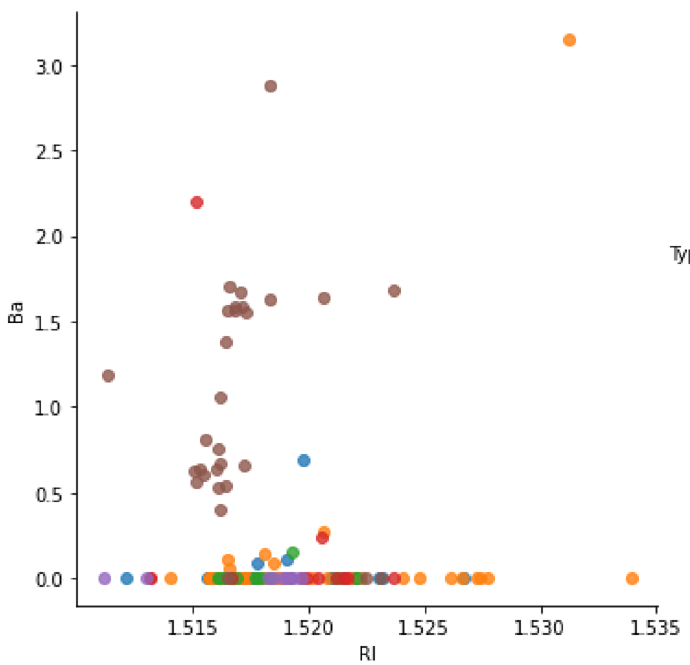
patrząc na wykresy jedynym jakimkolwiek wyróżniającym się typem szkła jest szkło reflektorów (7), głównie z powodu wyższej zawartości baru

In [7]:

```
sns.lmplot(x='RI', y='Ba', hue='Type of glass', data=glass_df, fit_reg=False)
```

Out[7]:

```
<seaborn.axisgrid.FacetGrid at 0x14985187070>
```

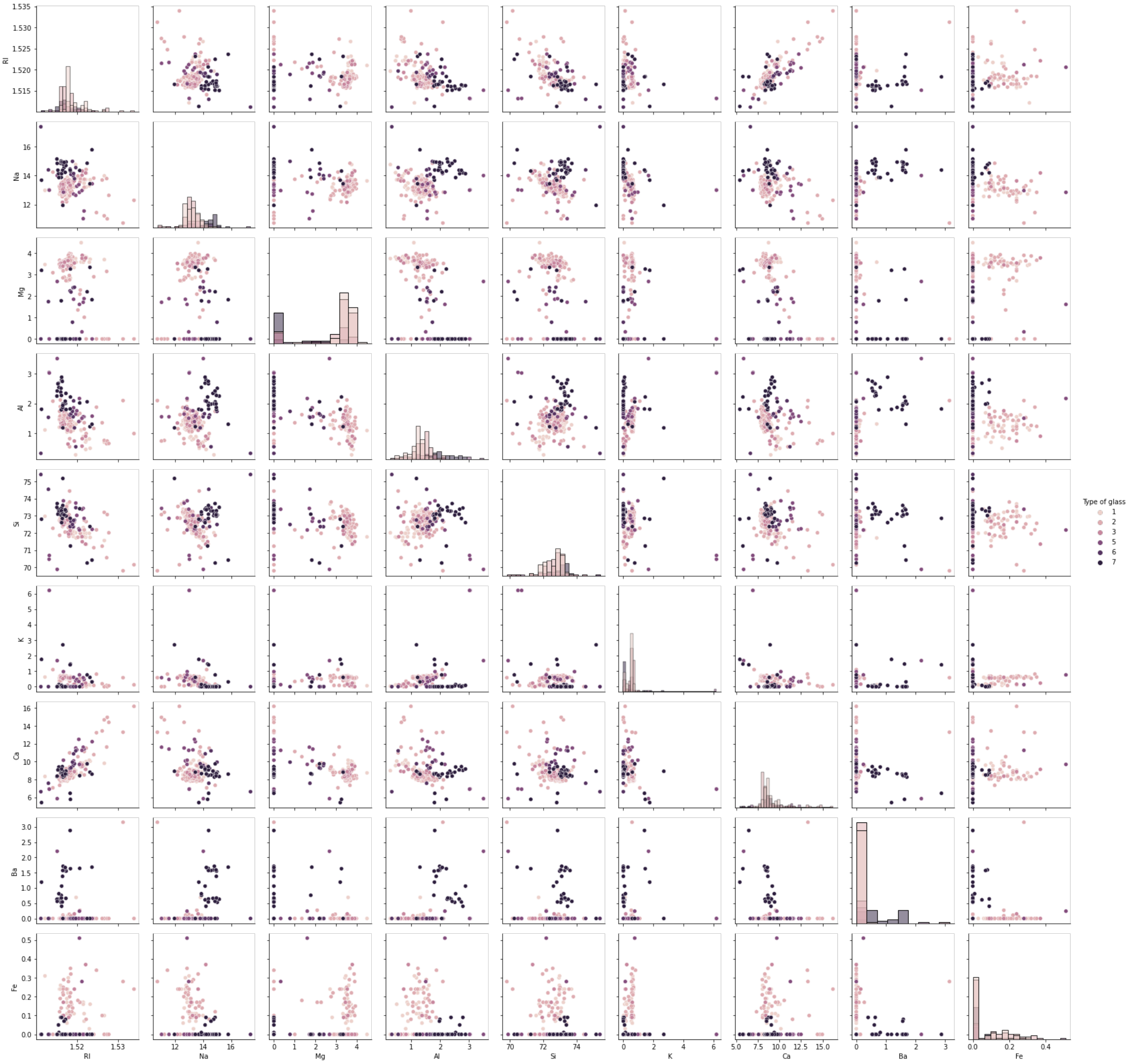


In [8]:

```python
glass_grid = sns.PairGrid(glass_df, hue='Type of glass')
glass_grid.map_diag(sns.histplot)
glass_grid.map_offdiag(sns.scatterplot)
glass_grid.add_legend()
```

Out[8]:

```
<seaborn.axisgrid.PairGrid at 0x149858df7f0>
```



Wine

In [9]:

```python
print(wine_df)
```

```
     class  Alcohol  Malic acid   Ash  Alcalinity of ash  Magnesium  \
0        1    14.23        1.71  2.43               15.6        127
1        1    13.20        1.78  2.14               11.2        100
2        1    13.16        2.36  2.67               18.6        101
3        1    14.37        1.95  2.50               16.8        113
4        1    13.24        2.59  2.87               21.0        118
..     ...      ...         ...   ...                ...        ...
173      3    13.71        5.65  2.45               20.5         95
174      3    13.40        3.91  2.48               23.0        102
175      3    13.27        4.28  2.26               20.0        120
```

```
176       3    13.17         2.59  2.37                  20.0        120
177       3    14.13         4.10  2.74                  24.5         96

     Total phenols  Flavanoids  Nonflavanoid phenols  Proanthocyanins  \
0             2.80        3.06                  0.28             2.29
1             2.65        2.76                  0.26             1.28
2             2.80        3.24                  0.30             2.81
3             3.85        3.49                  0.24             2.18
4             2.80        2.69                  0.39             1.82
..             ...         ...                   ...              ...
173           1.68        0.61                  0.52             1.06
174           1.80        0.75                  0.43             1.41
175           1.59        0.69                  0.43             1.35
176           1.65        0.68                  0.53             1.46
177           2.05        0.76                  0.56             1.35

     Color intensity   Hue  OD280/OD315 of diluted wines  Proline
0               5.64  1.04                          3.92     1065
1               4.38  1.05                          3.40     1050
2               5.68  1.03                          3.17     1185
3               7.80  0.86                          3.45     1480
4               4.32  1.04                          2.93      735
..               ...   ...                           ...      ...
173             7.70  0.64                          1.74      740
174             7.30  0.70                          1.56      750
175            10.20  0.59                          1.56      835
176             9.30  0.60                          1.62      840
177             9.20  0.61                          1.60      560

[178 rows x 14 columns]
```

kolumna `class` zawiera indeksy 1-3 oznaczające różne odmiany wina z liczbami instancji

1. 59
2. 71
3. 48

pozostałe kolumny zawierają informacje o właściwościach chemicznych wina

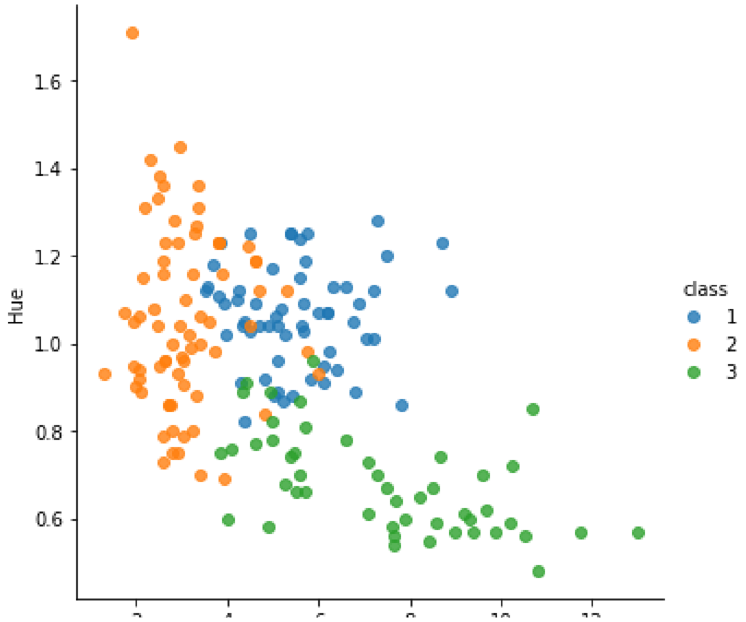
wszystkie 3 klasy są częściowo rozróżnialne patrząc na różne atrybuty, np. odmiana 1. ma większą zawartość proliny, odmiana 2. ma mało intensywny kolor i próbki odmiany 3. mają podobny do siebie odcień

In [10]:

```
sns.lmplot(x='Color intensity', y='Hue', hue='class', data=wine_df, fit_reg=False)
```

Out[10]:

```
<seaborn.axisgrid.FacetGrid at 0x14985b03190>
```

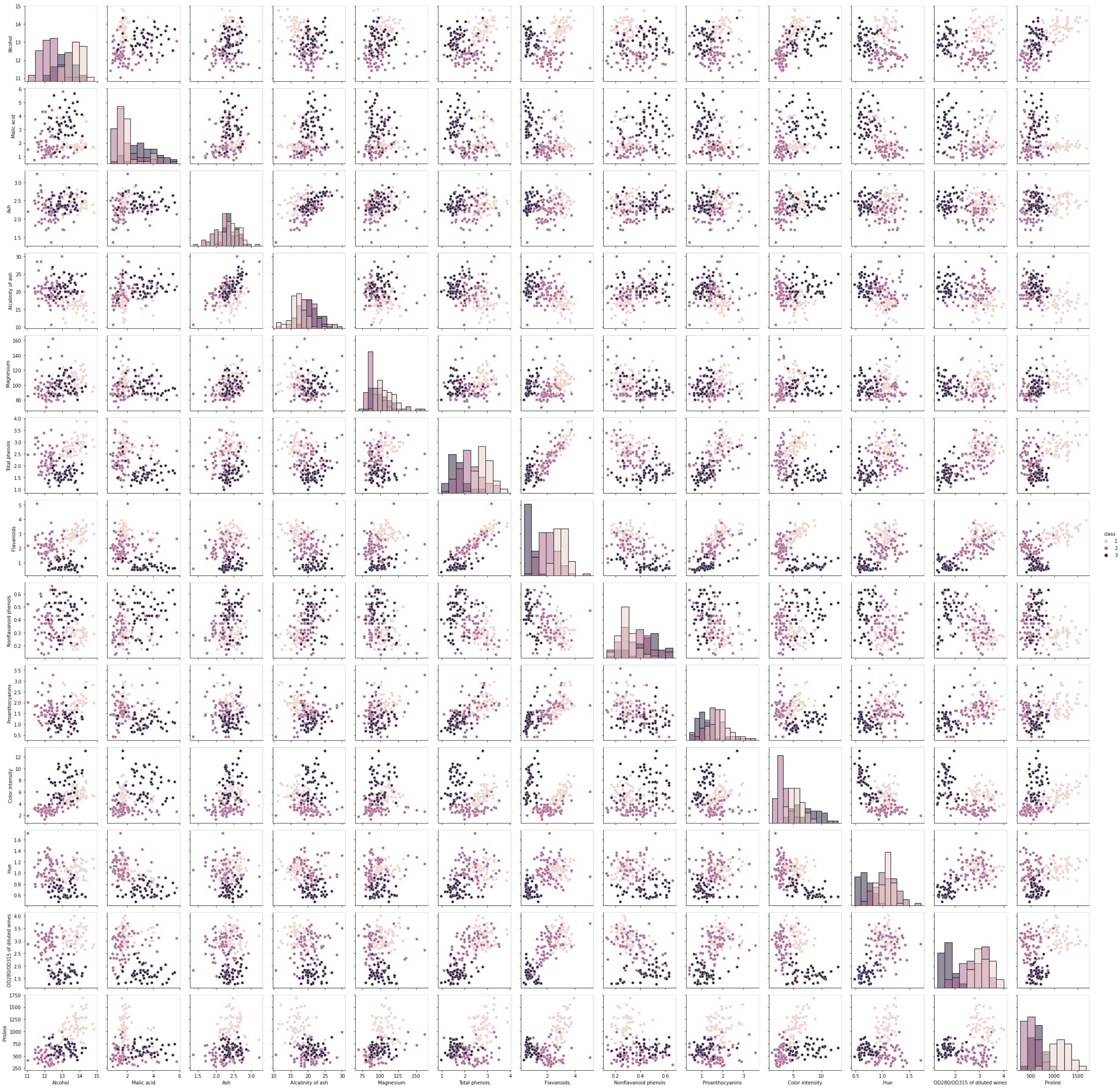


In [11]:

```
wine_grid = sns.PairGrid(wine_df, hue='class')
wine_grid.map_diag(sns.histplot)
wine_grid.map_offdiag(sns.scatterplot)
wine_grid.add_legend()
```

Out[11]:

<seaborn.axisgrid.PairGrid at 0x14989ea5880>



PCA

In [12]:

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

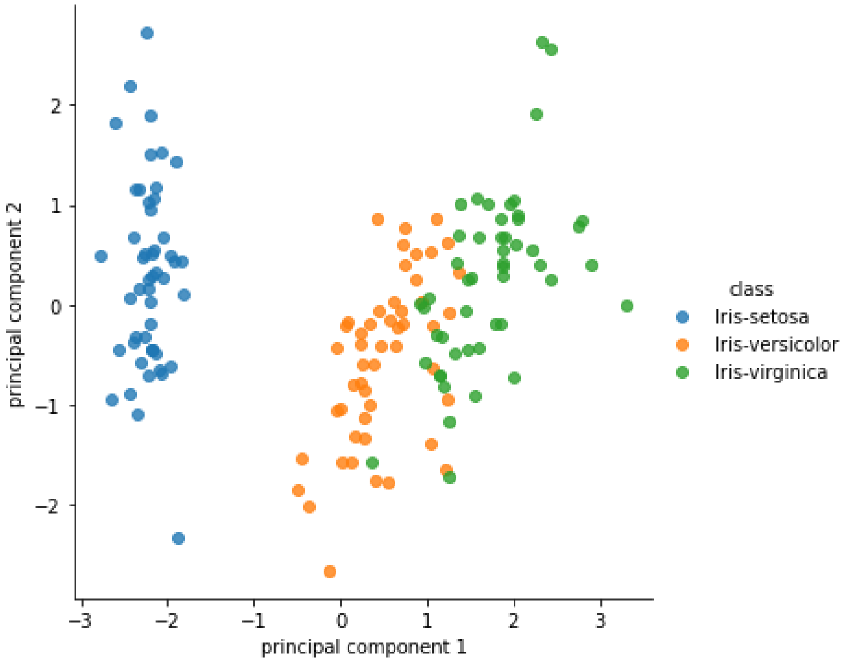
Iris

In [13]:

```
features = ['sepal length', 'sepal width', 'petal length', 'petal width']
x = iris_df.loc[:, features].values
y = iris_df.loc[:,['class']].values
x = StandardScaler().fit_transform(x)
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents, columns = ['principal component 1
', 'principal component 2'])
finalDf = pd.concat([principalDf, iris_df[['class']]], axis = 1)
sns.lmplot(x='principal component 1', y='principal component 2', hue='class', data=finalD
f, fit_reg=False)
```

Out[13]:

```
<seaborn.axisgrid.FacetGrid at 0x14992fa1190>
```



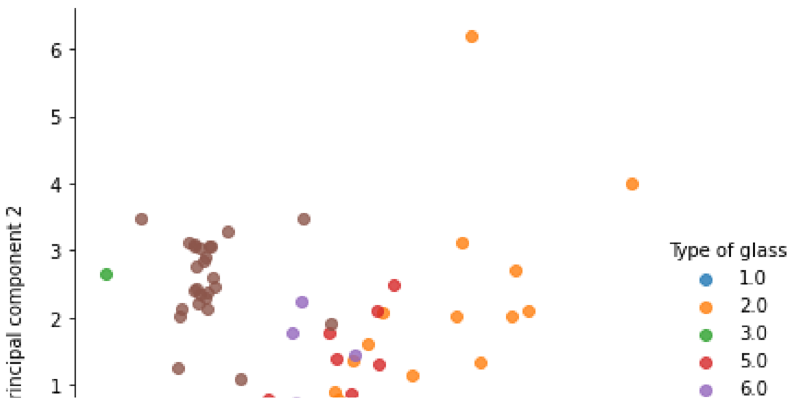
Glass

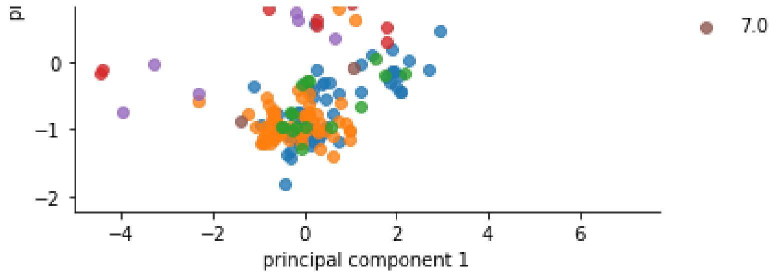
In [14]:

```
features = ['RI', 'Na', 'Mg', 'Al', 'Si', 'K', 'Ca', 'Ba', 'Fe']
x = glass_df.loc[:, features].values
y = glass_df.loc[:,['Type of glass']].values
x = StandardScaler().fit_transform(x)
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents, columns = ['principal component 1
', 'principal component 2'])
finalDf = pd.concat([principalDf, glass_df[['Type of glass']]], axis = 1)
sns.lmplot(x='principal component 1', y='principal component 2', hue='Type of glass', dat
a=finalDf, fit_reg=False)
```

Out[14]:

```
<seaborn.axisgrid.FacetGrid at 0x14992fa12e0>
```





Wine

In [15]:

```
features = ['Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline']
x = wine_df.loc[:, features].values
y = wine_df.loc[:,['class']].values
x = StandardScaler().fit_transform(x)
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents, columns = ['principal component 1', 'principal component 2'])
finalDf = pd.concat([principalDf, wine_df[['class']]], axis = 1)
sns.lmplot(x='principal component 1', y='principal component 2', hue='class', data=finalDf, fit_reg=False)
```

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x1499301e310>
```