oliveOilAnalysis-python

February 6, 2024

0.1 Required Packages

```
[ ]: #Import the necessary libraries
     import pandas as pd
     import seaborn as sb
     from matplotlib import pyplot as plt

     sb.set_context("notebook", font_scale=2.5) # Comment this line if you are␣
      ↪running directly in Python.
     %matplotlib inline
```

```
/var/folders/49/qhkpplds6xxghpsg8cyk9crc0000gn/T/ipykernel_17677/243197114.py:2:
DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of
pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better
interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

0.2 Load the Data

```
[ ]: # Loading into Pandas DataFrame
     oo_sensory = pd.read_csv("olive_oil_sensory.csv")
```

```
[ ]: # Loading into Pandas DataFrame
     oo_chemical = pd.read_csv("olive_oil_chemical.csv")
```

0.3 Combining Data

```
[ ]: combined = pd.merge(oo_sensory, oo_chemical, how='left', on='region')
     combined.head(16)
```

```
[ ]:   region  s_yellow  s_green  s_brown  s_glossy  s_transp  s_syrup  c_Acidity  \
     0      G1      21.4     73.4     10.1      79.7      75.2     50.3       0.73
```

```
1      G2      23.4     66.3      9.8     77.8      68.7     51.7       0.19
2      G3      32.7     53.5      8.7     82.3      83.2     45.4       0.26
3      G4      30.2     58.3     12.2     81.1      77.1     47.8       0.67
4      G5      51.8     32.5      8.0     72.4      65.3     46.5       0.52
5      I1      40.7     42.9     20.1     67.7      63.5     52.2       0.26
6      I2      53.8     30.4     11.5     77.8      77.3     45.2       0.24
7      I3      26.4     66.5     14.2     78.7      74.6     51.8       0.30
8      I4      65.7     12.1     10.3     81.6      79.6     48.3       0.35
9      I5      45.0     31.9     28.4     75.7      72.9     52.8       0.19
10     S1      70.9     12.2     10.8     87.7      88.1     44.5       0.15
11     S2      73.5      9.7      8.3     89.9      89.7     42.3       0.16
12     S3      68.1     12.0     10.8     78.4      75.1     46.4       0.27
13     S4      67.6     13.9     11.9     84.6      83.8     48.5       0.16
14     S5      71.4     10.6     10.8     88.1      88.5     46.7       0.24
15     S6      71.4     10.0     11.4     89.5      88.5     47.2       0.30

    c_Peroxide  c_K232  c_K270    c_DK
0         12.70   1.900  0.1390  0.003
1         12.30   1.678  0.1160 -0.004
2         10.30   1.629  0.1160 -0.005
3         13.70   1.701  0.1680 -0.002
4         11.20   1.539  0.1190 -0.001
5         18.70   2.117  0.1420  0.001
6         15.30   1.891  0.1160  0.000
7         18.50   1.908  0.1250  0.001
8         15.60   1.824  0.1040  0.000
9         19.40   2.222  0.1580 -0.003
10        10.50   1.522  0.1160 -0.004
11         8.14   1.527  0.1063 -0.002
12        12.50   1.555  0.0930 -0.002
13        11.00   1.573  0.0940 -0.003
14        10.80   1.331  0.0850 -0.003
15        11.40   1.415  0.0930 -0.004
```

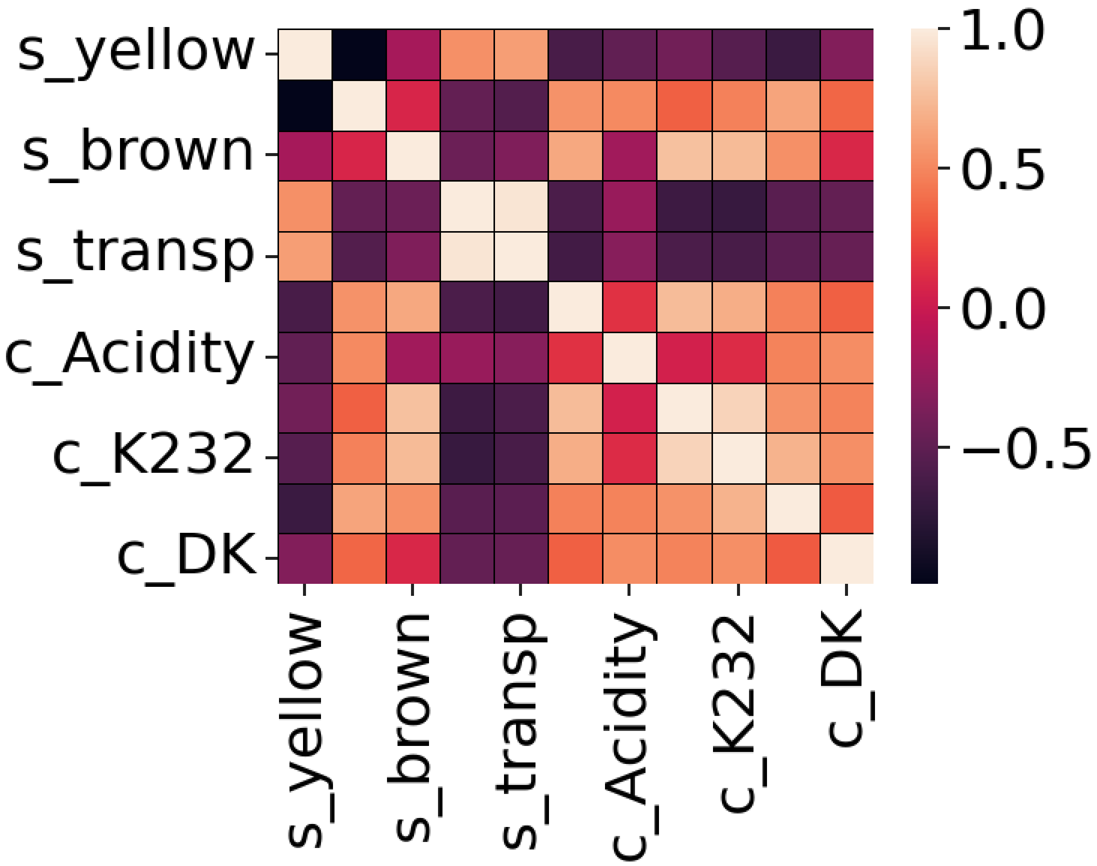
0.4 Delete the first Column

[ ]:

```
numeric = combined.drop('region', axis=1)
cMatrix = numeric.corr()
sb.heatmap(cMatrix, linewidths=0.5, linecolor="black")
```

[ ]:

```
<Axes: >
```



0.5 Describe

```
[ ]: numeric.describe().T
```

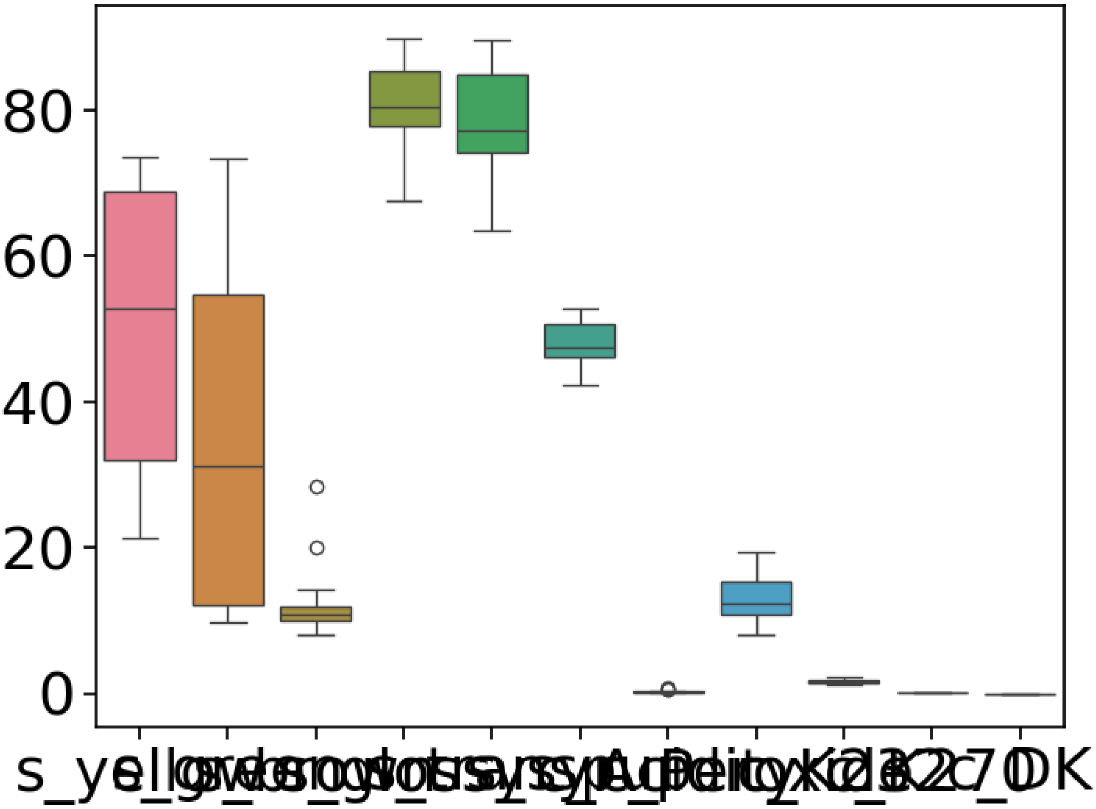
```
[ ]:            count       mean        std     min       25%       50%       75%  \
s_yellow     16.0  50.875000  19.458623  21.400  32.07500  52.8000  68.80000
s_green      16.0  33.512500  23.486986   9.700  12.07500  31.1500  54.70000
s_brown      16.0  12.331250   5.128706   8.000  10.02500  10.8000  11.97500
s_glossy     16.0  80.812500   6.188040  67.700  77.80000  80.4000  85.37500
s_transp     16.0  78.193750   8.307384  63.500  74.17500  77.2000  84.87500
s_syrup      16.0  47.975000   3.065398  42.300  46.15000  47.5000  50.65000
c_Acidity    16.0   0.311875   0.176568   0.150   0.19000   0.2600   0.31250
c_Peroxide   16.0  13.252500   3.345141   8.140  10.95000  12.4000  15.37500
c_K232       16.0   1.708250   0.248731   1.331   1.53600   1.6535   1.89325
c_K270       16.0   0.118144   0.023707   0.085   0.10150   0.1160   0.12850
c_DK         16.0  -0.001750   0.002236  -0.005  -0.00325  -0.0020   0.00000

               max
s_yellow    73.500
```

```
s_green      73.400
s_brown      28.400
s_glossy     89.900
s_transp     89.700
s_syrup      52.800
c_Acidity     0.730
c_Peroxide   19.400
c_K232        2.222
c_K270        0.168
c_DK          0.003
```

0.6 Boxplot

```
[ ]: plt.figure(figsize=(8, 6))
     sb.boxplot(data=numeric)
```

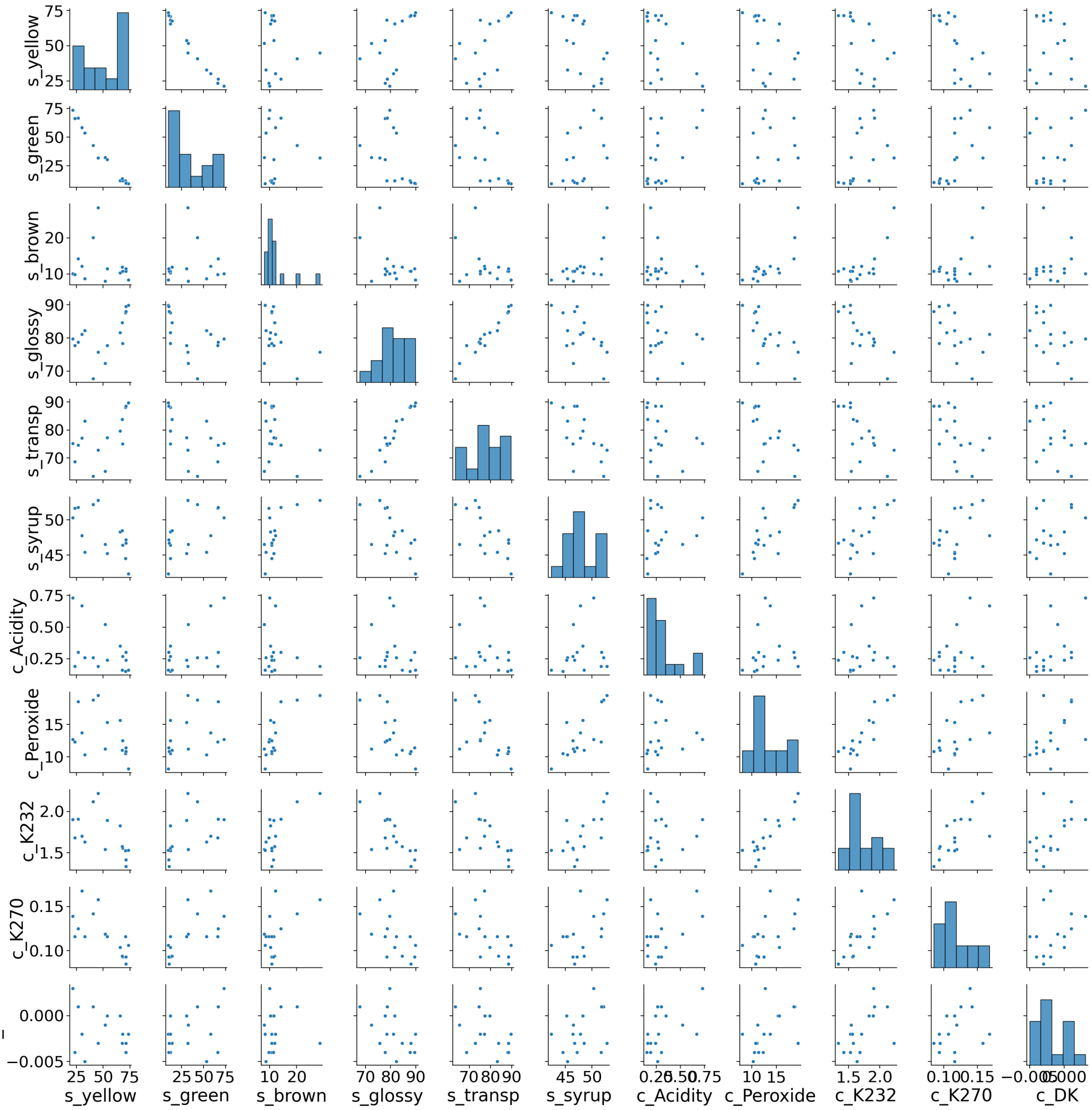
```
[ ]: <Axes: >
```



0.7 Pairplot

```
[ ]: sb.pairplot(numeric)
```

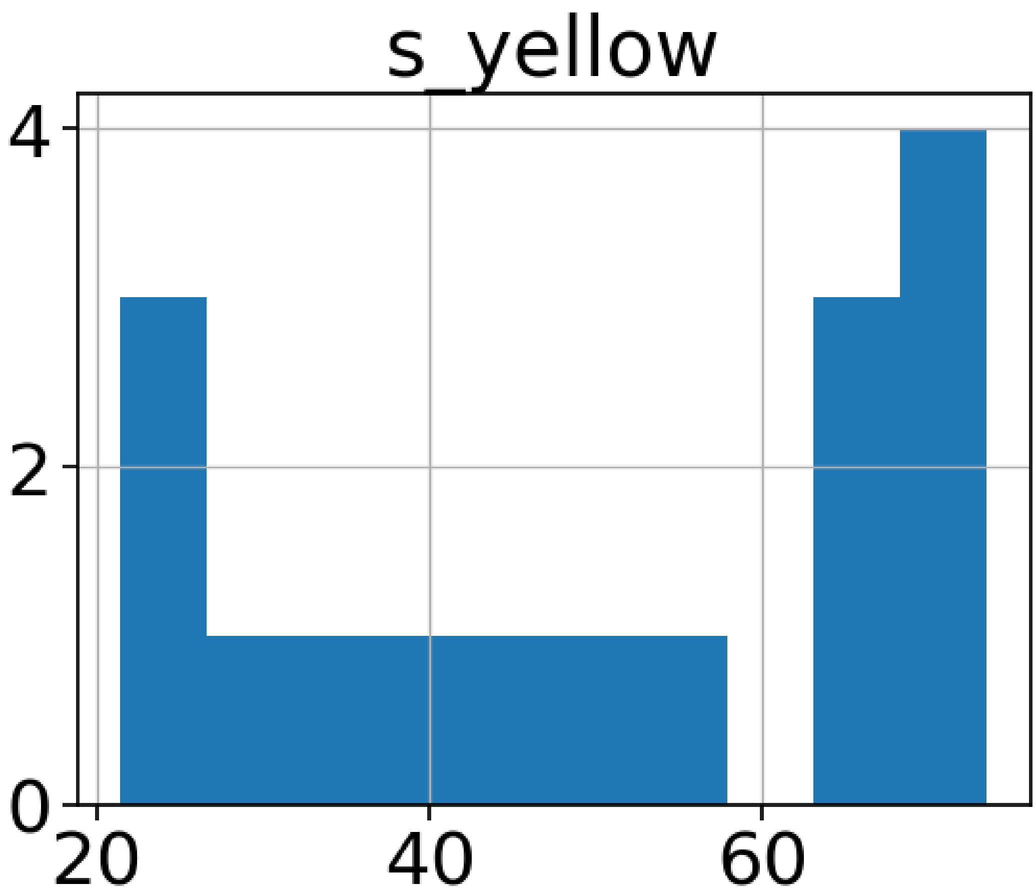
```
[ ]: <seaborn.axisgrid.PairGrid at 0x13c2826f0>
```



0.8 Histogram

```
[ ]: numeric.hist("s_yellow")
```

```
[ ]: array([[<Axes: title={'center': 's_yellow'}>]], dtype=object)
```



0.9 Mean, Median, Diff

```
[ ]: means = numeric.mean()
     print(means)
```

```
s_yellow      50.875000
s_green       33.512500
s_brown       12.331250
s_glossy      80.812500
s_transp      78.193750
s_syrup       47.975000
c_Acidity      0.311875
c_Peroxide    13.252500
c_K232         1.708250
c_K270         0.118144
c_DK          -0.001750
dtype: float64
```

```
[ ]: median = numeric.median()
     print(median)
```

```
s_yellow      52.8000
s_green       31.1500
s_brown       10.8000
s_glossy      80.4000
s_transp      77.2000
s_syrup       47.5000
c_Acidity      0.2600
c_Peroxide    12.4000
c_K232         1.6535
c_K270         0.1160
c_DK          -0.0020
dtype: float64
```

```
[ ]: for i in range(11):
         print(means.iloc[i] - median.iloc[i])
```

```
-1.9249999999999972
2.3625000000000043
1.53125
0.4124999999999943
0.9937500000000057
0.4750000000000014
0.051875000000000004
0.8525000000000009
0.054750000000000076
0.0021437499999999986
0.00025
```