final

June 8, 2020

1 Final Project for PSYC010 - Experimental Design, Methodology, and Data Analysis Procedures

Author: Matthew Kenney

1.1 Introduction:

As a computer science student, I have naturally picked up an interest in machine learning and data analytics, and have become fairly familiar in working with data science packages in python. In learning data science, however, I noticed that most of the learning resources which I had access to (courses at Dartmouth, online guides on data science for programmers, etc.) focused heavily on develivering a working end-product, rather than covering statistical theory in depth and strictly adhering to guidelines set by the statistical community.

Generally, focusing on delivering an end product at the expense of some assumption-checking and due-dilligence tends to be much safer in engineering than in research, because the objective of engineering is to create working tools wheras the objective of research is to uncover the nature of the objects under study. Nevertheless, I recognize that the best data scientists and machine learning engineers are likely the ones who have a deep understanding of both programming tools and the mathematics and statistics underlying those tools.

Taking this course has helped bridge many of the gaps in my understanding that I had as a result of learning about machine learning and data science without a strong foundation in statistics. As such, I thought it would be appropriate to complete a final project in which I used my programming background to conduct an analysis with an emphasis on assumption checking and statistical soundness.

1.1.1 The Analysis

Although we did not explicitly cover multiple regression in this class, I decided to conduct a multiple regression in this code and utilize many of the assumption checks and analyses that we *did* learn in class to augment my analysis. To conduct this analysis, I pulled a dataset which attempts to trace the link between measurable chemical properties in wine and percieved wine quality. Although a rather obscure dataset, I thought that this dataset satisfied by purposes perfectly, because it is composed of only ratio and interval measures in all independent variables and in the dependent variable (wine quality, rated on a scale from 1-10). While working with categorical variables is straightforward and well-documented in multiple-linear regression, it makes assumption checking

and analysis of regression a bit more complex, and so I sought to avoid categorical variables in this case.

Information on the dataset I choose to use can be found in the original paper that generated the data, and the data can be downloaded from the UCI machine learning repository at this link. To give a bit of an intution for the kind of information this dataset contains, however, I have listed all variables reccorded in this datset below.

Indpendent Variables:

Note: units of measure have been removed in this dataset, but are documented in the referenced paper that generated this data - fixed acidity - volatile acidity - citric acid - residual sugar - chlorides - free sulfur dioxide - total sulfur dioxide - density - pH - sulphates - alcohol

Dependent Variables: - quality (score between 0 and 10)

Note: quality was determined by a panel of oenologists at a wine judging event

1.1.2 Reporting Style:

The description of this project (Final Project Option 1) stated that we should follow the format of Writing Assignment 5 to report our results. I am taking some liberties here and reporting the results with rough adherance to the Writing Assignment 5 format and APA guidelines, and reporting these results throughout this Jupyter notebook rather than in a seperate document with text and figures/ tables seperated out. Although this format is, of course, unsuitable for publication, it was my opinion that keeping the code, analysis, and figures all in one cohesive document would make for an easier read and a more coherent representation of my final project.

My report is now presented below:

Report: The objective of this analysis is to c"onstruct a multiple regression equation to establish a numerical relationship between 11 chemical properties in wine (namely: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol) and the perceived quality of the wine. Quality scores (rated 0-10) were set by a panel of oenologists at a wine tasting event. In total, the dataset under study includes information on 4898 white wines. Prior to conducting the multiple regression analysis, descriptive statistics associated with the observed data and a variety of descriptive figures were generated. Example entries from the wine datset are viewable in **Table 1**, and descriptive statistics on each of the variables included in the dataset are reported in **Table 2**. A histogram (**Figure 1**), reporting the frequencies of various wine scores showed that wine scores followed a roughly normal distribution, and that wine scores ranged between a minimum score of 3 and a maximum score of 9, even though judges were allowed to assign scores between 0 and 10.

[1]:
```
# Analysis and Descriptives
import numpy as np
import pandas as pd
from scipy.stats import describe, norm
import statsmodels.api as sm
```

```
from statsmodels.graphics.gofplots import qqplot, ProbPlot

# Graphing
from simple_colors import *
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
wine = pd.read_csv('./winequality-white.csv', delimiter=';')
```

[3]:
```
print(black('Table 1: Wine Dataset Example Entries', ['bold']))
wine.head()
```

Table 1: Wine Dataset Example Entries

[3]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

[4]:
```
print(black('Table 2: Descriptive Statistics for Wine Dataset', ['bold']))
wine.describe().T
```

Table 2: Descriptive Statistics for Wine Dataset

[4]:

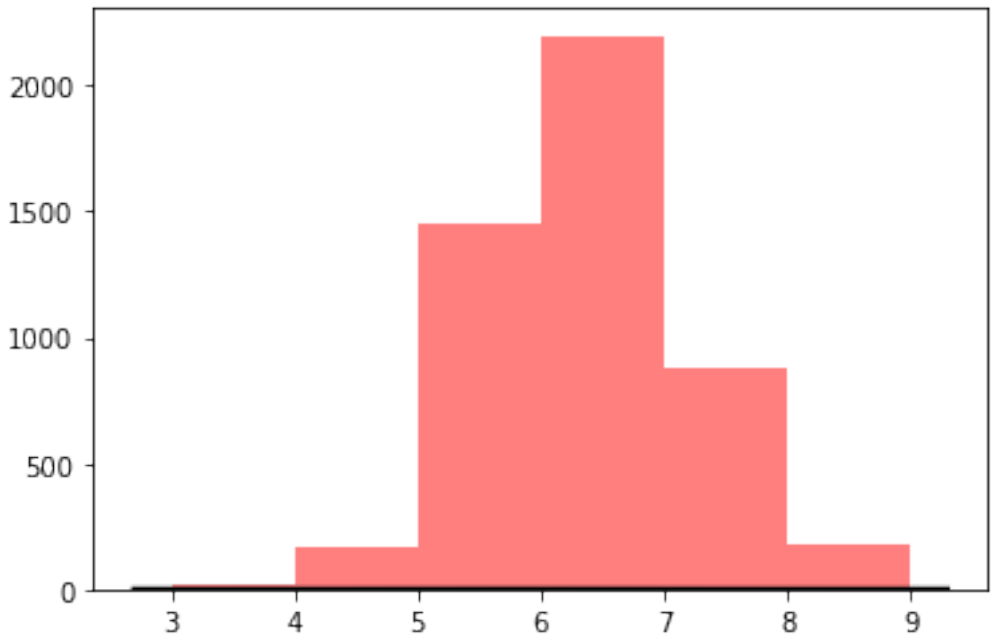
	count	mean	std	min	25%
fixed acidity	4898.0	6.854788	0.843868	3.80000	6.300000
volatile acidity	4898.0	0.278241	0.100795	0.08000	0.210000
citric acid	4898.0	0.334192	0.121020	0.00000	0.270000
residual sugar	4898.0	6.391415	5.072058	0.60000	1.700000
chlorides	4898.0	0.045772	0.021848	0.00900	0.036000
free sulfur dioxide	4898.0	35.308085	17.007137	2.00000	23.000000

total sulfur dioxide	4898.0	138.360657	42.498065	9.00000	108.000000
density	4898.0	0.994027	0.002991	0.98711	0.991723
pH	4898.0	3.188267	0.151001	2.72000	3.090000
sulphates	4898.0	0.489847	0.114126	0.22000	0.410000
alcohol	4898.0	10.514267	1.230621	8.00000	9.500000
quality	4898.0	5.877909	0.885639	3.00000	5.000000

	50%	75%	max
fixed acidity	6.80000	7.3000	14.20000
volatile acidity	0.26000	0.3200	1.10000
citric acid	0.32000	0.3900	1.66000
residual sugar	5.20000	9.9000	65.80000
chlorides	0.04300	0.0500	0.34600
free sulfur dioxide	34.00000	46.0000	289.00000
total sulfur dioxide	134.00000	167.0000	440.00000
density	0.99374	0.9961	1.03898
pH	3.18000	3.2800	3.82000
sulphates	0.47000	0.5500	1.08000
alcohol	10.40000	11.4000	14.20000
quality	6.00000	6.0000	9.00000

[5]:
```
print(black('Figure 1: Histogram of Wine Quality Scores', ['bold']))
# Distribution of wine scores
# Note: all scores fall within 3-9 range
plt.hist(wine['quality'], bins=[3,4,5,6,7,8,9], alpha=0.5, color='r')
mu, std = norm.fit(wine['quality'])
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2);
```

Figure 1: Histogram of Wine Quality Scores



1.2 Visualizing Assumptions

Multiple regression requires a number of assumptions to be met, including the following: - **Outliers**: outliers have been removed from the data as these can substantially affect model fit - **Linearity**: the relationship we are trying to model is linear, and with several predictors, they combine additively - **Collinearity**: The data does not express high intercorrelations or interassociations among the independent variables (otherwise, some of these variables are not truly independent and cannot be reliably used as predictors) - **Normality**: both the independent and dependent variables must roughly correspond to normal distributions - **Homoscedasticity**: variance is spread evenly among the entire range of the observed data. This assumption must hold for all predictor variables and the dependent variable

In order to assess whether or not these assumptions are met in the initial state of the wine dataset, a variety of figures were generated. These figures help to evaluate some of the above assumptions in a visual and intuitive manner. To test for outliers, a box-plot for each of the variables in this dataset was generated (see **Figure 2**). Box plots demonstrated that there were potential outlier variables in the data, with the 'fixed acitity', 'total sulfer dioxide', and the 'free sulfer dioxide' variables all showing a large number of values falling well beyond the interquartile range.

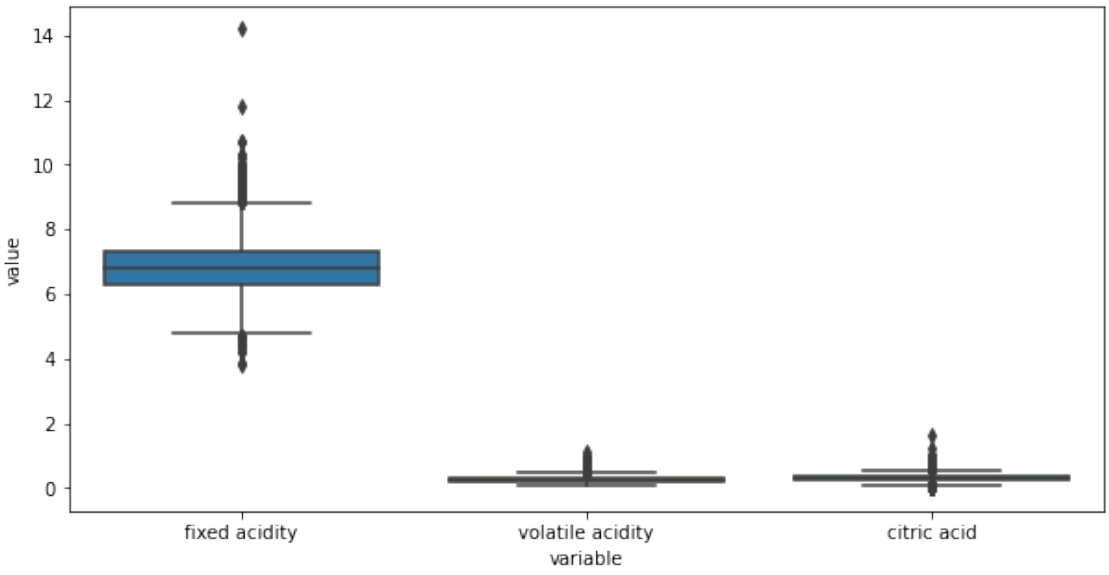
We also generated histograms for each of the variables in the dataset to assess normality. These histograms, shown in **Figure 3**, indicate that all of the variables in our dataset are rougly normal, although some of these variables, such as free sulfer dioxide, residual sugar, and chlorides have a definite skew. In addition to the histograms, we generated Q-Q plots to test for the normality of these variables (See **Figure 4**). Although some of the observed independent variables do not follow the normal curve perfectly, we chose not to reject any variables based on normality concerns because, ultimately, the P-P plot of standardised residuals came back nicely, demonstrating that

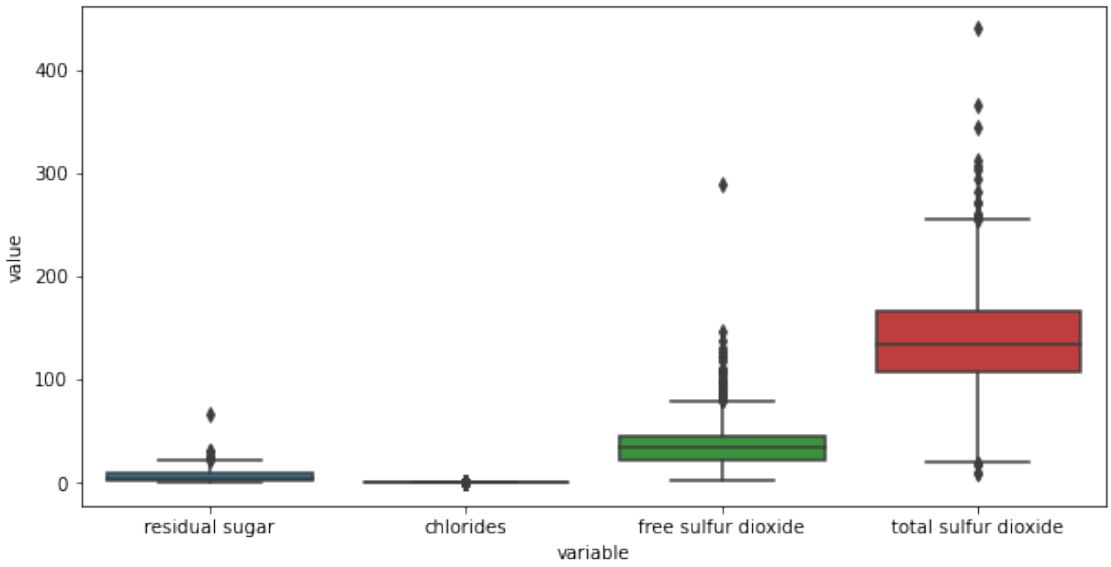
the data contained approximately normally distributed errors, even if some of the variables were slightly non-normal.

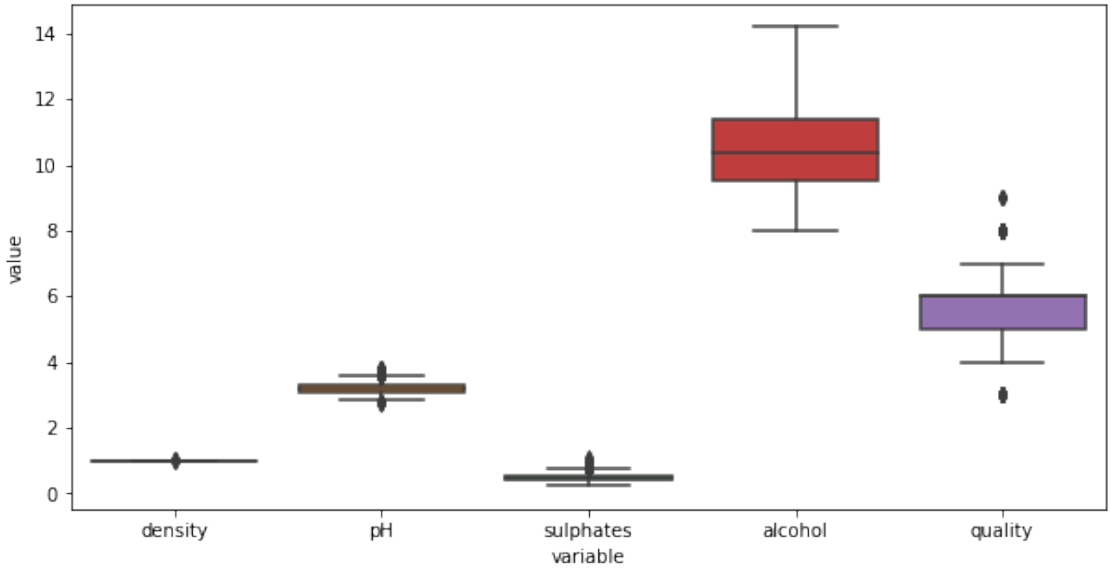
Finally, we assessed the colinearity of our variables using a correlation graph showing the coefficient of determination values associated with each variable pairing. This correlation graph (**Figure 5**) is designed to take a close look at each pairwise relationship between variables in our dataset. For the most part, the correlation graph confirmed the data does not show strong colinearities between variables. However, it is clear that there is a very strong pairwise relationship between residual sugar and density($r^2 = 0.70$), density and alcohol ($r^2 = 0.61$) and a sizable relationship between both free sulfer dioxide and total sulfer dioxide ($r^2 = 0.38$) and between total sulfer dioxide and density ($r^2 = 0.28$). The strong correlations between these variables do not come as a suprise, since they are clearly related chemical properties. Nevertheless, colinearities are problematic, leading to erroneous p-values and coefficients in the line of best-fit. *Because these variable pairs demonstrate colinearities, members of these paris are candidates for removal from our dataset.*

[6]:
```
# Boxplots:
print(black('Figure 2: Boxplots for All Variables', ['bold']))
fig, ax = plt.subplots(figsize=(10,5))
sns.boxplot(ax=ax, x="variable", y="value", data=pd.melt(wine.iloc[:, 0:3]))
plt.show()
fig, ax = plt.subplots(figsize=(10,5))
sns.boxplot(ax=ax, x="variable", y="value", data=pd.melt(wine.iloc[:, 3:7]))
plt.show()
fig, ax = plt.subplots(figsize=(10,5))
sns.boxplot(ax=ax, x="variable", y="value", data=pd.melt(wine.iloc[:, 7:]))
plt.show()
```

Figure 2: Boxplots for All Variables

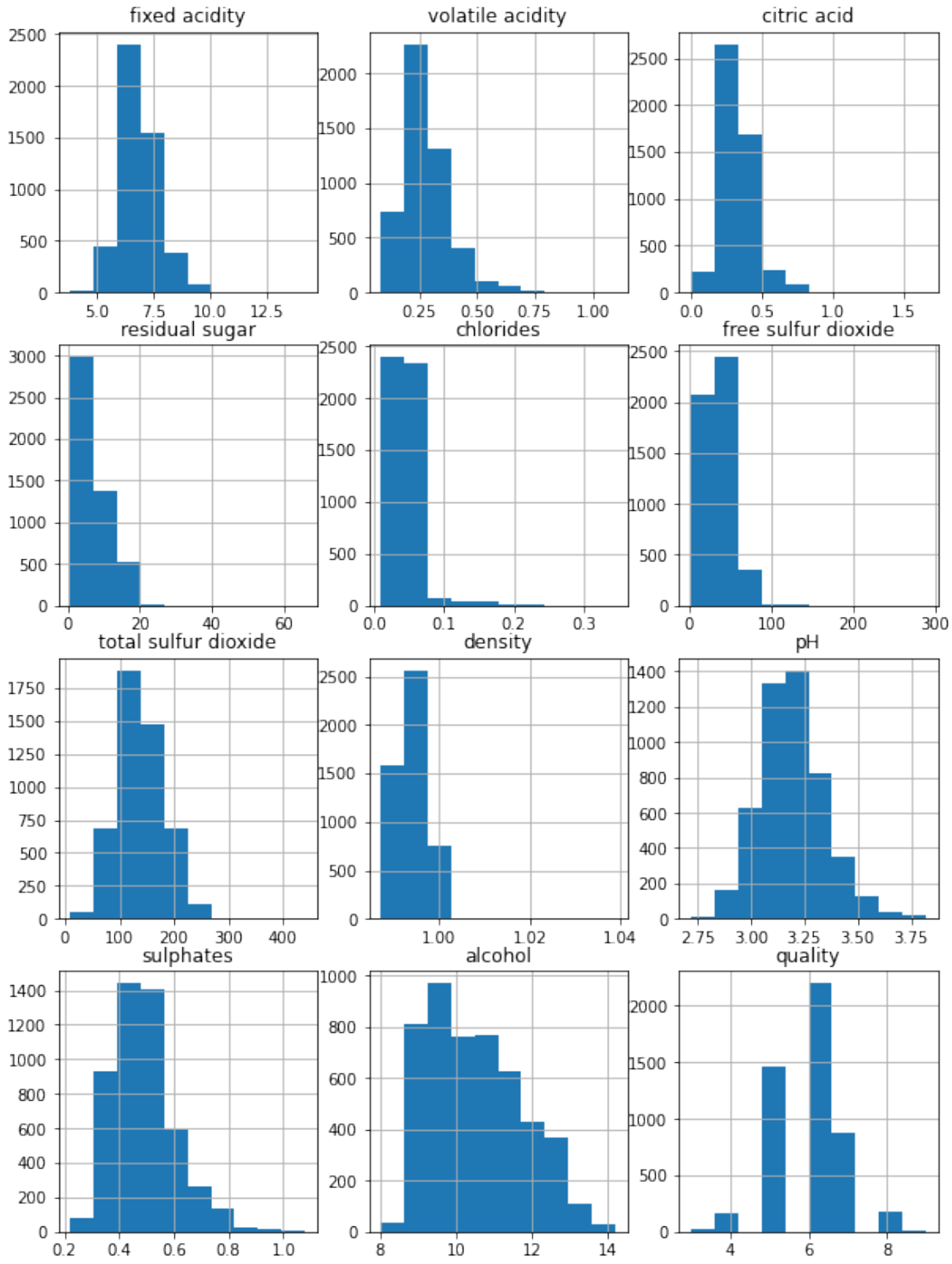






[7]:
```
print(black('Figure 3: Histograms for All Variables', ['bold']))
for n, col in enumerate(wine.columns):
    plt.subplot(4, 3, n+1)
    plt.title(col)
    wine[col].hist(ax = plt.gca(), figsize=(10,14))
```

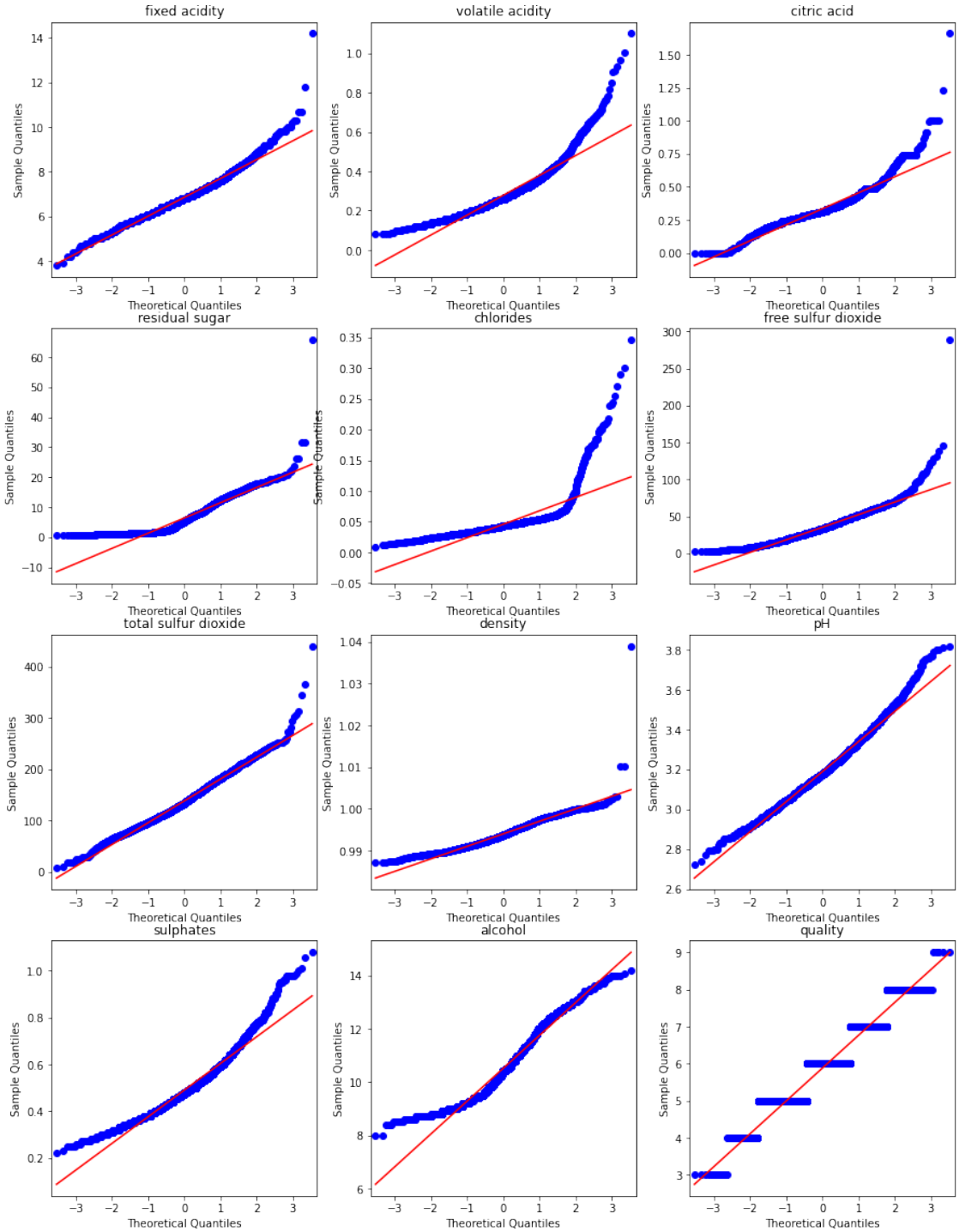
Figure 3: Histograms for All Variables



[8]:
```
print(black('Figure 4: Q-Q plots for all Variables', ['bold']))
f, axs = plt.subplots(nrows=4, ncols=3, figsize=(15,20))
for n, col in enumerate(wine.columns):
```

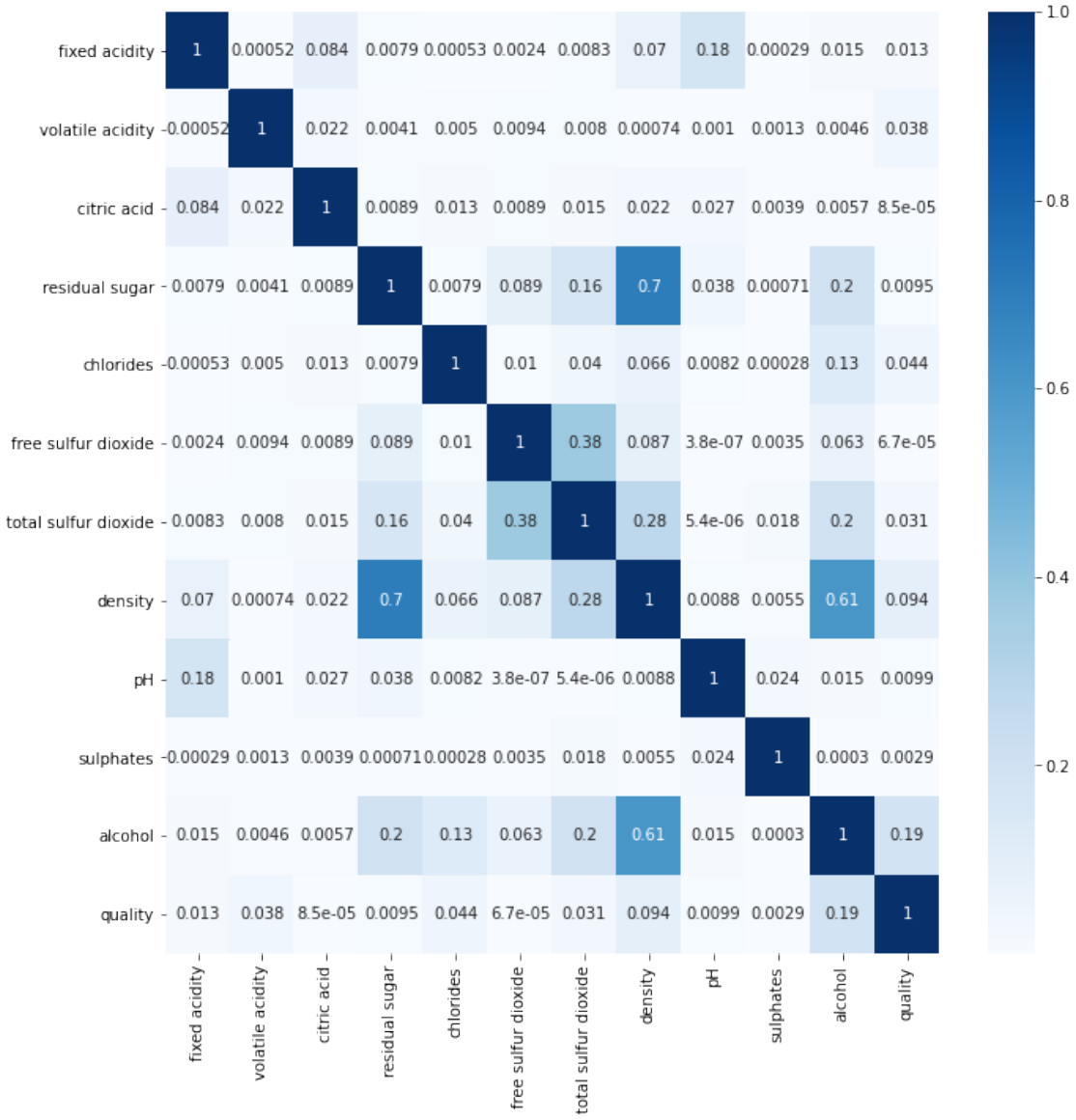
```
    ax = axs[n//3][n%3]
    ax.title.set_text(col)
    qqplot(wine[col], line='s', ax=ax)
```

Figure 4: Q-Q plots for all Variables



[9]:
```
print(black('Figure 5: Correlation Heatmap', ['bold']))
fig, ax = plt.subplots(figsize=(11,11))
# wine.corr() returns the r values representing the strength of the relationship
# between each pair of variables in our dataset
sns.heatmap(wine.corr()**2, annot=True, cmap='Blues', ax=ax);
```

Figure 5: Correlation Heatmap



1.2.1 Testing Assumptions:

In order to meet the assumptions of multiple linear regression, we ran a preliminary linear regression to determine which assumptions were failing at the outset of our analysis. Results of this regression are reported in **Table 3**. The result of this multiple regressions model clearly demonstrated deviations in our dataset from the desired assumptions. Specifically, we found that our dataset included outliers, and that our data (as we intuited from the pairwise correlation graph above) contained colinearities.

As a first step, because we saw a large condition number (7.71e+03), indicating collinearities in our dataset, we attempted to remove elements of the wine dataset which were shown to covary heavily with other variables. As we noticed in our preliminary analysis above, the density variable was shown to covary strongly with multiple other variables, including alcohol, total sulfer dioxide, and residual sugar. As such, we removed density from the wine dataset to ameliorate some of these collinearities. After removing density, the variable causing the most signficinant collinearity, we reconstructed the covariance matrix for viewing in **Figure 5**.

After colinearities were resolved, we ran a series of OLS multiple regression trials to remove outliers in our dataset (Note: The suggestion to run OLS iteratively until all samples classifying as outliers were removed was given in the guide I used to conduct this analysis). In order to determine which wines could be outliers in the wine dataset, we utilized residual values computed by our preliminary Ordinary Least Squares (OLS) multiple regression. The differences between the 'quality' values predicted by the OLS model and the true 'quality' values in the dataset represent the residual values in the case of our analysis. These residuals can then be standardized into z-scores (i.e. residual values are normalized for each wine in the dataset such that the mean residual is 0 and the standard deviation is 1). Once standardized residuals have been obtained, it is common practice in multiple regression workflows to remove data points which result in standardized residuals greater than or equal to 3.29 or less than or equal to -3.29 (Tabachnick & Fidell, 2007). These represent residuals in the z-distribution which have a less than 1% chance of occuring by random chance, and are therefore classified as outliers. As such, we removed all wines from the dataset matching this criteria. In total, 39 wines expressed standardized residuals with a madnitude equal to or greater than 3.29 and were therefore removed from the dataset.

Once outliers were removed, we ran OLS again to obtain our final multiple regression model. Along with this model, we generated a final matrix representing the Variance Inflation Factor (VIF) for each pairwise combination in our dataset (equal to $\frac{1}{1-r^2}$) to test the dataset once again for collinearities. To test for the assumption of homoscedasticity, we generated a scatterplot of standardized residuals. Finally, we and generated a Normal P-P Plot of Regression Standardized Residual.

[10]:
```
# Split data:
predictors = wine.drop(columns=['quality']) # independent variables are␣
 ↪everything but wine quality
labels = wine['quality'] # Dependent variable is wine quality
```

[11]:
```
# Remove non-normal columns
predictors = predictors.drop(columns=['chlorides', 'volatile acidity'])
```

[12]:
```
# PRELIMINARY OLS:
X = predictors
X = sm.add_constant(X)
y = labels
```

[13]:
```
# Conduct least squares multiple regression
results = sm.OLS(y, X).fit() # OLS = Ordinary Least Squares
print('Multiple regression finished.\n\nIndependent Variables: {}\n\Dependent␣
 ↪Variable:{}'.format(
      list(predictors.columns), 'quality'))
```

```
Multiple regression finished.

Independent Variables: ['fixed acidity', 'citric acid', 'residual sugar', 'free
sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates',
'alcohol']
\Dependent Variable:quality
```

[14]:
```
print(black('Table 3: Preliminary OLS Results', ['bold']))
results.summary()
```

```
Table 3: Preliminary OLS Results
```

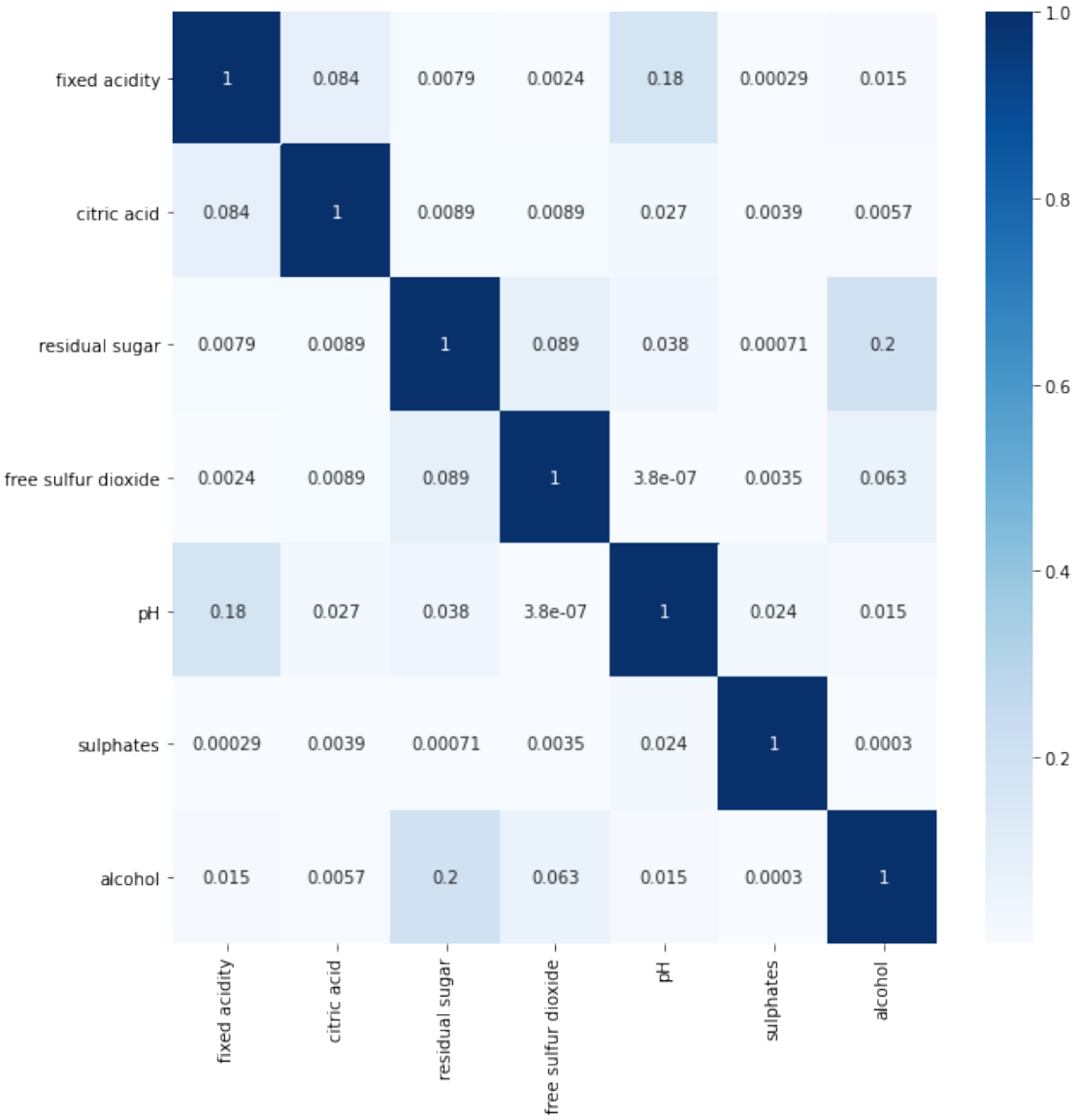
[14]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                quality   R-squared:                       0.242
Model:                            OLS   Adj. R-squared:                  0.240
Method:                 Least Squares   F-statistic:                     173.2
Date:                Mon, 08 Jun 2020   Prob (F-statistic):          6.15e-286
Time:                        12:26:56   Log-Likelihood:                -5676.6
No. Observations:                4898   AIC:                         1.137e+04
Df Residuals:                    4888   BIC:                         1.144e+04
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
========
                       coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
--------
const                189.1700     18.926      9.995      0.000     152.066
226.274
fixed acidity          0.1048      0.021      4.989      0.000       0.064
0.146
citric acid            0.2761      0.096      2.862      0.004       0.087
```

```
0.465
residual sugar           0.0928      0.008     12.285      0.000       0.078
0.108
free sulfur dioxide      0.0063      0.001      7.435      0.000       0.005
0.008
total sulfur dioxide    -0.0015      0.000     -4.026      0.000      -0.002
-0.001
density               -190.3994     19.180     -9.927      0.000    -228.000
-152.799
pH                       0.9140      0.106      8.630      0.000       0.706
1.122
sulphates                0.7483      0.103      7.279      0.000       0.547
0.950
alcohol                  0.1212      0.024      4.954      0.000       0.073
0.169
==============================================================================
Omnibus:                      120.572   Durbin-Watson:                   1.622
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              274.516
Skew:                           0.069   Prob(JB):                     2.45e-60
Kurtosis:                       4.152   Cond. No.                     3.67e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 3.67e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

[15]:
```
# Drop covarying attributes to reduce colinearity:
predictors = predictors.drop(columns=['density', 'total sulfur dioxide'])
print(black('Figure 6: Correlation Heatmap After Attribute Removal', ['bold']))
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(predictors.corr()**2, annot=True, cmap='Blues', ax=ax);
```

Figure 6: Correlation Heatmap After Attribute Removal



[16]:
```
#  OLS Trials To Remove Outliers:

i = 1
while(True):

    X = predictors
    X = sm.add_constant(X)
    y = labels
    results = sm.OLS(y, X).fit() # OLS = Ordinary Least Squares

    # If there are no outliers remaining:
```

```
    if (min(results.resid_pearson) > -3.29 and max(results.resid_pearson) < 3.
↪29):
        break

    else:
        # Remove outliers
        std_residuals = results.resid_pearson #Residuals, normalized to have␣
↪unit variance.

        # Remove all outlier wines:
        orig_num_wines = len(predictors)
        predictors = predictors[abs(std_residuals) < 3.29]
        labels = labels[abs(std_residuals) < 3.29]
        remaining_wines = len(predictors)
        print("Number of Outliner wines removed on trial {} = {}".format(i,␣
↪orig_num_wines - remaining_wines))

    i += 1
```

```
Number of Outliner wines removed on trial 1 = 27
Number of Outliner wines removed on trial 2 = 10
Number of Outliner wines removed on trial 3 = 2
```

1.2.2 Analyzing Results:

After preparing the data extensively to meet the assumptions required by multiple regresssion, we conducted the final analysis below. We conclude the data, after adjustments were made to remove unsuitable data attributes, met all prerequisite assumptions to conduct a multiple regression analysis. Tests to determine whether or not the data met the assumption of collinearity indicated that multicollinearity was not a concern (all VIF values were below 5 and close to 1, indicating little covariance between attributes - see **Table 5**). Although the Condition Number remains high, this is merely because of the large number of indepdendent variables in this multiple regression, and can be safely ignored. Likewise, the data met the assumption of indepdendent errors (Durbin-Watson Value = 1.656, see **Table 4**).

The histogram of standardized residuals (**Figure 7**) indicated that the data contained approximately normally distributed errors, and the normal P-P plot of standardized residuals (**Figure 8**) corroborated this finding.

The scatterplot of standardized residuals (**Figure 9**) showed that the data met assumptions of homogeneity of variance and linearity. Although this plot may appear misshapen, this is merely because the predicted value (wine quality) is a discrete variable with limited range (i.e. it is a score from 0-10). To satisfy assumption of homoscedasticity, this plot must show a roughly circular or patternless distribution of values, indicating an equal spread of variance in the residuals. Although the discrete nature of the predicted value produces streaks of points, they remain organized in a rougly spherical fashion and therefore demonstrate homoscedasticity.

The data also met the assumption of non-zero variances (see Descriptive statistics in **Table 2**).

Using the method of least squares, it was found that several chemical attributes of white wine - fixed acidity, citric acid content, residual sugar content, free sulfur dioxide contant, pH, sulphates content, and alcohol content - explain a significant ammount of the variance in the value of percieved wine quality ($F(7, 4851) = 238.9, p < 0.01, R^2 = 0.256, R^2 Adjusted = .255$).This analysis shows that all of the tested wine chemical properties were predictive of quality, albeit to different degress (see **Table 4** for coefficients, test statistics, and associated p values for each independent variable).

[17]:
```
# Conduct final least squares multiple regression
X = predictors
X = sm.add_constant(X)
y = labels
results = sm.OLS(y, X).fit() # OLS = Ordinary Least Squares
print('Multiple Regression finished.\n\nDependent variables: {}\n\npredicted␣
 ↪variable:{}'.format(
      list(predictors.columns), 'quality'))
```

```
Multiple Regression finished.

Dependent variables: ['fixed acidity', 'citric acid', 'residual sugar', 'free
sulfur dioxide', 'pH', 'sulphates', 'alcohol']

predicted variable:quality
```

[18]:
```
print(black('Table 4: Final OLS Results', ['bold']))
results.summary()
```

```
Table 4: Final OLS Results
```

[18]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                quality   R-squared:                       0.256
Model:                            OLS   Adj. R-squared:                  0.255
Method:                 Least Squares   F-statistic:                     238.9
Date:                Mon, 08 Jun 2020   Prob (F-statistic):          2.57e-306
Time:                        12:26:56   Log-Likelihood:                -5446.7
No. Observations:                4859   AIC:                         1.091e+04
Df Residuals:                    4851   BIC:                         1.096e+04
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
=======
                         coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const                  0.9187      0.332      2.770      0.006       0.268
```

```
1.569
fixed acidity          -0.0539      0.015     -3.705      0.000      -0.082
-0.025
citric acid             0.2189      0.093      2.355      0.019       0.037
0.401
residual sugar          0.0174      0.002      7.093      0.000       0.013
0.022
free sulfur dioxide     0.0059      0.001      8.525      0.000       0.005
0.007
pH                      0.2487      0.080      3.093      0.002       0.091
0.406
sulphates               0.3803      0.095      3.996      0.000       0.194
0.567
alcohol                 0.3762      0.010     38.173      0.000       0.357
0.396
==============================================================================
Omnibus:                       28.395   Durbin-Watson:                   1.656
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               30.338
Skew:                           0.155   Prob(JB):                     2.58e-07
Kurtosis:                       3.233   Cond. No.                     1.31e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.31e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

[19]:

```
# Matrix of VIF values for each pairwise combination
print(black('Table 5: VIF Scores for each pairwise combination', ['bold']))
1/ (1 - predictors.corr() ** 2) # Calculation for VIF Scores
```

Table 5: VIF Scores for each pairwise combination

[19]:

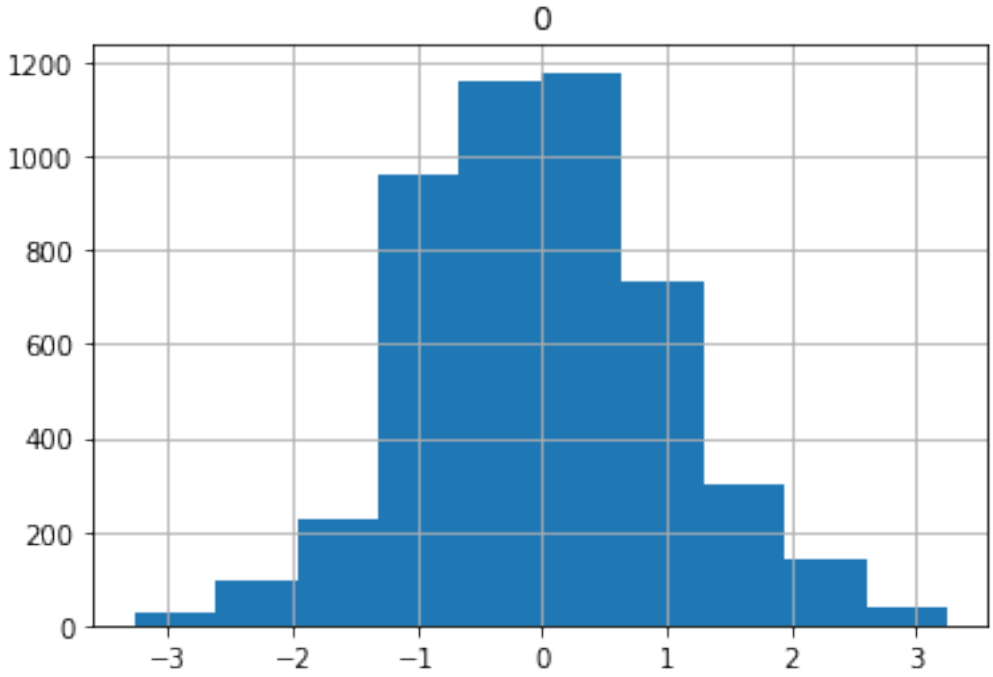
	fixed acidity	citric acid	residual sugar
fixed acidity	inf	1.091965	1.006992
citric acid	1.091965	inf	1.009234
residual sugar	1.006992	1.009234	inf
free sulfur dioxide	1.003029	1.010870	1.106629
pH	1.220777	1.028516	1.037412
sulphates	1.000316	1.003663	1.000749
alcohol	1.014063	1.006089	1.250321

	free sulfur dioxide	pH	sulphates	alcohol
fixed acidity	1.003029	1.220777	1.000316	1.014063
citric acid	1.010870	1.028516	1.003663	1.006089

residual sugar	1.106629	1.037412	1.000749	1.250321
free sulfur dioxide	inf	1.000010	1.003434	1.068582
pH	1.000010	inf	1.024271	1.014012
sulphates	1.003434	1.024271	inf	1.000291
alcohol	1.068582	1.014012	1.000291	inf

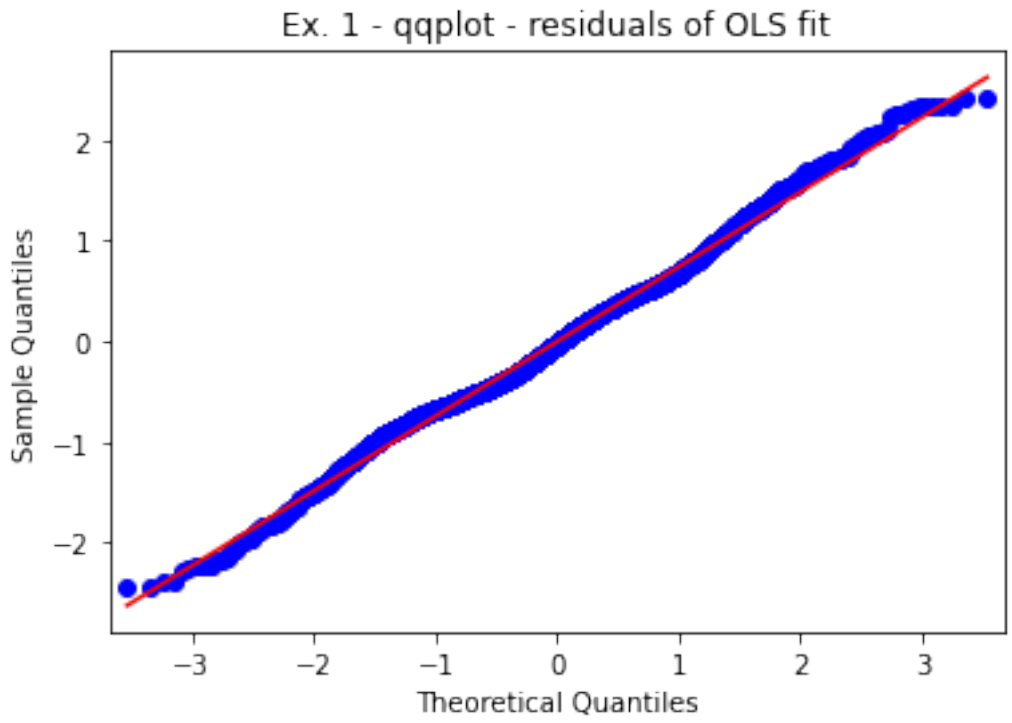
[20]:
```
print(black('Figure 7: Scatterplot of Standardised Residuals', ['bold']))
standard_residuals = pd.DataFrame(results.resid_pearson)
standard_residuals.hist();
```

Figure 7: Scatterplot of Standardised Residuals



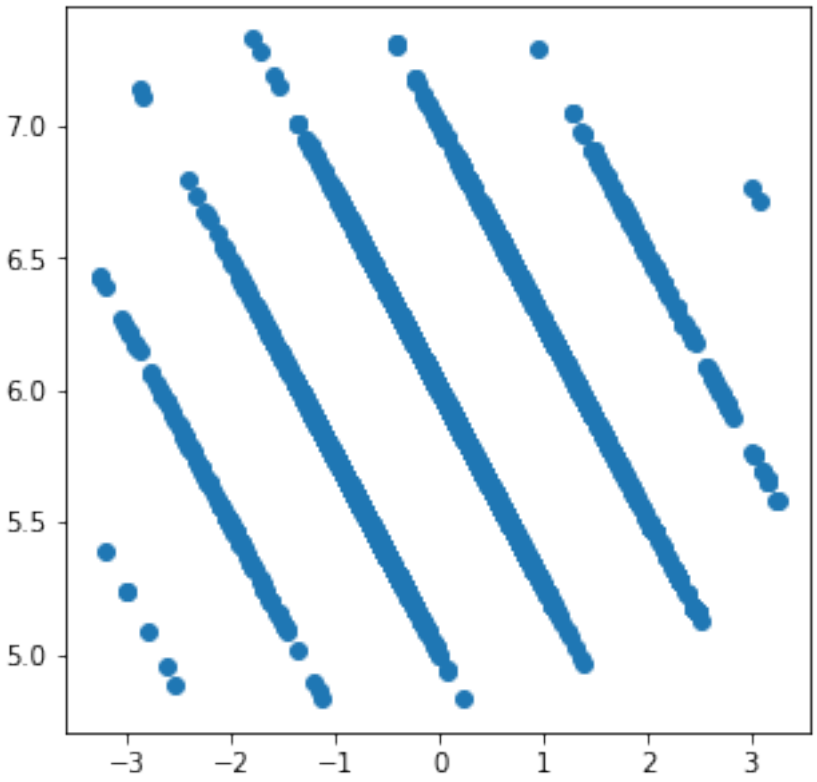
[21]:
```
print(black('Figure 8: Normal P-P Plot of Regression Standardized Residual.',␣
 ↪['bold']))
probplot = sm.ProbPlot(results.resid)
fig = probplot.qqplot(line='s')
h = plt.title('Ex. 1 - qqplot - residuals of OLS fit')
plt.show()
```

Figure 8: Normal P-P Plot of Regression Standardized Residual.



[22]:
```
print(black('Figure 9: Scatterplot of Standardised Residuals', ['bold']))
pred_vals = results.fittedvalues.copy()
residuals = results.resid_pearson
fig, ax = plt.subplots(figsize=(5,5))
_ = ax.scatter(residuals, pred_vals)
```

Figure 9: Scatterplot of Standardised Residuals



2 References:

- Multiple Regression Using Statsmodels
- Linear Regression Diagnostic in Python with StatsModels
- Reporting Multiple Regressions in APA format, Part One and Part Two
- Modeling wine preferences by data mining from physicochemical properties
- Tabachnick, B. G., & Fidell, L. S. (2007). Using multivariate statistics (5th ed.). Allyn & Bacon/Pearson Education.