Rauen_Assignment 5

Will Rauen

2024-04-11

Setup

For this invesatigation, the tidyverse, psych, and reshpae2 packages will be used to answer a research question related to Principal Component Analysis (PCA).

Data

```
wine_data <- read.csv("/Users/williamrauen/Desktop/DS 3100/Assignments/Assignment 5/Data/WineQT.csv")

dim(wine_data)
```

```
## [1] 1143   13
```

This data used for this investigation comes from an online dataset found on kaggle.com at:

https://www.kaggle.com/datasets/yasserh/wine-quality-dataset?resource=download

The dataset pertains to different aspects of wine such as its acidity, ph, alcohol percentage, quality, and sugar composition. There are 1143 rows in the dataset with each one representing a different wine with its own unique ID number, and 13 columns for each of its different aspects as mentioned before. Each of the 13 columns are continuous variables, and 13 is a high enough dimension to reduce, making this a dataset suitable for principal component analysis. Furthermore, there was no missingness in the data so we do not need to wrangle for missingness. We will make sure that the data is also adequate in the following section to ensure principal component analysis would be effective.

Data Adequacy

We will first use the Kaiser-Meyer-Olkin Test and Bartlett Test to determine id the data is adequate for principal component analysis

```
# Remove ID Column from Data Set
wine_test <- wine_data[,1:12]

#Kaiser-Meyer-Olkin Test
KMO(wine_test)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = wine_test)
## Overall MSA =  0.47
## MSA for each item = 
##        fixed.acidity     volatile.acidity          citric.acid 
##                 0.47                 0.58                 0.72 
##       residual.sugar            chlorides  free.sulfur.dioxide 
##                 0.24                 0.49                 0.50 
## total.sulfur.dioxide              density                   pH
##                 0.48                 0.39                 0.46
##            sulphates              alcohol              quality
##                 0.56                 0.31                 0.77
```

```r
#Bartlett Test
cortest.bartlett(wine_test)
```

```
## R was not square, finding R from data
```

```
## $chisq
## [1] 6349.014
##
## $p.value
## [1] 0
##
## $df
## [1] 66
```

Since the KMO test revealed a score less than 0.5, we will have to adjust our data to make it compatable for PCA. We can do this by removing variables with lower MSA scores, which in this case are the residual_sugar, chlorides, and alcohol variable, and the rerun the same tests again.

```r
#Remove ID column, Residual Sugar var., Alcohol var., and Chloride var.

wine_new <- wine_data[,c(1,2,3,6,7,8,9,10,12)]

#Kaiser-Meyer-Olkin Test
KMO(wine_new)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = wine_new)
## Overall MSA =  0.6
## MSA for each item =
##        fixed.acidity     volatile.acidity          citric.acid
##                 0.59                 0.67                 0.73
##  free.sulfur.dioxide total.sulfur.dioxide              density
##                 0.50                 0.44                 0.53
##                   pH            sulphates              quality
##                 0.67                 0.67                 0.58
```

```r
#Bartlett Test
cortest.bartlett(wine_new)
```

```
## R was not square, finding R from data
```

```
## $chisq
## [1] 4154.302
##
## $p.value
## [1] 0
##
## $df
## [1] 36
```

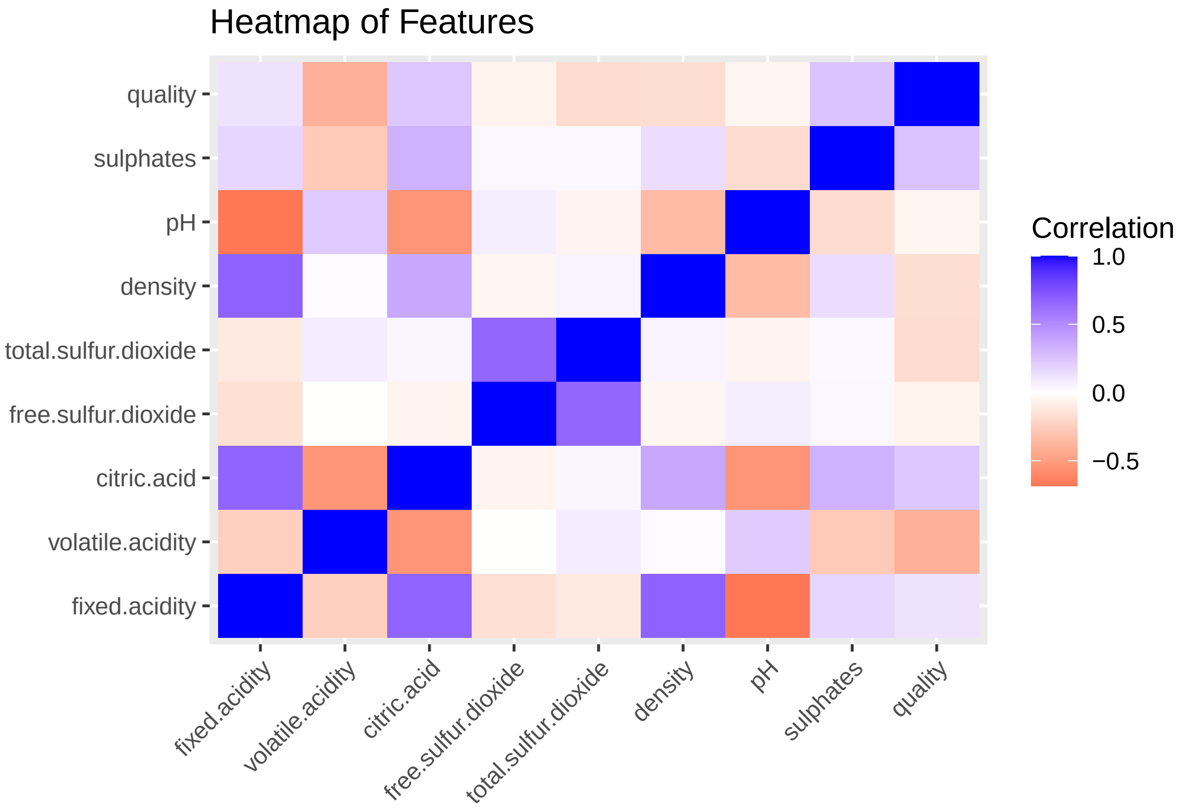
After re-running tests of data adequacy, we obtain a KMO test with a value of 0.60 which meets the minimum requirements for PCA. Furthermore, the Bartletts test tells us that we can reject the null hypothesis with a score extremely close to zero so the correlation matrix is not the identity. Even though we have a low KMO value, we do have a lot of data points so we should be good to move forward with PCA carefully.

Variables of Interest

There are now 9 variables of interest since one of the columns is the ID column which acts as the index and we removed the 3 other variables. The 9 variables of interest are fixed acidity, volatile acidity, citric acid content,density, total sulfur dioxide, free sulfur dioxide content, ph, sulphate content,and quality score. These are all continuous variables and are high enough dimension to attempt to meaningfully reduce to just 3, thus reinforcing its applicability to PCA.

We can also take a look at the heatmap of the correlation matrix to get a better visualization of the correlations between the variables we have.

```
wine_cor <- cor(wine_new, use = 'pairwise.complete')

#It can also be useful to visualize this using a heatmap
library(reshape2)
melted_wine_cor <- melt(wine_cor)
ggplot(data = melted_wine_cor, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  labs(x = '', y = '', fill = 'Correlation', title = 'Heatmap of Features') +
  scale_fill_gradient2(low = 'red', mid = 'white', high = 'blue', midpoint = 0) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Research Question

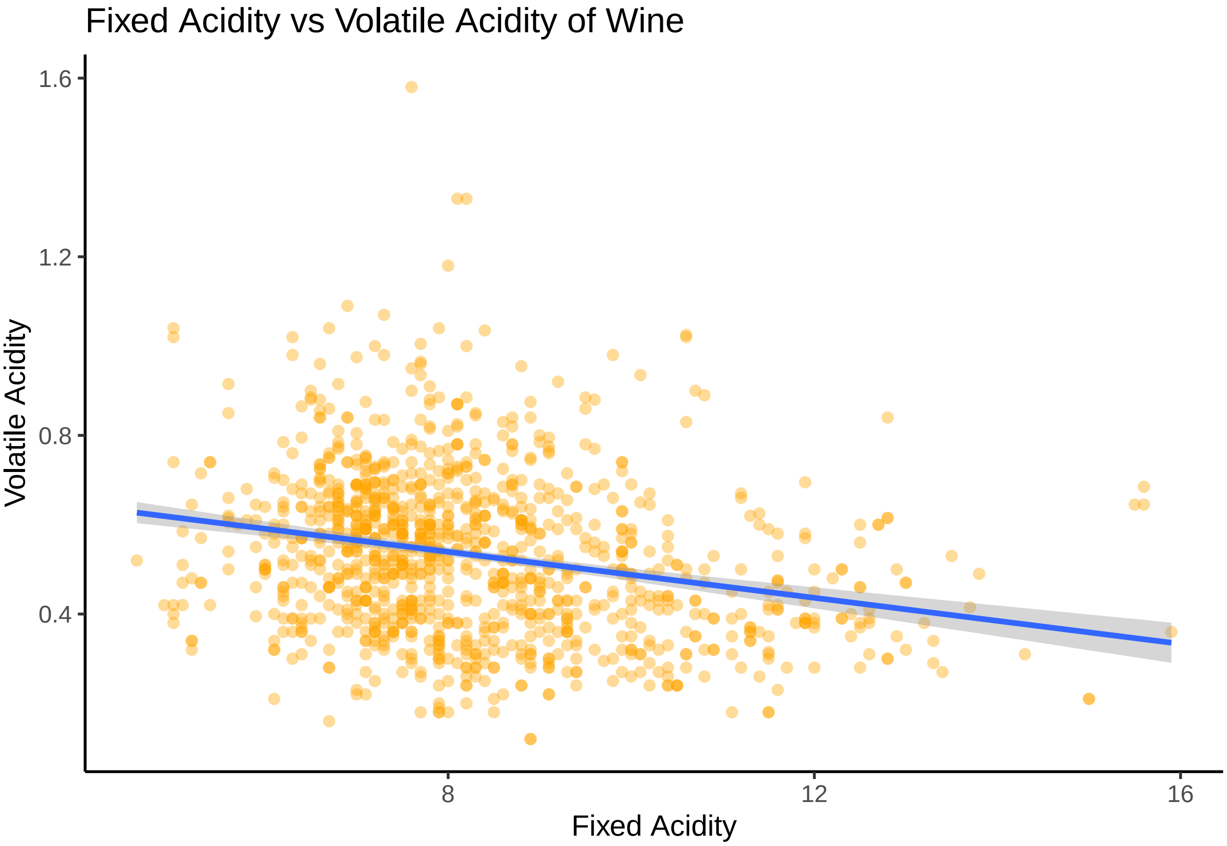
In this investigation, we will aim to determine if we can use PCA to successfully reduce the dimensionality of the data set to be reasonably explained by 3 components. If it is possible, we will also aim to see if there are any patterns between the variables that explain the 3 components.

Exploratory Data Analysis

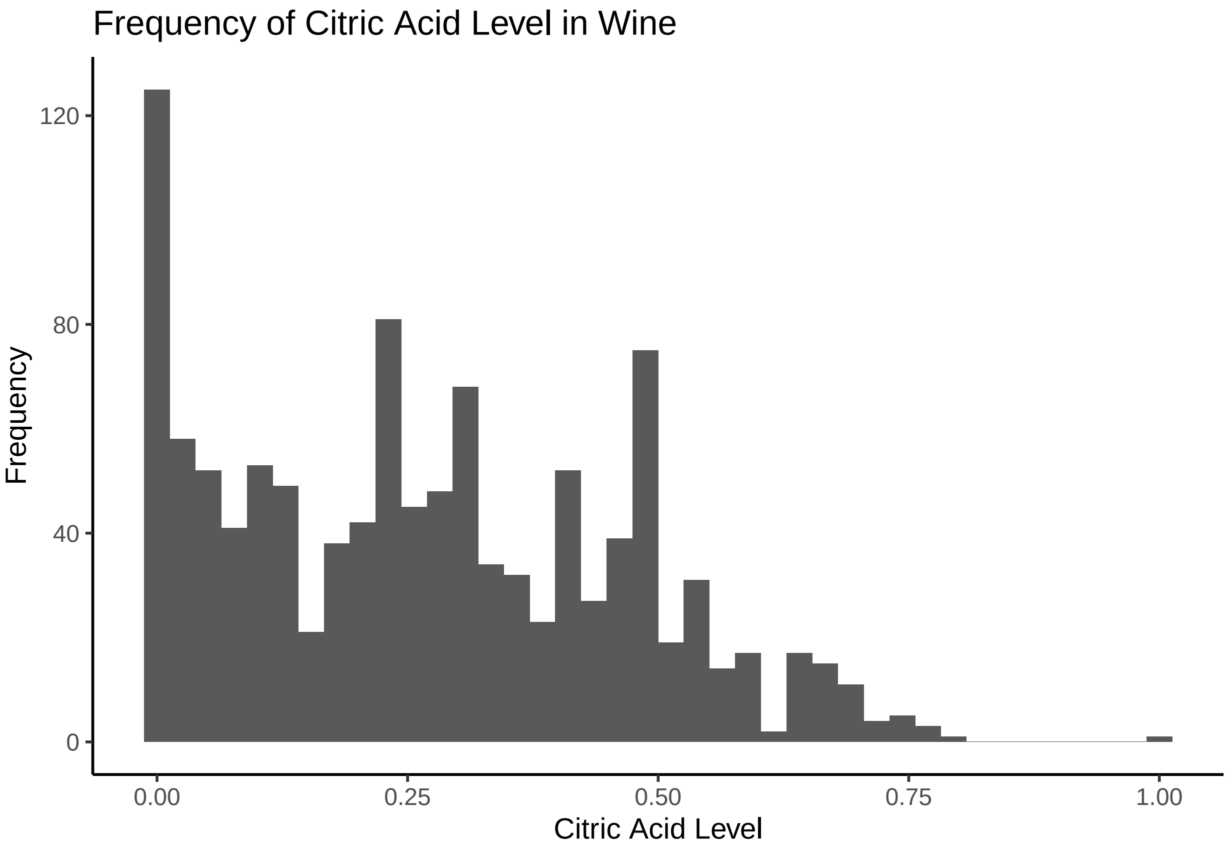
In this section, we will create multiple data visualizations to get a baseline understanding of our variables of interest.

```
# Acidity Scatter Plot
wine_new %>%
  ggplot(aes(x = fixed.acidity, y= volatile.acidity)) +
  geom_point(alpha = 0.4, color = "orange") +
  geom_smooth(method = lm, na.rm = TRUE) +
  theme_classic() +
  labs(title = "Fixed Acidity vs Volatile Acidity of Wine",
       x = "Fixed Acidity",
       y = "Volatile Acidity")
```

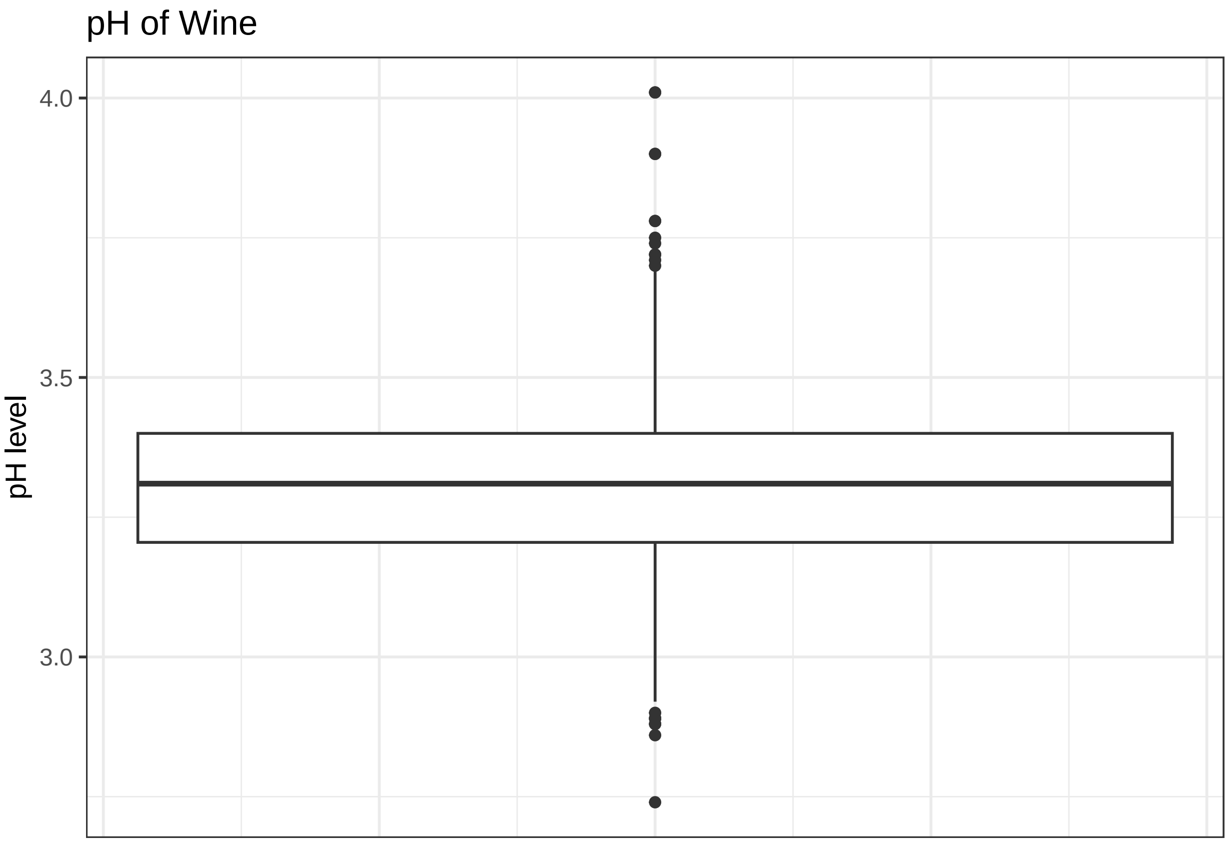
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Citric Acid Histogram
wine_new %>%
  ggplot(aes(x = citric.acid)) +
  geom_histogram(bins = 40) +
  theme_classic() +
  labs(title = "Frequency of Citric Acid Level in Wine",
       x = "Citric Acid Level",
       y = "Frequency")
```

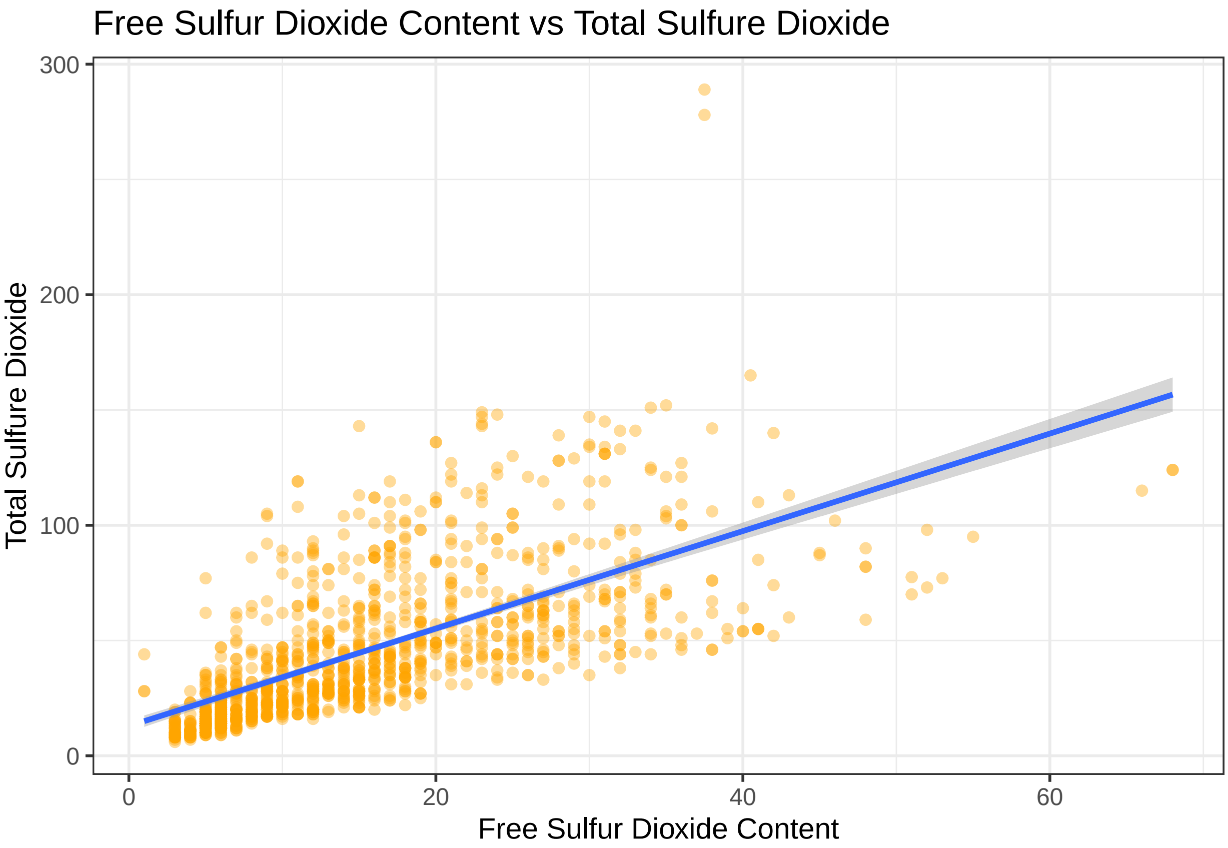


```
# pH Boxplot
wine_new %>%
  ggplot(aes(x = pH)) +
  geom_boxplot() +
  coord_flip()+
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  labs(title = "pH of Wine",
       x = "pH level")
```

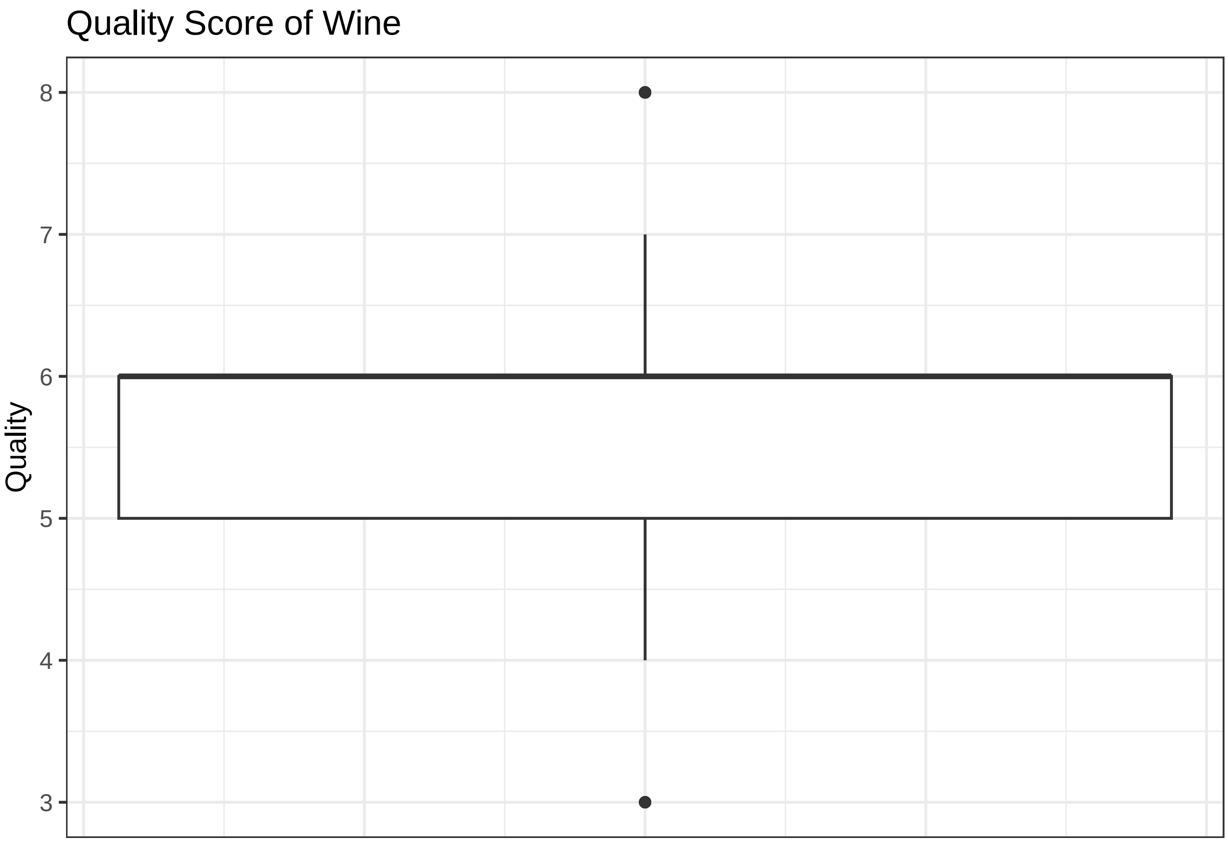


```
# Sulfur relationship Dot Plot
wine_new %>%
  ggplot(aes(x = free.sulfur.dioxide, y= total.sulfur.dioxide)) +
  geom_point(alpha = 0.4, color = "orange") +
  geom_smooth(method = lm, na.rm = TRUE) +
  theme_bw() +
  labs(title = "Free Sulfur Dioxide Content vs Total Sulfure Dioxide",
       x = "Free Sulfur Dioxide Content",
       y = "Total Sulfure Dioxide")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Quality Wine Boxplot
wine_new %>%
  ggplot(aes(x = quality)) +
  geom_boxplot() +
  coord_flip()+
  theme_bw() +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  labs(title = "Quality Score of Wine",
       x = "Quality")
```



Through our data exploration, we found that variables with similar relations (acidity, sulfur dioxide) could have relationships but do not seem too strongly supported. We also see that there are some statistical outliers on the boxplots, but those will be kept since they do not appear to influence too heavily.

Model Selection

In this section, we will conduct principal component analysis and use visualizations to determine how many principal components we should use.

```
#Extracting Eignevalues
wine_pca <- principal(wine_new, rotation = 'none')

#Printing Eigenvalues
print(sort(wine_pca$values, decreasing = T))
```

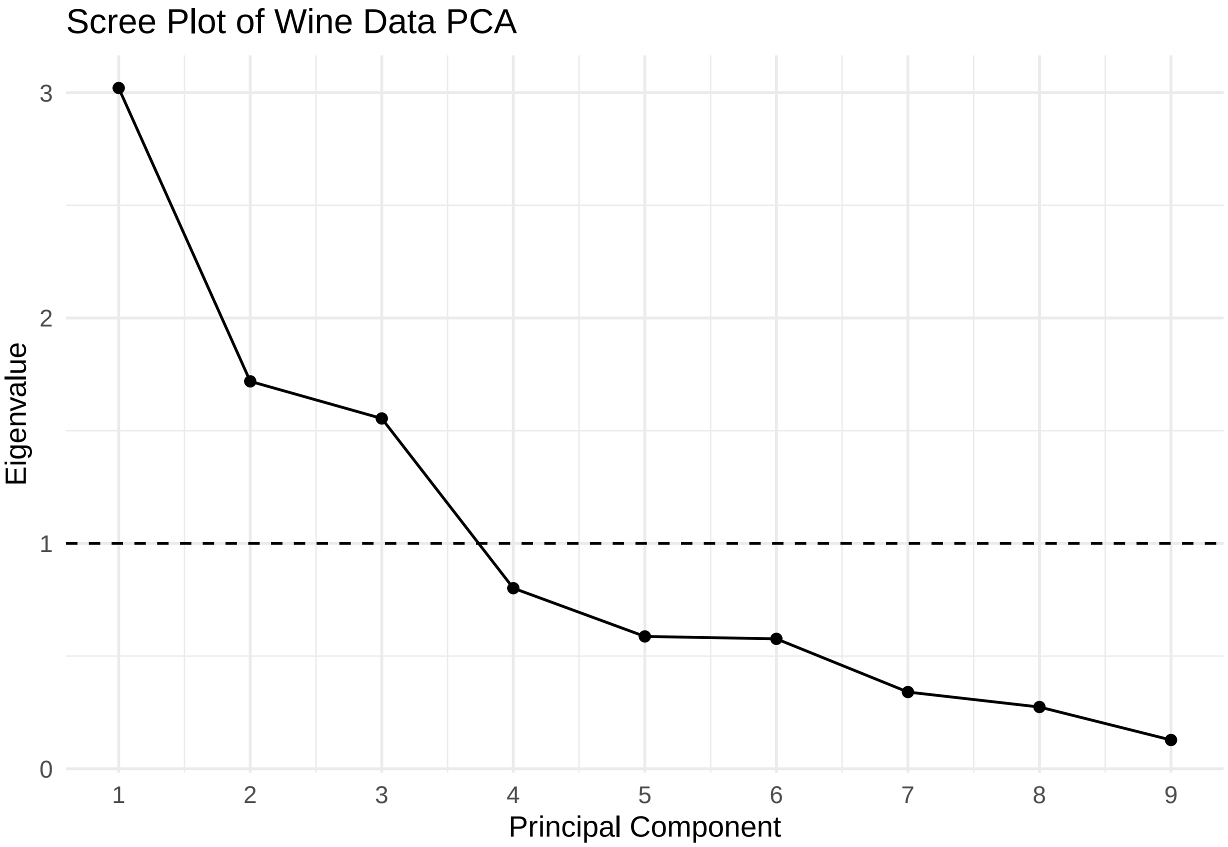
```
## [1] 3.0206165 1.7189173 1.5541966 0.8011970 0.5870978 0.5763667 0.3402694
## [8] 0.2738415 0.1274971
```

Using the PCA function, we found eigenvalues of each of our predictor variables. According to the Kaiser-Guttman rule, we should have as many principal components as variables who have these PCA values above 1. In this case, we find that we have 3 of such values above 1, but further analysis will be done.

Another way we can visualize how many components to use, is by creating a scree plot of the eigenvalues.

```
# Scree Plot
wine_scree <- data.frame(PCA = 1:length(wine_pca$values),
                         Eigenvalue = wine_pca$values)

# Scree plot Visualization
```

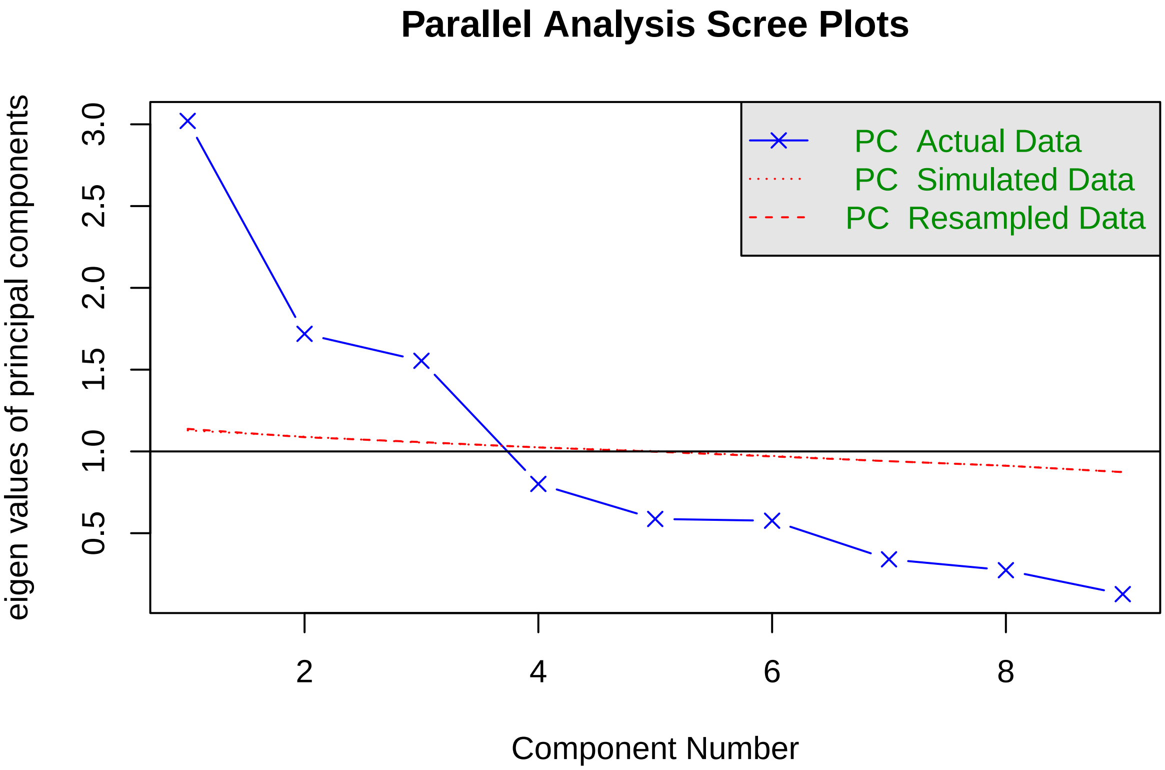
```
ggplot(wine_scree, aes(x = PCA, y = Eigenvalue)) +
  geom_line() +
  geom_point() +
  geom_hline(yintercept=1, linetype = 'dashed') +
  scale_x_continuous(breaks=c(1:24)) +
  theme_minimal() +
  labs(x = 'Principal Component',
       title = "Scree Plot of Wine Data PCA")
```



The Scree plot shows us to keep 3 principal components as you want to retain the components above the "bend". The bend on this scree plot occurs at the PCA = 4 value which supports our implication from above. However, we should also be aware of a bend at PCA = 3 that could be considered so we should move forward with another visualization.

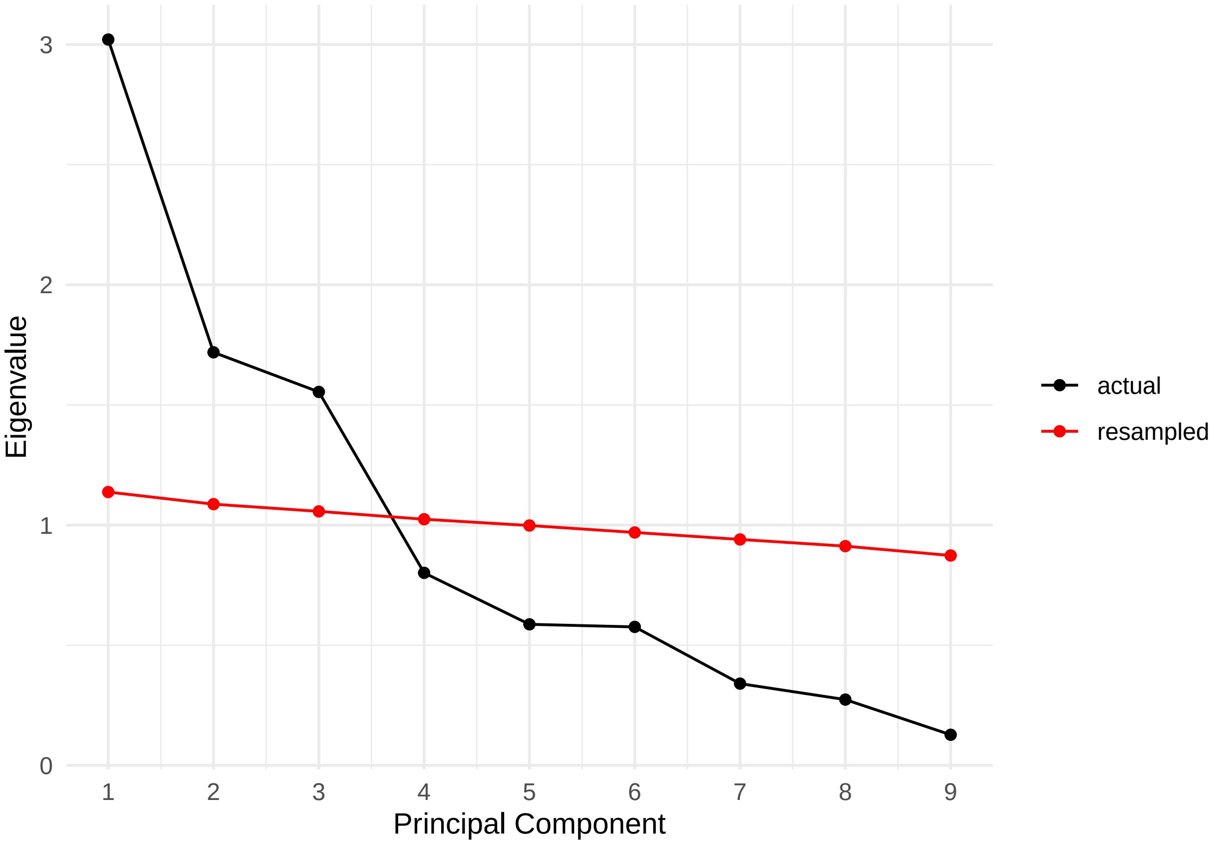
Our last visualization to determine the number principal components will be through the use of parallel analysis

```
# Set Seed for replicability
set.seed(123)

# Parallel Analysis Function
wine_pa <- fa.parallel(wine_new, fa = 'pc')
```



```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  3
```

```
# Plotting the Parallel Analysis
wine_pa_plot <- data.frame(PC = c(1:length(wine_pa$pc.values),
                                  1:length(wine_pa$pc.values)),
                      type = c(rep('actual', times = 9),
                               rep('resampled', times = 9)),
                      Eigenvalue = c(wine_pa$pc.values, wine_pa$pc.simr))

ggplot(wine_pa_plot,aes(x = PC, y = Eigenvalue, col = type)) +
  geom_line()+
  geom_point() +
  scale_x_continuous(breaks=c(1:24)) +
  theme_minimal() +
  labs(x = 'Principal Component', y = 'Eigenvalue') +
  scale_color_manual(name = '', values = c('black', 'red'))
```



Thus, parallel analysis would indicate using 3 components in the PCA. Thus, we will move forward with 3 components.

Rotation & PCA Scores

As aforementioned, we will be moving forward with a PCA with 3 components. However we can decide if we want to rotate our data in order to better separate our predictors amongst our components.

```
# Original PCA
wine_pca_no <- principal(wine_new, nfactors = 3, rotate = 'none')
wine_pca_no
```

```
## Principal Components Analysis
## Call: principal(r = wine_new, nfactors = 3, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                        PC1   PC2   PC3   h2   u2 com
## fixed.acidity         0.88  0.09 -0.31 0.88 0.12 1.3
## volatile.acidity     -0.54  0.21 -0.57 0.66 0.34 2.3
## citric.acid           0.86  0.07  0.15 0.77 0.23 1.1
## free.sulfur.dioxide  -0.17  0.76  0.45 0.82 0.18 1.7
## total.sulfur.dioxide -0.09  0.87  0.29 0.84 0.16 1.2
## density               0.59  0.32 -0.51 0.71 0.29 2.5
## pH                   -0.74 -0.19  0.17 0.61 0.39 1.2
## sulphates             0.43  0.00  0.41 0.36 0.64 2.0
## quality               0.30 -0.44  0.60 0.65 0.35 2.4
##
##                        PC1  PC2  PC3
## SS loadings           3.02 1.72 1.55
## Proportion Var        0.34 0.19 0.17
## Cumulative Var        0.34 0.53 0.70
## Proportion Explained  0.48 0.27 0.25
## Cumulative Proportion 0.48 0.75 1.00
##
## Mean item complexity =  1.7
## Test of the hypothesis that 3 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  587.98  with prob <  3.9e-118
##
## Fit based upon off diagonal values = 0.93
```

```r
#Orthogonal PCA
wine_pca_or <- principal(wine_new, nfactors = 3, rotate = 'varimax')
wine_pca_or
```

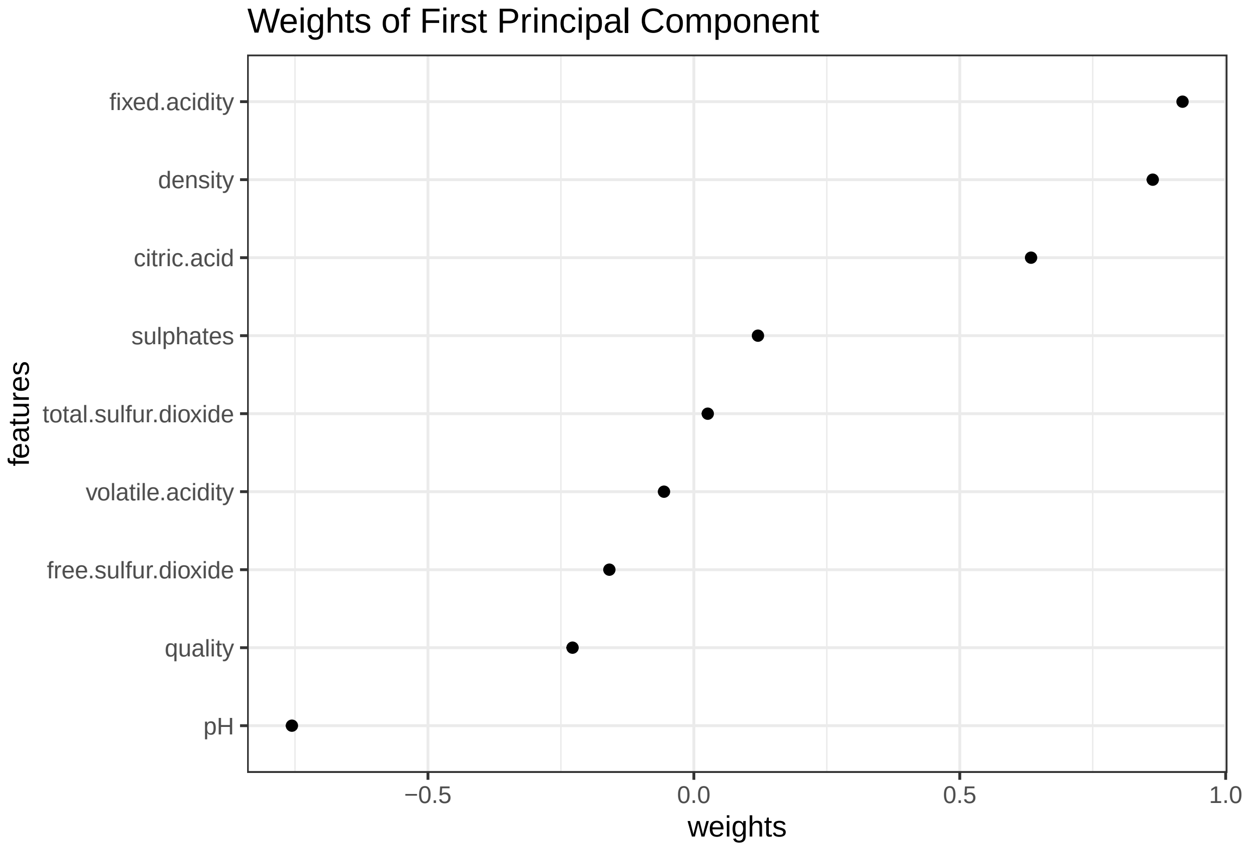
```
## Principal Components Analysis
## Call: principal(r = wine_new, nfactors = 3, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                        RC1   RC3   RC2   h2   u2 com
## fixed.acidity         0.91  0.18 -0.14 0.88 0.12 1.1
## volatile.acidity     -0.15 -0.80  0.02 0.66 0.34 1.1
## citric.acid           0.68  0.55  0.03 0.77 0.23 1.9
## free.sulfur.dioxide  -0.11  0.04  0.90 0.82 0.18 1.0
## total.sulfur.dioxide  0.05 -0.08  0.91 0.84 0.16 1.0
## density               0.82 -0.20  0.02 0.71 0.29 1.1
## pH                   -0.76 -0.20 -0.02 0.61 0.39 1.1
## sulphates             0.19  0.56  0.12 0.36 0.64 1.3
## quality              -0.14  0.77 -0.18 0.65 0.35 1.2
##
##                        RC1  RC3  RC2
## SS loadings           2.63 1.96 1.71
## Proportion Var        0.29 0.22 0.19
## Cumulative Var        0.29 0.51 0.70
## Proportion Explained  0.42 0.31 0.27
## Cumulative Proportion 0.42 0.73 1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 3 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  587.98  with prob <  3.9e-118
##
## Fit based upon off diagonal values = 0.93
```

```r
#Oblique PCA
wine_pca_ob <- principal(wine_new, nfactors = 3, rotate = 'promax')
wine_pca_ob
```

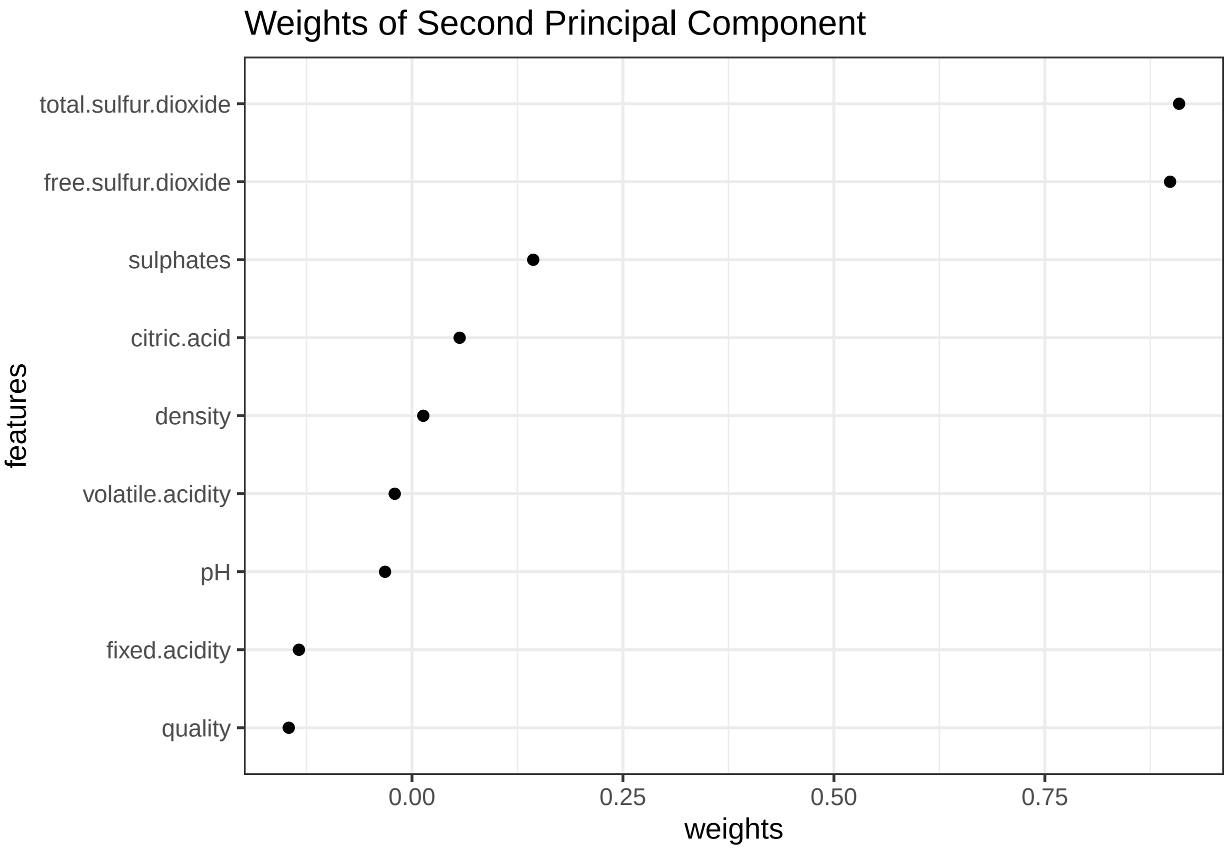
```
## Principal Components Analysis
## Call: principal(r = wine_new, nfactors = 3, rotate = "promax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                        RC1   RC3   RC2   h2   u2 com
## fixed.acidity         0.92  0.05 -0.13 0.88 0.12 1.0
## volatile.acidity     -0.06 -0.80 -0.02 0.66 0.34 1.0
## citric.acid           0.63  0.46  0.06 0.77 0.23 1.8
## free.sulfur.dioxide  -0.16  0.04  0.90 0.82 0.18 1.1
## total.sulfur.dioxide  0.03 -0.11  0.91 0.84 0.16 1.0
## density               0.86 -0.33  0.01 0.71 0.29 1.3
## pH                   -0.76 -0.09 -0.03 0.61 0.39 1.0
## sulphates             0.12  0.54  0.14 0.36 0.64 1.2
## quality              -0.23  0.82 -0.15 0.65 0.35 1.2
##
##                        RC1  RC3  RC2
## SS loadings           2.66 1.94 1.69
## Proportion Var        0.30 0.22 0.19
## Cumulative Var        0.30 0.51 0.70
## Proportion Explained  0.42 0.31 0.27
## Cumulative Proportion 0.42 0.73 1.00
##
##  With component correlations of
##      RC1   RC3   RC2
## RC1 1.00  0.26  0.04
## RC3 0.26  1.00 -0.01
## RC2 0.04 -0.01  1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 3 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  587.98  with prob <  3.9e-118
##
## Fit based upon off diagonal values = 0.93
```

After looking at each of the rotations of the PCA, it was interesting to note that orthogonal and oblique PCA had somewhat similar values and scores. Considering that I would think the sulfur dioxide variables would be correlated, and their scores were around 0.9 in the oblique and orthogonal rotations, I think the correct model is one of those two. Considering all of these components have some chemistry to do with one another like ph, density, acid levels, etc.., I think it would be safe to move forward with an oblique rotation which allows for some correlations across variables, but keeps the major correlations strong.

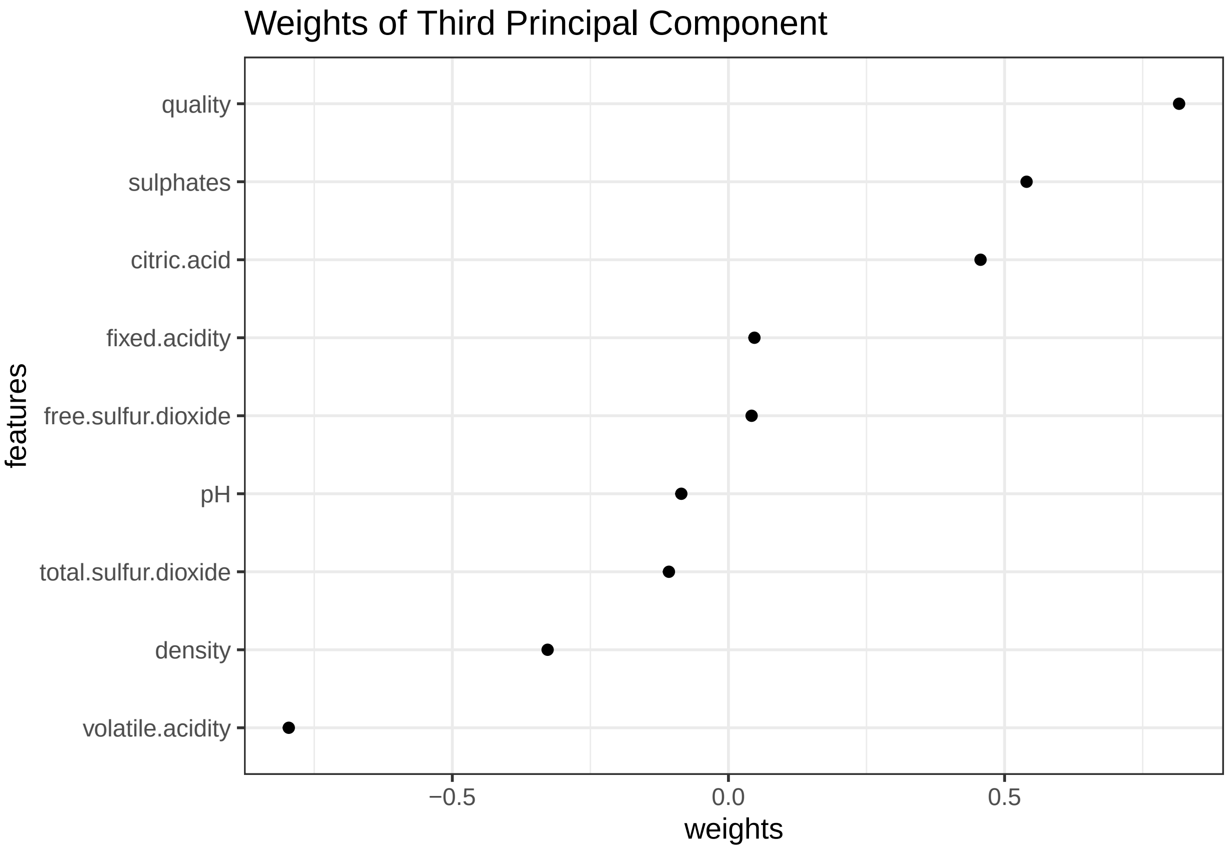
```
#Extract loadings
loadings <- wine_pca_ob$loadings

#Organize into matrix
loadings <- unclass(loadings)

#Convert to dataframe
loadings_df <- as.data.frame(loadings)

#make rownames a variable
loadings_df$features <- rownames(loadings_df)

#convert rownames to standard numbers
rownames(loadings_df) <- NULL

#Create plot of weights for first principal component
ggplot(data = loadings_df, aes(x = RC1, y = reorder(features, RC1))) +
  geom_point() +
  labs(y = 'features', x = 'weights', title = 'Weights of First Principal Component') +
  theme_bw()
```



```
#Create plot for second principal component
ggplot(data = loadings_df, aes(x = RC2, y = reorder(features, RC2))) +
  geom_point() +
  labs(y = 'features', x = 'weights', title = 'Weights of Second Principal Component') +
  theme_bw()
```



```r
#Create plot for third principal component
ggplot(data = loadings_df, aes(x = RC3, y = reorder(features, RC3))) +
  geom_point() +
  labs(y = 'features', x = 'weights', title = 'Weights of Third Principal Component') +
  theme_bw()
```



Our components seem to have a few measures that make sense to be split by. In order to determine their split, we consider the weights of varaibles with a score greater than 0.3 The first component shows that fixed acidity, citric acid and density all have the highest weights which might imply that levels of acidity could be the first component.The second component very clearly reveals its a component based on sulfur dioxide level. Lastly, the third component might reveal that sulphates and quality have some sort of relationship with one another. Perhaps a higher levels of sulphates implies that the wine will be higher quality by protecting the process of fermentation.

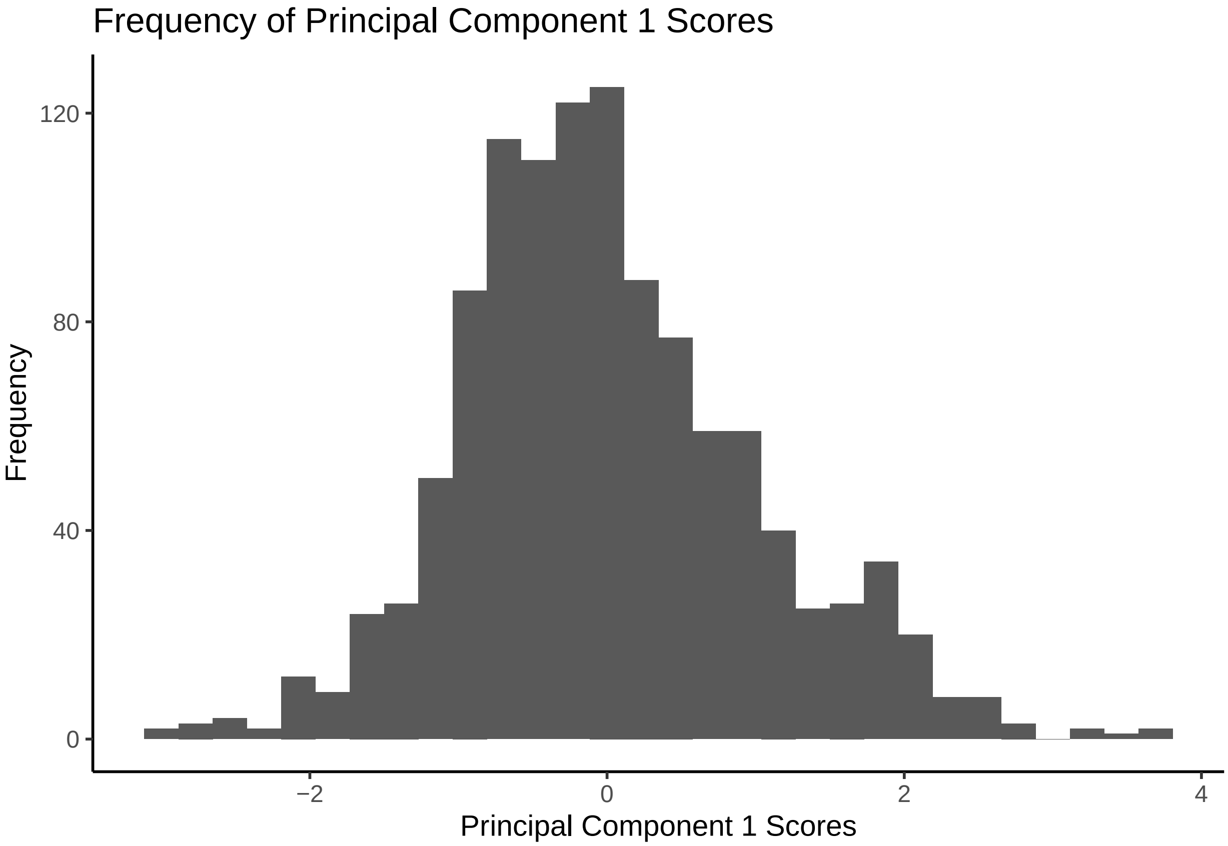
Nonetheless, we could use these components in the future for model creation using the scores from PCA as shown below.

```
wine_scores <- as.data.frame(wine_pca_ob$scores)
head(wine_scores)
```

```
##          RC1        RC3         RC2
## 1 -0.6485624 -1.3775953 -0.54660513
## 2 -0.1754682 -1.3986933  0.80960127
## 3 -0.1764044 -1.1833521  0.08330211
## 4  1.3862117  0.9209274  0.31912523
## 5 -0.6485624 -1.3775953 -0.54660513
## 6 -0.6454412 -1.2844487 -0.33879422
```

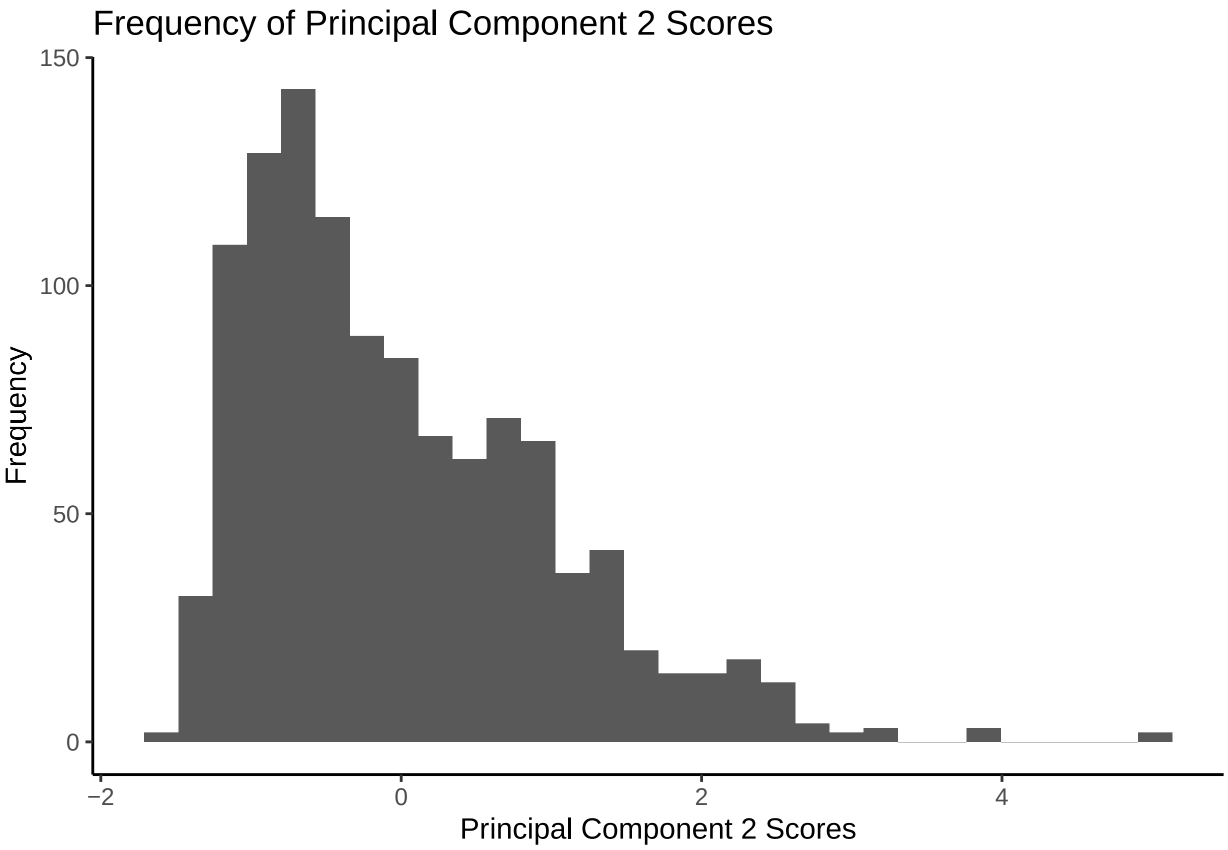
```
# Visualizations of Principal Component Scores

wine_scores %>%
  ggplot(aes(x = RC1)) +
  geom_histogram() +
  theme_classic() +
  labs(title = "Frequency of Principal Component 1 Scores",
       x = "Principal Component 1 Scores",
       y = "Frequency")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



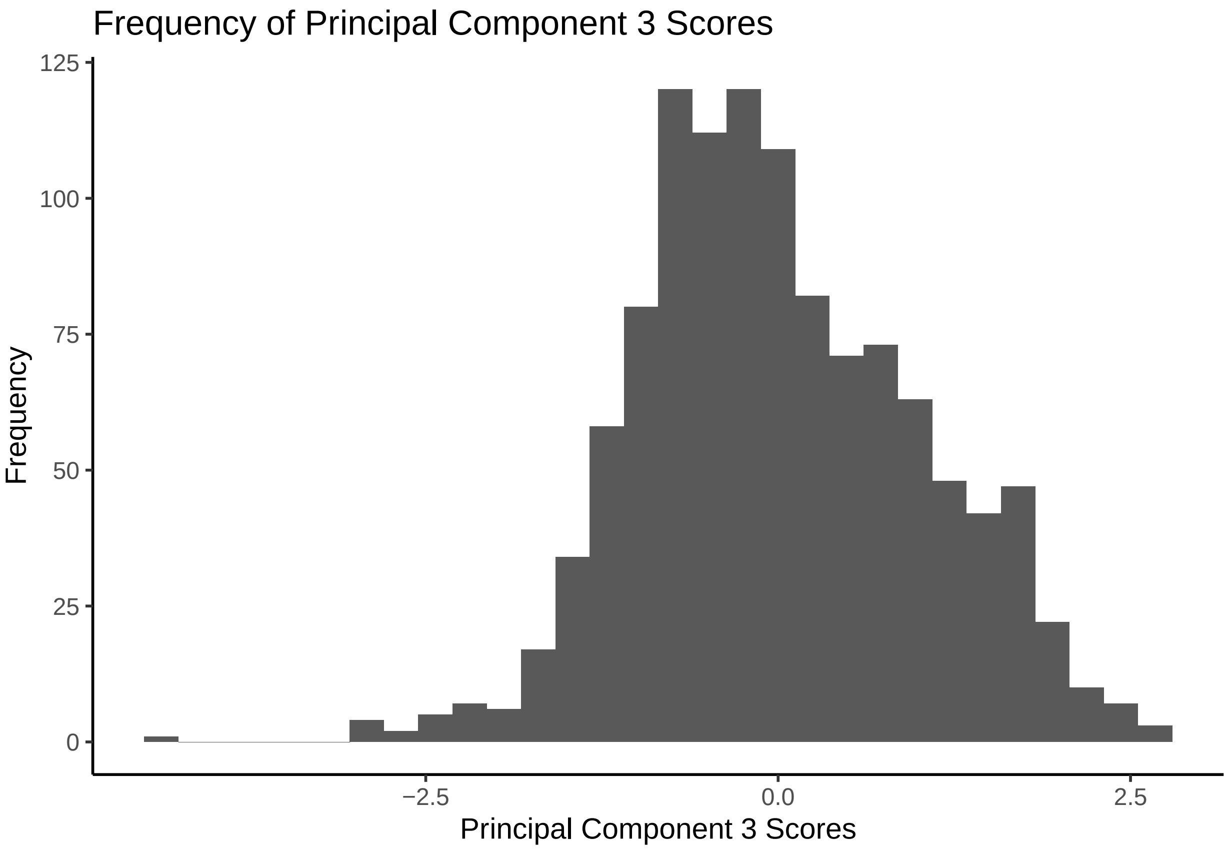
```
wine_scores %>%
  ggplot(aes(x = RC2)) +
  geom_histogram() +
  theme_classic() +
  labs(title = "Frequency of Principal Component 2 Scores",
       x = "Principal Component 2 Scores",
       y = "Frequency")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
wine_scores %>%
  ggplot(aes(x = RC3)) +
  geom_histogram() +
  theme_classic() +
  labs(title = "Frequency of Principal Component 3 Scores",
       x = "Principal Component 3 Scores",
       y = "Frequency")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The visualizations of the principal component scores show that RC1 and RC3 are approximately normal while RC2 might be considered skewed to the right.

Conclusion

The purpose of this investigation was to determine if our data related to wine could have its dimensions reduced into 3 principal components. We originally had 12 predictor variables, but reduced that to 9 in order to move forward with our PCA analysis. Using the Kaiser-Meyer-Olkin test, we were able to acheive a score of 0.6: above the bare minimum of 0.5 for PCA. This could be good reason to reject that this PCA was meaningful; however, we did have a high number of data points so we went on with the investigation.

After determining the number of predictors, we found the eigenvalue of our new prediction matrix. According to the Kaiser-Guttman rule, Scree Plot, and Parallel Analysis, our findings indicated that a PCA with 3 components was the correct amount to move forward with. To figure out what rotation to use, I looked at each of the 3 rotations I created and determined that an oblique rotation would be the best one to use in the context of my variables. This rotation allowed for the chemistry of my variables to stay somewhat related while also keeping the rotation that were significant in the orthogonal rotation to remain. I then graphed the weights of these components and the components scores themselves to give the reader a better visual understanding

Ultimately, we were able to use PCA to reduce the dimensionality of our data set to 3 components. However, the strength and adequacy of our data could mean that our findings are not as significant as we think. In the future, we may want to find a data set with even more predictor varaibles that can satisfy the tests of data adequacy without predictor elimination.