```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn as sk
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error


df = pd.read_csv("winequality-red.csv")
df.sample
```

```
    <bound method NDFrame.sample of       fixed acidity  volatile acidity  citric acid
    residual sugar  chlorides  \
    0               7.4             0.700         0.00            1.9      0.076
    1               7.8             0.880         0.00            2.6      0.098
    2               7.8             0.760         0.04            2.3      0.092
    3              11.2             0.280         0.56            1.9      0.075
    4               7.4             0.700         0.00            1.9      0.076
    ...             ...               ...          ...            ...        ...
    1594            6.2             0.600         0.08            2.0      0.090
    1595            5.9             0.550         0.10            2.2      0.062
    1596            6.3             0.510         0.13            2.3      0.076
    1597            5.9             0.645         0.12            2.0      0.075
    1598            6.0             0.310         0.47            3.6      0.067

          free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
    0                    11.0                  34.0  0.99780  3.51       0.56
    1                    25.0                  67.0  0.99680  3.20       0.68
    2                    15.0                  54.0  0.99700  3.26       0.65
    3                    17.0                  60.0  0.99800  3.16       0.58
    4                    11.0                  34.0  0.99780  3.51       0.56
    ...                   ...                   ...      ...   ...        ...
    1594                 32.0                  44.0  0.99490  3.45       0.58
    1595                 39.0                  51.0  0.99512  3.52       0.76
    1596                 29.0                  40.0  0.99574  3.42       0.75
    1597                 32.0                  44.0  0.99547  3.57       0.71
    1598                 18.0                  42.0  0.99549  3.39       0.66

          alcohol  quality
    0         9.4        5
    1         9.8        5
    2         9.8        5
    3         9.8        6
    4         9.4        5
    ...       ...      ...
    1594     10.5        5
    1595     11.2        6
    1596     11.0        6
    1597     10.2        5
    1598     11.0        6
```

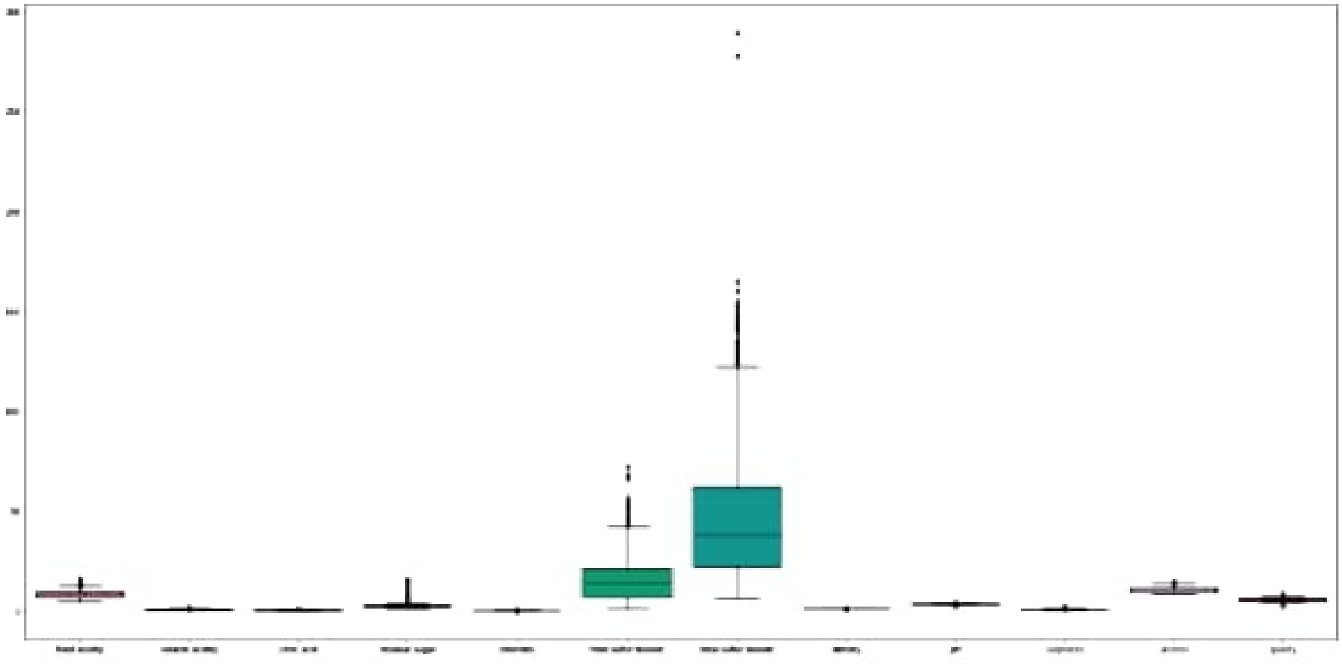
```
[1599 rows x 12 columns]>
```

```
df.describe
```

```
<bound method NDFrame.describe of       fixed acidity  volatile acidity  citric acid
residual sugar  chlorides  \
0               7.4             0.700         0.00             1.9      0.076
1               7.8             0.880         0.00             2.6      0.098
2               7.8             0.760         0.04             2.3      0.092
3              11.2             0.280         0.56             1.9      0.075
4               7.4             0.700         0.00             1.9      0.076
...             ...               ...          ...             ...        ...
1594            6.2             0.600         0.08             2.0      0.090
1595            5.9             0.550         0.10             2.2      0.062
1596            6.3             0.510         0.13             2.3      0.076
1597            5.9             0.645         0.12             2.0      0.075
1598            6.0             0.310         0.47             3.6      0.067

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                    11.0                  34.0  0.99780  3.51       0.56
1                    25.0                  67.0  0.99680  3.20       0.68
2                    15.0                  54.0  0.99700  3.26       0.65
3                    17.0                  60.0  0.99800  3.16       0.58
4                    11.0                  34.0  0.99780  3.51       0.56
...                   ...                   ...      ...   ...        ...
1594                 32.0                  44.0  0.99490  3.45       0.58
1595                 39.0                  51.0  0.99512  3.52       0.76
1596                 29.0                  40.0  0.99574  3.42       0.75
1597                 32.0                  44.0  0.99547  3.57       0.71
1598                 18.0                  42.0  0.99549  3.39       0.66

      alcohol  quality
0         9.4        5
1         9.8        5
2         9.8        5
3         9.8        6
4         9.4        5
...       ...      ...
1594     10.5        5
1595     11.2        6
1596     11.0        6
1597     10.2        5
1598     11.0        6

[1599 rows x 12 columns]>
```

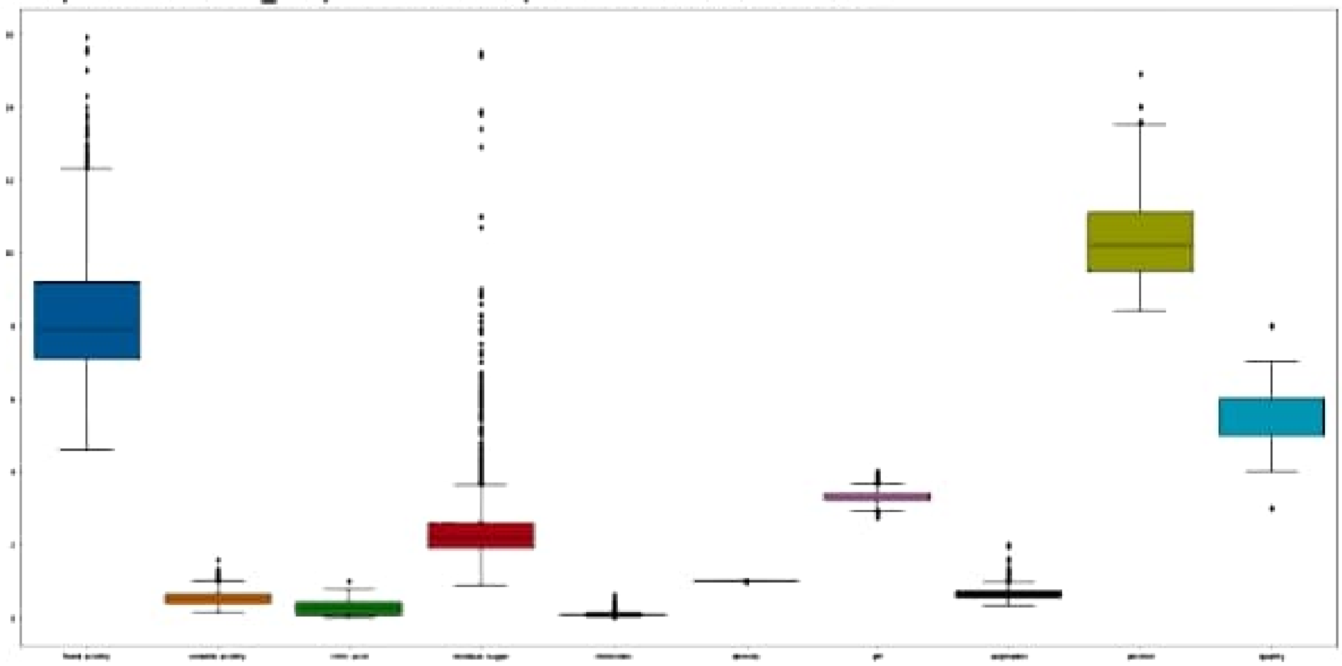
```
plt.figure(figsize=(30,15))
sns.boxplot(data=df)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fa6bdc65550>



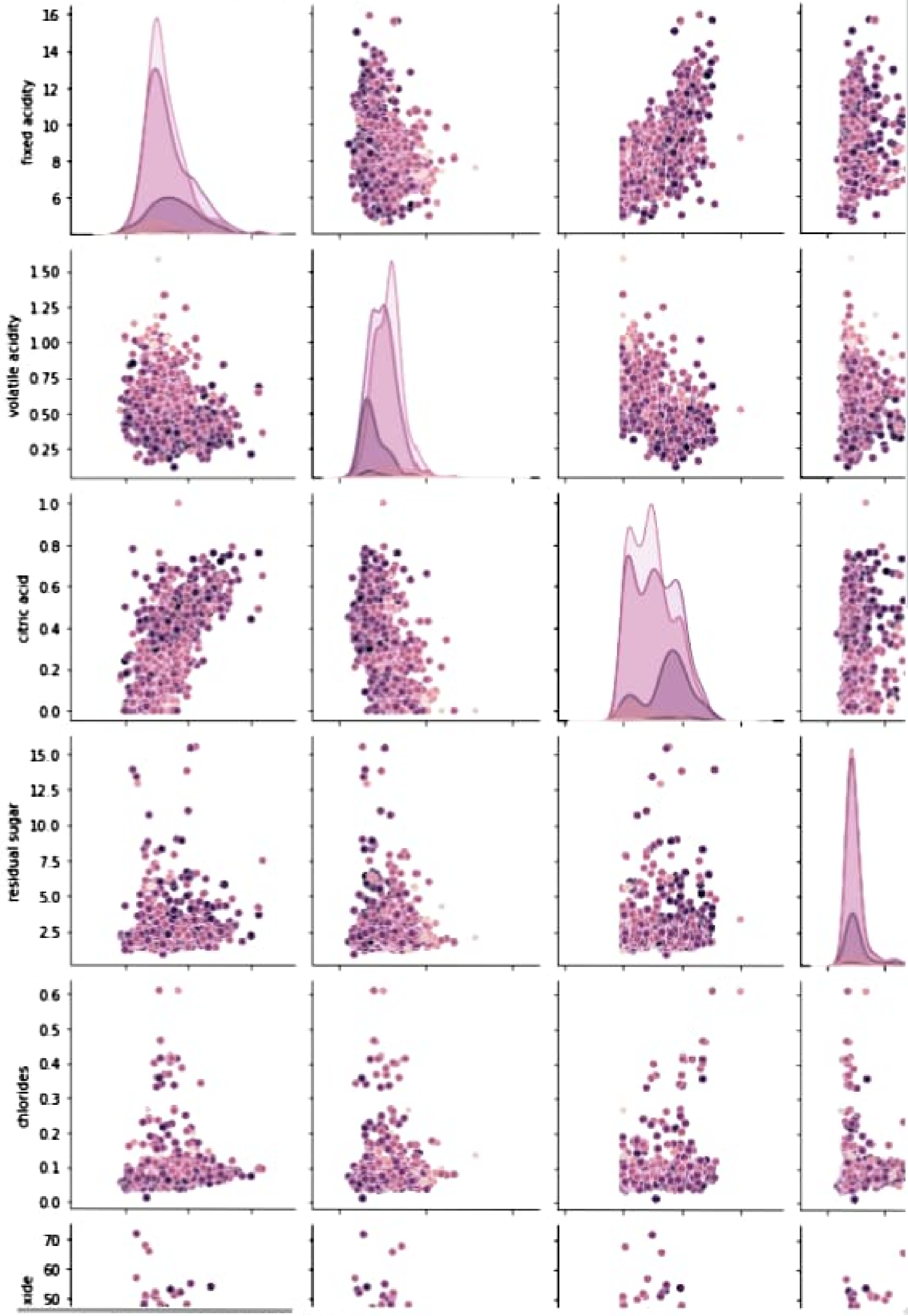
```
plt.figure(figsize=(30,15))
close = df[['fixed acidity','volatile acidity','citric acid','residual sugar','chlorides','de
sns.boxplot(data=close)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fa6c7d04a90>



```
plt.figure(figsize=(30,15))
sns.pairplot(data=df, hue='quality')
plt.show
```

```
<function matplotlib.pyplot.show(*args, **kw)>
<Figure size 2160x1080 with 0 Axes>
```



```
plt.figure(figsize=(30,15))
sns.heatmap(df.corr(), vmin=-1, vmax=1, cmap='Set2', annot=True)
plt.show()
```

```
## Highest to the lowest correlation in order to Quality: Alcohol (0.48), Volatile Acidity (-
```

```
a = df.iloc[:, :-1].values
b = df.iloc[:, -1].values
```

```
x_train,x_test,y_train,y_test = train_test_split(a, b, test_size=0.2, random_state = 0)
```

```
lr = LinearRegression()
lr.fit(x_train, y_train)
```

```
    LinearRegression()
```

```
print(lr.predict([[15,0.01,0,5,0.001,30,50,0.95,3,0.9,15]]))
```

```
[9.82988592]
```

```
a = sm.add_constant(x_train)
project = sm.OLS(y_train,a).fit()
print(project.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.365
Model:                            OLS   Adj. R-squared:                  0.360
Method:                 Least Squares   F-statistic:                     66.34
Date:                Sat, 29 Oct 2022   Prob (F-statistic):          6.26e-117
Time:                        21:42:08   Log-Likelihood:                -1268.8
No. Observations:                1279   AIC:                             2562.
Df Residuals:                    1267   BIC:                             2624.
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         34.9987     23.831      1.469      0.142     -11.755      81.752
x1             0.0413      0.029      1.416      0.157      -0.016       0.098
x2            -1.1495      0.133     -8.631      0.000      -1.411      -0.888
x3            -0.1779      0.165     -1.077      0.282      -0.502       0.146
x4             0.0279      0.017      1.670      0.095      -0.005       0.061
x5            -1.8734      0.466     -4.024      0.000      -2.787      -0.960
x6             0.0027      0.002      1.097      0.273      -0.002       0.007
x7            -0.0028      0.001     -3.448      0.001      -0.004      -0.001
x8           -31.5167     24.325     -1.296      0.195     -79.238      16.205
x9            -0.2545      0.216     -1.179      0.239      -0.678       0.169
x10            0.9240      0.126      7.362      0.000       0.678       1.170
x11            0.2678      0.030      9.031      0.000       0.210       0.326
==============================================================================
Omnibus:                       21.104   Durbin-Watson:                   2.098
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               29.312
Skew:                          -0.182   Prob(JB):                     4.32e-07
Kurtosis:                       3.646   Cond. No.                     1.14e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specif
[2] The condition number is large, 1.14e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
y_pred = lr.predict(x_test)
print("MSE:", mean_squared_error(y_test, y_pred))
```

```
MSE: 0.38447119782012323
```