# EX_01

January 19, 2023

[1]:
```
# preamble imports
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import statistics
```

# 1 Assignment 1 - Descriptive Statistics

### Patrick Pfenning

### 01/11/23

## 1.1 Basic Python

[2]:
```
"""Using the statistics module"""
# Set data
data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]
# Find mean
statistics.mean(data)
```

[2]:
```
3
```

[3]:
```
# Find median
statistics.median(data)
```

[3]:
```
3.0
```

[4]:
```
# Find mode
statistics.mode(data)
```

[4]:
```
4
```

[5]:
```
# Summarize data
from dataclasses import dataclass, field

"""Python 3.10 has no describe function, so I made my own class to do the same.
 ↪"""
```

```python

@dataclass
class DescribeResults:
    """Class for describing data"""
    data: list = field(repr=False, default_factory=list)
    # initialize calculations
    mean: float = field(init=False)
    median: float = field(init=False)
    mode: float = field(init=False)
    variance: float = field(init=False)
    stdev: float = field(init=False)
    minmax: tuple = field(init=False)
    sum: float = field(init=False)

    def __post_init__(self):
        """Set Calculations"""
        self.mean = statistics.mean(self.data)
        self.median = statistics.median(self.data)
        self.mode = statistics.mode(self.data)
        self.variance = statistics.variance(self.data)
        self.stdev = statistics.stdev(self.data)
        self.minmax = (min(data), max(data))
        self.sum = sum(data)

DescribeResults(data)
```

[5]:
```
DescribeResults(mean=3, median=3.0, mode=4, variance=1.1111111111111112,
stdev=1.0540925533894598, minmax=(1, 4), sum=30)
```

[6]:
```python
"""Using the numpy module"""
# mean
np.mean(data)
```

[6]:
```
3.0
```

[7]:
```python
# median
np.median(data)
```

[7]:
```
3.0
```

[8]:
```python
# std. deviation
np.std(data)
```

[8]:
```
1.0
```

## 1.2 Data Preparation

Before telling our story, we must first clean our data.

[9]:

```
# get data
df = pd.read_csv('../data/wine.csv').set_index('unique_id', drop=True)
# transpose for better visibility
df.T.iloc[:, :5]
```

[9]:

```
unique_id                593      617      782      990     822
class                   1.00     1.00     1.00     1.00    1.00
alcohol_percentage     14.23    13.20    13.16    14.37   13.24
malic_acid              1.71     1.78     2.36     1.95    2.59
ash                     2.43     2.14     2.67     2.50    2.87
alcalinity             15.60    11.20    18.60    16.80   21.00
magnesium             127.00   100.00   101.00   113.00  118.00
phenols                 2.80     2.65     2.80     3.85    2.80
flavanoids              3.06     2.76     3.24     3.49    2.69
nonflavanoids           0.28     0.26     0.30     0.24    0.39
proanthocyanins         2.29     1.28     2.81     2.18    1.82
color                   5.64     4.38     5.68     7.80    4.32
hue                     1.04     1.05     1.03     0.86    1.04
price_usd            1065.00  1050.00  1185.00  1480.00  735.00
```

[10]:

```
# standardization (maps quantitative values to a bell curve with mean 0)
df_norm = df.iloc[:, 1:].copy()
df_norm = (df_norm-df_norm.mean())/df_norm.std()
df_norm['class'] = df['class']
df_norm = df_norm[df.columns]
df_norm.T.iloc[:, :5]
```

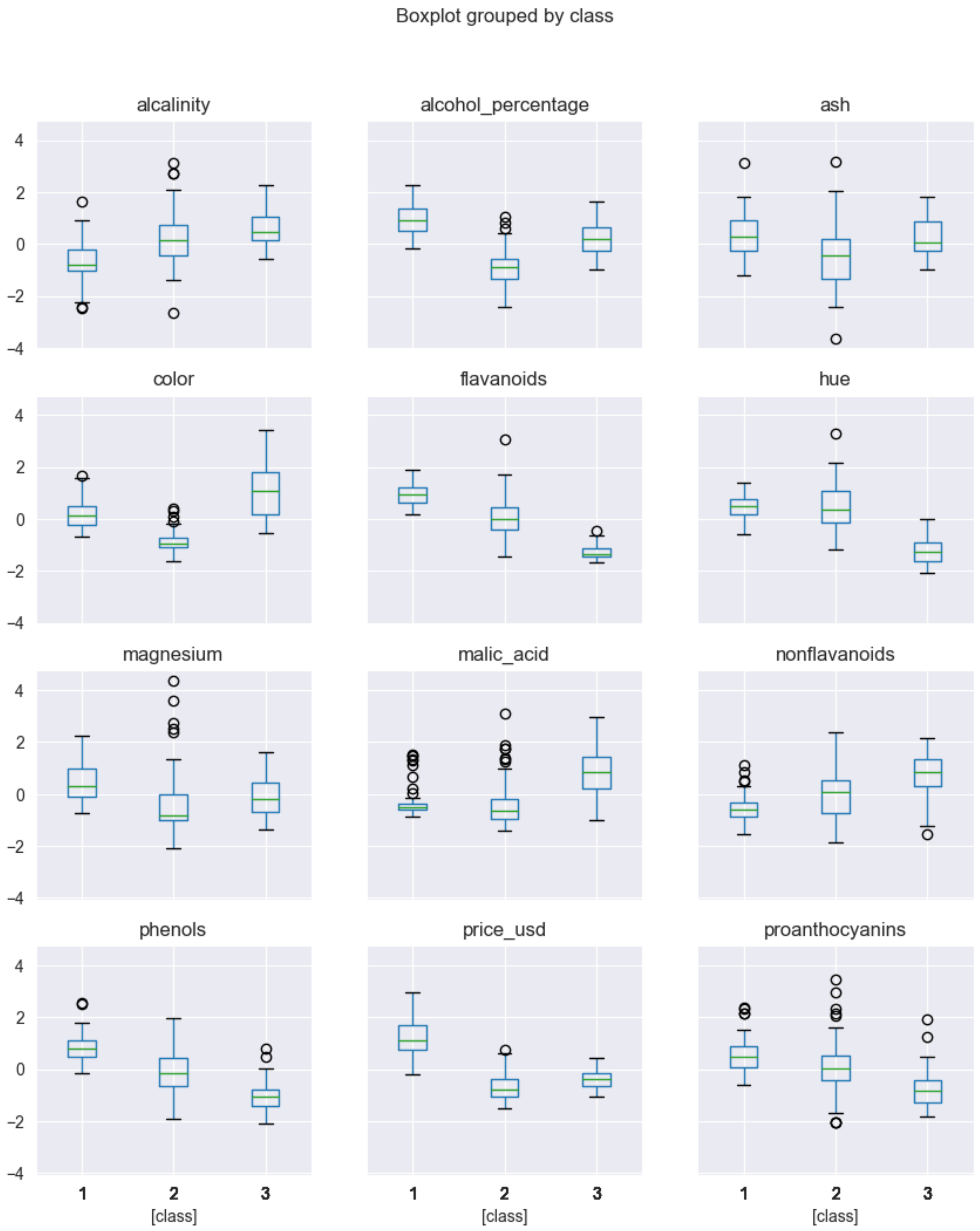
[10]:

```
unique_id                 593       617       782       990       822
class                1.000000  1.000000  1.000000  1.000000  1.000000
alcohol_percentage   1.514341  0.245597  0.196325  1.686791  0.294868
malic_acid          -0.560668 -0.498009  0.021172 -0.345835  0.227053
ash                  0.231400 -0.825667  1.106214  0.486554  1.835226
alcalinity          -1.166303 -2.483841 -0.267982 -0.806975  0.450674
magnesium            1.908522  0.018094  0.088110  0.928300  1.278379
phenols              0.806722  0.567048  0.806722  2.484437  0.806722
flavanoids           1.031908  0.731565  1.212114  1.462399  0.661485
nonflavanoids       -0.657708 -0.818411 -0.497005 -0.979113  0.226158
proanthocyanins      1.221438 -0.543189  2.129959  1.029251  0.400275
color                0.251009 -0.292496  0.268263  1.182732 -0.318377
hue                  0.361158  0.404908  0.317409 -0.426341  0.361158
price_usd            1.010159  0.962526  1.391224  2.328007 -0.037767
```

[18]:

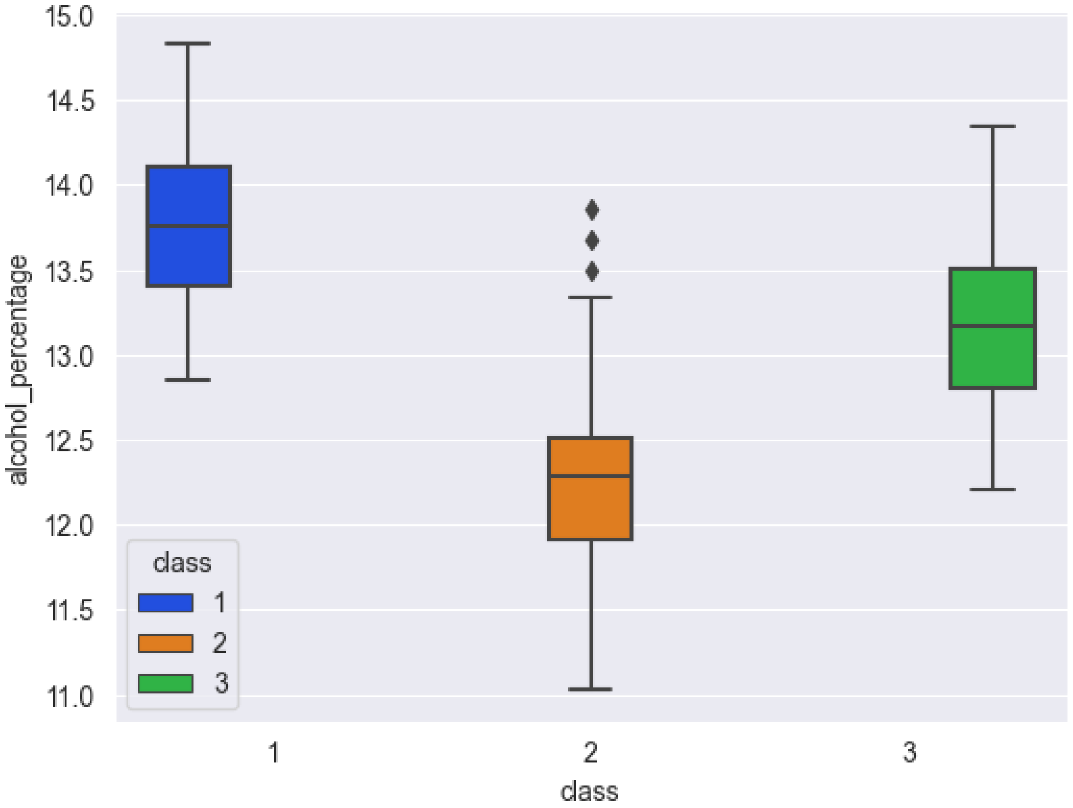
```
"""Boxplot of Standardized Data by Class"""
df_norm.boxplot(by='class', layout=(4, 3), figsize=(10, 12))
```

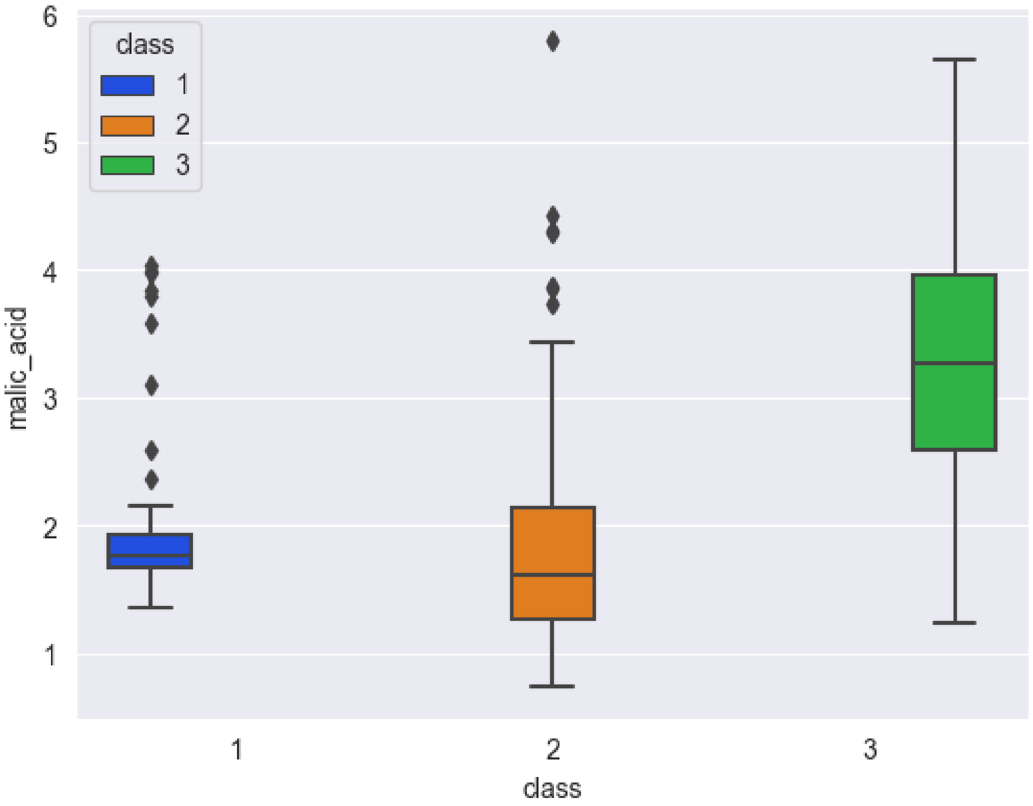
```
plt.show()
```

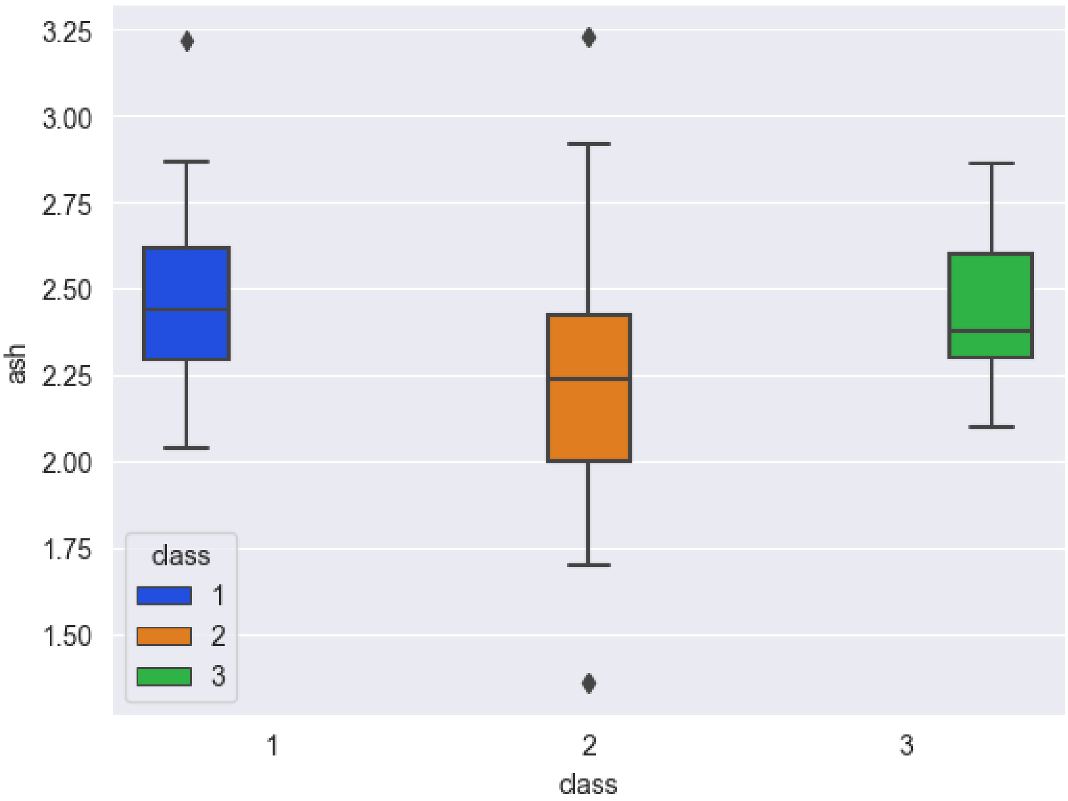


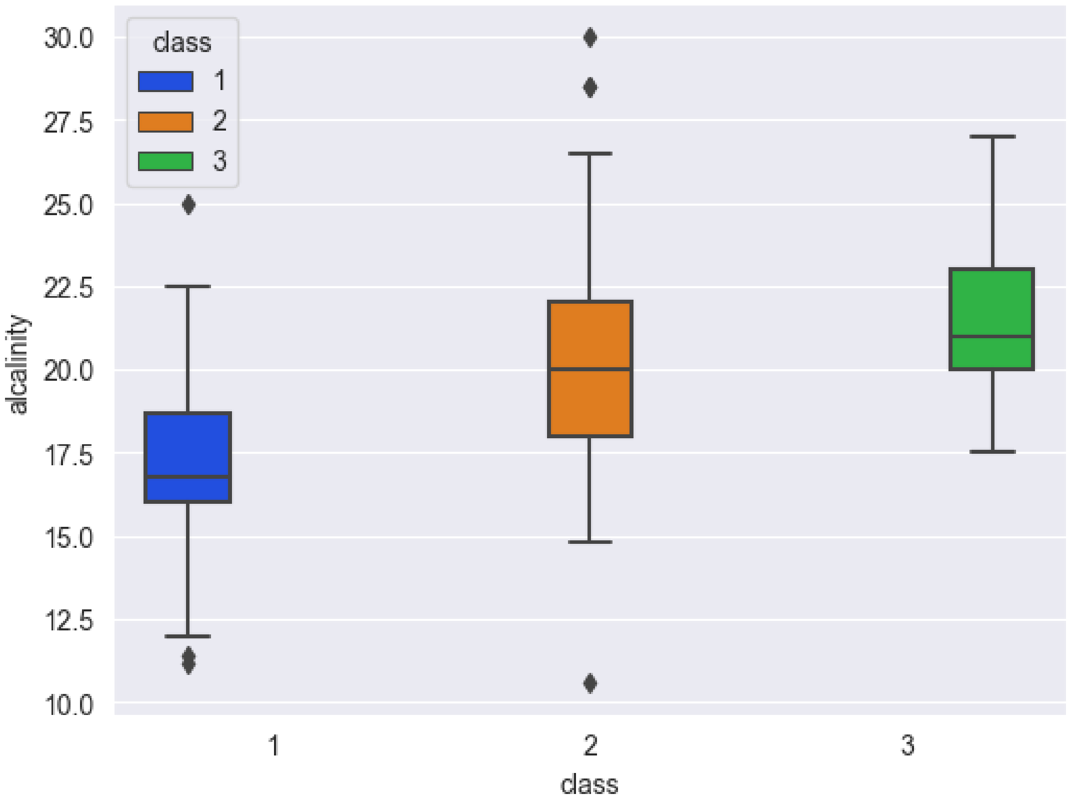
[21]:
```
"""Boxplot of Raw Data by Class"""
import seaborn as sns
```

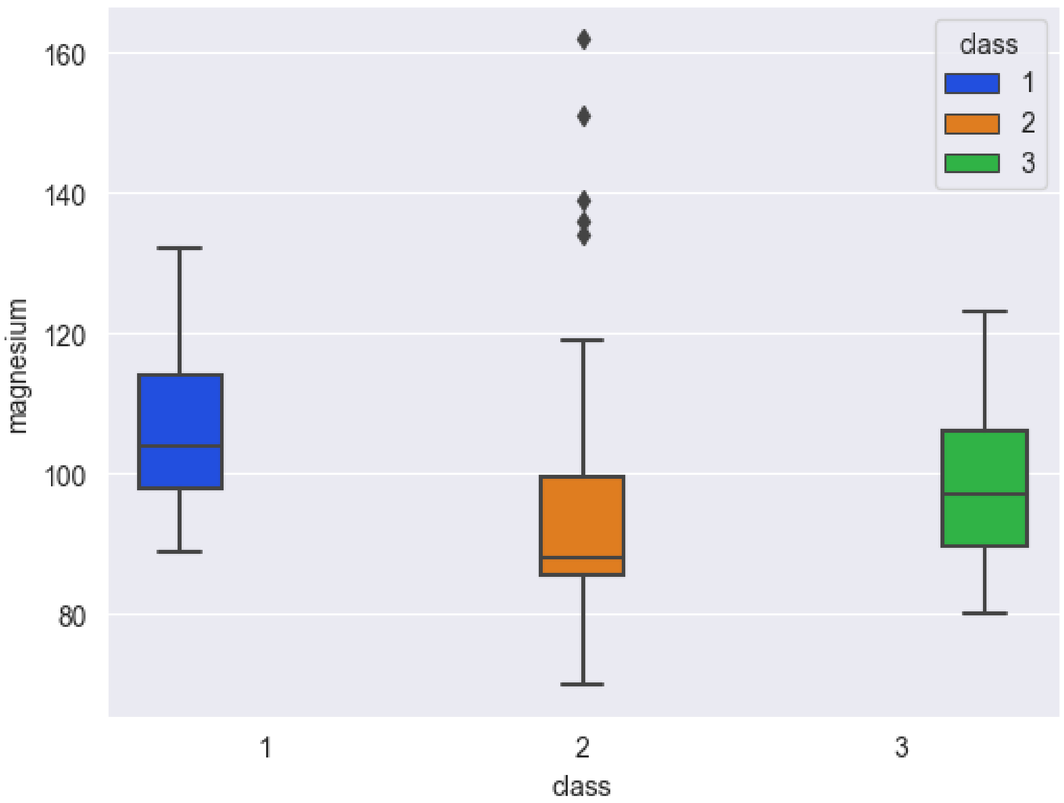
```
for col in df.columns[1:]:
    sns.boxplot(df, x='class', y=col, hue='class', palette='bright')
    plt.show()
```

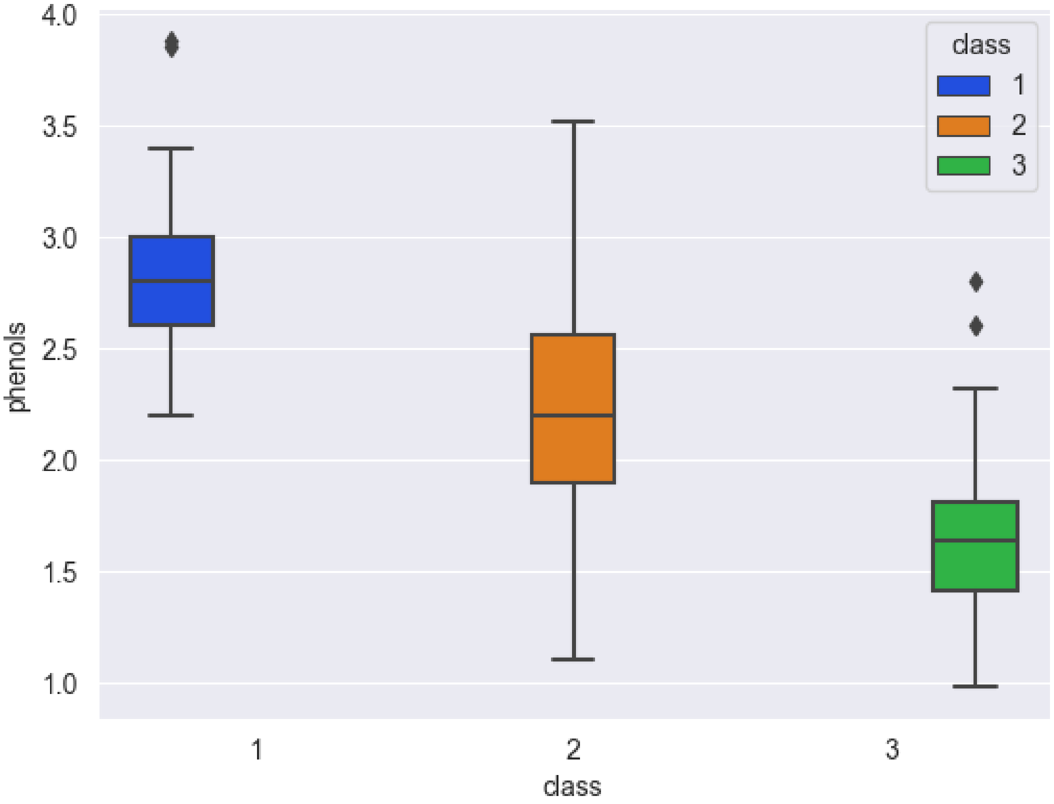


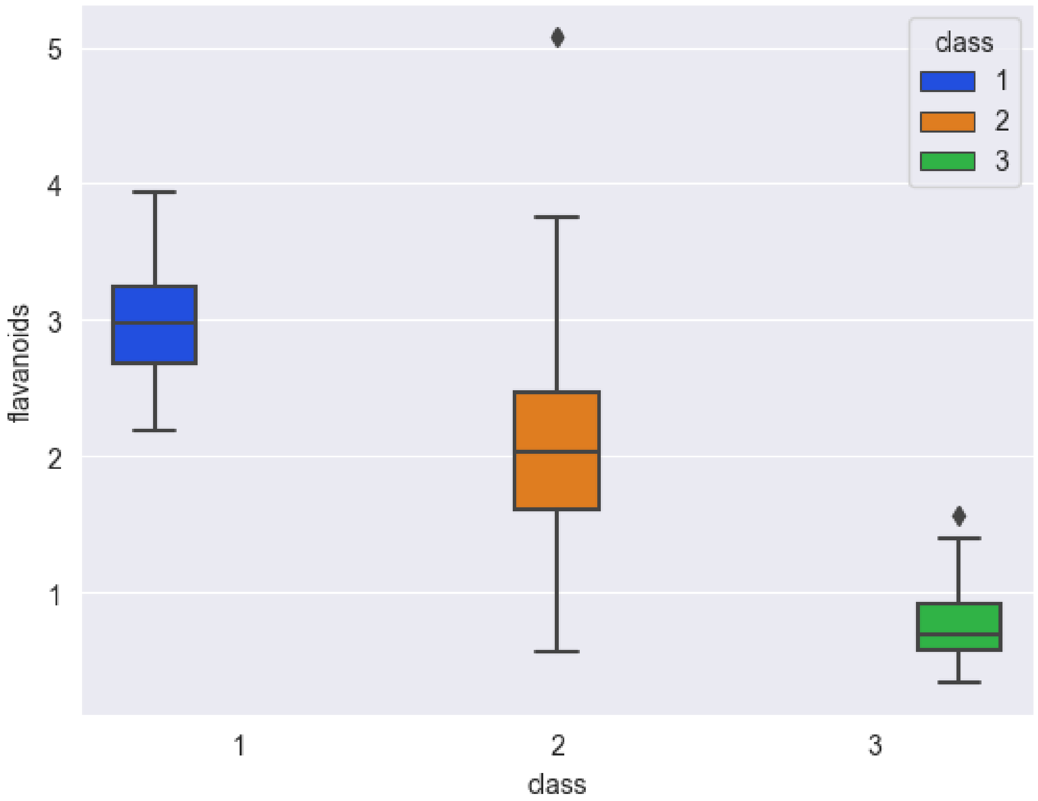

class
1
2
3
6
5
4
3
2
1
malic_acid
1
2
3
class

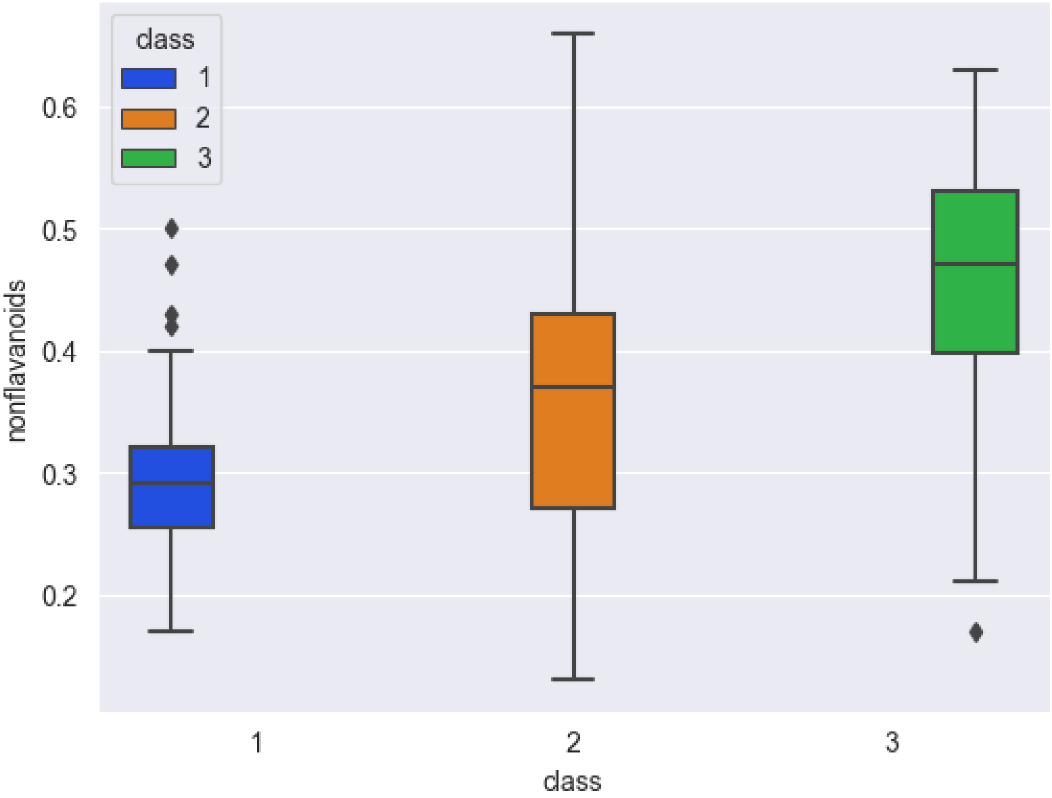

3.25
3.00
2.75
2.50
2.25
2.00
1.75
1.50
ash
class
1
2
3
1
2
3
class

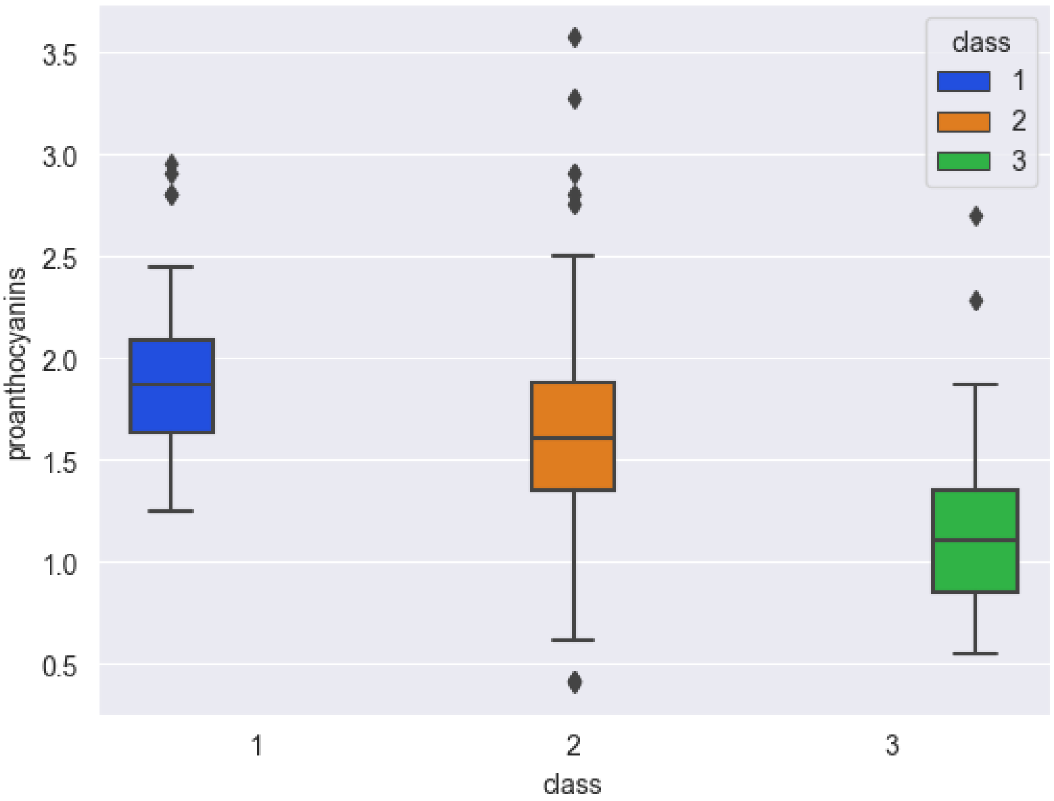

class
1
2
3
alcalinity
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
1
2
3
class

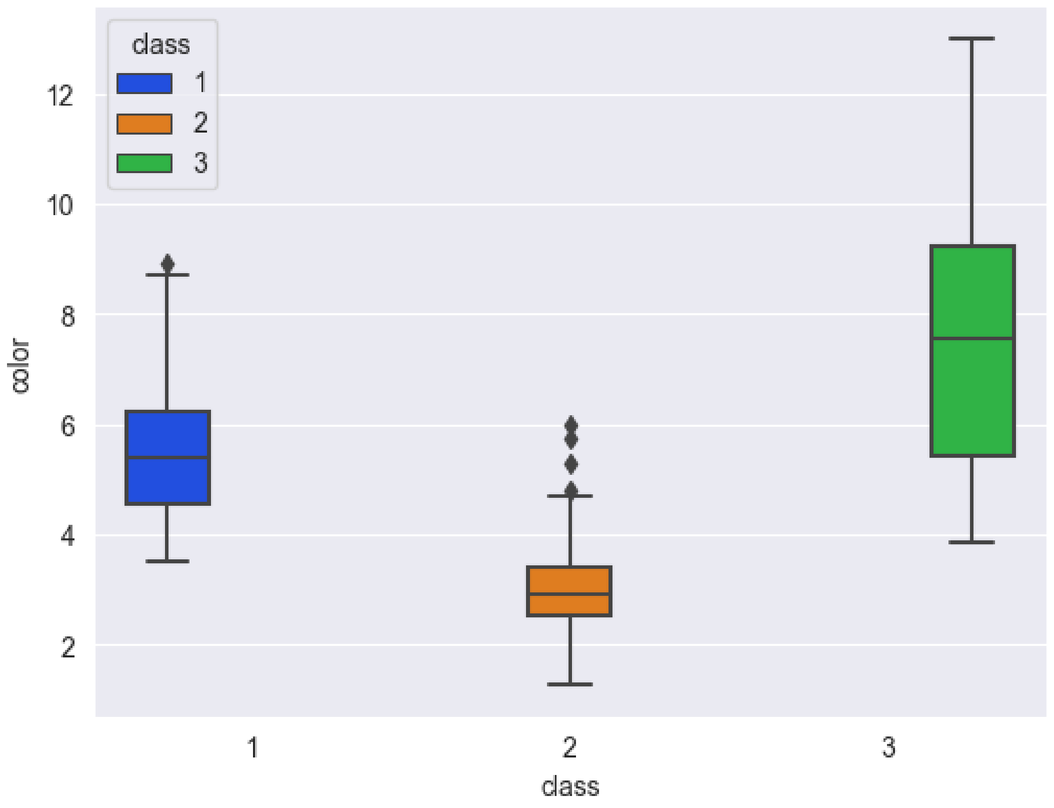

160
140
120
100
80
magnesium
1
2
3
class
class
1
2
3

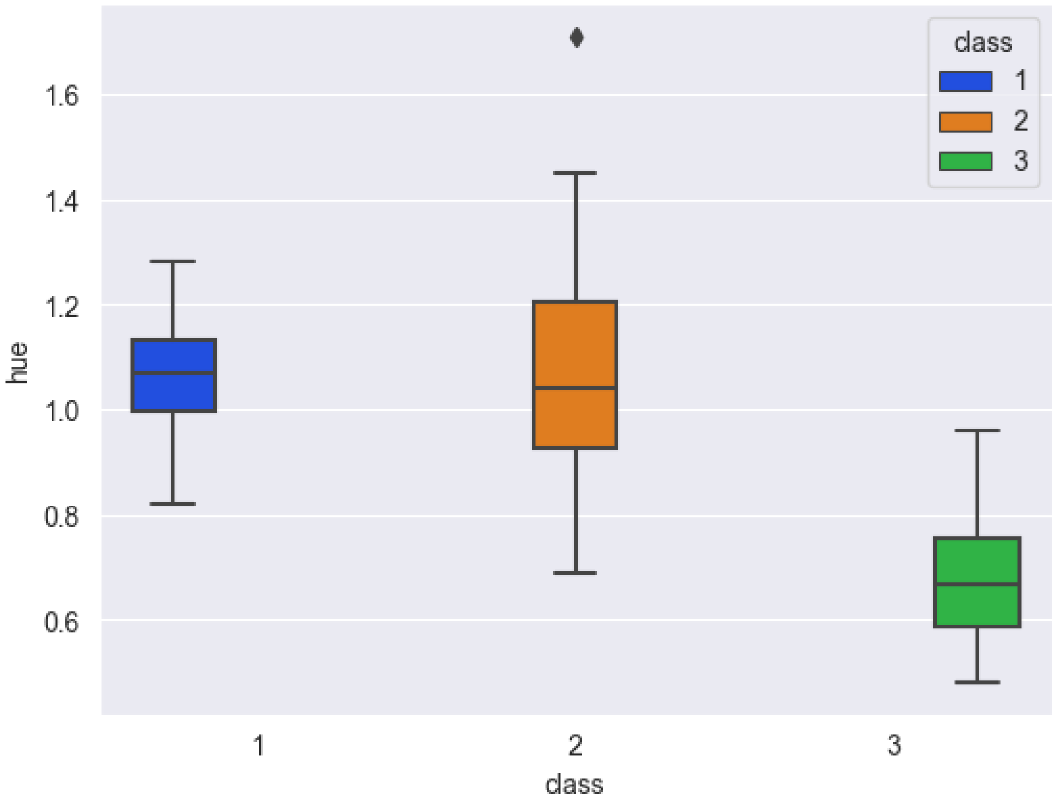

4.0
3.5
3.0
2.5
2.0
1.5
1.0
phenols
1
2
3
class
class
1
2
3

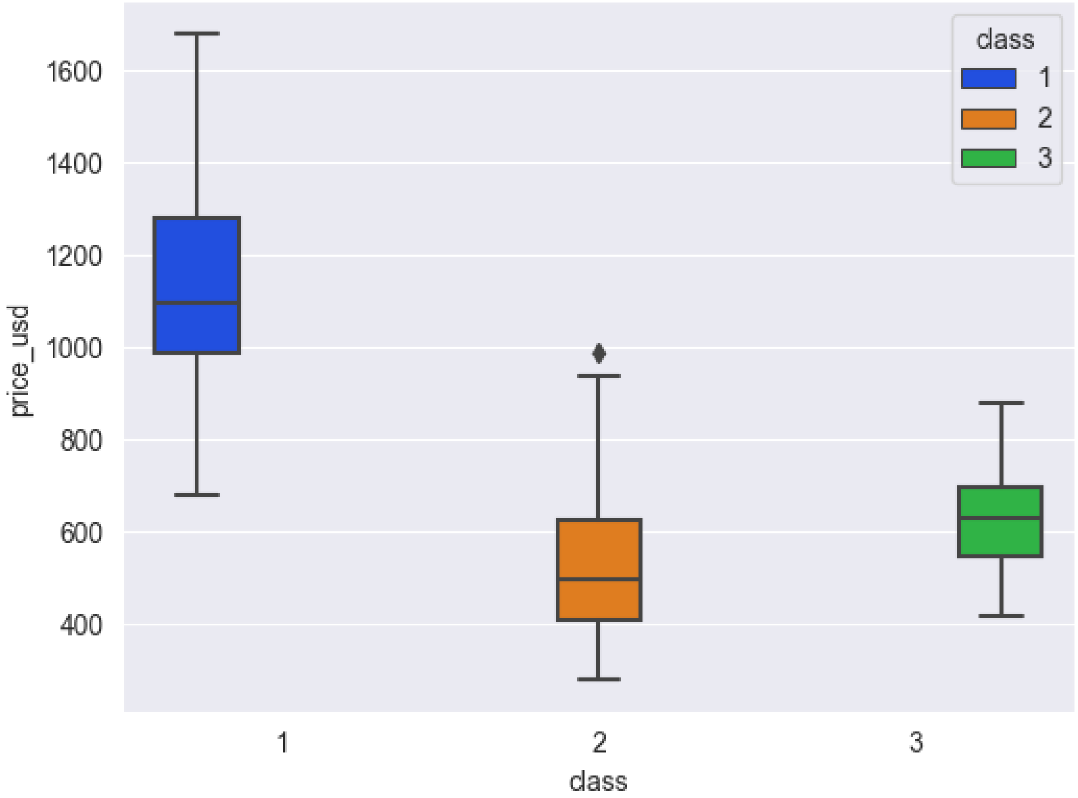

class
1
2
3
5
4
3
2
1
flavanoids
1
2
3
class


class
1
2
3
0.6
0.5
0.4
0.3
0.2
nonflavanoids
1
2
3
class


3.5
3.0
2.5
2.0
1.5
1.0
0.5
proanthocyanins
1
2
3
class
class
🟦 1
🟧 2
🟩 3


class
1
2
3
12
10
8
6
4
2
color
1
2
3
class


class
1
2
3
1.6
1.4
1.2
1.0
0.8
0.6
hue
1
2
3
class



```
[22]: """Describe each measure by classification"""
      grp = df.groupby('class').describe().T.reset_index(names=['feature', 'measure'])
      desc = {k:v.drop(columns='feature').set_index('measure',drop=True) for k,v in␣
       ↪grp.groupby('feature')}
      desc
```

```
[22]: {'alcalinity': class            1          2          3
       measure
       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,
       'alcohol_percentage': class            1          2          3
       measure
       count     59.000000  71.000000  48.000000
       mean      13.744746  12.278732  13.153750
       std        0.462125   0.537964   0.530241
```

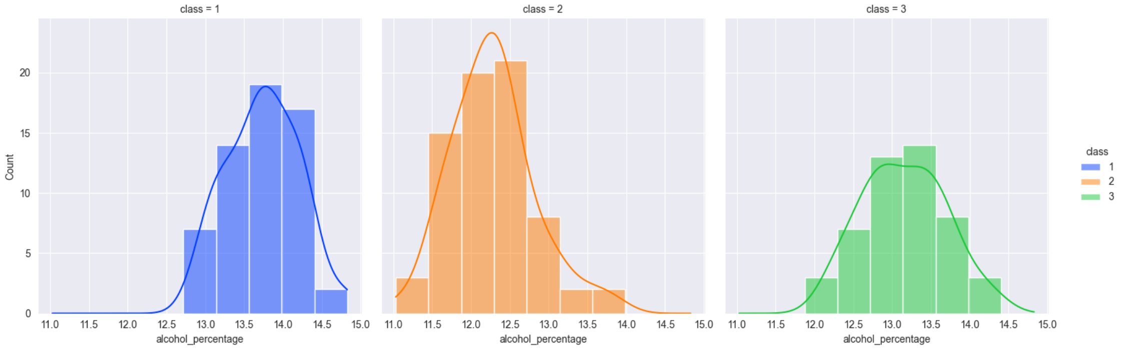
```
min      12.850000  11.030000  12.200000
25%      13.400000  11.915000  12.805000
50%      13.750000  12.290000  13.165000
75%      14.100000  12.515000  13.505000
max      14.830000  13.860000  14.340000,
'ash': class            1          2          3
measure
count    59.000000  71.000000  48.000000
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,
'color': class            1          2          3
measure
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
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,
'flavanoids': class            1          2          3
measure
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,
'hue': class            1          2          3
measure
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,
'magnesium': class             1           2           3
measure
```

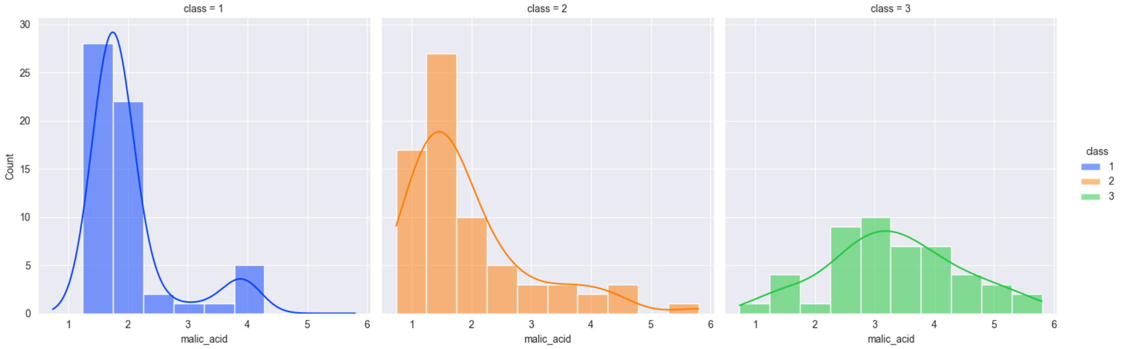
```
count     59.000000   71.000000   48.000000
mean     106.338983   94.549296   99.312500
std       10.498949   16.753497   10.890473
min       89.000000   70.000000   80.000000
25%       98.000000   85.500000   89.750000
50%      104.000000   88.000000   97.000000
75%      114.000000   99.500000  106.000000
max      132.000000  162.000000  123.000000,
'malic_acid': class            1          2          3
measure
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,
'nonflavanoids': class            1          2         3
measure
count    59.000000  71.000000  48.00000
mean      0.290000   0.363662   0.44750
std       0.070049   0.123961   0.12414
min       0.170000   0.130000   0.17000
25%       0.255000   0.270000   0.39750
50%       0.290000   0.370000   0.47000
75%       0.320000   0.430000   0.53000
max       0.500000   0.660000   0.63000,
'phenols': class            1          2          3
measure
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,
'price_usd': class              1           2           3
measure
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
```

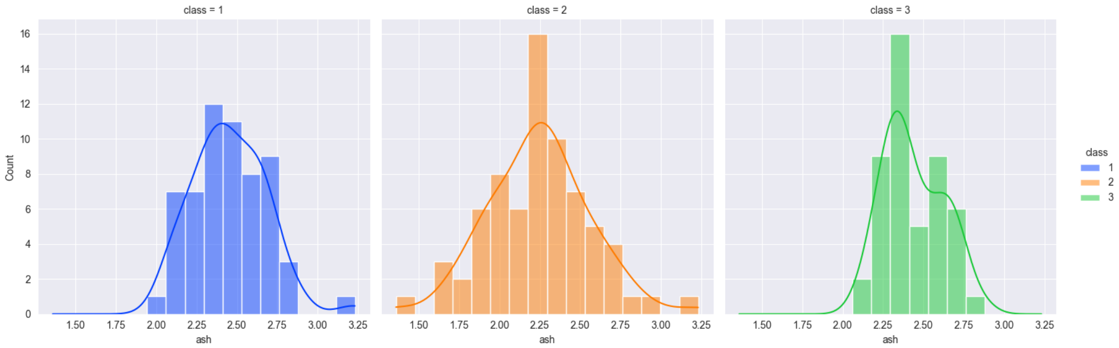
```
max      1680.000000  985.000000  880.000000,
'proanthocyanins': class            1          2          3
measure
count    59.000000  71.000000  48.000000
mean      1.899322   1.630282   1.153542
std       0.412109   0.602068   0.408836
min       1.250000   0.410000   0.550000
25%       1.640000   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}
```

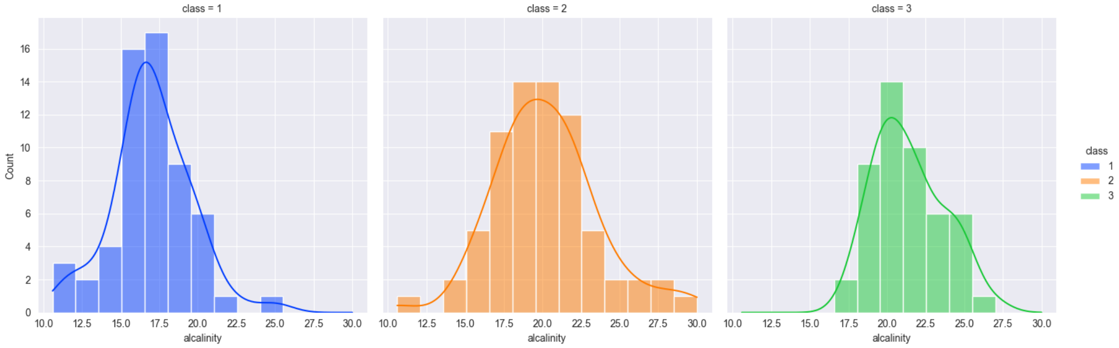
[15]:

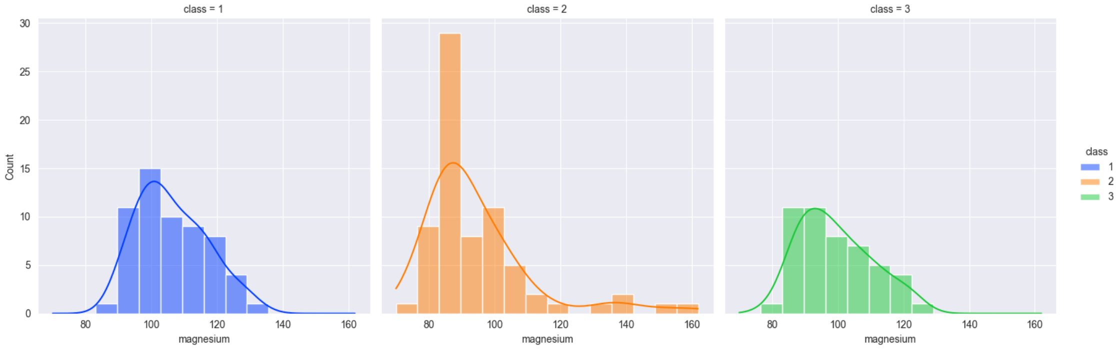
```python
import seaborn as sns
for col in df.columns[1:]:
    sns.displot(df, x=col, col='class', kde=True, hue='class', palette='bright')
    plt.show()
```

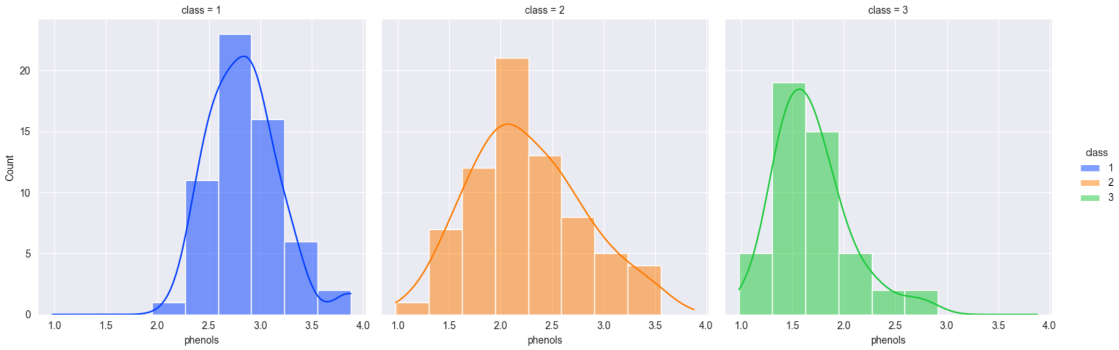


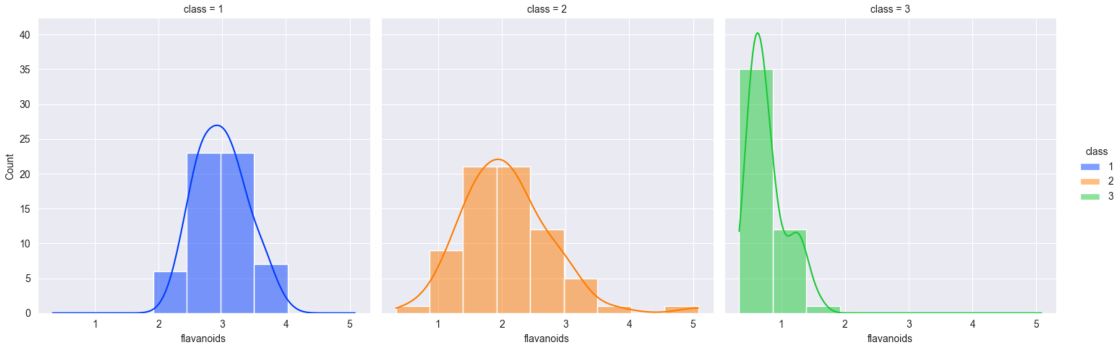


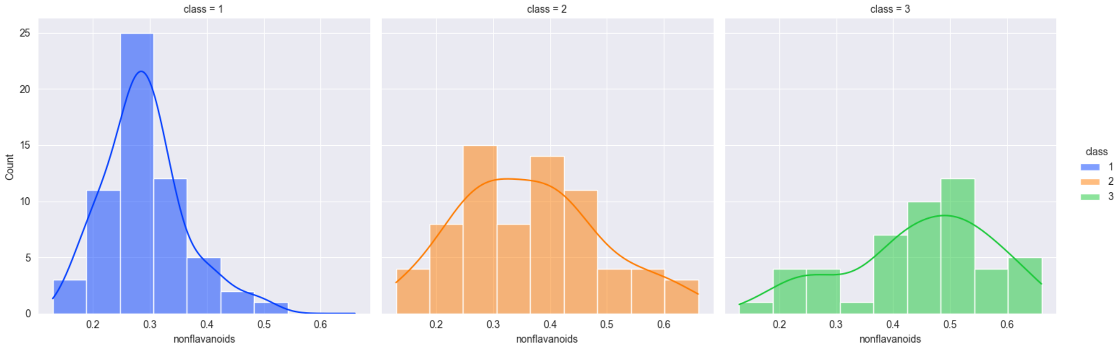

class = 1
class = 2
class = 3
Count
0
2
4
6
8
10
12
14
16
1.50
1.75
2.00
2.25
2.50
2.75
3.00
3.25
ash
class
1
2
3

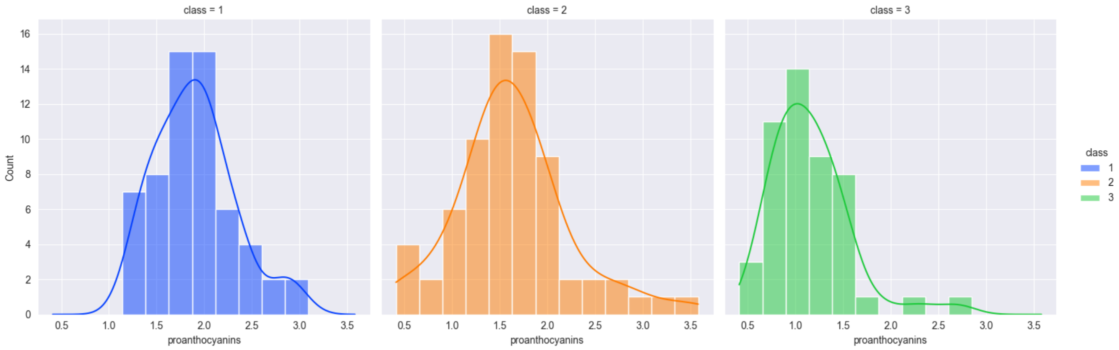

class = 1
class = 2
class = 3
Count
0
2
4
6
8
10
12
14
16
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0
alcalinity
class
1
2
3

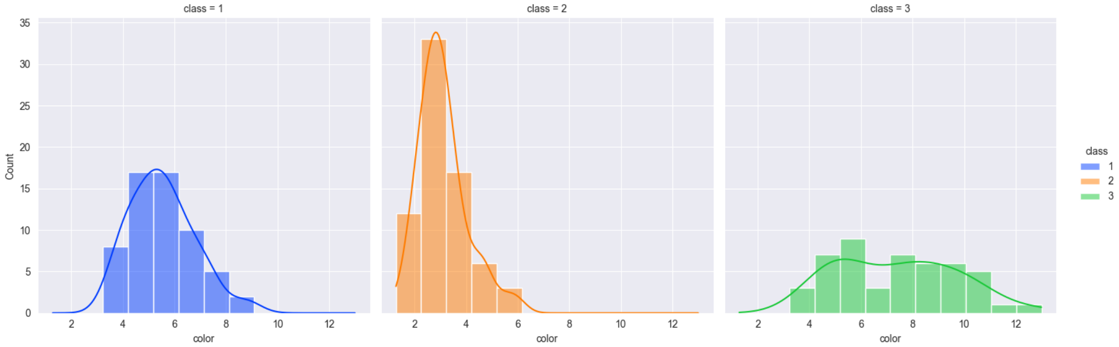

class = 1
class = 2
class = 3
Count
0
5
10
15
20
25
30
80
100
120
140
160
magnesium
class
1
2
3

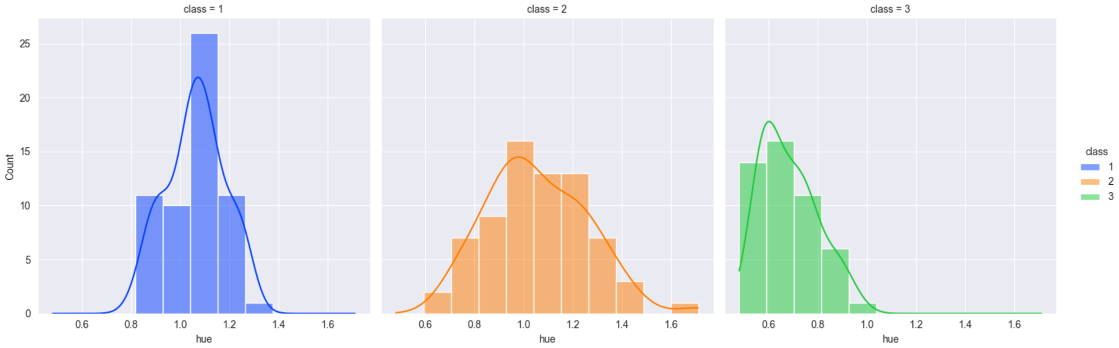

class = 1
class = 2
class = 3
Count
0
5
10
15
20
1.0
1.5
2.0
2.5
3.0
3.5
4.0
phenols
class
1
2
3

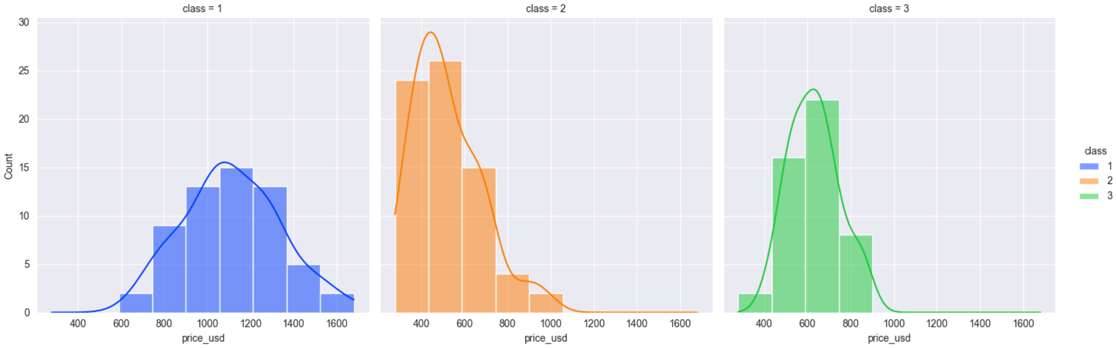

class = 1
class = 2
class = 3
Count
0
5
10
15
20
25
30
35
40
1
2
3
4
5
flavanoids
class
1
2
3


class = 1
class = 2
class = 3
Count
0
5
10
15
20
25
0.2
0.3
0.4
0.5
0.6
nonflavanoids
class
1
2
3


class = 1
class = 2
class = 3
Count
0
2
4
6
8
10
12
14
16
0.5
1.0
1.5
2.0
2.5
3.0
3.5
proanthocyanins
class
1
2
3


class = 1
class = 2
class = 3
Count
0
5
10
15
20
25
30
35
2
4
6
8
10
12
color
class
1
2
3


class = 1
class = 2
class = 3
Count
0
5
10
15
20
25
0.6
0.8
1.0
1.2
1.4
1.6
hue
class
1
2
3


class = 1
class = 2
class = 3
Count
0
5
10
15
20
25
30
400
600
800
1000
1200
1400
1600
price_usd
class
1
2
3