Wine Quality Analysis

1 Data

The dataset I chose is about the red variants of the Portuguese "Vinho Verde" wine. It is taken from the UCI ML Repository. [1] It consists of the following features: `fixed acidity` (acids that naturally occur in the grape and ferment the wine), `volatile acidity` (acids evaporating at low temperatures), `citric acid` (acid supplement boosting acidity in wine), `residual sugar` (amount of sugar remaining after fermantation stops defining the sweetness of wine), `chlorides` (amount of salts in the wine), `free sulfur dioxide` (free form of SO2 with antimicrobial and antioxidant properties), `total sulfur dioxide` (amount of free and bound forms of SO2, sulfites in the wine used as preservatives), `density` (amount of wine juice depending on the amount of sugar and alcohol present), `pH` (measure of acidity of the wine, the lower the pH, the more acidic the wine is), `sulphates` (amount of potassium sulphate as a wine additive acting as antimicrobial and antioxidant agent), `alcohol` (alcohol content in a given volume of wine) and `quality` (score between 0 and 10 by wine experts). Statistical analysis of this dataset can bring valuable insights to winemakers and businesses. Winemakers can adjust the production process of wine by considering the relationships between the physicochemical properties of wine and the quality measured by the experts. In the same manner, businesses can get valuable insights into understanding consumer preferences. Marketers will also get information on the characteristics they need to target and market more.

2 Data Visualization

In figure 1 you can see the scatter plot of the data depicting all the relationships between all the pairs of our variables. For instance, fixed acidity and density have a positive relationship while fixed acidity and pH have negative relationship. Based on the KDE of the variables, we can assume that `density` and `pH` are normally distributed. Most of the other variables are skewed to the right.

3 Linear Regression

The dependent variable and the main objective of the analysis will be the wine `quality` which is a score given by experts ranging from 0 to 10. Before doing linear regression, the features were normalized. The quality of the wine is the most interesting feature to analyze as it is based on people's preferences and perceptions of a good wine. Figure 2 shows the Linear Regression Model results we got by having all the other variables as exploratory ones. The R-squared for this model is 0.361 and the Adj. R-squared is 0.356 suggesting that almost

[1] The dataset can be accessed at https://archive.ics.uci.edu/dataset/186/wine+quality

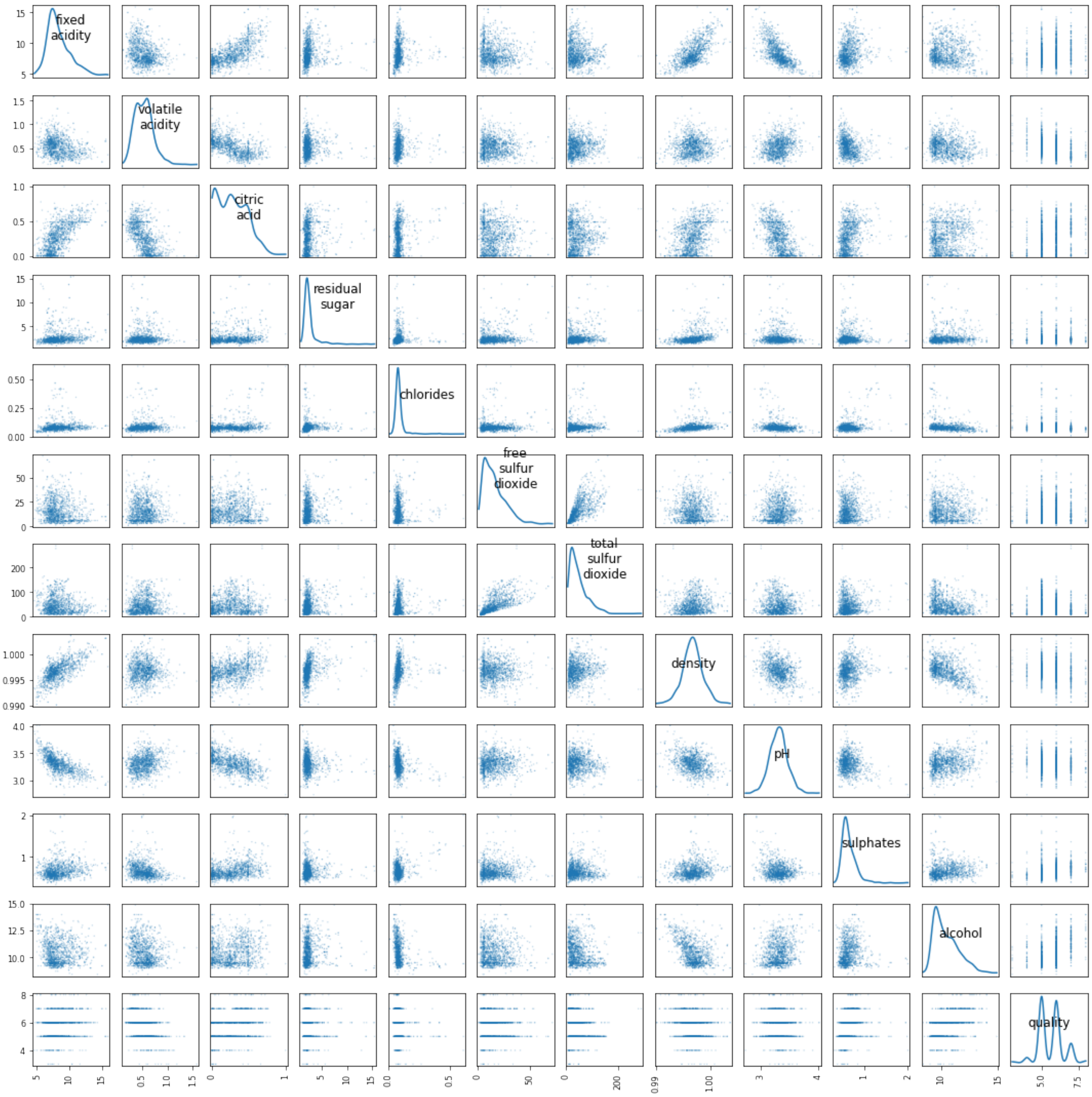
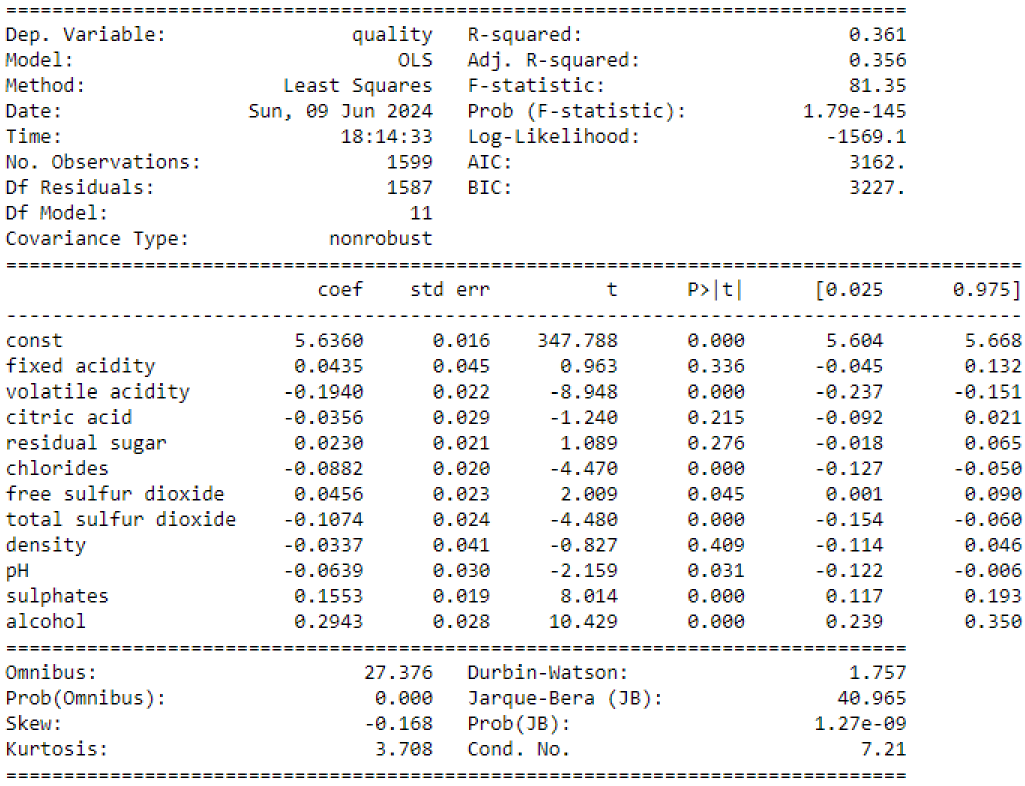


Figure 1: Scatter Plot of the data

36% of the variability of the data is explained. The non-significant features based on this model are fixed acidity, citric acid, residual sugar, and density. We also see some positive relationship between alcohol and quality, also between sulphates and quality, as well as a negative relationship between volatile acidity and quality.



```
==============================================================================
Dep. Variable:                quality   R-squared:                       0.361
Model:                            OLS   Adj. R-squared:                  0.356
Method:                 Least Squares   F-statistic:                     81.35
Date:                Sun, 09 Jun 2024   Prob (F-statistic):          1.79e-145
Time:                        18:14:33   Log-Likelihood:                -1569.1
No. Observations:                1599   AIC:                             3162.
Df Residuals:                    1587   BIC:                             3227.
Df Model:                          11
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
const                    5.6360      0.016    347.788      0.000       5.604       5.668
fixed acidity            0.0435      0.045      0.963      0.336      -0.045       0.132
volatile acidity        -0.1940      0.022     -8.948      0.000      -0.237      -0.151
citric acid             -0.0356      0.029     -1.240      0.215      -0.092       0.021
residual sugar           0.0230      0.021      1.089      0.276      -0.018       0.065
chlorides               -0.0882      0.020     -4.470      0.000      -0.127      -0.050
free sulfur dioxide      0.0456      0.023      2.009      0.045       0.001       0.090
total sulfur dioxide    -0.1074      0.024     -4.480      0.000      -0.154      -0.060
density                 -0.0337      0.041     -0.827      0.409      -0.114       0.046
pH                      -0.0639      0.030     -2.159      0.031      -0.122      -0.006
sulphates                0.1553      0.019      8.014      0.000       0.117       0.193
alcohol                  0.2943      0.028     10.429      0.000       0.239       0.350
==============================================================================
Omnibus:                       27.376   Durbin-Watson:                   1.757
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               40.965
Skew:                          -0.168   Prob(JB):                     1.27e-09
Kurtosis:                       3.708   Cond. No.                         7.21
==============================================================================
```

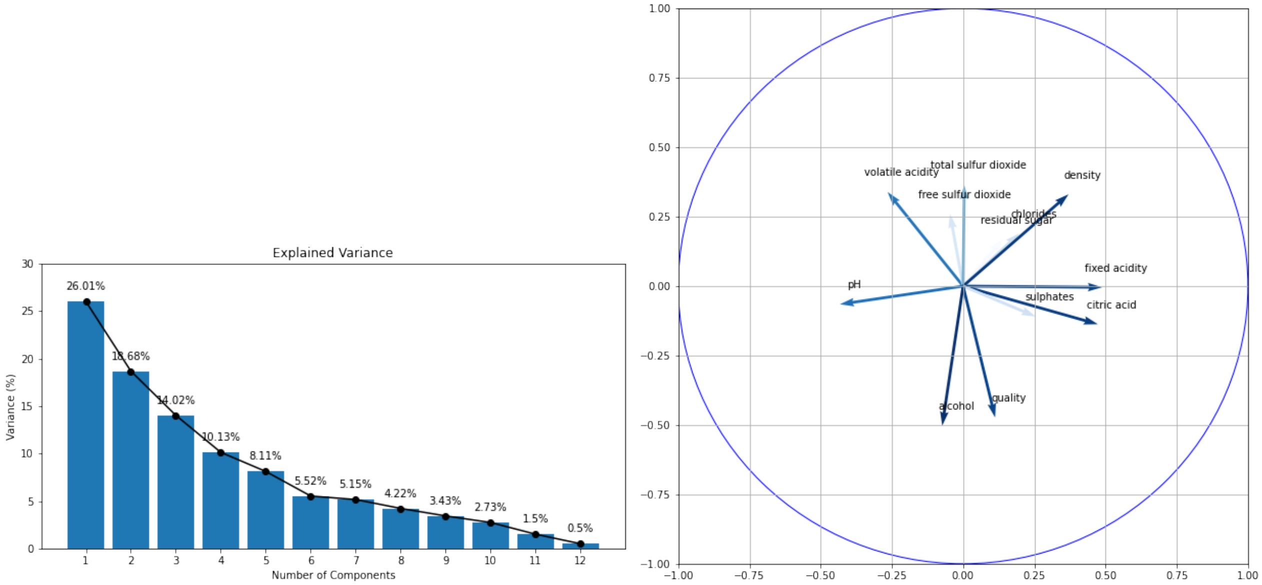
Figure 2: OLS Regression Results

4 Variable Selection

To select the variables linear regression was done on the data iteratively and at each step an insignificant variable with the highest p-value was dropped, the iterations were stopped when there was no insignificant variable left (the significance level was fixed to be 5%). As a result, the features density, fixed acidity, residual sugar, and citric acid were dropped which didn't seem to have any impact on quality based on our null model either. The R-squared and Adj. R-squared of this model are 0.359 and 0.357 respectively which proves that this model is better in terms of simplicity and no significant change in R-squared.

5 PCA

Now let us do PCA on the whole data and see how the principal components can be used to understand the underlying patterns in the dataset. As shown in figure 3a the first four components explain 26.01%, 18.68%, 14.02%, and 10.13% of the variability of our data, respectively. The respective eigenvalues for all the components are 3.1211677, 2.24188204, 1.68291969, 1.21502087, 0.97326362, 0.66259224, 0.6183178, 0.50587256, 0.41130754, 0.32791939, 0.18021863, 0.05951792. We can recommend keeping the first four components since it is common practice to consider components with eigenvalues greater than 1. Looking at the Scree plot we cannot tell that this is the most favorable case as we do not have a very significantly decreasing pattern for the first several components, but the fact that the first four components explain around 70% of the variability of the data is not bad.



(a) PCA Scree Plot

(b) PCA Biplot

Figure 3: Scree Plot and Biplot

Figure 3b depicts the PCA biplot which contains information on how strongly each variable influences the principal components (in this case the first two principal components). First, let us notice that the arrows in the biplot are not very long meaning there are no vivid contributors on the first two components. The most impactful variables contributing to the first two components are alcohol, density, fixed acidity, citric acid, and quality. The variables fixed acidity, citric acid, pH strongly influence the first component indicating that these features have more variability captured by the first component. On the other hand, alcohol and quality have a strong contribution to the second principal component. We can also get nice insights into how the features are correlated. Alcohol and quality are positively correlated with each other, implying that higher alcohol content tends to be associated with higher quality ratings in wine. Alcohol and total sulfur dioxide have a strong negative correlation, implying that as alcohol content increases, the total sulfur dioxide levels tend to decrease. pH and fixed acidity have again a strong negative correlation which is expected as acidity and pH are inversely related. Fixed acidity and citric acid are positively correlated. The plot suggests that there is almost no correlation between quality and pH, as well as between quality and fixed acidity, indicating that variations in pH and fixed acidity do not significantly affect the perceived quality of the wine. Quality has strong negative correlations with total sulfur dioxide, free sulfur dioxide, and volatile acidity. This implies that higher levels of these components are associated with lower quality ratings, reflecting their potential negative impact on the sensory attributes of wine. However, since the arrows for these dioxides are shorter (meaning they do not contribute strongly to the first component), this can be a misinterpretation.

6 Projection of the data on the two components

Figure 5 depicts the projection of the data onto the first two principal components. The data is mostly centered around the mean. The variance of the points across the two components is almost the same, meaning they explain the variance of the data to almost the same extent. The plot suggests that the variance of the data is not explained that much although we have considered the first two components. The isolated points far from the mean may possibly be outliers.

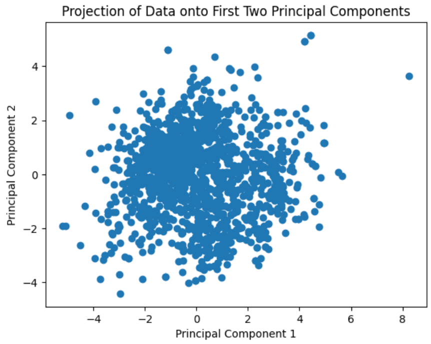


Figure 4: Projection of the data onto first two components

7 LASSO

I have also applied LASSO Regression on our data. And the features that LASSO Regression selected were volatile acidity, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, pH, sulphates, alcohol. Surprisingly, for example, residual sugar was not selected in the Linear Regression Model, while it is selected here.

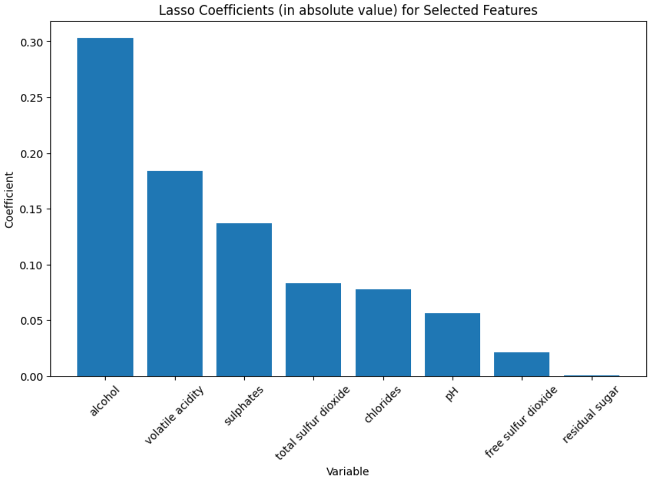


Figure 5: LASSO Coefficients (in absolute value) for Selected Features

A Code

Please find my codes at https://github.com/lusdavtyan/Wine-Quality-Analysis.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import textwrap
import statsmodels.api as sm
from matplotlib.colors import Normalize
```

In [2]:

```
original = pd.read_csv('winequality-red.csv', sep=';')
```

In [3]:

```
df = original
df
```

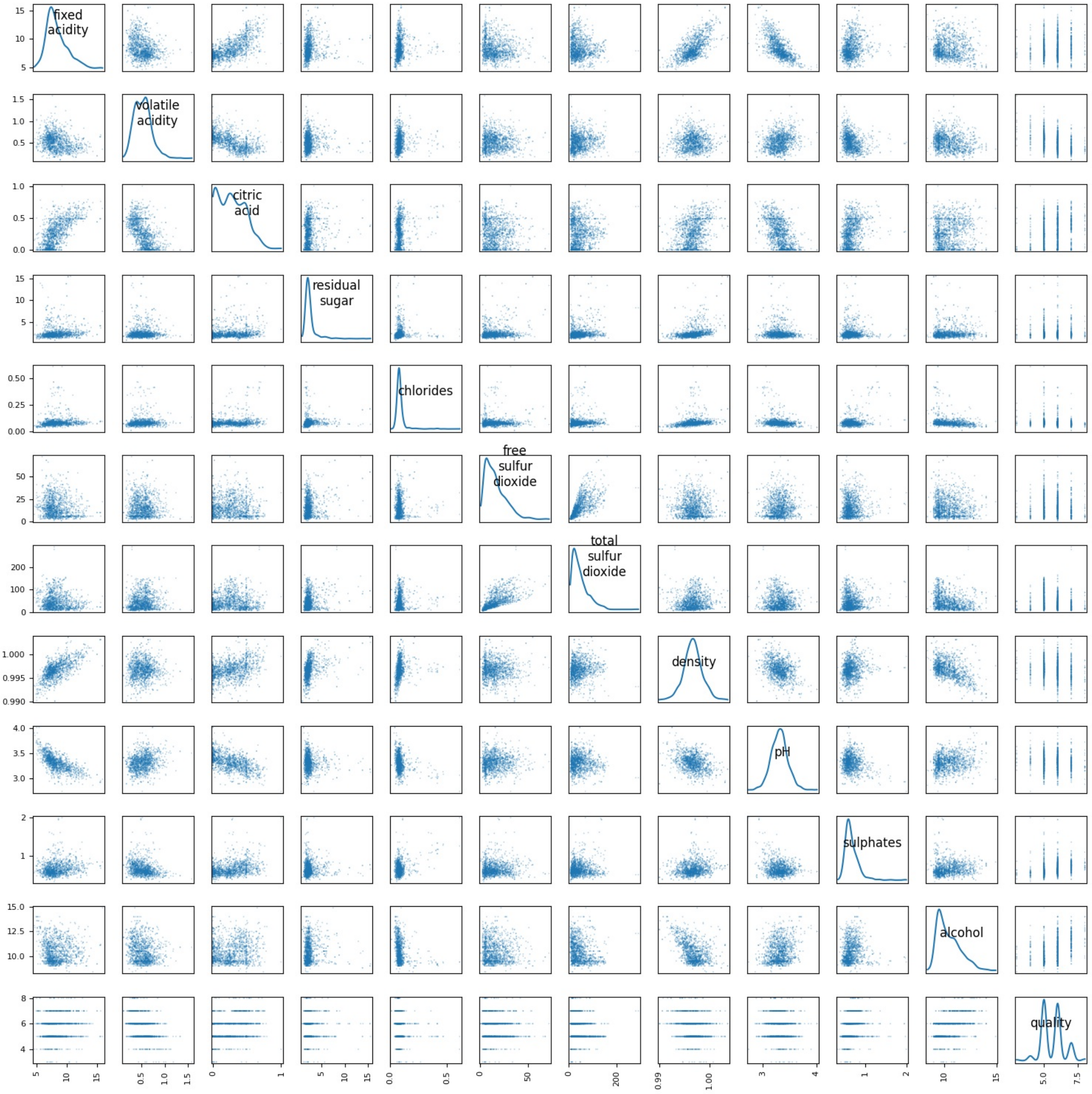
Out[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

In [4]:

```
axes = pd.plotting.scatter_matrix(df, alpha=0.3, s=7, figsize=(15, 15), diagonal='kde')
plot_labels = [label.replace(' ', '\n') for label in df.columns]

for i, ax in enumerate(axes.flatten()):
    ax.yaxis.label.set_ha('right')
    if i % (len(df.columns) + 1) == 0:
        ax.text(0.5, 0.5, plot_labels[i // len(df.columns)], horizontalalignment='center', verticalalignment='bottom', transform=ax.transAxes, fontsize=12)
    ax.set_xlabel('')
    ax.set_ylabel('')

plt.tight_layout()
plt.show()
```



In [5]:

```
X = df.drop('quality', axis=1)
y = df['quality']
# normalize features
X = (X - X.mean()) / X.std()
```

In [6]:

```
X = sm.add_constant(X)
full_model = sm.OLS(y, X).fit()
print(full_model.summary())
print()
def stepwise_regression(X, y, significance_level=0.05):
    X = sm.add_constant(X)

    model = sm.OLS(y, X).fit()

    while True:
        pvalues = model.pvalues.drop(index=['const'])

        if all(pvalues < significance_level):
            break

        max_pvalue = pvalues.max()
        if max_pvalue < significance_level:
            break

        max_pvalue_variable = pvalues.idxmax()
        print(f"Dropping variable '{max_pvalue_variable}' with p-value {max_pvalue}")
        X = X.drop(columns=[max_pvalue_variable])

        model = sm.OLS(y, X).fit()

    return model

final_model = stepwise_regression(X, y)
print()
print("Selected Model")
print(final_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                quality   R-squared:                       0.361
Model:                            OLS   Adj. R-squared:                  0.356
Method:                 Least Squares   F-statistic:                     81.35
Date:                Mon, 10 Jun 2024   Prob (F-statistic):          1.79e-145
Time:                        00:39:35   Log-Likelihood:                -1569.1
No. Observations:                1599   AIC:                             3162.
Df Residuals:                    1587   BIC:                             3227.
Df Model:                          11
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
const                    5.6360      0.016    347.788      0.000       5.604       5.668
fixed acidity            0.0435      0.045      0.963      0.336      -0.045       0.132
volatile acidity        -0.1940      0.022     -8.948      0.000      -0.237      -0.151
citric acid             -0.0356      0.029     -1.240      0.215      -0.092       0.021
residual sugar           0.0230      0.021      1.089      0.276      -0.018       0.065
chlorides               -0.0882      0.020     -4.470      0.000      -0.127      -0.050
free sulfur dioxide      0.0456      0.023      2.009      0.045       0.001       0.090
total sulfur dioxide    -0.1074      0.024     -4.480      0.000      -0.154      -0.060
density                 -0.0337      0.041     -0.827      0.409      -0.114       0.046
pH                      -0.0639      0.030     -2.159      0.031      -0.122      -0.006
sulphates                0.1553      0.019      8.014      0.000       0.117       0.193
alcohol                  0.2943      0.028     10.429      0.000       0.239       0.350
==============================================================================
Omnibus:                       27.376   Durbin-Watson:                   1.757
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               40.965
Skew:                          -0.168   Prob(JB):                     1.27e-09
Kurtosis:                       3.708   Cond. No.                         7.21
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Dropping variable 'density' with p-value 0.40860789719340285
Dropping variable 'fixed acidity' with p-value 0.6121192274702094
Dropping variable 'residual sugar' with p-value 0.4342760865135191
Dropping variable 'citric acid' with p-value 0.2875672160577794

Selected Model
                            OLS Regression Results
==============================================================================
Dep. Variable:                quality   R-squared:                       0.359
Model:                            OLS   Adj. R-squared:                  0.357
Method:                 Least Squares   F-statistic:                     127.6
Date:                Mon, 10 Jun 2024   Prob (F-statistic):          5.32e-149
Time:                        00:39:35   Log-Likelihood:                -1570.5
No. Observations:                1599   AIC:                             3157.
Df Residuals:                    1591   BIC:                             3200.
Df Model:                           7
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
const                    5.6360      0.016    347.932      0.000       5.604       5.668
volatile acidity        -0.1813      0.018    -10.043      0.000      -0.217      -0.146
chlorides               -0.0950      0.019     -5.076      0.000      -0.132      -0.058
free sulfur dioxide      0.0531      0.022      2.389      0.017       0.010       0.097
total sulfur dioxide    -0.1145      0.023     -5.070      0.000      -0.159      -0.070
pH                      -0.0745      0.018     -4.106      0.000      -0.110      -0.039
sulphates                0.1496      0.019      8.031      0.000       0.113       0.186
alcohol                  0.3083      0.018     17.225      0.000       0.273       0.343
==============================================================================
Omnibus:                       24.204   Durbin-Watson:                   1.750
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               35.245
Skew:                          -0.156   Prob(JB):                     2.22e-08
Kurtosis:                       3.657   Cond. No.                         2.44
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [7]:

```
y = (y - y.mean()) / y.std()
```

In [8]:

```
X.drop('const', axis=1, inplace=True)
df = pd.concat([X, y], axis=1)
```

In [9]:

```
# PCA
pca = PCA()
principalComponents = pca.fit_transform(df)
```

In [10]:

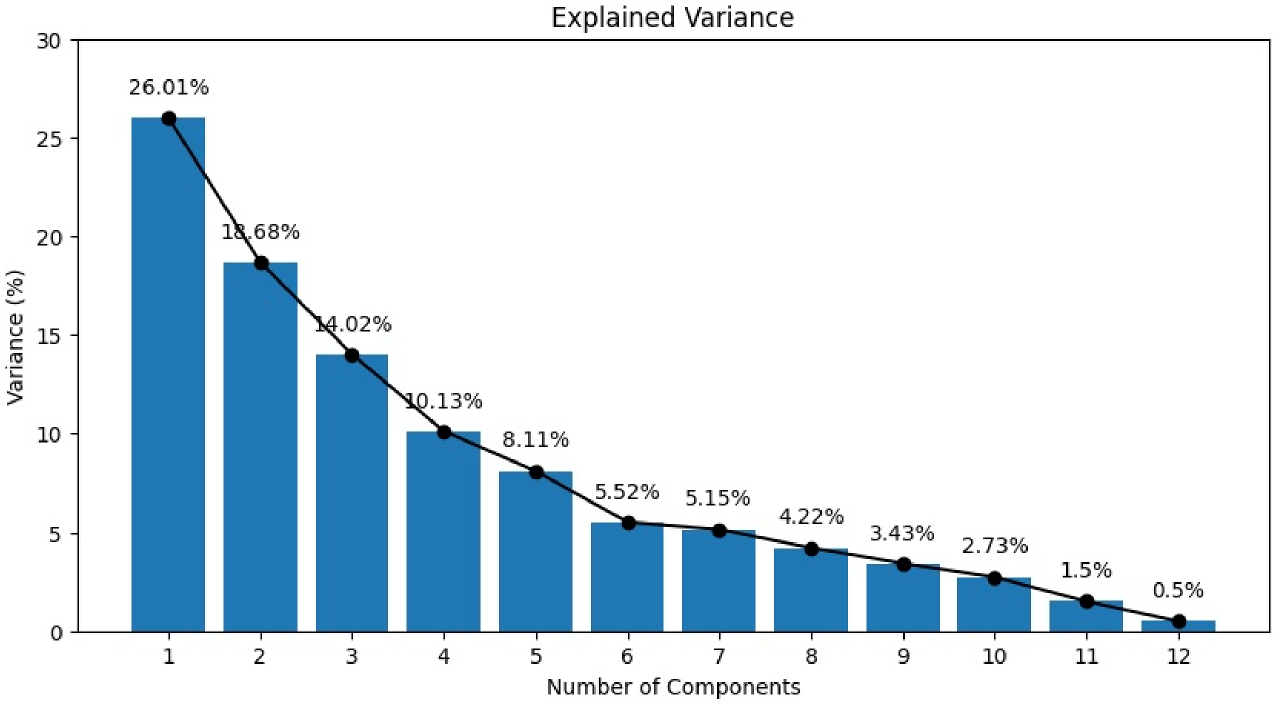
```
# eigenvalues
eigenvalues = pca.explained_variance_
eigenvalues
```

Out[10]:

```
array([3.1211677 , 2.24188204, 1.68291969, 1.21502087, 0.97326362,
       0.66259224, 0.6183178 , 0.50587256, 0.41130754, 0.32791939,
       0.18021863, 0.05951792])
```

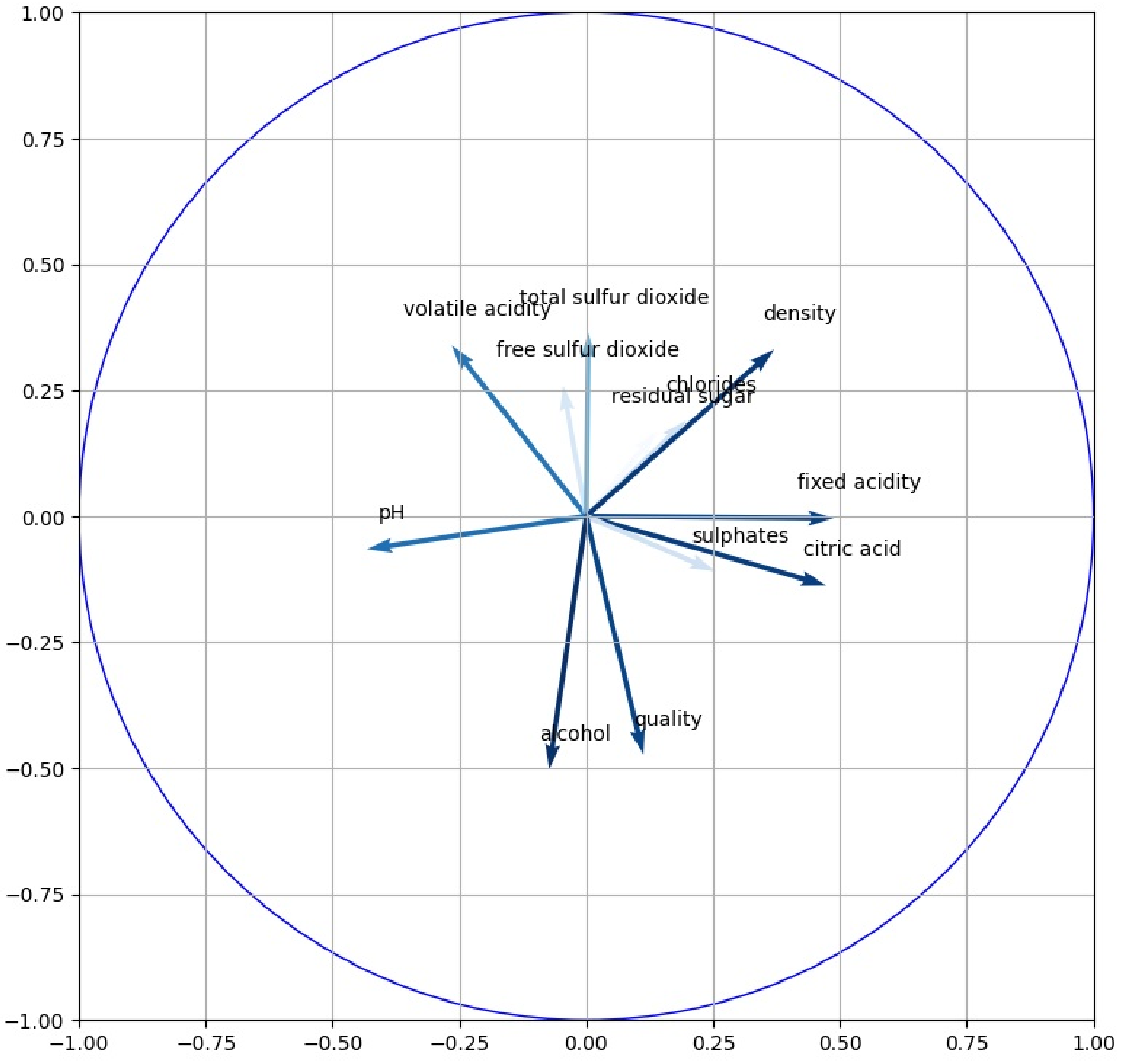
In [11]:

```
explained_var_ratio = pca.explained_variance_ratio_ * 100

plt.figure(figsize=(10,5))
bars = plt.bar(range(1,len(explained_var_ratio)+1), explained_var_ratio)

for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval + 1,
             f'{round(yval,2)}%', ha='center', va='bottom')

plt.plot(range(1,len(explained_var_ratio)+1), explained_var_ratio, color='black', marker='o')

plt.xlabel('Number of Components')
plt.ylabel('Variance (%)')
plt.title('Explained Variance')
plt.ylim(0,30)
plt.xticks(range(1,len(explained_var_ratio)+1))
plt.show()
```



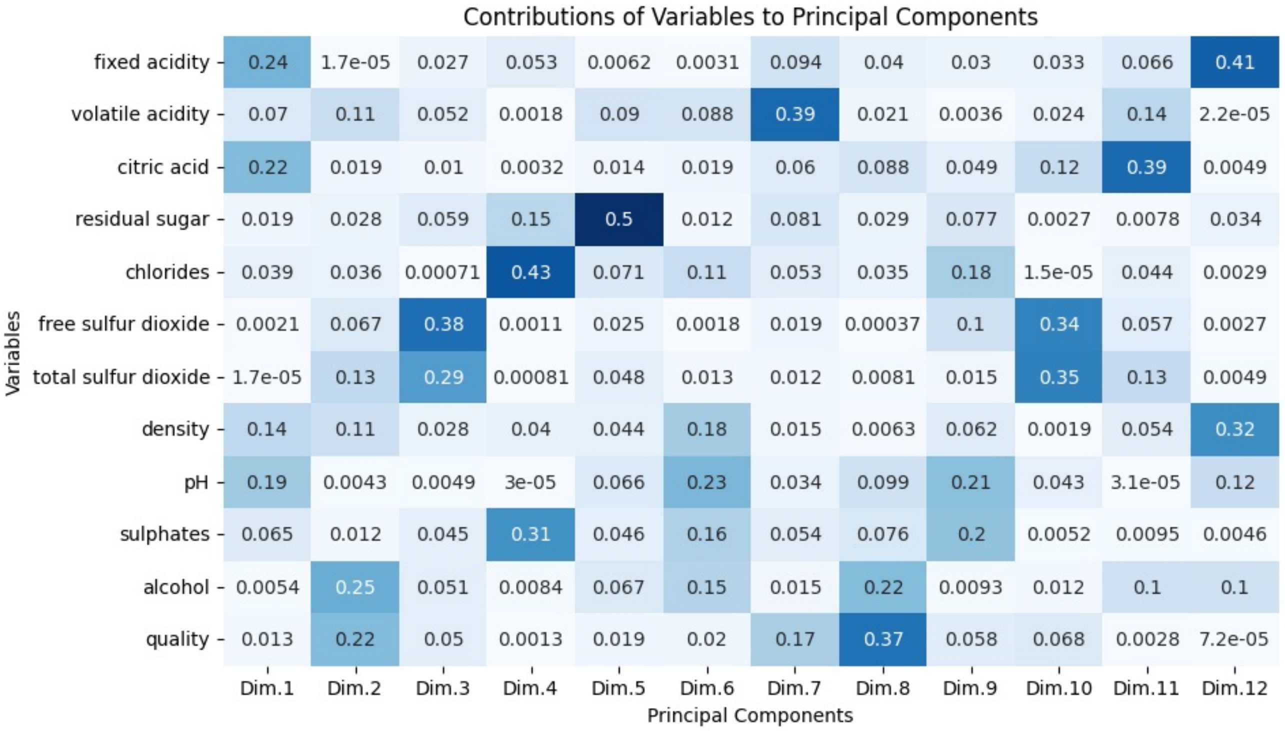
In [12]:

```
arrow_width = 0.005
cmap_name = 'Blues'

distances = np.sqrt(pca.components_[0, :]**2 + pca.components_[1, :]**2)
norm = Normalize(vmin=min(distances), vmax=max(distances))
colors = plt.get_cmap(cmap_name)(norm(distances))

plt.figure(figsize=(9, 9))
plt.xlim(-1, 1)
plt.ylim(-1, 1)

plt.quiver(
    np.zeros(pca.components_.shape[1]),
    np.zeros(pca.components_.shape[1]),
    pca.components_[0, :],
    pca.components_[1, :],
    angles='xy',
    scale_units='xy',
    scale=1,
    color=colors,
    width=arrow_width
)

for i, j, z in zip(pca.components_[0, :] + 0.05, pca.components_[1, :] + 0.05, df.columns):
    plt.text(i, j, z, ha='center', va='bottom')

circle = plt.Circle((0, 0), 1, facecolor='none', edgecolor='b')
plt.gca().add_artist(circle)

plt.grid()
plt.show()
```



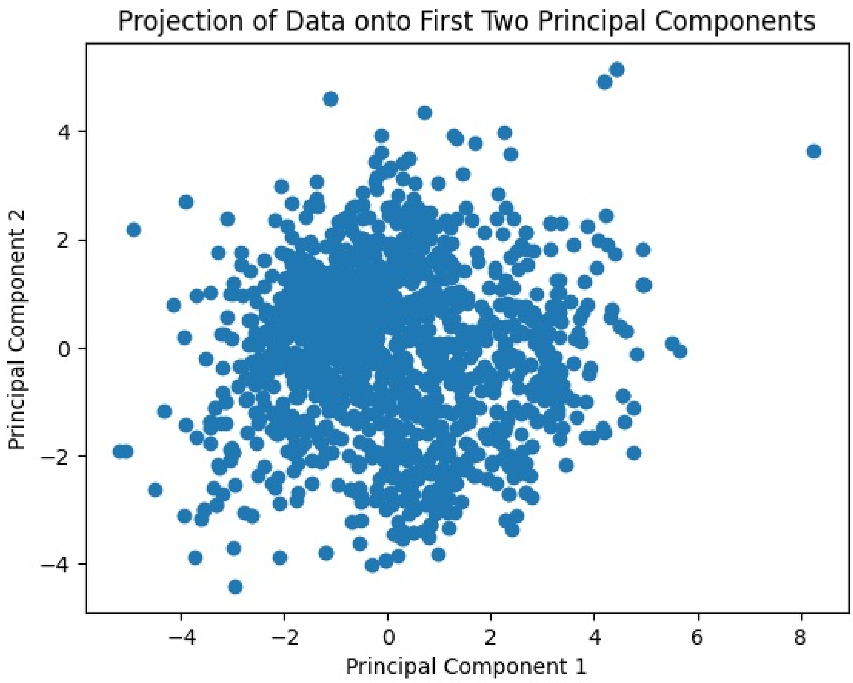
In [13]:

```
loadings = pca.components_.T ** 2
contributions = loadings / loadings.sum(axis=0)

contrib_df = pd.DataFrame(contributions, columns=[f'Dim.{i+1}' for i in range(loadings.shape[1])], index=df.colum
ns)

plt.figure(figsize=(10, 6))
sns.heatmap(contrib_df, annot=True, cmap='Blues', cbar=False)
plt.title('Contributions of Variables to Principal Components')
plt.ylabel('Variables')
plt.xlabel('Principal Components')
plt.xticks(rotation=0)
plt.yticks(rotation=0)
plt.show()
```



In [14]:

```
transformed_data = pca.transform(df)

pc1 = transformed_data[:, 0]
pc2 = transformed_data[:, 1]

plt.scatter(pc1, pc2)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Projection of Data onto First Two Principal Components')
plt.show()
```



In [15]:

```
df = original
df
```

Out[15]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	3.45	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	3.52	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	3.42	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

1599 rows × 12 columns

In [16]:

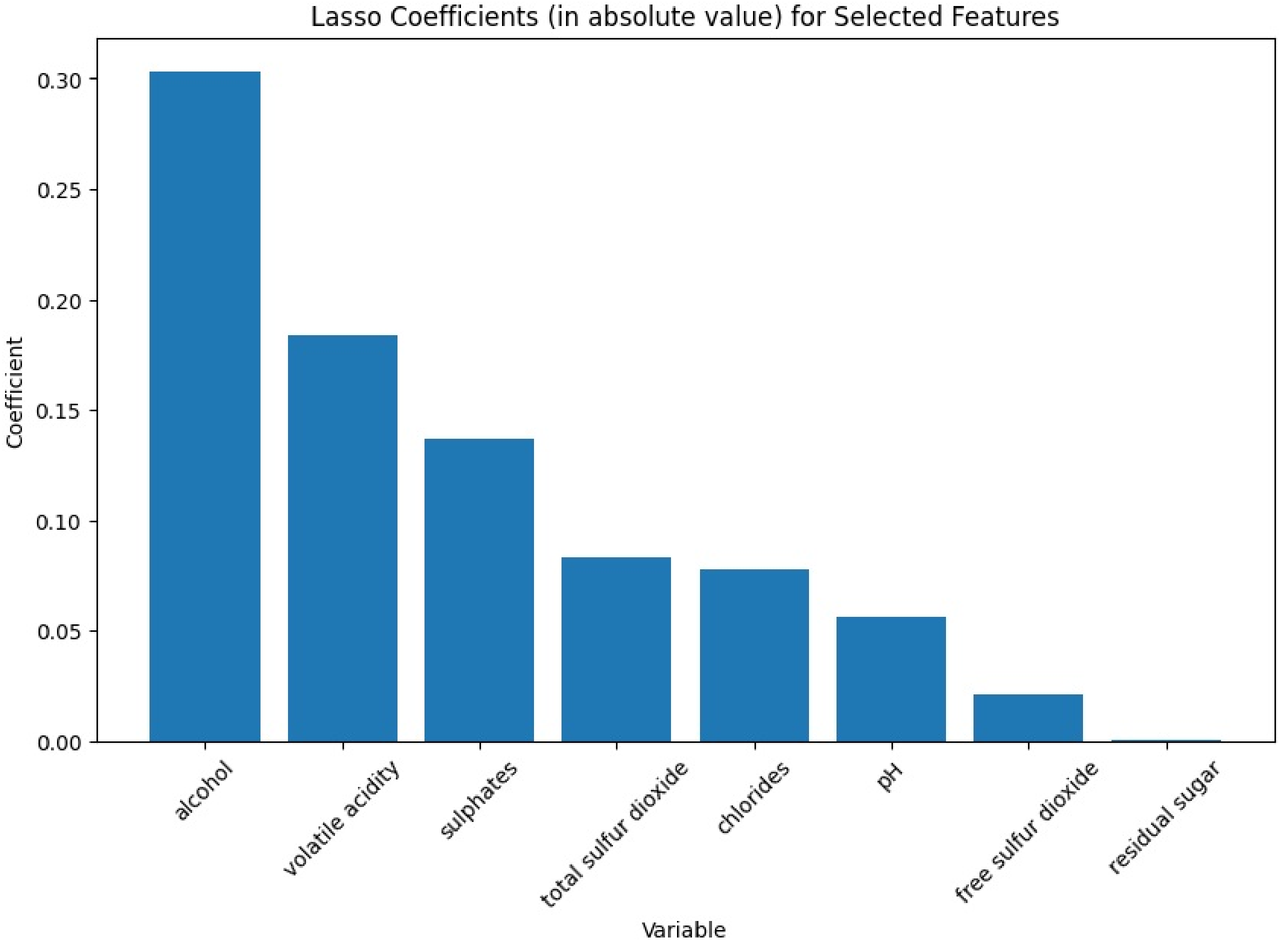
```
from sklearn.linear_model import Lasso

X = df.drop('quality', axis=1)
y = df['quality']
# normalize features
X = (X - X.mean()) / X.std()
lasso = Lasso(alpha=0.01)
lasso.fit(X, y)

selected_variables = X.columns[lasso.coef_ != 0]
selected_variables
```

Out[16]:

```
Index(['volatile acidity', 'residual sugar', 'chlorides',
       'free sulfur dioxide', 'total sulfur dioxide', 'pH', 'sulphates',
       'alcohol'],
      dtype='object')
```

In [18]:

```
coefficients = np.abs(lasso.coef_[lasso.coef_ != 0])

pairs = list(zip(selected_variables, coefficients))

pairs.sort(key=lambda x: x[1], reverse=True)

sorted_variables, sorted_coefficients = zip(*pairs)

plt.figure(figsize=(10, 6))
plt.bar(sorted_variables, sorted_coefficients)
plt.xlabel('Variable')
plt.ylabel('Coefficient')
plt.title('Lasso Coefficients (in absolute value) for Selected Features')
plt.xticks(rotation=45)
plt.show()
```


Lasso Coefficients (in absolute value) for Selected Features
Coefficient
alcohol
volatile acidity
sulphates
total sulfur dioxide
chlorides
pH
free sulfur dioxide
residual sugar
Variable