Table of Contents

[Correlation and Covariance 1](#_Toc30467269)

[Covariance 2](#_Toc30467270)

[Correlation 2](#_Toc30467271)

[Causation and Correlation 3](#_Toc30467272)

[Multiple Linear Regression 4](#_Toc30467273)

[A Brief Exploratory Data Analysis (EDA) 4](#_Toc30467274)

[Coefficient of Determination () 9](#_Toc30467275)

[Adjusted 9](#_Toc30467276)

[F-Statistic 10](#_Toc30467277)

[F-Statistic (p-value) 10](#_Toc30467278)

[Log-likelihood 10](#_Toc30467279)

[AIC 11](#_Toc30467280)

[BIC 11](#_Toc30467281)

[Coefficient p-Values 12](#_Toc30467282)

[Root Mean Squared Error (RMSE) 12](#_Toc30467283)

[Collinearity 13](#_Toc30467284)

[Visualizing Actual, Predicted and Error Residuals 15](#_Toc30467285)

[Quantile-Quantile Plot of Residuals 16](#_Toc30467286)

[Model Selection 17](#_Toc30467287)

[Future Considerations 17](#_Toc30467288)

[Model Equation 17](#_Toc30467289)

[Model Interpretation 18](#_Toc30467290)

[Data Scoring 18](#_Toc30467291)

[Anscombe’s Quartet 19](#_Toc30467292)

## Correlation and Covariance

Covariance and correlation are not the same, but they are closely related to each other. This section examines these two statistical measures with equations, explanations, and real-life examples.

Both covariance and correlation show direction. However, correlation is also able to identify the strength of the relationship.

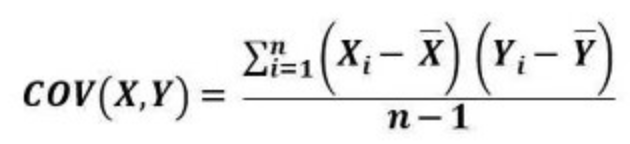
Figure : Covariance and Correlation Direction



### Covariance

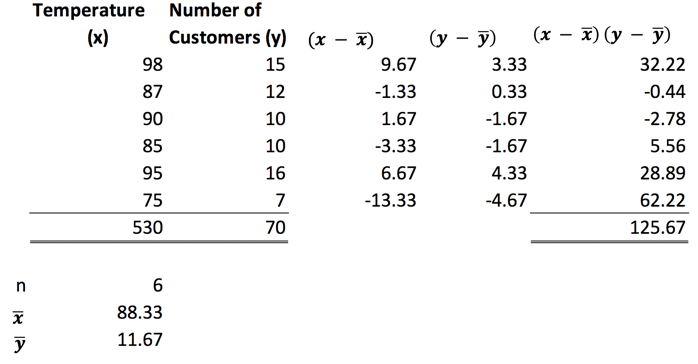
Covariance measures how two variables change together. It tells you if there is a relationship between two variables and which direction that relationship is in. **Positive covariance**, indicates that as one variable increases the other also increases. A **negative covariance**, implies that as one variable increases the other decreases.

Equation : Covariance



Example : Covariance for Ice Cream Sales During Warmer Months

This example examines a trend for more ice cream sales during warmer months. Before make any decisions around going into the icecream business, you want to be sure that the relationship is real.



cov(x,y) = = = 25.13

**Conclusion:**

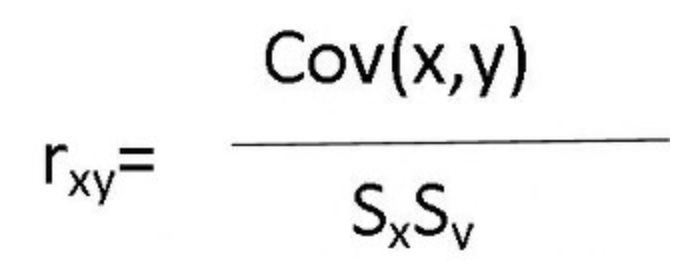
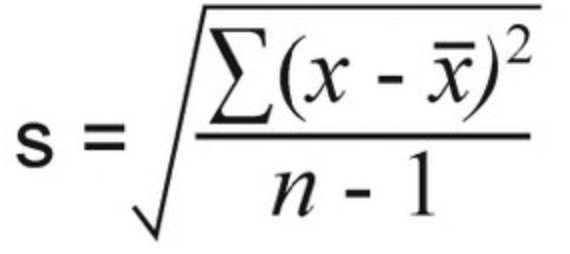
The covariance is positive so there is a positive relationship between sales and warmer months.

### Correlation

Correlation, like covariance, is a measure of how two variables change in relation to each other, but correlation goes one step further than covariance since correlation tells how strong the relationship is.

To determine the strength of a relationship, you must use the formula for correlation coefficient. This formula will result in a number between -1 and 1. 1 indicates a perfect correlation. -1 represents a perfect inverse correlation. Zero indicates no relationship exists between the two variables.

Equation : Correlation

 **where** sample standard deviation is: 

Example : Correlation

This example builds on the covariance calculations from Example 1 to also show the correlation between sales and the warmer months.  
=

= = = 8.14 = = = 3.39

= = 0.912

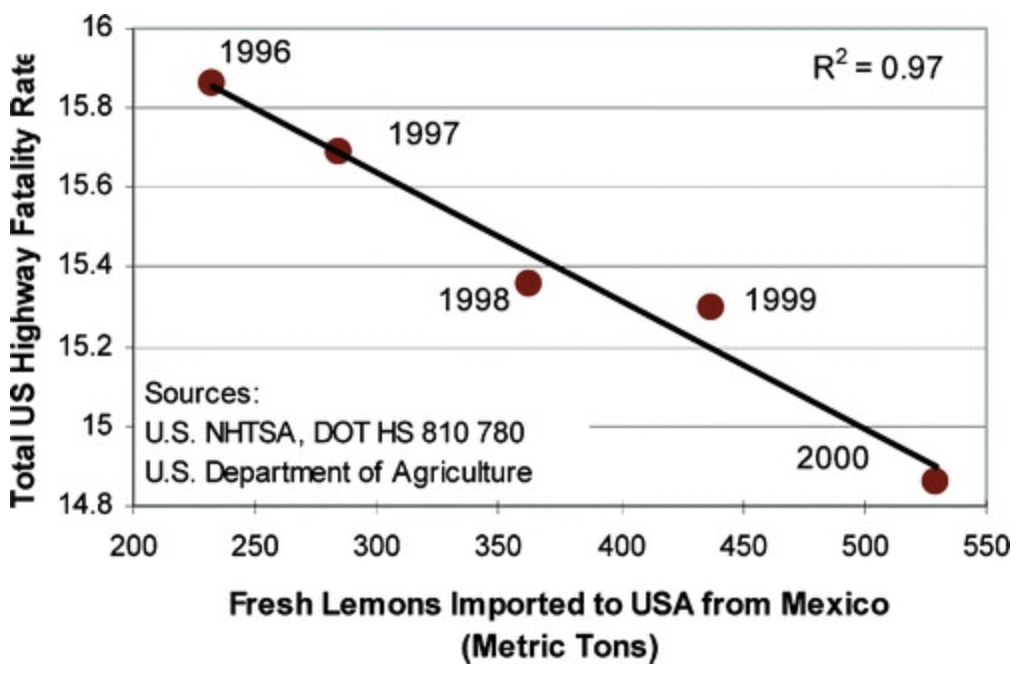
**Conclusion:**

With correlation, not only have we established that there is a positive relationship between ice cream sales and warmer months, we have also shown a strong relationship since 0.912 is close to 1. We need to be ready to purchase more ice cream stock for the summer.

### Causation and Correlation

It is important to emphasize that even though two variables may appear to be correlated it does not necessarily mean that they are related in any way. For example, the US highway fatality rate may statistically be correlated with fresh lemon imports but these two variables are actually completely unrelated. (see Figure 2).

Figure : Statistically Correlated but Completely Unrelated.



When building data models, be careful not to assume that relationships between variables actually exist even when their rise and fall coincides. Remember always:

**CORRELATION DOES NOT IMPLY CAUSATION**

This rule may seem sensible but sometimes it is difficult to tell whether a true relationship exists or not. For example, many illnesses may appear to be correlated to diet and hereditary traits yet their true causes really are still unknown.

## Multiple Linear Regression

Last day we discussed ordinary least square regression (OLS) with one predictor variable. Basically, we were using OLS to generate an equation that defined a between a best-fit straight-line relation between X and Y. This same routine can also include multiple predictor variables to develop a more powerful predictive model for a target variable.

### A Brief Exploratory Data Analysis (EDA)

Example : Half-Baked Exploratory Data Analysis

This example begins the study of how to build a model to predict wine quality. In an attempt to keep this document brief but substantial I will not conduct a full-blown exploratory data analysis of the wine quality data. However, to quickly understand the data I will show a snapshot of the data, some basic statistics about the variables and their correlations. Table 1 shows a glimpse of the data for each column within the data set.

Table : Wine Quality DataFrame

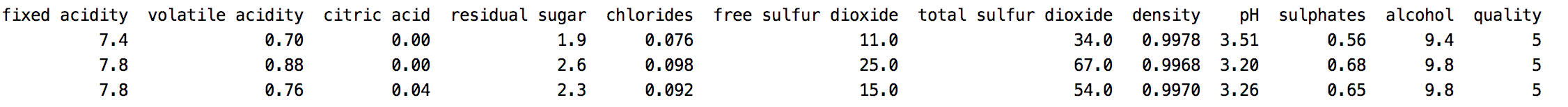


Table 2 shows a summary of numeric values within the wine data set. Fortunately for us there are no missing values. We can tell since all counts are the same at 1599.

Table : Statistical Summary

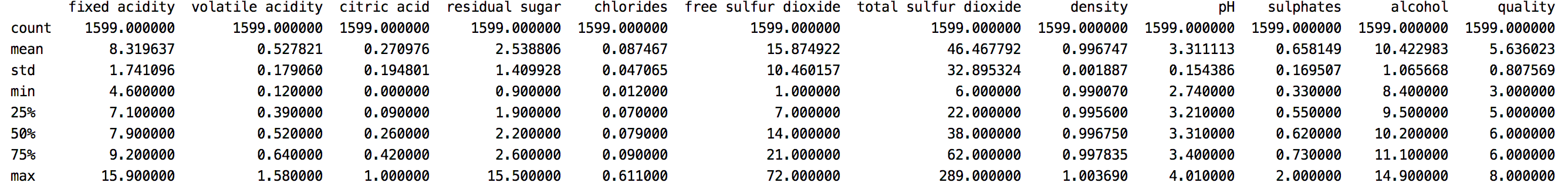
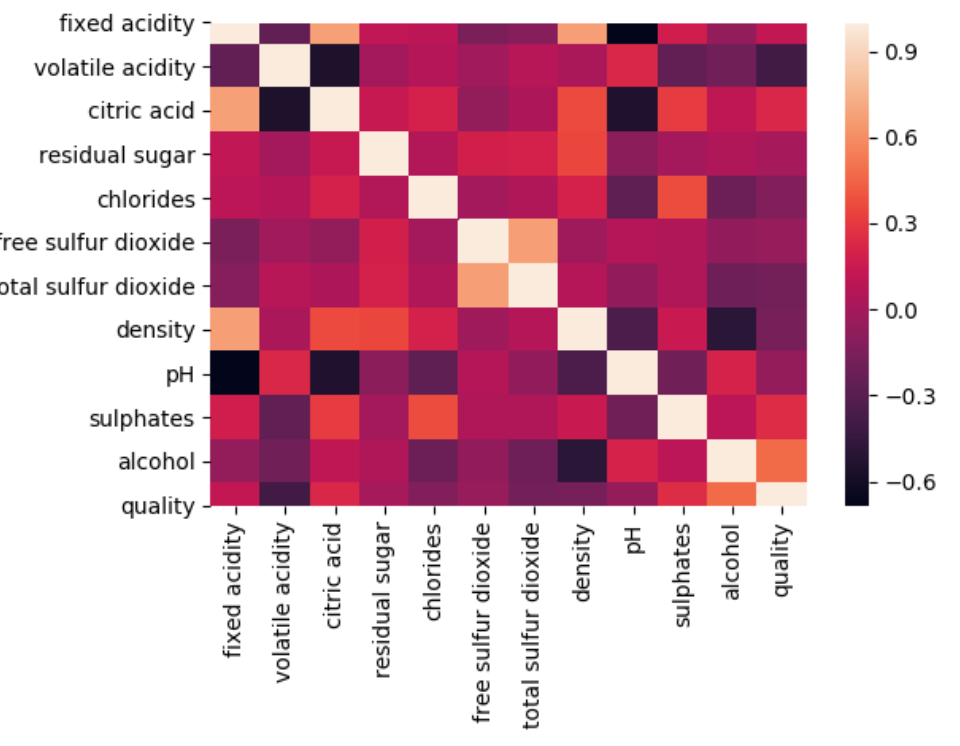


Figure 3 shows a heatmap to help us understand how all variables are correlated. At a glance it looks like alcohol, sulphates and citric acid may be decent predictors of quality but we will have to explore this further.

Figure : Scatter Matrix of Predictor Variables

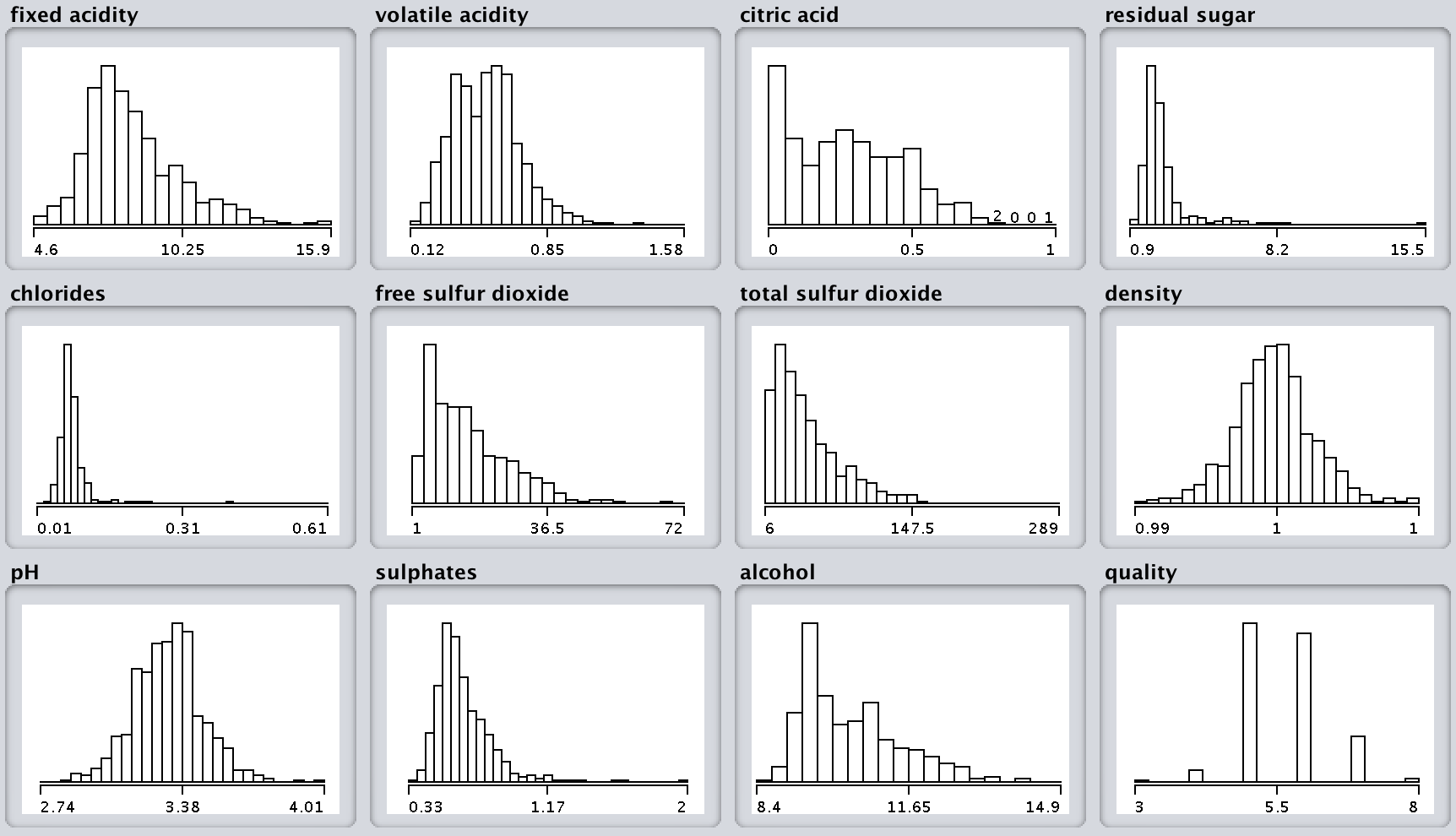


Here is the code used to present the data frame, descriptive statistics and heatmap:

|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  import seaborn as sns  PATH = "/Users/pm/Desktop/DayDocs/2019\_2020/PythonForDataAnalytics/workingData/"  CSV\_DATA = "winequality.csv"  dataset = pd.read\_csv(PATH + CSV\_DATA,  skiprows=1, # Don't include header row as part of data.  encoding = "ISO-8859-1", sep=',',  names=('fixed acidity', 'volatile acidity', 'citric acid',  'residual sugar', 'chlorides', 'free sulfur dioxide',  'total sulfur dioxide', 'density', 'pH', 'sulphates',  'alcohol', 'quality'))  # Show all columns.  pd.set\_option('display.max\_columns', None)  # Increase number of columns that display on one line.  pd.set\_option('display.width', 1000)  print(dataset.head(3))  print(dataset.describe())  X = dataset[[ 'fixed acidity', 'volatile acidity', 'citric acid',  'residual sugar', 'chlorides', 'free sulfur dioxide',  'total sulfur dioxide', 'density', 'pH', 'sulphates',  'alcohol', 'quality']]  # Compute the correlation matrix  corr = dataset.corr()  # plot the heatmap  sns.heatmap(corr,  xticklabels=corr.columns,  yticklabels=corr.columns)  plt.show() |

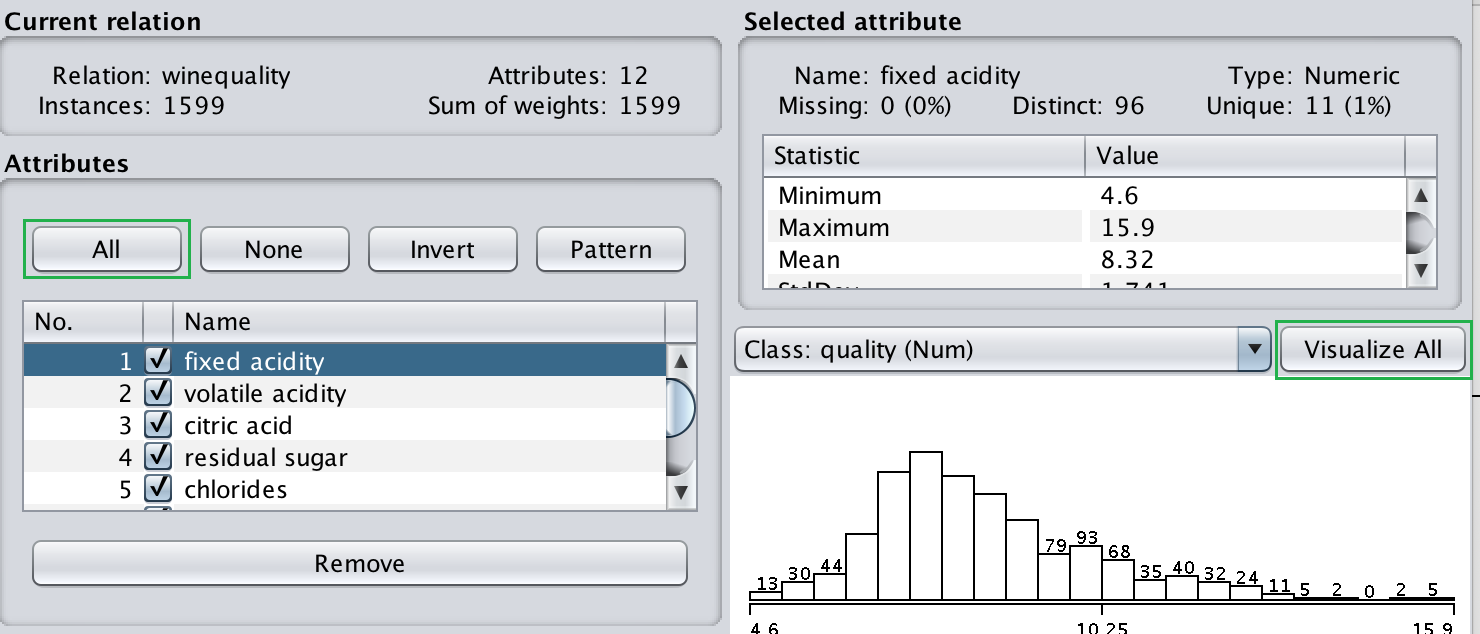
It also helps to visualize the distributions of each variable (refer to Figure 4). The variables appear to be normal-like but they not perfectly normal. Citric acid appears to have a normal-like distribution but it does spike on the left. Most importantly, the wine quality distribution is discrete and it is not perfectly normal.

Figure : Wine Quality Variable Distributions



I used Weka to generate these diagrams.

<https://www.cs.waikato.ac.nz/ml/weka/>



Example : Calculating the Wine Quality Frequencies

Wine quality ranges between 3 and 8. The wine quality variable is discrete and most ratings in the set are between 4 and 5. Very little data exists for other ratings. (This could be a problem). It would be good to know the frequencies of the wine quality ratings. Table 3 shows the quality frequencies.

Table : Wine Quality Frequencies

|  |
| --- |
| 3 10  8 18  4 53  7 199  6 638  5 681 |

To generate the frequency table for wine quality, add this code to the end of the code in Example 3:

|  |
| --- |
| # Show counts.  print(dataset['quality'].value\_counts(ascending=True)) |

Example : Multiple Linear Regression for Wine Quality: Model A

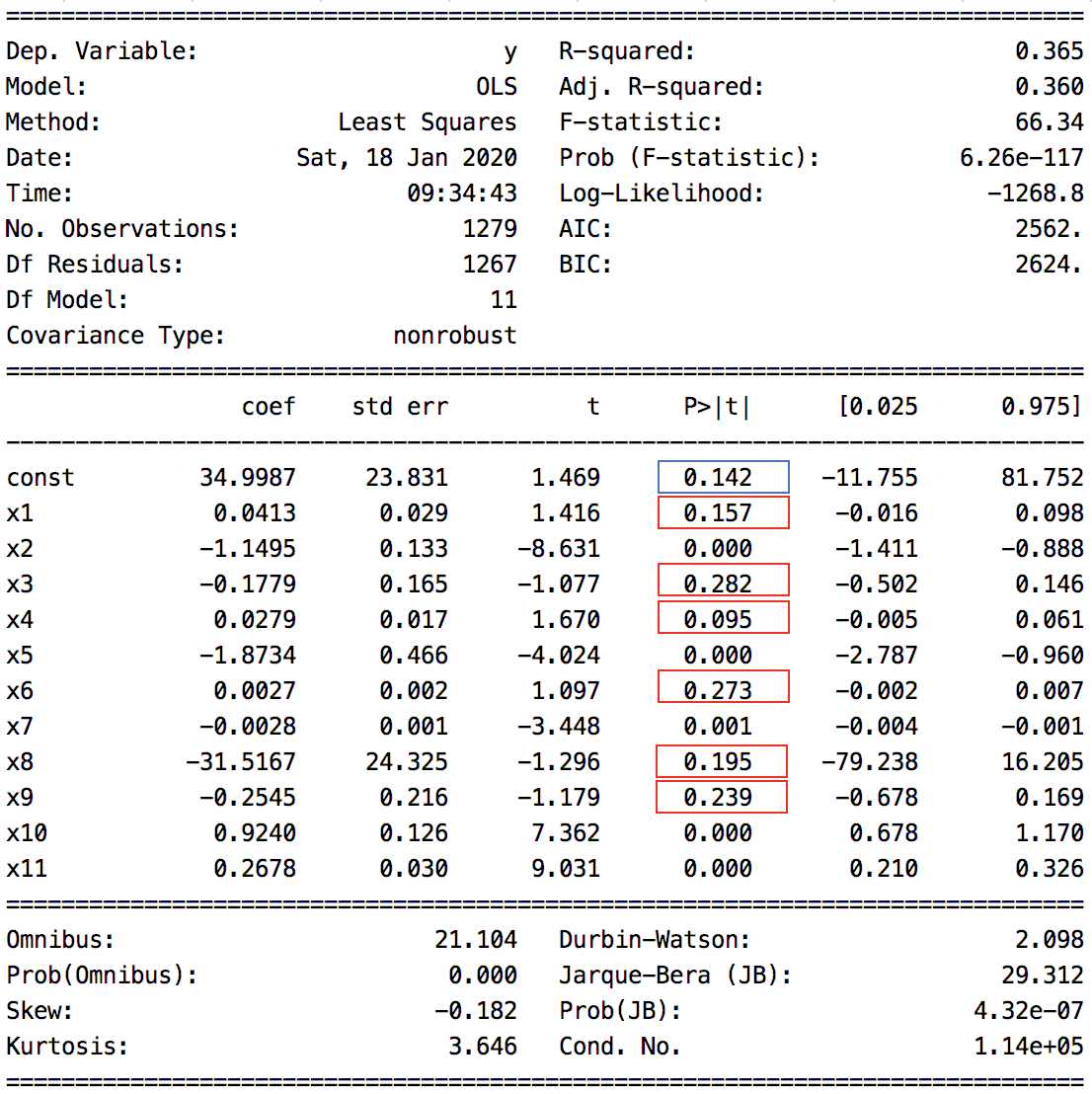
This example demonstrates how to build a crude multiple regression model for predicting wine quality. The example helps to demonstrate steps used to adjust and interpret the model. Actually though, the model that will be generated is not very useful but we will explain why.

Here is some initial code which reads all data from a wine quality data set. The code then sets up a multiple regression model to predict quality with all other available columns. Note how y is defined as our target variable and our predictor variables are assigned to X.

|  |
| --- |
| import pandas as pd  from sklearn.model\_selection import train\_test\_split  from sklearn.linear\_model import LinearRegression  from sklearn import metrics  import statsmodels.api as sm  import numpy as np  PATH = "/Users/pm/Desktop/DayDocs/2019\_2020/PythonForDataAnalytics/workingData/"  CSV\_DATA = "winequality.csv"  dataset = pd.read\_csv(PATH + CSV\_DATA,  skiprows=1, # Don't include header row as part of data.  encoding = "ISO-8859-1", sep=',',  names=('fixed acidity', 'volatile acidity', 'citric acid',  'residual sugar', 'chlorides', 'free sulfur dioxide',  'total sulfur dioxide', 'density', 'pH', 'sulphates',  'alcohol', 'quality'))  # Show all columns.  pd.set\_option('display.max\_columns', None)  # Increase number of columns that display on one line.  pd.set\_option('display.width', 1000)  print(dataset.head())  print(dataset.describe())  X = dataset[['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',  'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',  'pH', 'sulphates','alcohol']].values  # Adding an intercept \*\*\* This is requried \*\*\*. Don't forget this step.  # The intercept centers the error residuals around zero  # which helps to avoid over-fitting.  X = sm.add\_constant(X)  y = dataset['quality'].values  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)  model = sm.OLS(y\_train, X\_train).fit()  predictions = model.predict(X\_test) # make the predictions by the model  print(model.summary())  print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, predictions))) |

The model output is shown in Figure 5. Explanations of the main statistics are explained after Figure 5.

Figure : Wine Quality Model Output

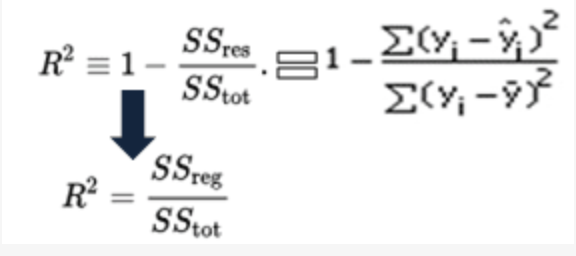


Root Mean Squared Error: 0.6200574149384108

### Coefficient of Determination ()

As discussed last day, the coefficient of determination, defines the proportion of variance that is accounted for by the model.

Equation :



The result in Figure 5 shows that = 0.365.

### Adjusted

shows the proportion of variance accounted for by the model but it also includes a penalty for the number of predictors, *p*. Calculating manually gives:

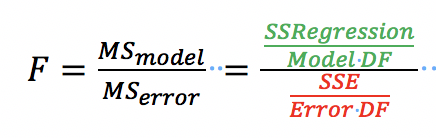
= 1 - = 1 - = 1 - = 0.35948 0.360

#### Correlation

The square root of the coefficient of determination, is actually the **correlation** between the model. This implies a correlation between the model and the wine quality of = = 0.604 which is decent.

### F-Statistic

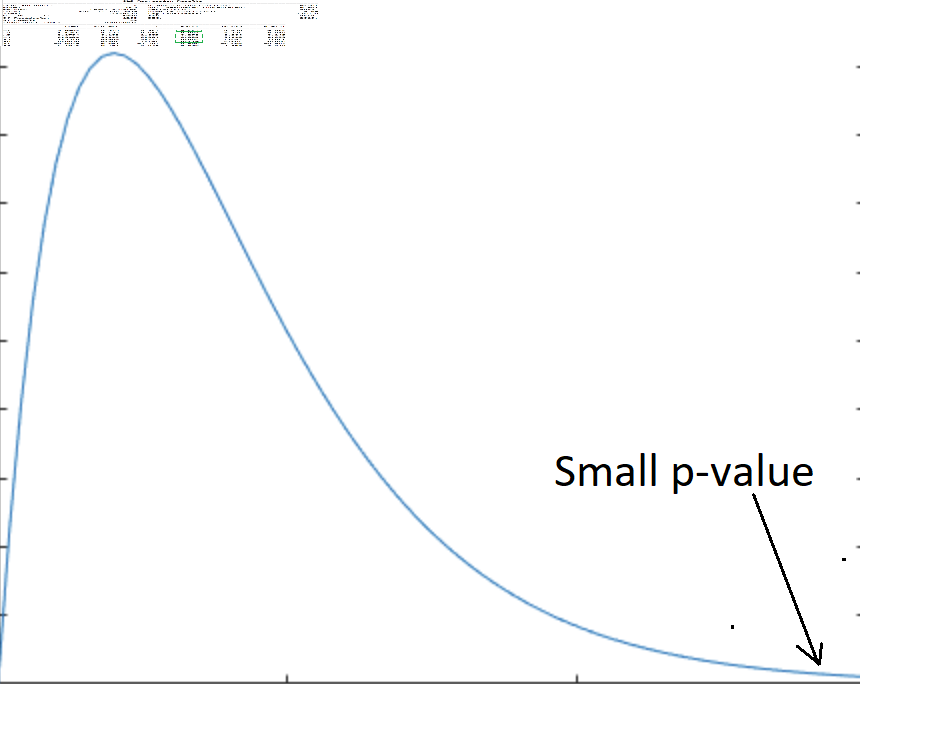
The large F-Statistic, 8683, does indicate that a large proportion of target variable variance is described by the model relative to the variance due to error.

 TotalDF=N-1, ModelDF=# params, ErrorDF=TotalDF-ModelDF

### F-Statistic (p-value)

The small probability, **p-value**, for the F-statistic suggests a rejection of the null hypothesis on the Chi-Square distribution.

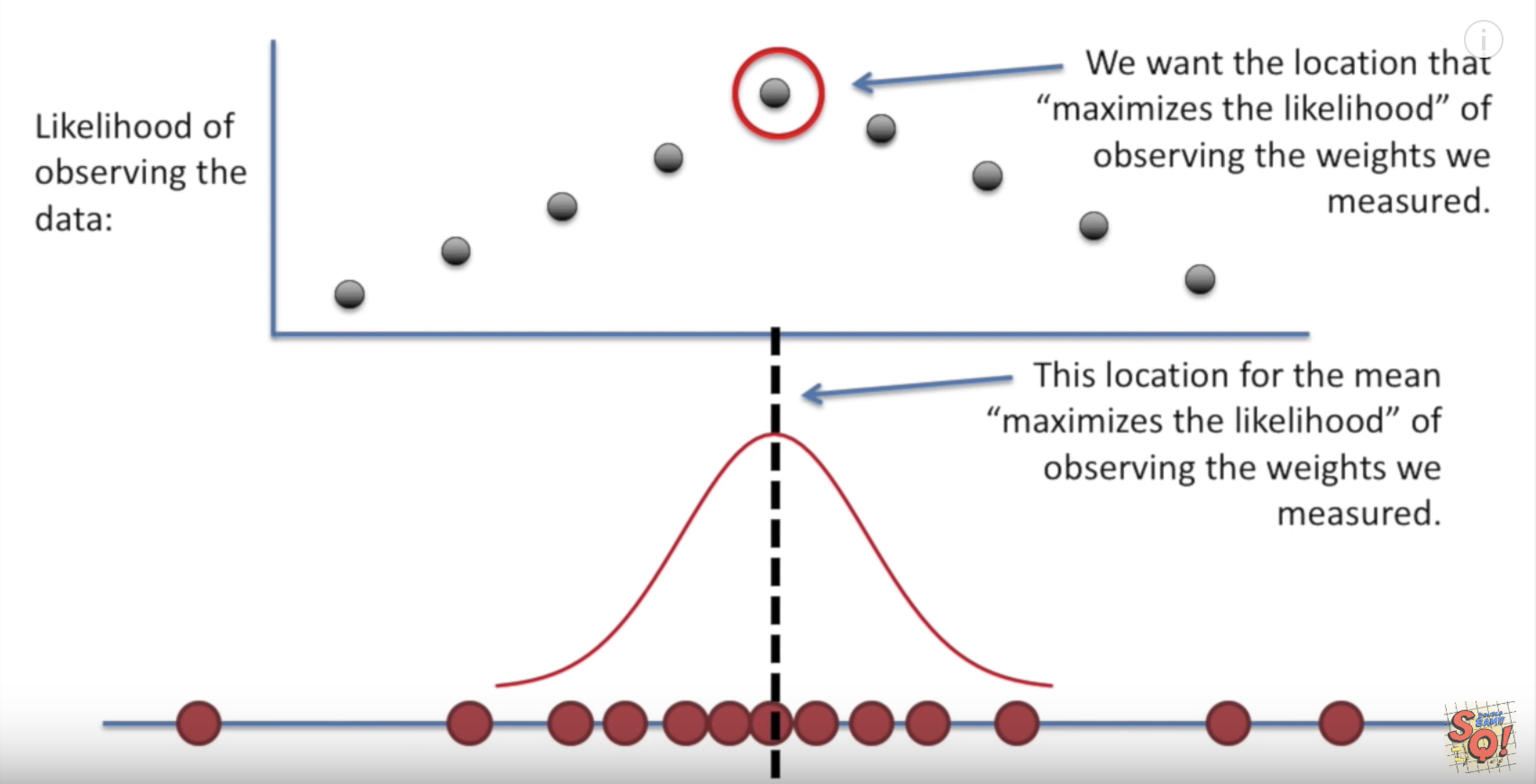
: The model offers no predictive value.



It appears that we can safely reject the null hypothesis, .

### Log-likelihood

The log-likelihood describes the cumulative sum of probability for all sample values under the best-fitting normal distribution. A larger log-likelihood is better.



See <https://www.youtube.com/watch?v=XepXtl9YKwc>

### AIC

To help fight bias from over-fitting, the AIC (Akaike Information Criterion) is a measure of validity which combines the log-likelihood score with a penalty for larger numbers of model variables. When comparing models, the desirable model will have a lower AIC score. The AIC does not use an absolute score.

Equation : AIC

**AIC = -2\*Log-likelihood(LL) + 2\*k**

k is number of predictor variables + 1 for the intercept

For our example, the AIC measure is

AIC = -2 \* -1268.8 + 2\*(11+1) 2562

### BIC

The Bayesian Information Criterion is used to evaluate models based on the likelihood function. The model with the lowest BIC is preferred. Similar to the AIC score, the BIC score also does not use an absolute value. The BIC score is based on the likelihood function and the AIC. BIC has a higher penalty for total number of predictor variables than the AIC.

k is number of predictor variables + 1 for the intercept.

Equation : BIC

**BIC = -2 \* LL + log(N) \* k**

For Example 5;

BIC = -2\*(-1268.8) + log(1279)\*12 = 2539.8 + 78.69 2623

### Coefficient p-Values

The model summary output also shows the p-values for the hypothesis tests for the model coefficients. Where:

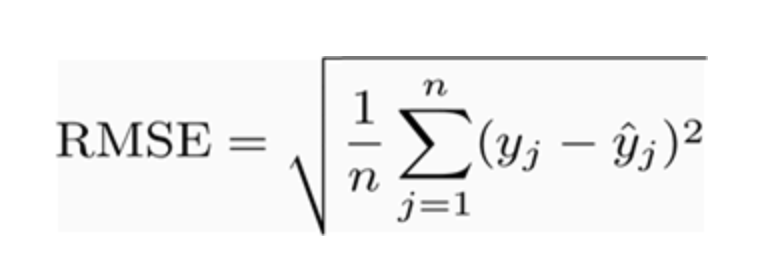
: The coefficient offers no predictive significance.

|  |  |
| --- | --- |
|  | On that note, the probabilities (**p-values**) for x1 (fixed acidity), x3 (citric acid), x4 (residual sugar), x6 (free sulfur dioxide), x8 (density) and x9 (pH) are high.  In other words, these coefficients are statistically insignificant. We might try to eliminate the coefficients to improve the model. |

### Root Mean Squared Error (RMSE)

The root mean square error determines the average dispersion between the actual and predicted values. A lower RMSE (deviation) is preferred.

Equation : Root Mean Square Error



Root Mean Squared Error: 0.6200574149384108

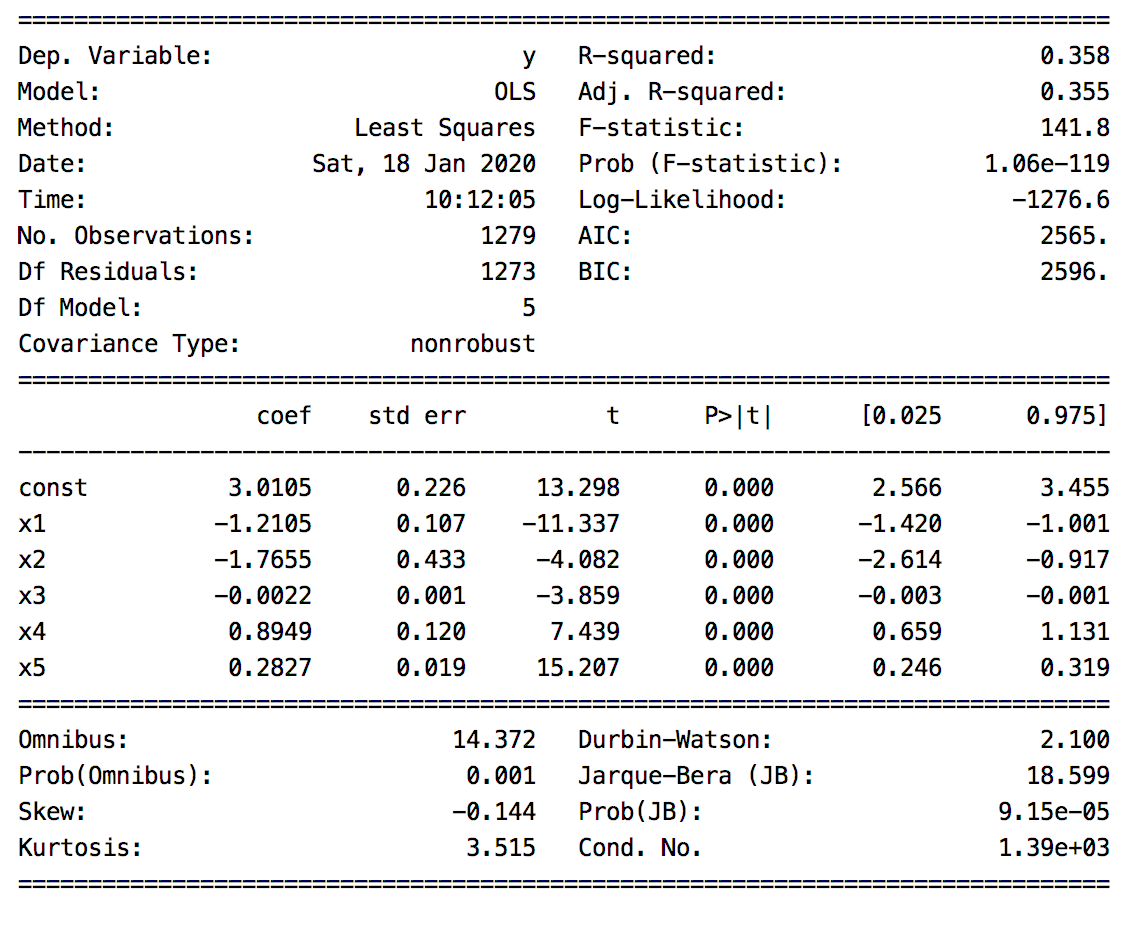
Example : Multiple Linear Regression for Wine Quality: Model B

In our first version of the model in Example 5 we found that coefficients for x1 (fixed acidity), x3 (citric acid), x4 (residual sugar), x6 (free sulfur dioxide), x8 (density) and x9 (pH) are statistically insignificant. The finding suggests that we can eliminate these variables from our model. Adjust the example code so our predictor subset excludes the statistically insignificant variables.

|  |
| --- |
| X = dataset[[ 'volatile acidity', 'chlorides', 'total sulfur dioxide',  'sulphates','alcohol']].values |

Our model summary actually shows a coefficient of determination, , of 0.3*58* which is lower than in model A in Example 5.

Figure : Output from Model B



Warnings:

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

[2] The condition number is large, 1.39e+03. This might indicate that there are

strong multicollinearity or other numerical problems.

Root Mean Squared Error: 0.6259157889490549

All predictor variables in the second model appear to be statistically significant.

### Collinearity

Both Example 5 and Example 6 are warning us about collinearity. Collinearity refers to unwanted interdependence between predictor variables. Ideally predictor variables are independent of each other. If predictor variables are not independent of each other a small change in input can have a large effect on the output. This could mean the RMSE will be higher and predictions will be less reliable.

We will talk more about collinearity later. As far as this wine quality model is concerned, collinearity could adversely affect the model. The high collinearity should be documented and explained to a wine quality expert who understands the chemical relationships being studied.

Sometimes high collinearity is acceptable. For example, SAT and LSAT scores on college entrance exams are highly correlated. Both test results indicate that the applicant is prepared for a rigorous academic program. Collinearity, could be high for these predictor variables yet we may still want both variables in the model.

Other times, high collinearity may be unacceptable. Collinearity can be a real problem for complex models with many predictor variables. For example, a stock portfolio may rely on a complex inter-relationship between sectors. A portfolio model may be incorrectly constructed under the assumption that oil and housing prices rise together yet an unexpected change in one of these sectors could have a detrimental effect on the prediction.

In any case, it is important to document collinearity. The model validation and RMSE will help to understand how collinearity affects the outcome of a model. It is also important to understand the relationships between the variable models to judge whether collinearity is a factor that needs to be addressed.

We will discuss collinearity more later.

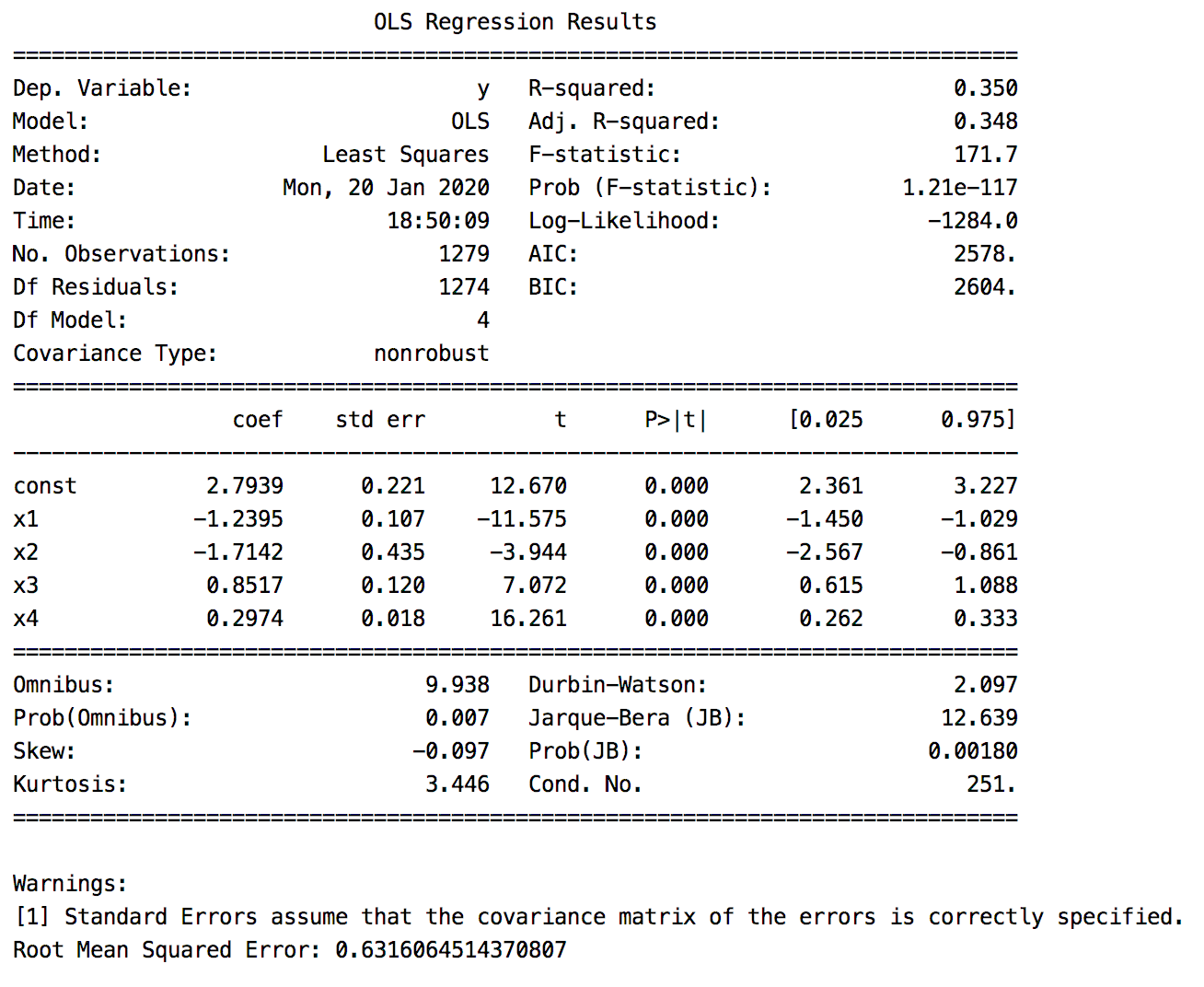
Example : Multiple Linear Regression for Wine Quality: Model C

The coefficient for X3 (total sulfur dioxide) of -0.0022 is so small it really does nothing to contribute to the target variable. This example runs the regression again but with only 'volatile acidity', 'chlorides', 'sulphates' and 'alcohol' as predictor variables. To build this example, replace the definition for X in Example 6 with this version and then run the regression again:

|  |
| --- |
| X = dataset[[ 'volatile acidity', 'chlorides', 'sulphates','alcohol']].values |

\

The results for model C have not changed very much when compared to model B but the multicollinearity warning has disappeared.



#### Comparison of Models

We can observe a summary of statistical measures for all models in Table 4. Model C is the simplest model but it also performs reasonably well compared to Model A and Model B. The RMSE is expected to increase with less predictor variables but the increased deviation for Model C is small at 0.006 greater than for Model B. The BIC score for model C is also second place but very close to being most favourable.

Table : Model Comparison

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Variables | #Vars |  |  | AIC | BIC | RMSE |
| A | fixed acidity, volatile acidity, citric acid, residual sugar,  chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol | 11 | 0.365 | 0.360 | 2562 | 2624 | 0.620 |
| B | volatile acidity, chlorides, total sulfur dioxide, sulphates, alcohol | 5 | 0.358 | 0.355 | 2565 | 2596 | 0.626 |
| C | volatile acidity, chlorides, sulphates, alcohol | 4 | 0.350 | 0.348 | 2578 | 2604 | 0.632 |

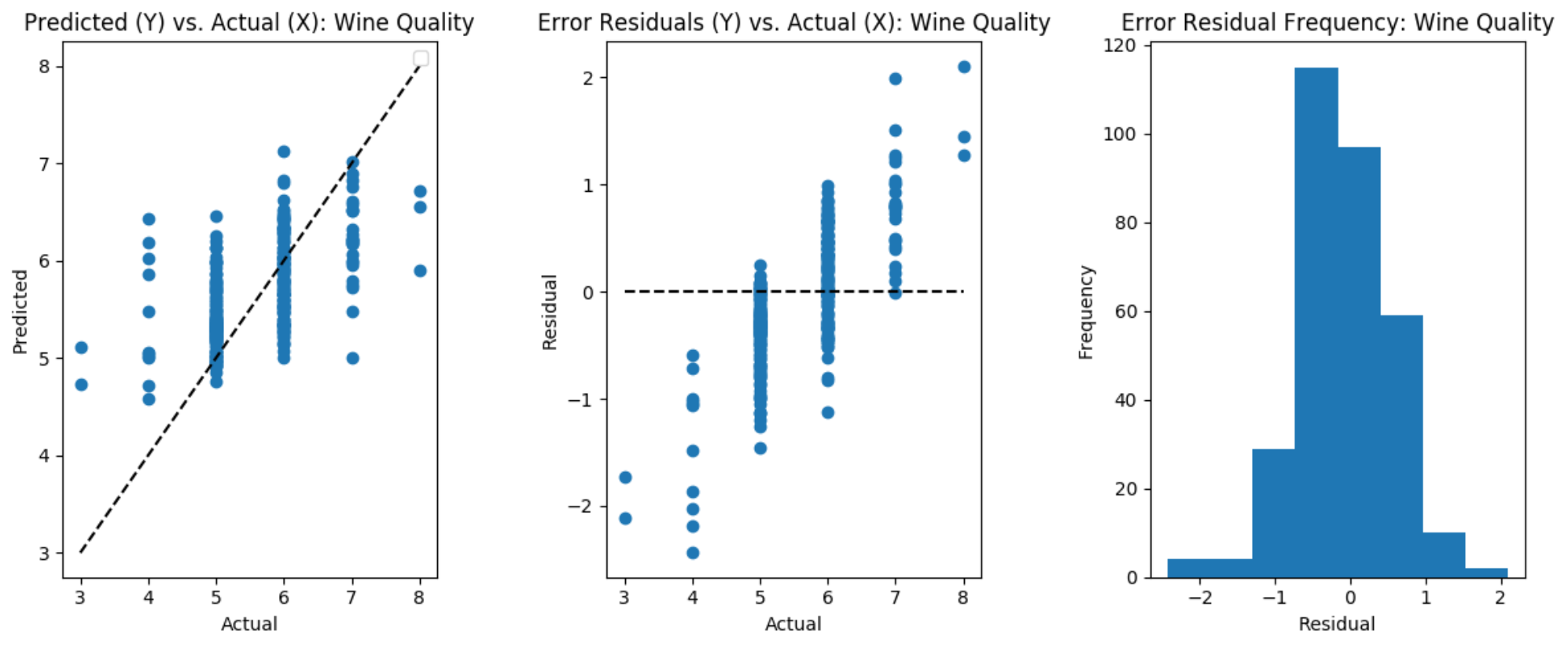
### Visualizing Actual, Predicted and Error Residuals

Another important step in validating the model involves visualization of actual values and residual error values against predicted values.

Example : Graphing Actual, Predicted and Residual Values

This section includes important visualizations to help understand our wine quality model. Figure 7 highlights several problems with the model. Through visualization we can see the model is really only useful for predicting when the quality is 6. It tends to over predict quality for lower actual quality values. The model under estimates higher quality wines. As well, very little data exists for higher and lower quality ratings.

Figure : (a) Predicted vs. Actual Wine Quality (b) Error Residuals vs. Actual Wine Quality (c) Error Residual Frequency



Here is the code that was used to draw these validation plots. This code can be added to the end of the regression code:

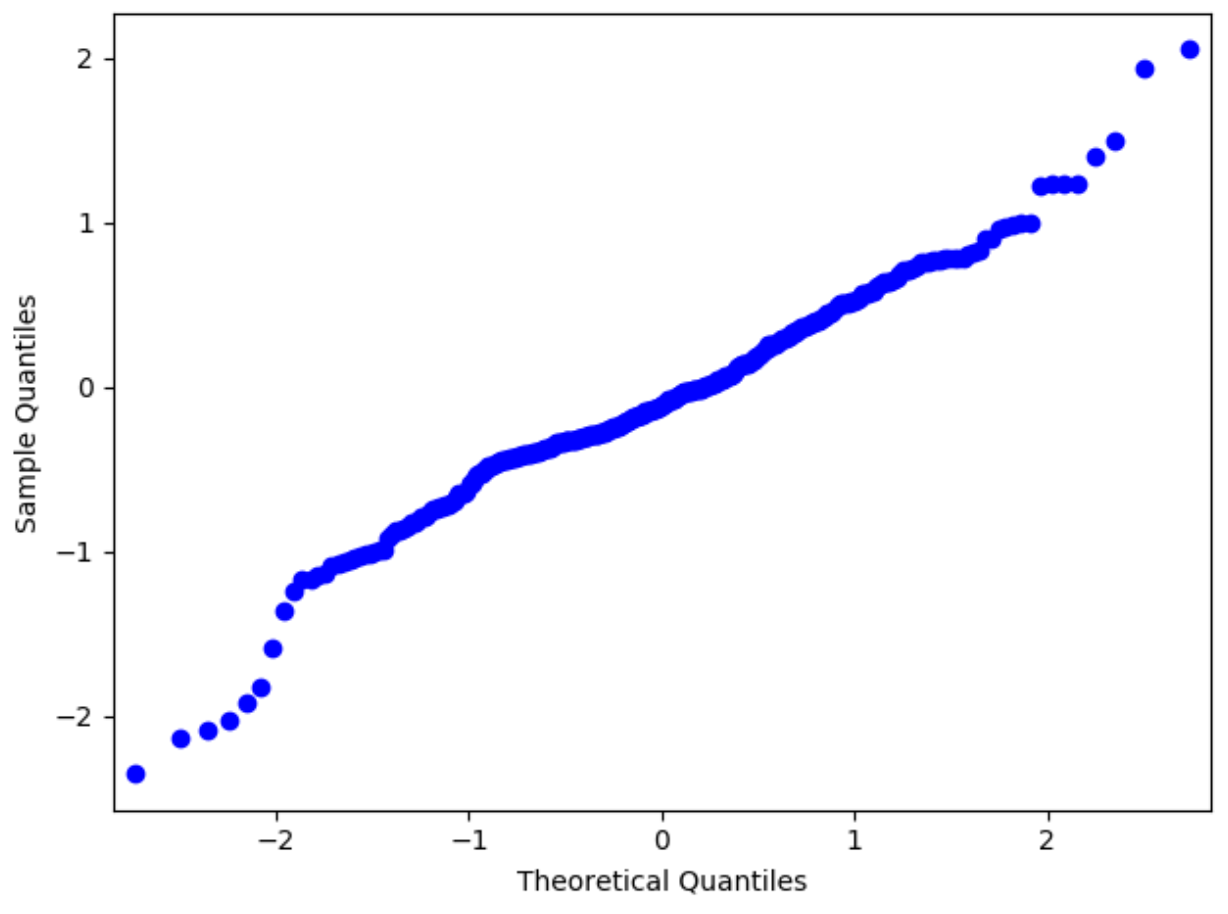
|  |
| --- |
| import matplotlib.pyplot as plt  def plotPredictionVsActual(plt, title, y\_test, predictions):  plt.scatter(y\_test, predictions)  plt.legend()  plt.xlabel("Actual")  plt.ylabel("Predicted")  plt.title('Predicted (Y) vs. Actual (X): ' + title)  plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'k--')  def plotResidualsVsActual(plt, title, y\_test, predictions):  residuals = y\_test - predictions  plt.scatter(y\_test, residuals, label='Residuals vs Actual')  plt.xlabel("Actual")  plt.ylabel("Residual")  plt.title('Error Residuals (Y) vs. Actual (X): ' + title)  plt.plot([y\_test.min(), y\_test.max()], [0, 0], 'k--')  def plotResidualHistogram(plt, title, y\_test, predictions, bins):  residuals = y\_test - predictions  plt.xlabel("Residual")  plt.ylabel("Frequency")  plt.hist(residuals, label='Residuals vs Actual', bins=bins)  plt.title('Error Residual Frequency: ' + title)  plt.plot()  def drawValidationPlots(title, bins, y\_test, predictions):  # Define number of rows and columns for graph display.  plt.subplots(nrows=1, ncols=3, figsize=(12,5))    plt.subplot(1, 3, 1) # Specfy total rows, columns and image #  plotPredictionVsActual(plt, title, y\_test, predictions)  plt.subplot(1, 3, 2) # Specfy total rows, columns and image #  plotResidualsVsActual(plt, title, y\_test, predictions)  plt.subplot(1, 3, 3) # Specfy total rows, columns and image #  plotResidualHistogram(plt, title, y\_test, predictions, bins)  plt.show()  BINS = 8  TITLE = "Wine Quality"  drawValidationPlots(TITLE, BINS, y\_test, predictions) |

### Quantile-Quantile Plot of Residuals

QQ plots of residuals map quantile probability for the residuals relative to the quantiles of a classic normal distribution. Ideally the output of a QQ plot should be a straight line. Dips at both sides of the plot indicate overly thin tails and more concentration at the center. The residuals do not follow a normal distribution.

Figure : QQ Plot of Error Residuals

The QQ plot highlights normality.



To draw the QQ plot in Figure 7, add this code after the code in Example 6.

|  |
| --- |
| from statsmodels.graphics.gofplots import qqplot  def plotQQ(plt, title, y\_test, predictions):  residuals = y\_test - predictions  plt.title("Quantile-Quantial Residuals - " + title)  qqplot(residuals)  plotQQ(plt, TITLE, y\_test, predictions)  plt.show() |

### Model Selection

Currently I prefer model C since it is the simplest model and since it performs relatively close to model B which is only slightly better statistically. The choice is still not ideal though. Like all models, Model C is really only useful for predicting when the quality is 6. It tends to over predict quality for lower actual quality values. The model under estimates higher quality wines.

### Future Considerations

More data is needed particularly for higher and lower quality wines to enable a better model. Transformations before modeling may also help to normalize the predictor variables. A different type of model may be needed since the target variable is discrete. Possibly other properties related to wine quality need to be considered for the model if they can be obtained. The wine quality rating is subjective as well so predictions can be very difficult. It may not be possible to improve the prediction much.

### Model Equation

The model is generated from the coefficient output from the regression in Example 7. Figure 9 shows the coefficients from our last regression for Model C translate into an equation.

Figure : Model Coefficients from the Selected Model

Translating our coefficient output to and equation implies;

|  |  |
| --- | --- |
|  |  |

**= 2.7939 1.7142 +**

**+ 0.2974 + Error**

### Model Interpretation

Whenever possible, it helps to interpret the model in descriptive terms. Based on our equation, we can say the model starts with a default quality rating of 2.7939. Reductions of volatile acidity and chlorides improve the rating. Increases in sulphates and alcohol also improve the rating.

## Data Scoring

You will have noticed when evaluating our model that we calculated the RMSE based on predicted values.

predictions = model.predict(X\_test)

np.sqrt(metrics.mean\_squared\_error(y\_test, predictions))

Here is a sample of the test after running the instructions:

print(X\_test)

print(y\_test)

print(predictions)

#### X test

[[ 1. 0.47 0.171 0.76 10.8 ]

[ 1. 0.82 0.095 0.53 9.6 ]

[ 1. 0.29 0.063 0.84 11.7 ]

#### Y test actual

[6 5 7

#### Y predicted

[5.77777936 4.92142727 6.52185395

Example : Manual Predictions

We can build our model to make predictions from the equation that we generated earlier and test out our model.

= 2.7939 1.7142

+ + 0.2974 + Error

|  |
| --- |
| def getWineQuality(actualQuality, volatileAcidity, chlorides, sulphates, alcohol):  wineQuality = 2.7939 - 1.2395 \*volatileAcidity - 1.7142 \*chlorides + \  + 0.8517\*sulphates + 0.2974\*alcohol;  print("Wine Quality actual: " + str(actualQuality) + " predicted: "  + str(wineQuality))  getWineQuality(6, 0.47, 0.171, 0.76, 10.8)  getWineQuality(5, 0.82, 0.095, 0.53, 9.6)  getWineQuality(7, 0.29, 0.063, 0.84, 11.7) |

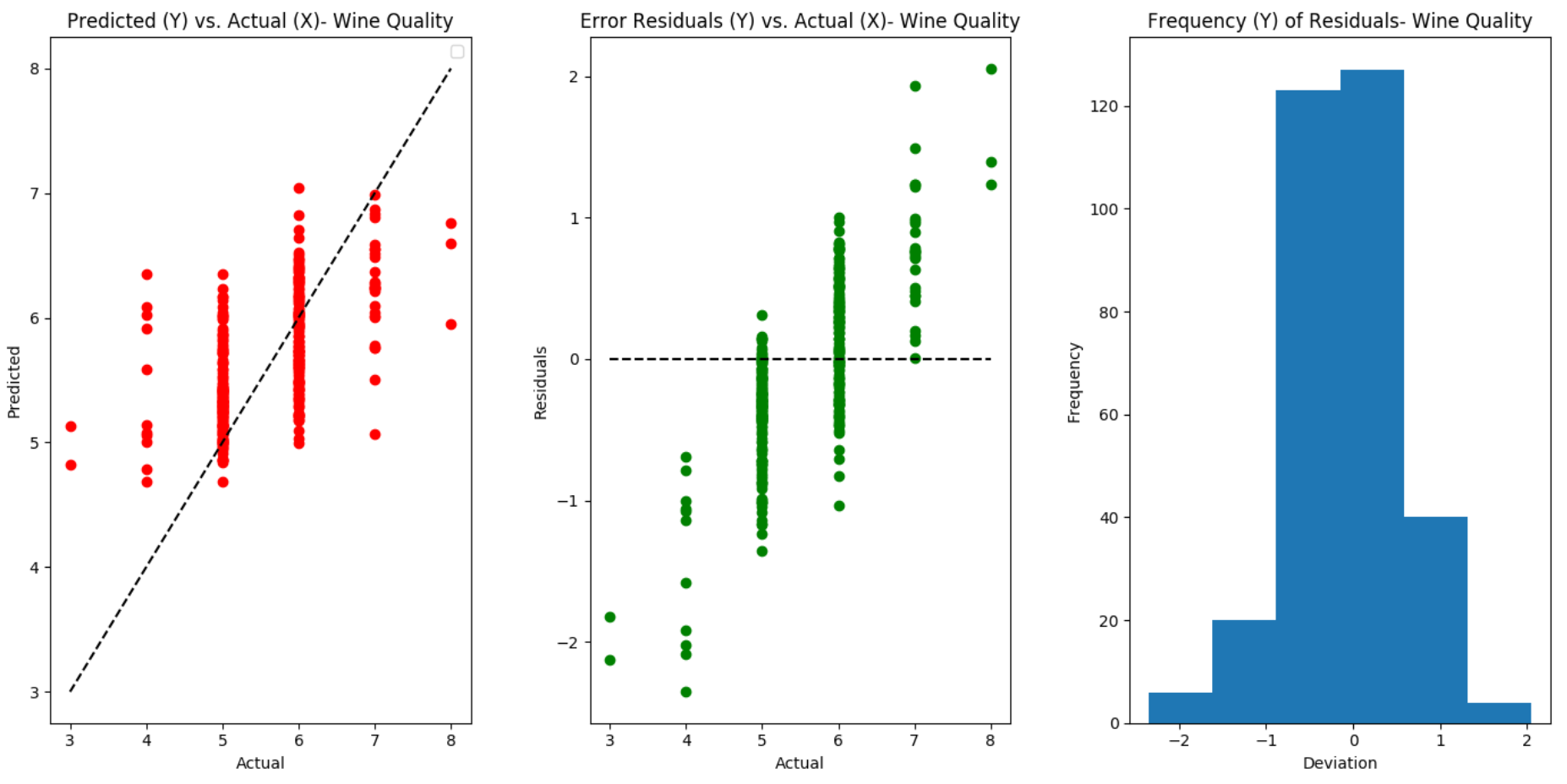
The output shows a fairly accurate series of estimates:

|  |
| --- |
| Wine Quality actual: 6 predicted: 5.7774188  Wine Quality actual: 5 predicted: 4.921101999999999  Wine Quality actual: 7 predicted: 6.5214584 |

## Anscombe’s Quartet

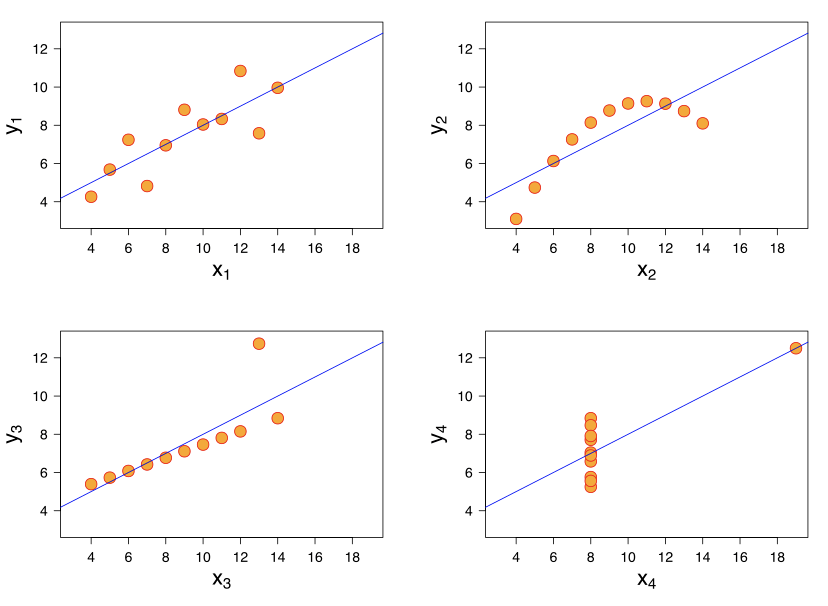
If we consider the predicted versus actual wine quality results we can observe how poorly the straight line fits our data (see Figure 10).

Figure : Poor Fitting Straight Line for Wine Quality Predictions



This poor fitting straight line demonstrates the dilemma that is highlighted by the infamous Anscombe’s Quartet as shown in Figure 11. The second, third and fourth plots in Figure 11 highlight situations where the best fitting straight line fails to serve as a predictor.

Figure : Anscombe’s Quartet



Please include your code for the at the end of this document. Keep it neat and tidy for your records and reference later. **(5 marks)**

Exercise (1 mark)

Using the USA Housing data set show the first three rows of the DataFrame:

|  |
| --- |
|  |

Exercise (1 mark)

Show a statistical summary for the numerical columns:

|  |
| --- |
|  |

Exercise (2 marks)

Show a correlation heatmap for the numeric columns.

|  |
| --- |
|  |

Exercise (4 marks)

Show the distributions using Weka for the numeric data.

|  |
| --- |
|  |

Exercise (8 marks)

Perform linear regression to create three separate models that predict price. Ignore multicollinearity. The first model can contain all of the fields for predictor variables except *Address* which is non-numeric and *Price* which is our target variable. Show a model comparison table like the one in table 4 for all three models.

|  |
| --- |
|  |

Exercise (4 marks)

Explain which model you are selecting and why.

|  |
| --- |
|  |

Exercise (4 marks)

Show the OLS Regression Results output for your selected model.

|  |
| --- |
|  |

Exercise (6 marks)

For the selected model, manually compute the AIC and BIC scores but use the log-likelihood score that is included in the output to perform these calculations. Computing the LL manually is difficult and not worth the effort.

|  |
| --- |
|  |

Exercise (6 marks)

Graph the (a) Predicted vs. Actual Wine Quality (b) Error Residuals vs. Actual Wine Quality (c) Error Residual Frequency plots for the selected model. The scatter should indicate a much better fit than the model developed for the wine quality set. Show them here:

|  |
| --- |
|  |

Exercise (2 marks)

Draw a QQ plot for the residuals. Show it here:

|  |
| --- |
|  |

Exercise (6 marks)

Developer your equation for the selected model and write an interpretation in English for it.

|  |
| --- |
|  |

Exercise (6 marks)

Manually calculate predictions for three separate samples based on your chosen model equation.

|  |
| --- |
|  |

Here is some code to start with:

|  |
| --- |
| import pandas as pd  from sklearn.model\_selection import train\_test\_split  import statsmodels.api as sm  PATH = "/Users/pm/Desktop/DayDocs/2019\_2020/PythonForDataAnalytics/workingData/"  CSV\_DATA = "USA\_Housing.csv"  dataset = pd.read\_csv(PATH + CSV\_DATA,  skiprows=1, # Don't include header row as part of data.  encoding = "ISO-8859-1", sep=',',  names=('Avg. Area Income','Avg. Area House Age', 'Avg. Area Number of Rooms','Avg. Area Number of Bedrooms', "Area Population", 'Price', "Address"))  # Show all columns.  pd.set\_option('display.max\_columns', None)  pd.set\_option('display.width', 1000)  print(dataset.head()) |