```
import pandas as pd
import numpy as np
from scipy.stats import zscore
import matplotlib.pyplot as plt


from sklearn.datasets import load_wine

# Load the Wine dataset
wine = load_wine()
X = pd.DataFrame(wine.data, columns=wine.feature_names)
y = pd.Series(wine.target, name='target')


# Combining X and y into a single DataFrame for the analysis
df = X.copy()
df['target'] = y


# 1. Display the shape of the dataset
shape_original = X.shape
print(shape_original)
```

```
(178, 13)
```

```
# 2. Print the header and the last few rows
header = df.head()
tail = df.tail()
print(header)
print(tail)
```

```
   alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
0    14.23        1.71  2.43               15.6      127.0           2.80
1    13.20        1.78  2.14               11.2      100.0           2.65
2    13.16        2.36  2.67               18.6      101.0           2.80
3    14.37        1.95  2.50               16.8      113.0           3.85
4    13.24        2.59  2.87               21.0      118.0           2.80

   flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
0        3.06                  0.28             2.29             5.64  1.04
1        2.76                  0.26             1.28             4.38  1.05
2        3.24                  0.30             2.81             5.68  1.03
3        3.49                  0.24             2.18             7.80  0.86
4        2.69                  0.39             1.82             4.32  1.04

   od280/od315_of_diluted_wines  proline  target
0                          3.92   1065.0       0
1                          3.40   1050.0       0
2                          3.17   1185.0       0
3                          3.45   1480.0       0
4                          2.93    735.0       0
     alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
173    13.71        5.65  2.45               20.5       95.0           1.68
174    13.40        3.91  2.48               23.0      102.0           1.80
175    13.27        4.28  2.26               20.0      120.0           1.59
176    13.17        2.59  2.37               20.0      120.0           1.65
177    14.13        4.10  2.74               24.5       96.0           2.05

     flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
173        0.61                  0.52             1.06              7.7  0.64
174        0.75                  0.43             1.41              7.3  0.70
175        0.69                  0.43             1.35             10.2  0.59
176        0.68                  0.53             1.46              9.3  0.60
177        0.76                  0.56             1.35              9.2  0.61

     od280/od315_of_diluted_wines  proline  target
173                          1.74    740.0       2
174                          1.56    750.0       2
175                          1.56    835.0       2
176                          1.62    840.0       2
177                          1.60    560.0       2
```

```
# 3. Print a summary of the dataset's statistical details
summary_statistics = df.describe()
print(summary_statistics)
```

```
            alcohol  malic_acid         ash  alcalinity_of_ash   magnesium  \
count  178.000000  178.000000  178.000000         178.000000  178.000000
mean    13.000618    2.336348    2.366517          19.494944   99.741573
```

```
std      0.811827    1.117146    0.274344           3.339564   14.282484
min     11.030000    0.740000    1.360000          10.600000   70.000000
25%     12.362500    1.602500    2.210000          17.200000   88.000000
50%     13.050000    1.865000    2.360000          19.500000   98.000000
75%     13.677500    3.082500    2.557500          21.500000  107.000000
max     14.830000    5.800000    3.230000          30.000000  162.000000

       total_phenols  flavanoids  nonflavanoid_phenols  proanthocyanins  \
count     178.000000  178.000000            178.000000       178.000000
mean        2.295112    2.029270              0.361854         1.590899
std         0.625851    0.998859              0.124453         0.572359
min         0.980000    0.340000              0.130000         0.410000
25%         1.742500    1.205000              0.270000         1.250000
50%         2.355000    2.135000              0.340000         1.555000
75%         2.800000    2.875000              0.437500         1.950000
max         3.880000    5.080000              0.660000         3.580000

       color_intensity         hue  od280/od315_of_diluted_wines      proline  \
count       178.000000  178.000000                    178.000000   178.000000
mean          5.058090    0.957449                      2.611685   746.893258
std           2.318286    0.228572                      0.709990   314.907474
min           1.280000    0.480000                      1.270000   278.000000
25%           3.220000    0.782500                      1.937500   500.500000
50%           4.690000    0.965000                      2.780000   673.500000
75%           6.200000    1.120000                      3.170000   985.000000
max          13.000000    1.710000                      4.000000  1680.000000

           target
count  178.000000
mean     0.938202
std      0.775035
min      0.000000
25%      0.000000
50%      1.000000
75%      2.000000
max      2.000000
```

```
# 4. Display a concise summary of the dataframe
info_summary = df.info()
print(info_summary)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   alcohol                       178 non-null    float64
 1   malic_acid                    178 non-null    float64
 2   ash                           178 non-null    float64
 3   alcalinity_of_ash             178 non-null    float64
 4   magnesium                     178 non-null    float64
 5   total_phenols                 178 non-null    float64
 6   flavanoids                    178 non-null    float64
 7   nonflavanoid_phenols          178 non-null    float64
 8   proanthocyanins               178 non-null    float64
 9   color_intensity               178 non-null    float64
 10  hue                           178 non-null    float64
 11  od280/od315_of_diluted_wines  178 non-null    float64
 12  proline                       178 non-null    float64
 13  target                        178 non-null    int64
dtypes: float64(13), int64(1)
memory usage: 19.6 KB
None
```

```
# 5. Add an index column and display its new shape
df.reset_index(inplace=True)
shape_with_index = df.shape
print(shape_with_index)
```

```
(178, 15)
```

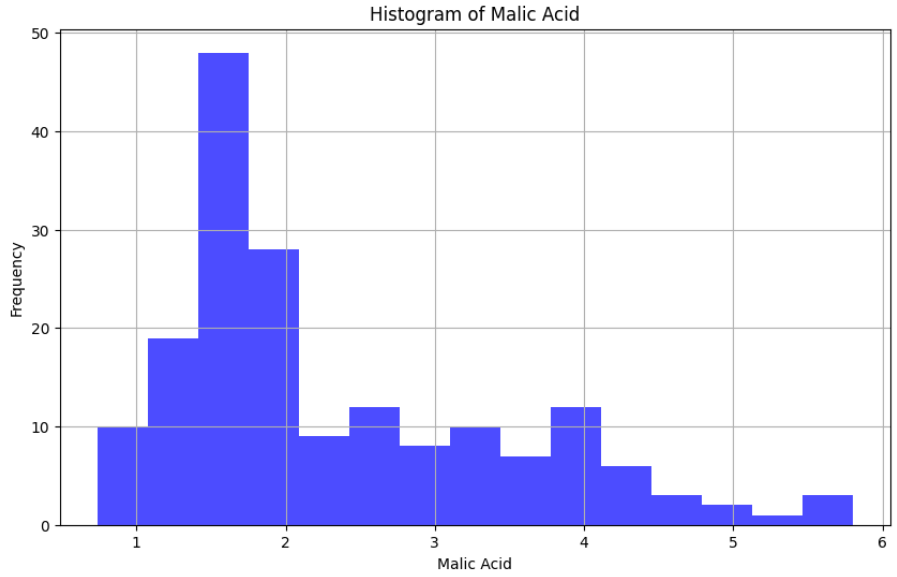
```
# 6. Choose a numerical field and print its unique values (using 'alcohol' as an example)
unique_values_alcohol = df['alcohol'].unique()
print(unique_values_alcohol)
```

```
[14.23 13.2  13.16 14.37 13.24 14.2  14.39 14.06 14.83 13.86 14.1  14.12
 13.75 14.75 14.38 13.63 14.3  13.83 14.19 13.64 12.93 13.71 12.85 13.5
 13.05 13.39 13.3  13.87 14.02 13.73 13.58 13.68 13.76 13.51 13.48 13.28
 13.07 14.22 13.56 13.41 13.88 14.21 13.9  13.94 13.82 13.77 13.74 13.29
 13.72 12.37 12.33 12.64 13.67 12.17 13.11 13.34 12.21 12.29 13.49 12.99
 11.96 11.66 13.03 11.84 12.7  12.   12.72 12.08 12.67 12.16 11.65 11.64
```

```
       12.69 11.62 12.47 11.81 12.6  12.34 11.82 12.51 12.42 12.25 12.22 11.61
       11.46 12.52 11.76 11.41 11.03 12.77 11.45 11.56 11.87 12.07 12.43 11.79
       12.04 12.86 12.88 12.81 12.53 12.84 13.36 13.52 13.62 12.87 13.32 13.08
       12.79 13.23 12.58 13.17 13.84 12.45 14.34 12.36 13.69 12.96 13.78 13.45
       12.82 13.4  12.2  14.16 13.27 14.13]
```

```
# 7. Replace an extreme set of values in the dataset with NaN (using max value in 'alcohol' as example)
df.loc[df['alcohol'] == df['alcohol'].max(), 'alcohol'] = np.NaN
```

```
# 8. Plot a histogram of a numerical variable (using 'malic_acid')
plt.figure(figsize=(10, 6))
plt.hist(df['malic_acid'], bins=15, color='blue', alpha=0.7)
plt.title('Histogram of Malic Acid')
plt.xlabel('Malic Acid')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

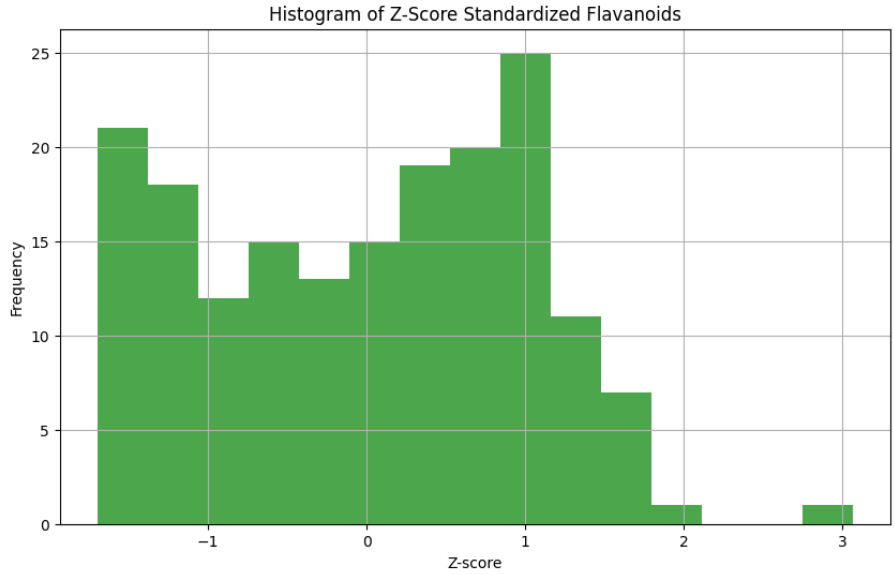



```
# 9. Represent 'target' as a numerical field
tail_with_new_column = df.tail()
print(tail_with_new_column)
```

```
     index  alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  \
173    173    13.71        5.65  2.45               20.5       95.0
174    174    13.40        3.91  2.48               23.0      102.0
175    175    13.27        4.28  2.26               20.0      120.0
176    176    13.17        2.59  2.37               20.0      120.0
177    177    14.13        4.10  2.74               24.5       96.0

     total_phenols  flavanoids  nonflavanoid_phenols  proanthocyanins  \
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  target  \
173              7.7  0.64                          1.74    740.0       2
174              7.3  0.70                          1.56    750.0       2
175             10.2  0.59                          1.56    835.0       2
176              9.3  0.60                          1.62    840.0       2
177              9.2  0.61                          1.60    560.0       2

     zscore_flavanoids
173          -1.424900
174          -1.284344
175          -1.344582
```

```
176          -1.354622
177          -1.274305
```

```
# 10. Standardize a numerical field ('flavanoids') and show its histogram
df['zscore_flavanoids'] = zscore(df['flavanoids'])
plt.figure(figsize=(10, 6))
plt.hist(df['zscore_flavanoids'], bins=15, color='green', alpha=0.7)
plt.title('Histogram of Z-Score Standardized Flavanoids')
plt.xlabel('Z-score')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```




```
# 11. Identify outliers (using z-score > 3 or < -3 as criterion for 'flavanoids')
outliers = df[np.abs(zscore(df['flavanoids'])) > 3]
head_outliers = outliers.head()
print(head_outliers)
```

```
     index  alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  \
121    121    11.56        2.05  3.23               28.5      119.0

     total_phenols  flavanoids  nonflavanoid_phenols  proanthocyanins  \
121           3.18        5.08                  0.47             1.87

     color_intensity   hue  od280/od315_of_diluted_wines  proline  target  \
121              6.0  0.93                          3.69    465.0       1

     zscore_flavanoids
121           3.062832
```

```
# 12. Sort the dataset by 'alcohol' and display 15 interesting fields
df_sorted = df.sort_values(by='alcohol')
interesting_fields = df_sorted.iloc[:, :15].head(15)
print(interesting_fields)
```

```
     index  alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  \
115    115    11.03        1.51  2.20               21.5       85.0
113    113    11.41        0.74  2.50               21.0       88.0
120    120    11.45        2.40  2.42               20.0       96.0
110    110    11.46        3.74  1.82               19.5      107.0
121    121    11.56        2.05  3.23               28.5      119.0
109    109    11.61        1.35  2.70               20.0       94.0
94      94    11.62        1.99  2.28               18.0       98.0
88      88    11.64        2.06  2.46               21.6       84.0
87      87    11.65        1.67  2.62               26.0       88.0
75      75    11.66        1.88  1.92               16.0       97.0
112    112    11.76        2.68  2.92               20.0      103.0
127    127    11.79        2.13  2.78               28.5       92.0
```

```
96      96    11.81        2.12  2.74              21.5      134.0
103    103    11.82        1.72  1.88              19.5       86.0
116    116    11.82        1.47  1.99              20.8       86.0

     total_phenols  flavanoids  nonflavanoid_phenols  proanthocyanins  \
115           2.46        2.17                  0.52             2.01
113           2.48        2.01                  0.42             1.44
120           2.90        2.79                  0.32             1.83
110           3.18        2.58                  0.24             3.58
121           3.18        5.08                  0.47             1.87
109           2.74        2.92                  0.29             2.49
94            3.02        2.26                  0.17             1.35
88            1.95        1.69                  0.48             1.35
87            1.92        1.61                  0.40             1.34
75            1.61        1.57                  0.34             1.15
112           1.75        2.03                  0.60             1.05
127           2.13        2.24                  0.58             1.76
96            1.60        0.99                  0.14             1.56
103           2.50        1.64                  0.37             1.42
116           1.98        1.60                  0.30             1.53

     color_intensity   hue  od280/od315_of_diluted_wines  proline  target
115             1.90  1.71                          2.87    407.0       1
113             3.08  1.10                          2.31    434.0       1
120             3.25  0.80                          3.39    625.0       1
110             2.90  0.75                          2.81    562.0       1
121             6.00  0.93                          3.69    465.0       1
109             2.65  0.96                          3.26    680.0       1
94              3.25  1.16                          2.96    345.0       1
88              2.80  1.00                          2.75    680.0       1
87              2.60  1.36                          3.21    562.0       1
75              3.80  1.23                          2.14    428.0       1
112             3.80  1.23                          2.50    607.0       1
127             3.00  0.97                          2.44    466.0       1
96              2.50  0.95                          2.26    625.0       1
103             2.06  0.94                          2.44    415.0       1
116             1.95  0.95                          3.33    495.0       1
```