```r
# Loading Packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(readxl)
library(ggthemes)
library(skimr)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(FSA)
```

```
## Registered S3 methods overwritten by 'FSA':
##   method       from
##   confint.boot car
```

```
##   hist.boot    car
## ## FSA v0.9.5. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
##
## Attaching package: 'FSA'
##
## The following object is masked from 'package:car':
##
##     bootCase
```

```
library(RVAideMemoire)
```

```
## *** Package RVAideMemoire v 0.9-83-7 ***
##
## Attaching package: 'RVAideMemoire'
##
## The following object is masked from 'package:FSA':
##
##     se
```

```
# Importing the data set
red_wine <- read_excel("winequality-red.xlsx")
white_wine <- read_excel("winequality-white.xlsx")
```

```
# Viewing the first 10 rows of the red wine dataframe.
head(red_wine,10)
```

```
## # A tibble: 10 x 12
##    `fixed acidity` `volatile acidity` `citric acid` `residual sugar` chlorides
##              <dbl>              <dbl>         <dbl>            <dbl>     <dbl>
##  1             7.4               0.7             0              1.9     0.076
##  2             7.8               0.88            0              2.6     0.098
##  3             7.8               0.76            0.04           2.3     0.092
##  4            11.2               0.28            0.56           1.9     0.075
##  5             7.4               0.7             0              1.9     0.076
##  6             7.4               0.66            0              1.8     0.075
##  7             7.9               0.6             0.06           1.6     0.069
##  8             7.3               0.65            0              1.2     0.065
##  9             7.8               0.58            0.02           2       0.073
## 10             7.5               0.5             0.36           6.1     0.071
## # i 7 more variables: `free sulfur dioxide` <dbl>,
## #   `total sulfur dioxide` <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
## #   alcohol <dbl>, quality <dbl>
```

```
# Viewing the first 10 rows of the white wine dataframe.
head(white_wine,10)
```

```
## # A tibble: 10 x 12
##    `fixed acidity` `volatile acidity` `citric acid` `residual sugar` chlorides
##              <dbl>              <dbl>         <dbl>            <dbl>     <dbl>
##  1               7               0.27          0.36             20.7     0.045
##  2             6.3               0.3           0.34              1.6     0.049
```

```
##  3            8.1              0.28         0.4               6.9     0.05
##  4            7.2              0.23         0.32              8.5     0.058
##  5            7.2              0.23         0.32              8.5     0.058
##  6            8.1              0.28         0.4               6.9     0.05
##  7            6.2              0.32         0.16              7       0.045
##  8            7                0.27         0.36             20.7     0.045
##  9            6.3              0.3          0.34              1.6     0.049
## 10            8.1              0.22         0.43              1.5     0.044
## # i 7 more variables: `free sulfur dioxide` <dbl>,
## #   `total sulfur dioxide` <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
## #   alcohol <dbl>, quality <dbl>
```

```
# Structure of both dataframes.
str(red_wine)
```

```
## tibble [1,599 x 12] (S3: tbl_df/tbl/data.frame)
##  $ fixed acidity       : num [1:1599] 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile acidity    : num [1:1599] 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric acid         : num [1:1599] 0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual sugar      : num [1:1599] 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num [1:1599] 0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ..
##  $ free sulfur dioxide : num [1:1599] 11 25 15 17 11 13 15 15 9 17 ...
##  $ total sulfur dioxide: num [1:1599] 34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num [1:1599] 0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num [1:1599] 3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num [1:1599] 0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num [1:1599] 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : num [1:1599] 5 5 5 6 5 5 5 7 7 5 ...
```

```
str(white_wine)
```

```
## tibble [4,898 x 12] (S3: tbl_df/tbl/data.frame)
##  $ fixed acidity       : num [1:4898] 7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
##  $ volatile acidity    : num [1:4898] 0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
##  $ citric acid         : num [1:4898] 0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
##  $ residual sugar      : num [1:4898] 20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
##  $ chlorides           : num [1:4898] 0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
##  $ free sulfur dioxide : num [1:4898] 45 14 30 47 47 30 30 45 14 28 ...
##  $ total sulfur dioxide: num [1:4898] 170 132 97 186 186 97 136 170 132 129 ...
##  $ density             : num [1:4898] 1.001 0.994 0.995 0.996 0.996 ...
##  $ pH                  : num [1:4898] 3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
##  $ sulphates           : num [1:4898] 0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
##  $ alcohol             : num [1:4898] 8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
##  $ quality             : num [1:4898] 6 6 6 6 6 6 6 6 6 6 ...
```

```
# checking for missing values
sum(is.na(red_wine))
```

```
## [1] 0
```

```r
sum(is.na(white_wine))
```

```
## [1] 0
```

```r
# Checking for duplicates
sum(duplicated(red_wine))
```

```
## [1] 240
```

```r
sum(duplicated(white_wine))
```

```
## [1] 937
```

```r
# Removing duplicates
red_wine <- red_wine[!duplicated(red_wine), ]
white_wine <- white_wine[!duplicated(white_wine), ]
```

```r
# Structure of both data frames.
str(red_wine)
```

```
## tibble [1,359 x 12] (S3: tbl_df/tbl/data.frame)
##  $ fixed acidity       : num [1:1359] 7.4 7.8 7.8 11.2 7.4 7.9 7.3 7.8 7.5 6.7 ...
##  $ volatile acidity    : num [1:1359] 0.7 0.88 0.76 0.28 0.66 0.6 0.65 0.58 0.5 0.58 ...
##  $ citric acid         : num [1:1359] 0 0 0.04 0.56 0 0.06 0 0.02 0.36 0.08 ...
##  $ residual sugar      : num [1:1359] 1.9 2.6 2.3 1.9 1.8 1.6 1.2 2 6.1 1.8 ...
##  $ chlorides           : num [1:1359] 0.076 0.098 0.092 0.075 0.075 0.069 0.065 0.073 0.071 0.097 ..
##  $ free sulfur dioxide : num [1:1359] 11 25 15 17 13 15 15 9 17 15 ...
##  $ total sulfur dioxide: num [1:1359] 34 67 54 60 40 59 21 18 102 65 ...
##  $ density             : num [1:1359] 0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num [1:1359] 3.51 3.2 3.26 3.16 3.51 3.3 3.39 3.36 3.35 3.28 ...
##  $ sulphates           : num [1:1359] 0.56 0.68 0.65 0.58 0.56 0.46 0.47 0.57 0.8 0.54 ...
##  $ alcohol             : num [1:1359] 9.4 9.8 9.8 9.8 9.4 9.4 10 9.5 10.5 9.2 ...
##  $ quality             : num [1:1359] 5 5 5 6 5 5 7 7 5 5 ...
```

```r
str(white_wine)
```

```
## tibble [3,961 x 12] (S3: tbl_df/tbl/data.frame)
##  $ fixed acidity       : num [1:3961] 7 6.3 8.1 7.2 6.2 8.1 8.1 8.6 7.9 6.6 ...
##  $ volatile acidity    : num [1:3961] 0.27 0.3 0.28 0.23 0.32 0.22 0.27 0.23 0.18 0.16 ...
##  $ citric acid         : num [1:3961] 0.36 0.34 0.4 0.32 0.16 0.43 0.41 0.4 0.37 0.4 ...
##  $ residual sugar      : num [1:3961] 20.7 1.6 6.9 8.5 7 1.5 1.45 4.2 1.2 1.5 ...
##  $ chlorides           : num [1:3961] 0.045 0.049 0.05 0.058 0.045 0.044 0.033 0.035 0.04 0.044 ...
##  $ free sulfur dioxide : num [1:3961] 45 14 30 47 30 28 11 17 16 48 ...
##  $ total sulfur dioxide: num [1:3961] 170 132 97 186 136 129 63 109 75 143 ...
##  $ density             : num [1:3961] 1.001 0.994 0.995 0.996 0.995 ...
##  $ pH                  : num [1:3961] 3 3.3 3.26 3.19 3.18 3.22 2.99 3.14 3.18 3.54 ...
##  $ sulphates           : num [1:3961] 0.45 0.49 0.44 0.4 0.47 0.45 0.56 0.53 0.63 0.52 ...
##  $ alcohol             : num [1:3961] 8.8 9.5 10.1 9.9 9.6 11 12 9.7 10.8 12.4 ...
##  $ quality             : num [1:3961] 6 6 6 6 6 6 5 5 5 7 ...
```

They both have the same column names and column length.

```r
# Merging the data frames for easier analysis
red_wine <- mutate(red_wine, wine_type = as.factor("Red"))
white_wine <- mutate(white_wine, wine_type = as.factor("White"))
wine_data <- bind_rows(red_wine, white_wine)
```

```r
#converting quality to ordinal categorical
wine_data$quality <- factor(wine_data$quality, ordered = TRUE)
```

```r
#Checking the head and tail of the newly merged dataframe
head(wine_data,5)
```

```
## # A tibble: 5 x 13
##   `fixed acidity` `volatile acidity` `citric acid` `residual sugar` chlorides
##             <dbl>              <dbl>         <dbl>            <dbl>     <dbl>
## 1             7.4               0.7             0              1.9     0.076
## 2             7.8               0.88            0              2.6     0.098
## 3             7.8               0.76            0.04           2.3     0.092
## 4            11.2               0.28            0.56           1.9     0.075
## 5             7.4               0.66            0              1.8     0.075
## # i 8 more variables: `free sulfur dioxide` <dbl>,
## #   `total sulfur dioxide` <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
## #   alcohol <dbl>, quality <ord>, wine_type <fct>
```

```r
tail(wine_data,5)
```

```
## # A tibble: 5 x 13
##   `fixed acidity` `volatile acidity` `citric acid` `residual sugar` chlorides
##             <dbl>              <dbl>         <dbl>            <dbl>     <dbl>
## 1             6.2               0.21            0.29           1.6     0.039
## 2             6.6               0.32            0.36           8       0.047
## 3             6.5               0.24            0.19           1.2     0.041
## 4             5.5               0.29            0.3            1.1     0.022
## 5             6                 0.21            0.38           0.8     0.02
## # i 8 more variables: `free sulfur dioxide` <dbl>,
## #   `total sulfur dioxide` <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
## #   alcohol <dbl>, quality <ord>, wine_type <fct>
```

```r
# checking the structure of the wine_data data frame
str(wine_data)
```

```
## tibble [5,320 x 13] (S3: tbl_df/tbl/data.frame)
##  $ fixed acidity       : num [1:5320] 7.4 7.8 7.8 11.2 7.4 7.9 7.3 7.8 7.5 6.7 ...
##  $ volatile acidity    : num [1:5320] 0.7 0.88 0.76 0.28 0.66 0.6 0.65 0.58 0.5 0.58 ...
##  $ citric acid         : num [1:5320] 0 0 0.04 0.56 0 0.06 0 0.02 0.36 0.08 ...
##  $ residual sugar      : num [1:5320] 1.9 2.6 2.3 1.9 1.8 1.6 1.2 2 6.1 1.8 ...
##  $ chlorides           : num [1:5320] 0.076 0.098 0.092 0.075 0.075 0.069 0.065 0.073 0.071 0.097 ..
##  $ free sulfur dioxide : num [1:5320] 11 25 15 17 13 15 15 9 17 15 ...
##  $ total sulfur dioxide: num [1:5320] 34 67 54 60 40 59 21 18 102 65 ...
##  $ density             : num [1:5320] 0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num [1:5320] 3.51 3.2 3.26 3.16 3.51 3.3 3.39 3.36 3.35 3.28 ...
##  $ sulphates           : num [1:5320] 0.56 0.68 0.65 0.58 0.56 0.46 0.47 0.57 0.8 0.54 ...
```

```
##  $ alcohol             : num [1:5320] 9.4 9.8 9.8 9.8 9.4 9.4 10 9.5 10.5 9.2 ...
##  $ quality             : Ord.factor w/ 7 levels "3"<"4"<"5"<"6"<..: 3 3 3 4 3 3 5 5 3 3 ...
##  $ wine_type           : Factor w/ 2 levels "Red","White": 1 1 1 1 1 1 1 1 1 1 ...
```

The red wine had 1359 rows and 12 columns The white wine had 3961 rows and 12 columns The new data frame (wine_data) has 5320 row and 13 columns. The quality column is now a categorical variable with 7 levels from 3-9 The last column represents the wine type, red or white.

```r
# Summary statistics
summary(wine_data)
```

```
##  fixed acidity    volatile acidity  citric acid     residual sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2400   1st Qu.: 1.800
##  Median : 7.000   Median :0.3000   Median :0.3100   Median : 2.700
##  Mean   : 7.215   Mean   :0.3441   Mean   :0.3185   Mean   : 5.048
##  3rd Qu.: 7.700   3rd Qu.:0.4100   3rd Qu.:0.4000   3rd Qu.: 7.500
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##
##    chlorides       free sulfur dioxide total sulfur dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 16.00      1st Qu.: 74.0        1st Qu.:0.9922
##  Median :0.04700   Median : 28.00      Median :116.0        Median :0.9947
##  Mean   :0.05669   Mean   : 30.04      Mean   :114.1        Mean   :0.9945
##  3rd Qu.:0.06600   3rd Qu.: 41.00      3rd Qu.:153.2        3rd Qu.:0.9968
##  Max.   :0.61100   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##
##        pH          sulphates         alcohol      quality  wine_type
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   3:  30   Red  :1359
##  1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   4: 206   White:3961
##  Median :3.210   Median :0.5100   Median :10.40   5:1752
##  Mean   :3.225   Mean   :0.5334   Mean   :10.55   6:2323
##  3rd Qu.:3.330   3rd Qu.:0.6000   3rd Qu.:11.40   7: 856
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   8: 148
##                                                   9:   5
```

From the summary statistics: There are 30 observations with a rating of 3 There are 206 observations with a rating of 4 There are 1752 observations with a rating of 5 There are 2323 observations with a rating of 6 There are 856 observations with a rating of 7 There are 148 observations with a rating of 8 There are 5 observations with a rating of 9

```r
# Veiwing the summary statistics (mean and median) based on wine type
summary_stats <- wine_data %>%
  group_by(wine_type) %>%
  summarise(across(where(is.numeric), list(
    mean = mean,
    median = median
  ), .names = "{.col}_{.fn}"))
print(summary_stats)
```

```
## # A tibble: 2 x 23
##   wine_type `fixed acidity_mean` `fixed acidity_median` `volatile acidity_mean`
```

```
##   <fct>                    <dbl>                  <dbl>                  <dbl>
## 1 Red                       8.31                   7.9                  0.529
## 2 White                     6.84                   6.8                  0.281
## # i 19 more variables: `volatile acidity_median` <dbl>,
## #   `citric acid_mean` <dbl>, `citric acid_median` <dbl>,
## #   `residual sugar_mean` <dbl>, `residual sugar_median` <dbl>,
## #   chlorides_mean <dbl>, chlorides_median <dbl>,
## #   `free sulfur dioxide_mean` <dbl>, `free sulfur dioxide_median` <dbl>,
## #   `total sulfur dioxide_mean` <dbl>, `total sulfur dioxide_median` <dbl>,
## #   density_mean <dbl>, density_median <dbl>, pH_mean <dbl>, ...
```

```r
# summary statistics using the skim function
wine_data %>%
  group_by(wine_type) %>%
  skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	5320
Number of columns	13
Column type frequency:	
factor	1
numeric	11
Group variables	wine_type

Variable type: factor

skim_variable	wine_type	n_missing	complete_rate	ordered	n_unique	top_counts
quality	Red	0	1	TRUE	6	5: 577, 6: 535, 7: 167, 4: 53
quality	White	0	1	TRUE	7	6: 1788, 5: 1175, 7: 689, 4: 153

Variable type: numeric

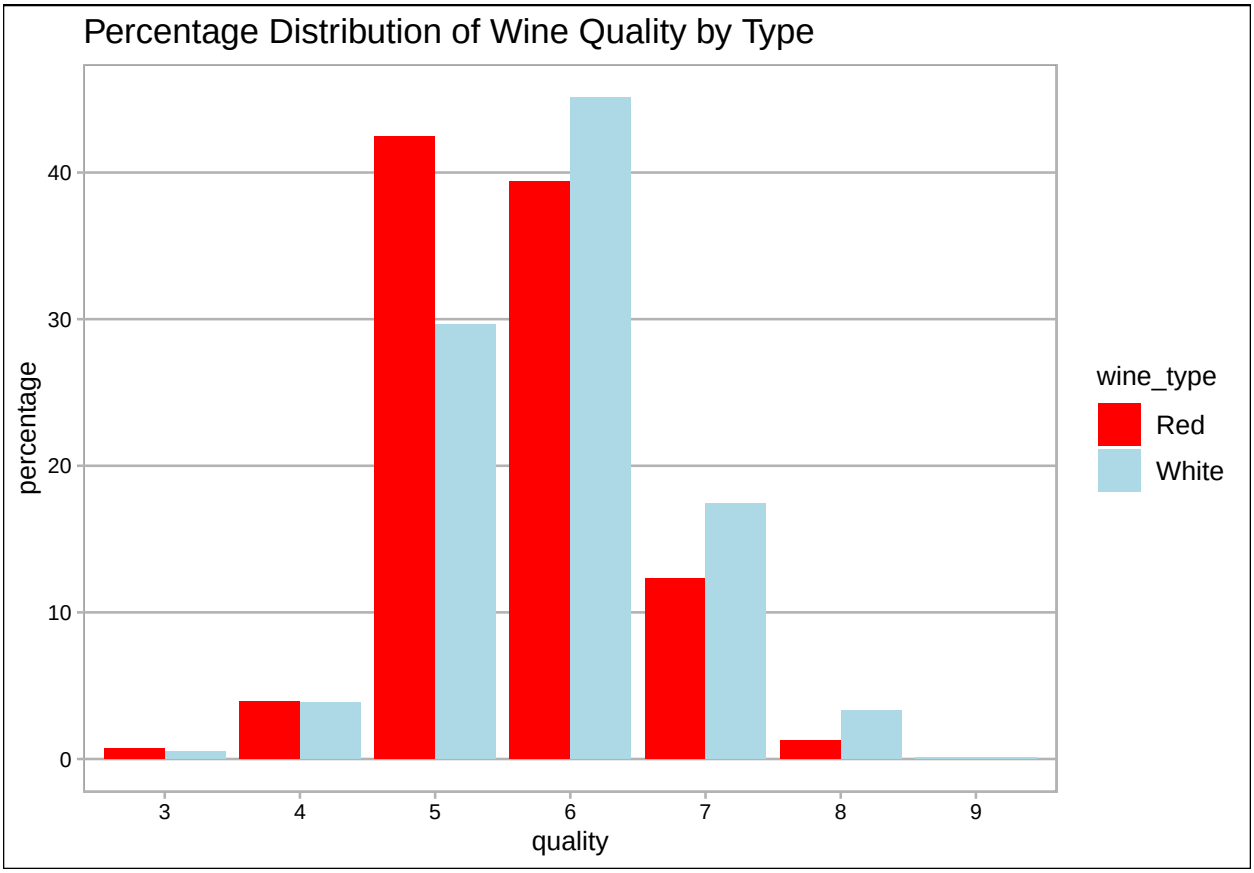
skim_variable	wine_type	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
fixed acidity	Red	0	1	8.31	1.74	4.60	7.10	7.90	9.20	15.90	
fixed acidity	White	0	1	6.84	0.87	3.80	6.30	6.80	7.30	14.20	
volatile acidity	Red	0	1	0.53	0.18	0.12	0.39	0.52	0.64	1.58	
volatile acidity	White	0	1	0.28	0.10	0.08	0.21	0.26	0.33	1.10	
citric acid	Red	0	1	0.27	0.20	0.00	0.09	0.26	0.43	1.00	
citric acid	White	0	1	0.33	0.12	0.00	0.27	0.32	0.39	1.66	
residual sugar	Red	0	1	2.52	1.35	0.90	1.90	2.20	2.60	15.50	
residual sugar	White	0	1	5.91	4.86	0.60	1.60	4.70	8.90	65.80	
chlorides	Red	0	1	0.09	0.05	0.01	0.07	0.08	0.09	0.61	
chlorides	White	0	1	0.05	0.02	0.01	0.04	0.04	0.05	0.35	
free sulfur dioxide	Red	0	1	15.89	10.45	1.00	7.00	14.00	21.00	72.00	

skim_variable	wine_type	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
free sulfur dioxide	White	0	1	34.89	17.21	2.00	23.00	33.00	45.00	289.00	
total sulfur dioxide	Red	0	1	46.83	33.41	6.00	22.00	38.00	63.00	289.00	
total sulfur dioxide	White	0	1	137.19	43.13	9.00	106.00	133.00	166.00	440.00	
density	Red	0	1	1.00	0.00	0.99	1.00	1.00	1.00	1.00	
density	White	0	1	0.99	0.00	0.99	0.99	0.99	1.00	1.04	
pH	Red	0	1	3.31	0.16	2.74	3.21	3.31	3.40	4.01	
pH	White	0	1	3.20	0.15	2.72	3.09	3.18	3.29	3.82	
sulphates	Red	0	1	0.66	0.17	0.33	0.55	0.62	0.73	2.00	
sulphates	White	0	1	0.49	0.11	0.22	0.41	0.48	0.55	1.08	
alcohol	Red	0	1	10.43	1.08	8.40	9.50	10.20	11.10	14.90	
alcohol	White	0	1	10.59	1.22	8.00	9.50	10.40	11.40	14.20	

```
# Percentage distribution of quality based on the wine types
quality_percentage <- wine_data %>%
  group_by(wine_type, quality) %>%
  summarize(count = n(), .groups = "drop") %>%
  group_by(wine_type) %>%
  mutate(percentage = count / sum(count) * 100)
quality_percentage
```

```
## # A tibble: 13 x 4
## # Groups:   wine_type [2]
##    wine_type quality count percentage
##    <fct>     <ord>   <int>      <dbl>
##  1 Red       3          10      0.736
##  2 Red       4          53      3.90
##  3 Red       5         577     42.5
##  4 Red       6         535     39.4
##  5 Red       7         167     12.3
##  6 Red       8          17      1.25
##  7 White     3          20      0.505
##  8 White     4         153      3.86
##  9 White     5        1175     29.7
## 10 White     6        1788     45.1
## 11 White     7         689     17.4
## 12 White     8         131      3.31
## 13 White     9           5      0.126
```

```
# Plotting the percentage distribution
ggplot(quality_percentage, aes(x = quality, y = percentage, fill = wine_type)) +
  geom_col(position = "dodge") +
  labs(title = "Percentage Distribution of Wine Quality by Type") +
  scale_fill_manual(values = c("Red" = "red", "White" = "lightblue")) +  theme_calc()
```




```r
# creating a function to plot box plot by wine type
plot_boxplot <- function(data, column_name) {
  ggplot(data, aes(x = wine_type, y = .data[[column_name]], fill = wine_type)) +
    geom_boxplot() +
    labs(title = paste("Boxplot of", column_name, "by Wine Type")) +
    scale_fill_manual(values = c("Red" = "red", "White" = "lightblue")) +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
}
```

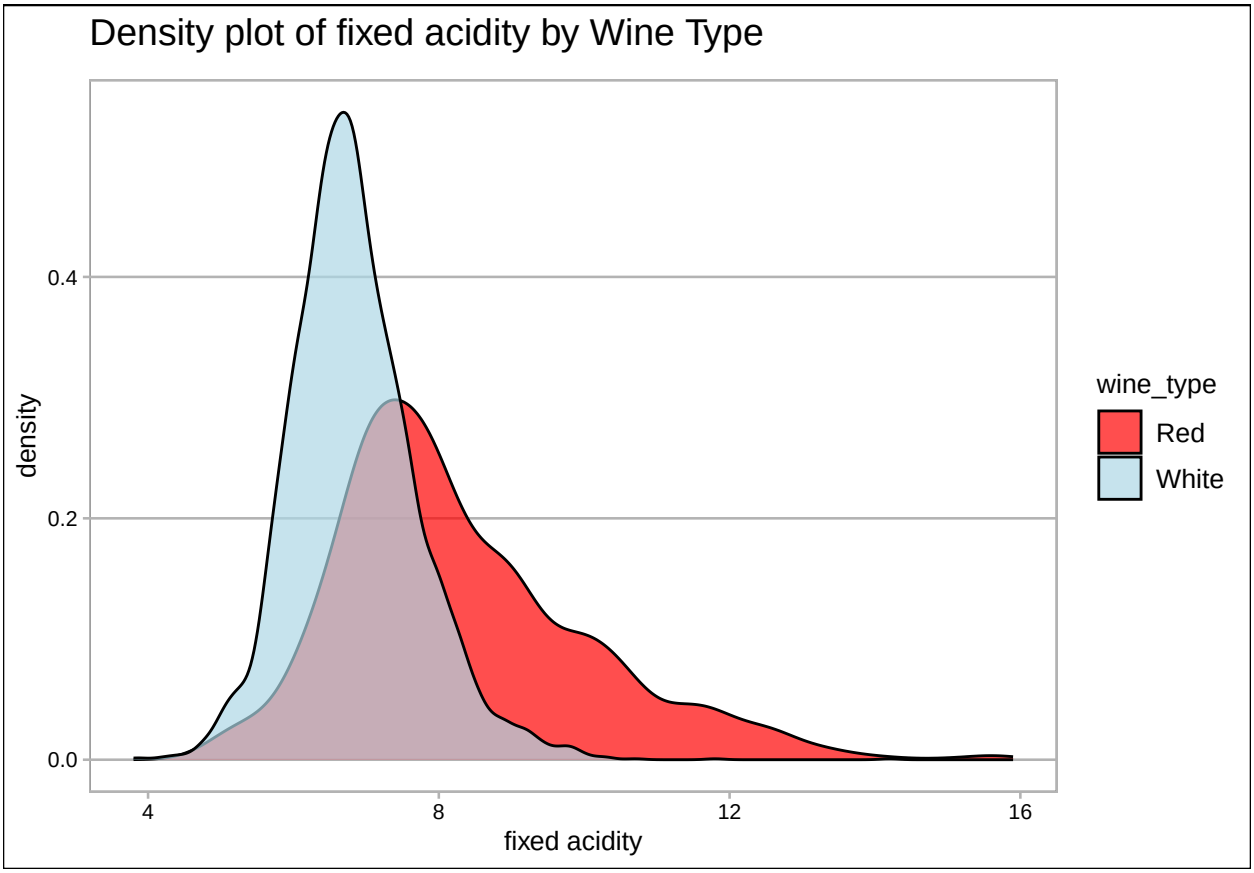
```r
# creating a function to plot box plot by quality and wine type
plot_boxplot_quality <- function(data, column_name) {
  ggplot(data, aes(x = quality, y = .data[[column_name]], fill = wine_type)) +
    geom_boxplot() +
    labs(title = paste("Boxplot of", column_name, "by quality and Wine Type")) +
    scale_fill_manual(values = c("Red" = "red", "White" = "lightblue")) +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
}
```

```r
# creating a function to plot density plot
plot_density_plot <- function(data, column_name) {
  ggplot(data, aes(x = .data[[column_name]], fill = wine_type)) +
    geom_density(alpha = 0.7) +
    labs(title = paste("Density plot of", column_name, "by Wine Type")) +
    scale_fill_manual(values = c("Red" = "red", "White" = "lightblue")) +
```

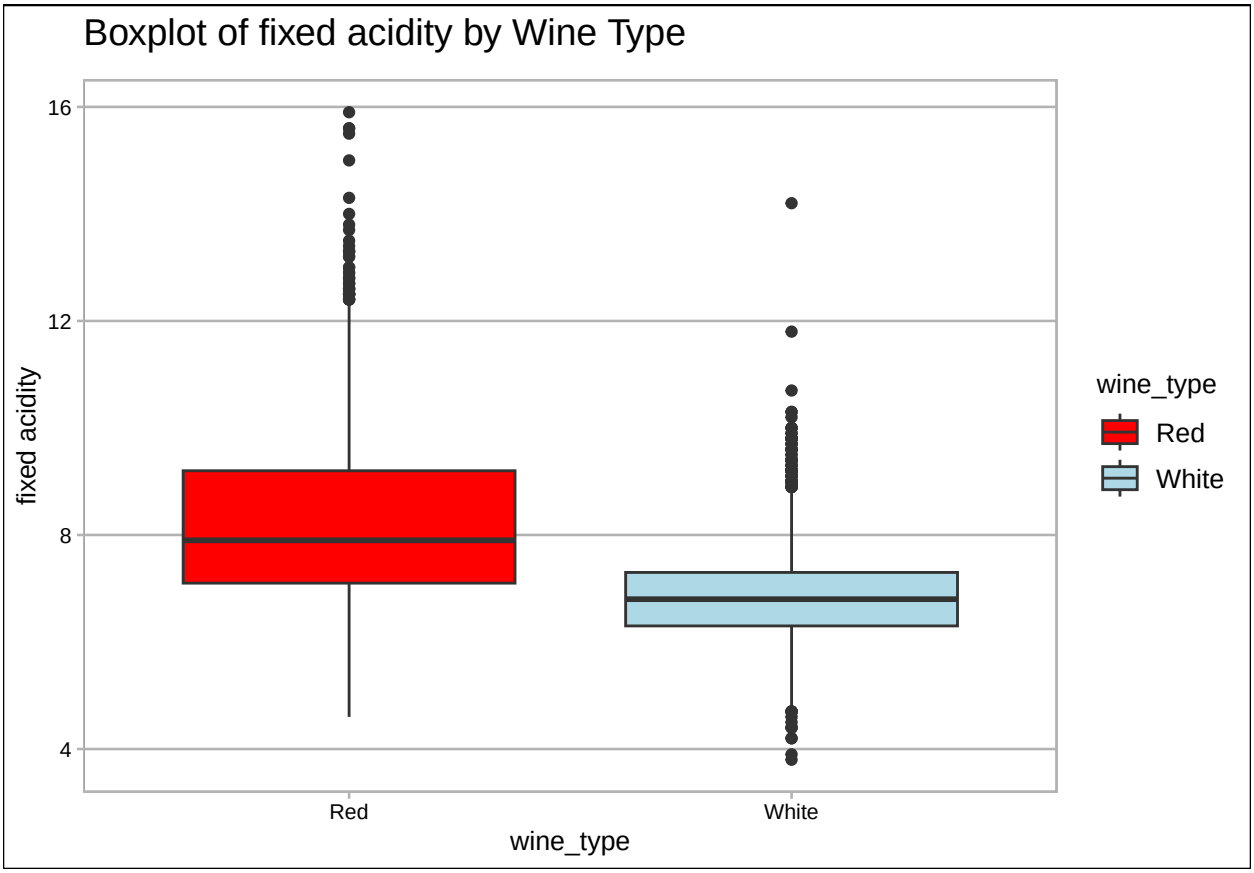
```
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
}
```

```
# Function to perform normality test.
# Creating a function to plot QQ plot and perform Shapiro-Wilks test
normality_tester <- function(data, column_name, color, wine_type) {
  column_data <- data[[column_name]]

  # QQ plot
  qqplot <- ggplot(data, aes(sample = column_data)) +
    stat_qq(color = color) +
    stat_qq_line(color = "black") +
    labs(
      title = paste("QQ Plot for", column_name, "(", wine_type, ")"),
      x = "Theoretical",
      y = "Sample"
    ) +
    theme_calc()+
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
  print(qqplot)

  #Shapiro-Wilks test
  shapiro_result <- shapiro.test(column_data)
  return(shapiro_result)
}
```

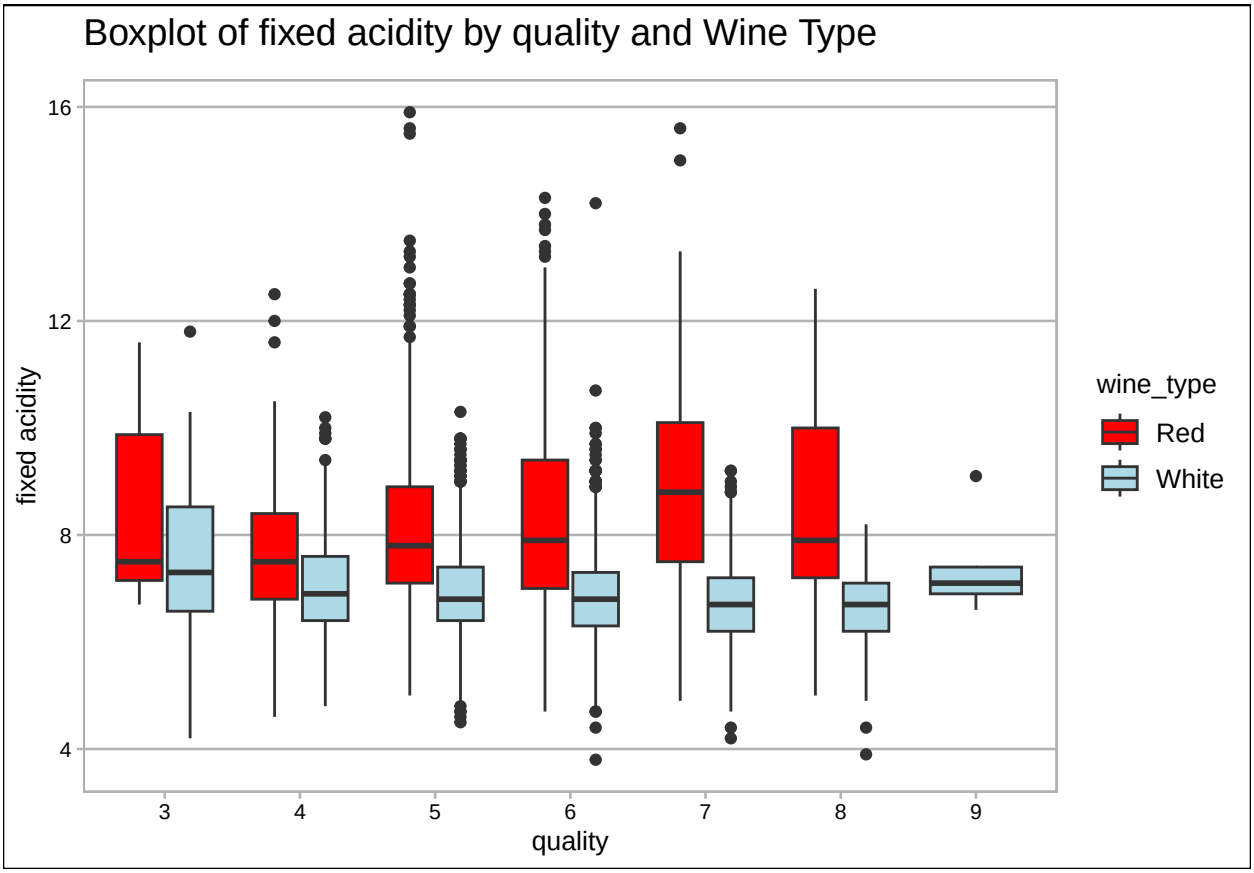
```
#fixed acidity
plot_density_plot(wine_data, 'fixed acidity')
```



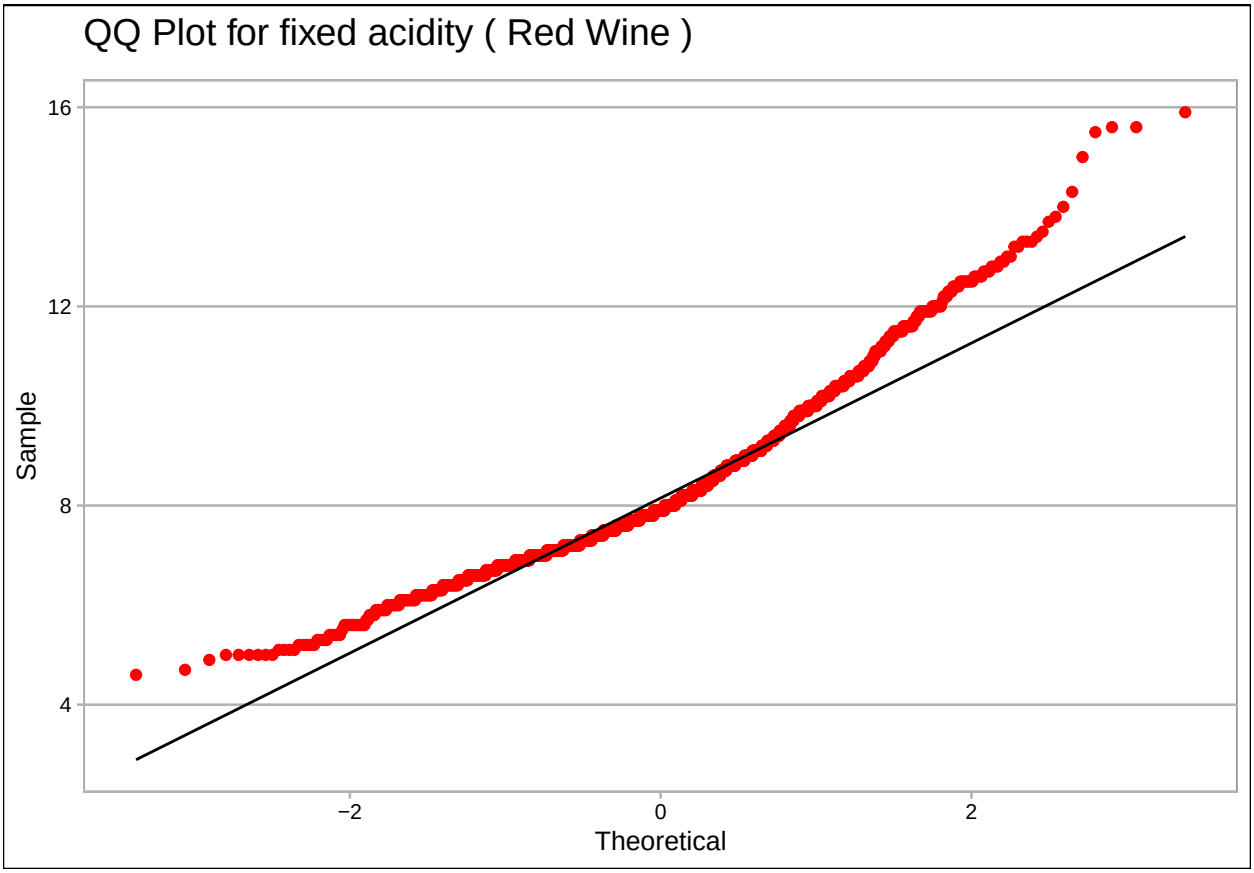

```
plot_boxplot(wine_data, 'fixed acidity')
```




```
plot_boxplot_quality(wine_data, 'fixed acidity')
```

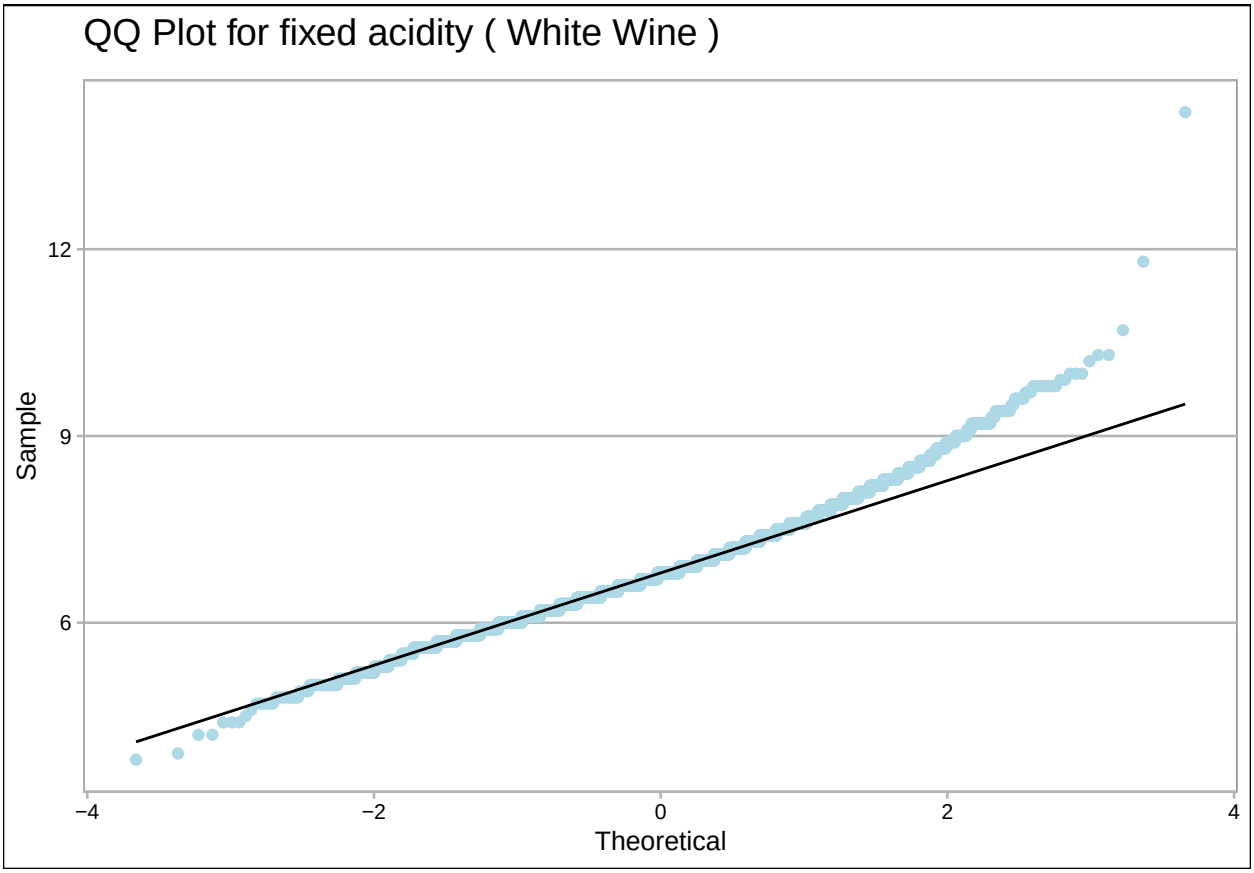



```
#performing normality check for red wine (fixed acidity)
normality_tester(red_wine, 'fixed acidity', 'red','Red Wine')
```




```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.94684, p-value < 2.2e-16
```

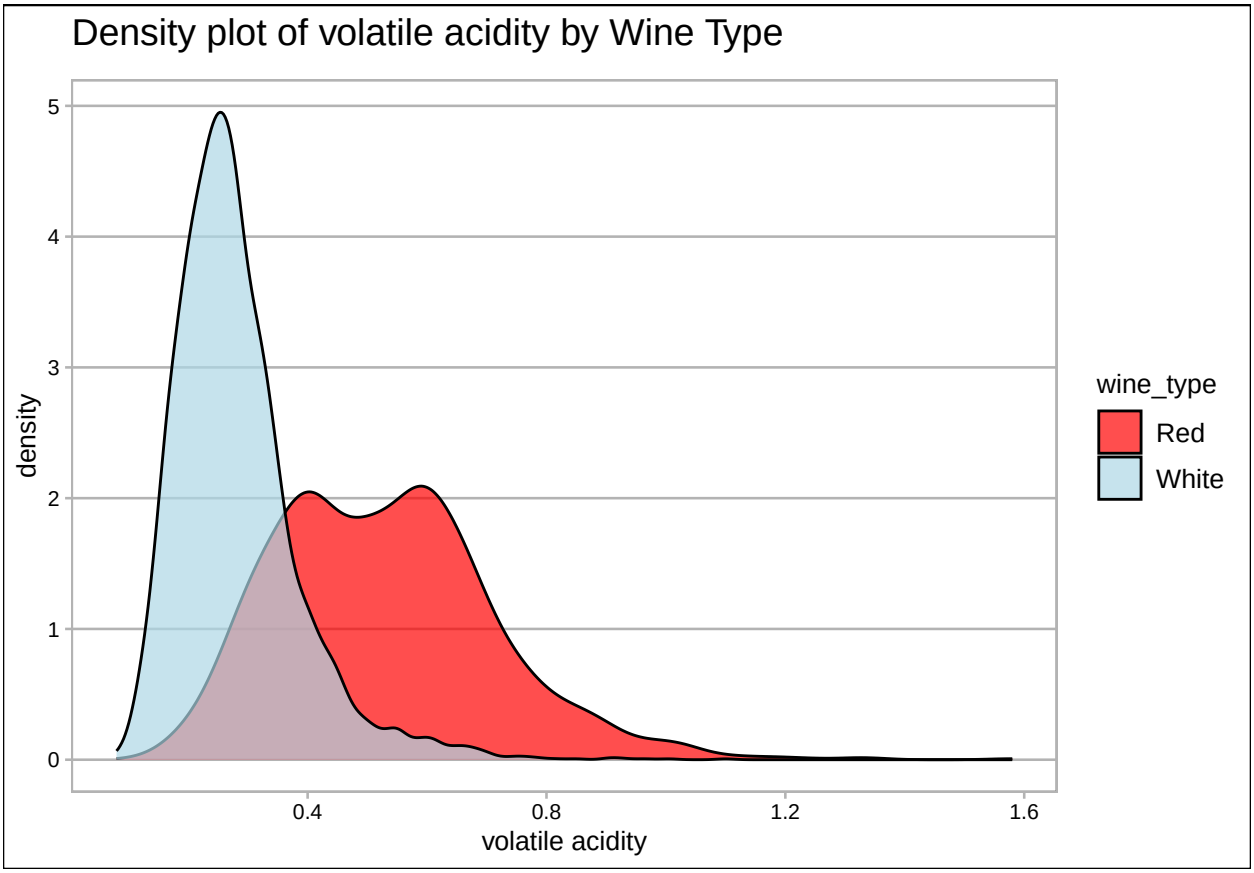
```
#performing normality check for white wine (fixed acidity)
normality_tester(white_wine, 'fixed acidity', 'lightblue', "White Wine")
```



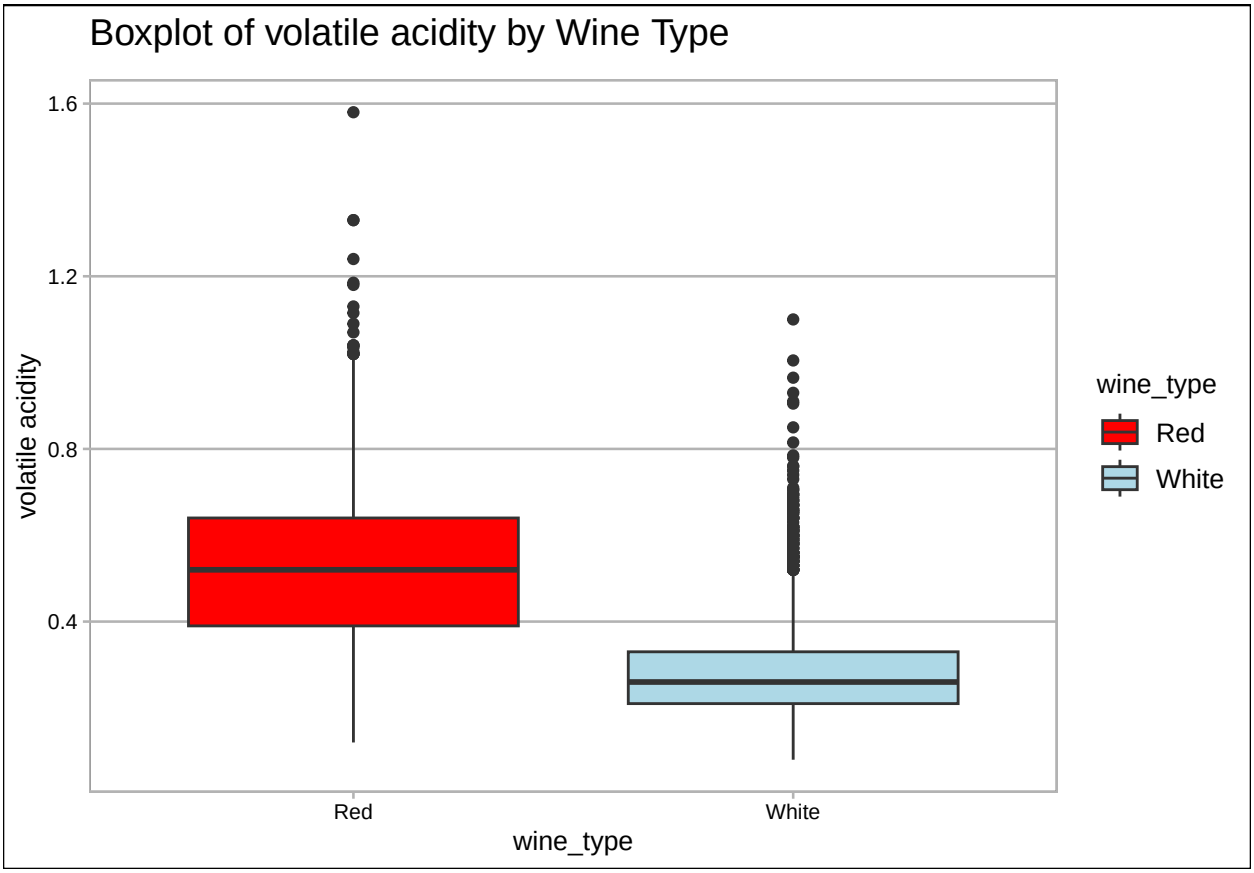

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.97418, p-value < 2.2e-16
```

From the normality rest, the P value is less than 0.05, therefore it doesn't follow a normal distribution

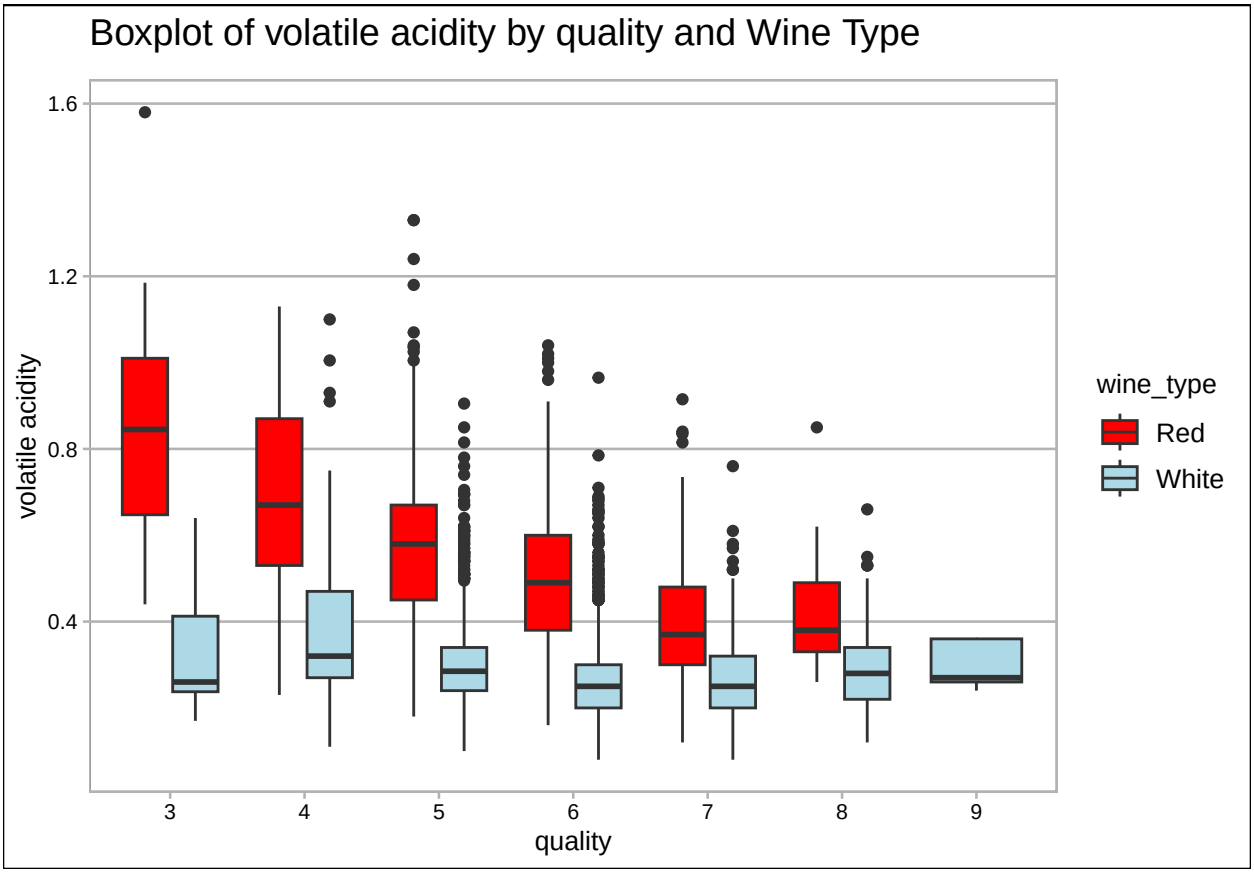
```
# volatile acidity
plot_density_plot(wine_data, 'volatile acidity')
```



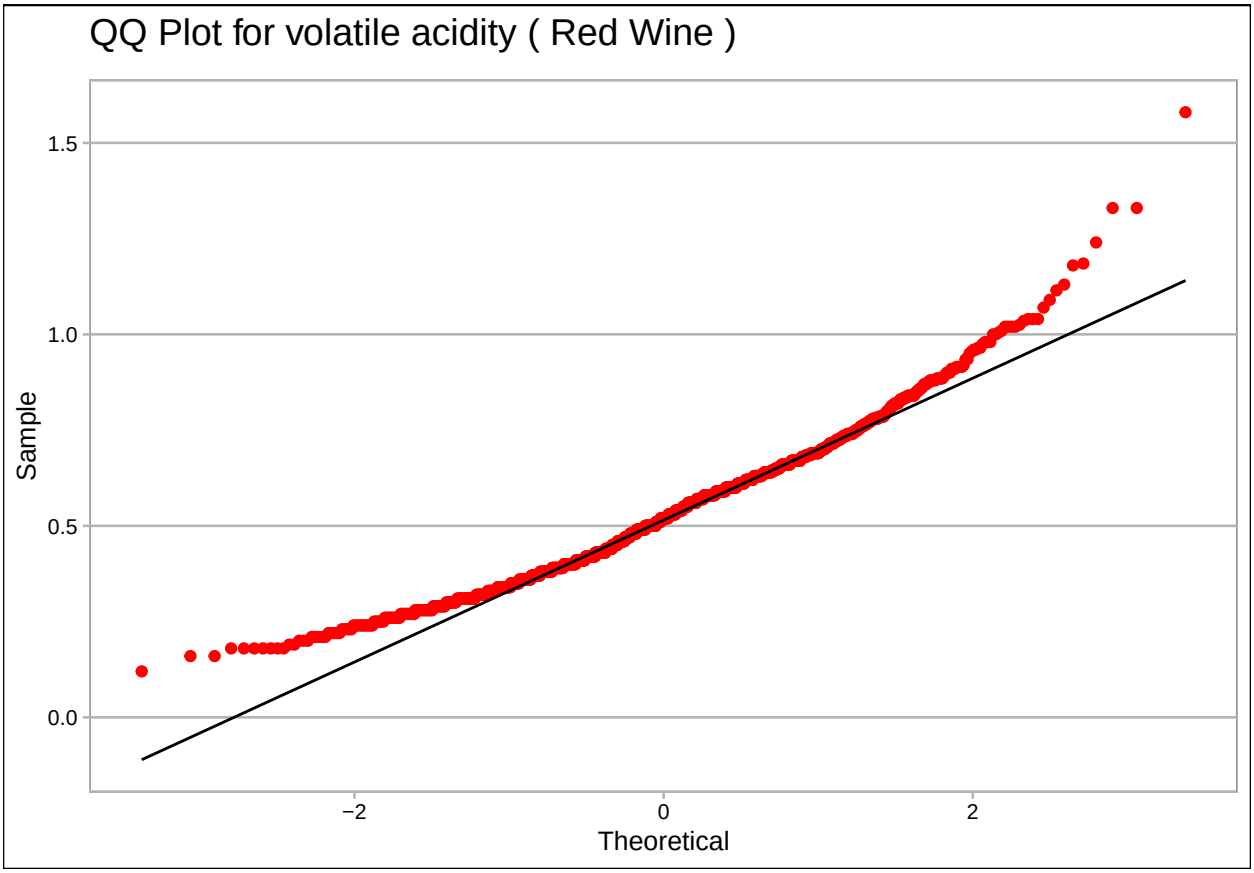

```
plot_boxplot(wine_data, 'volatile acidity')
```




```
plot_boxplot_quality(wine_data, 'volatile acidity')
```

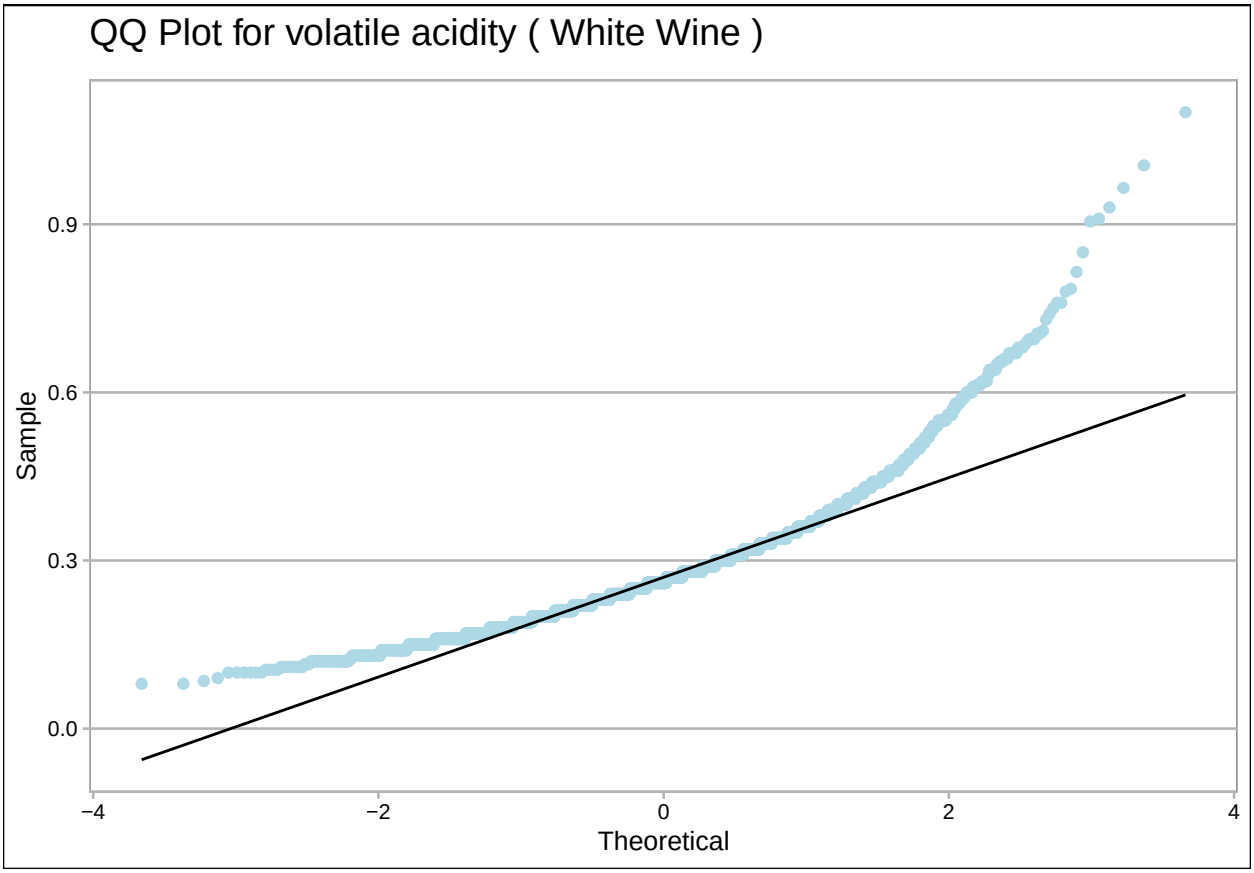



```
#performing normality check for red wine (volatile acidity)
normality_tester(red_wine, 'volatile acidity', 'red','Red Wine')
```



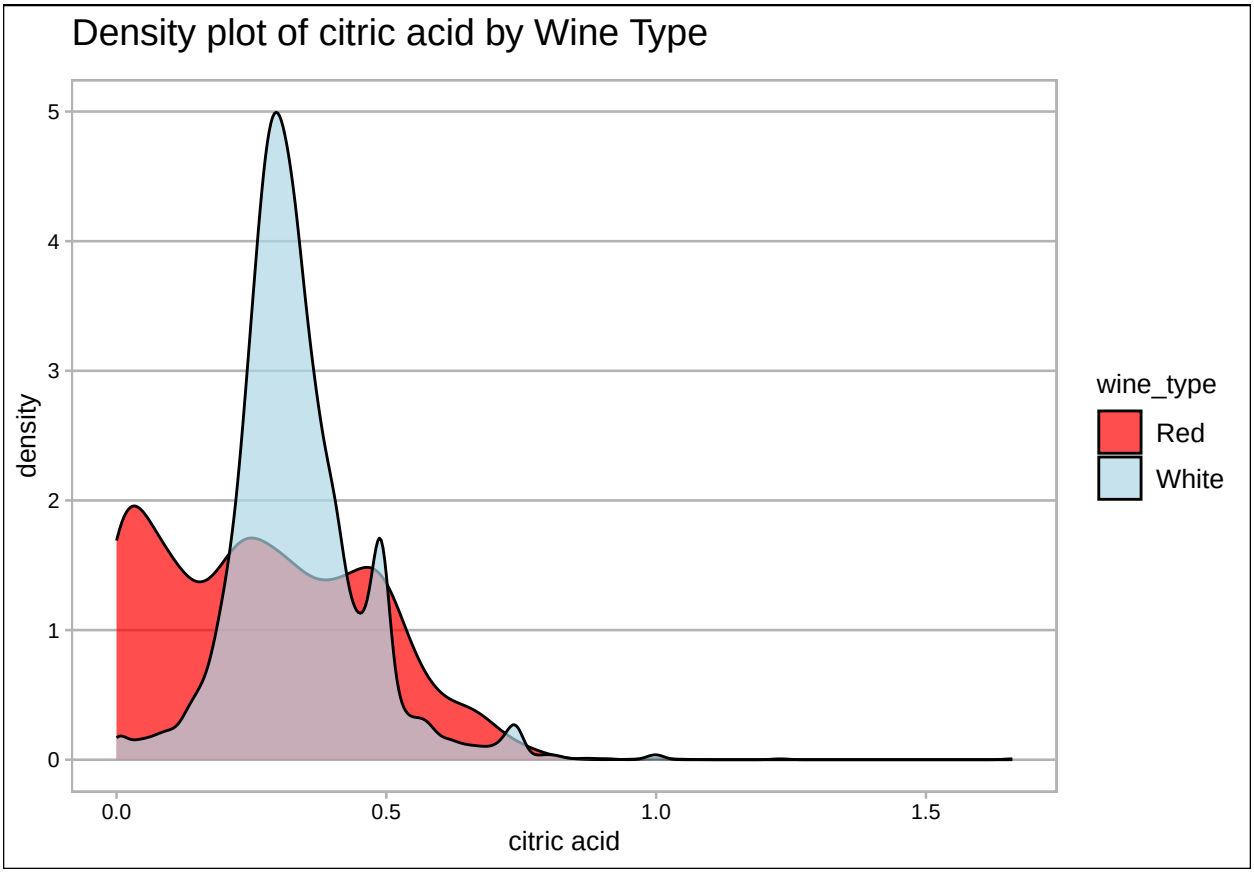

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.97018, p-value = 3.931e-16
```

```
#performing normality check for white wine (volatile acidity)
normality_tester(white_wine, 'volatile acidity', 'lightblue', "White Wine")
```

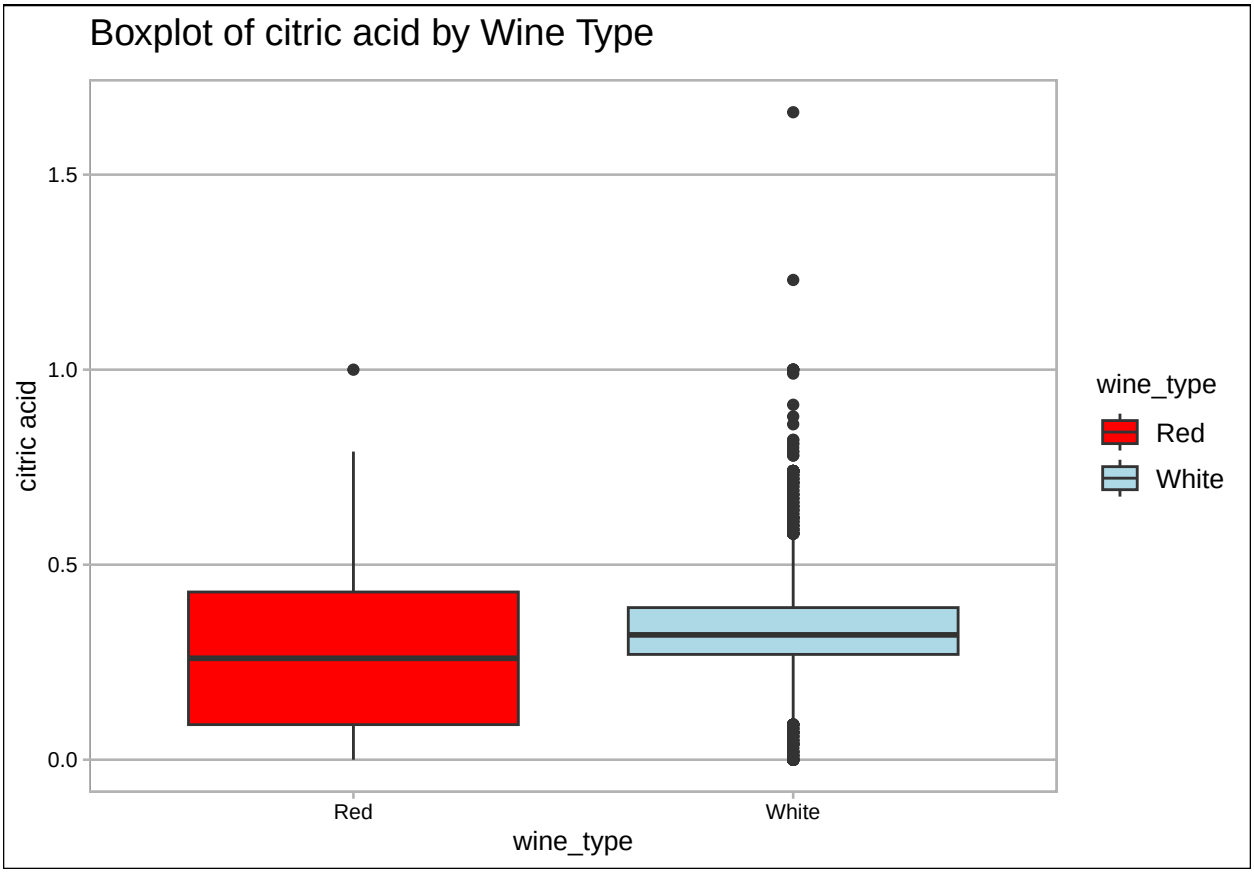



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.89753, p-value < 2.2e-16
```

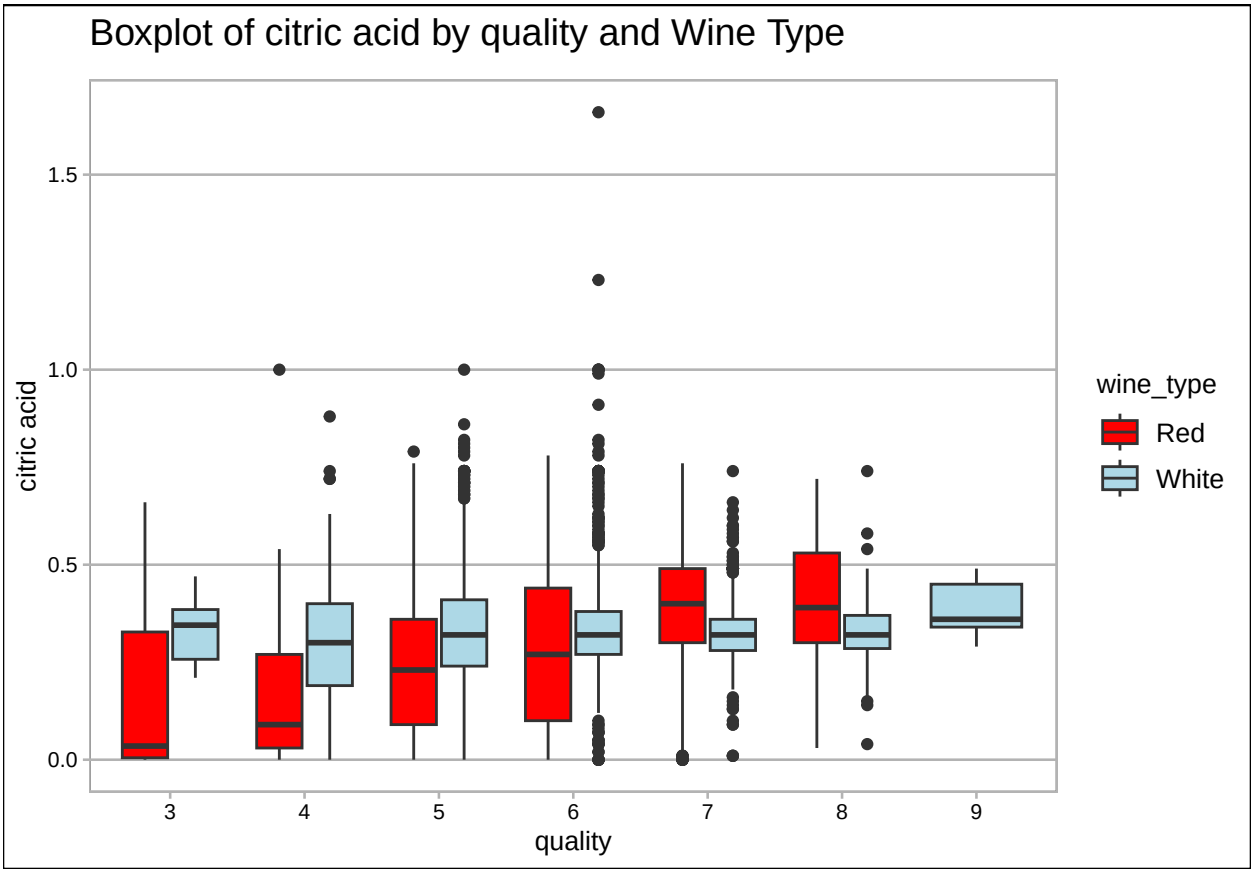
```
# citric acid
plot_density_plot(wine_data, 'citric acid')
```



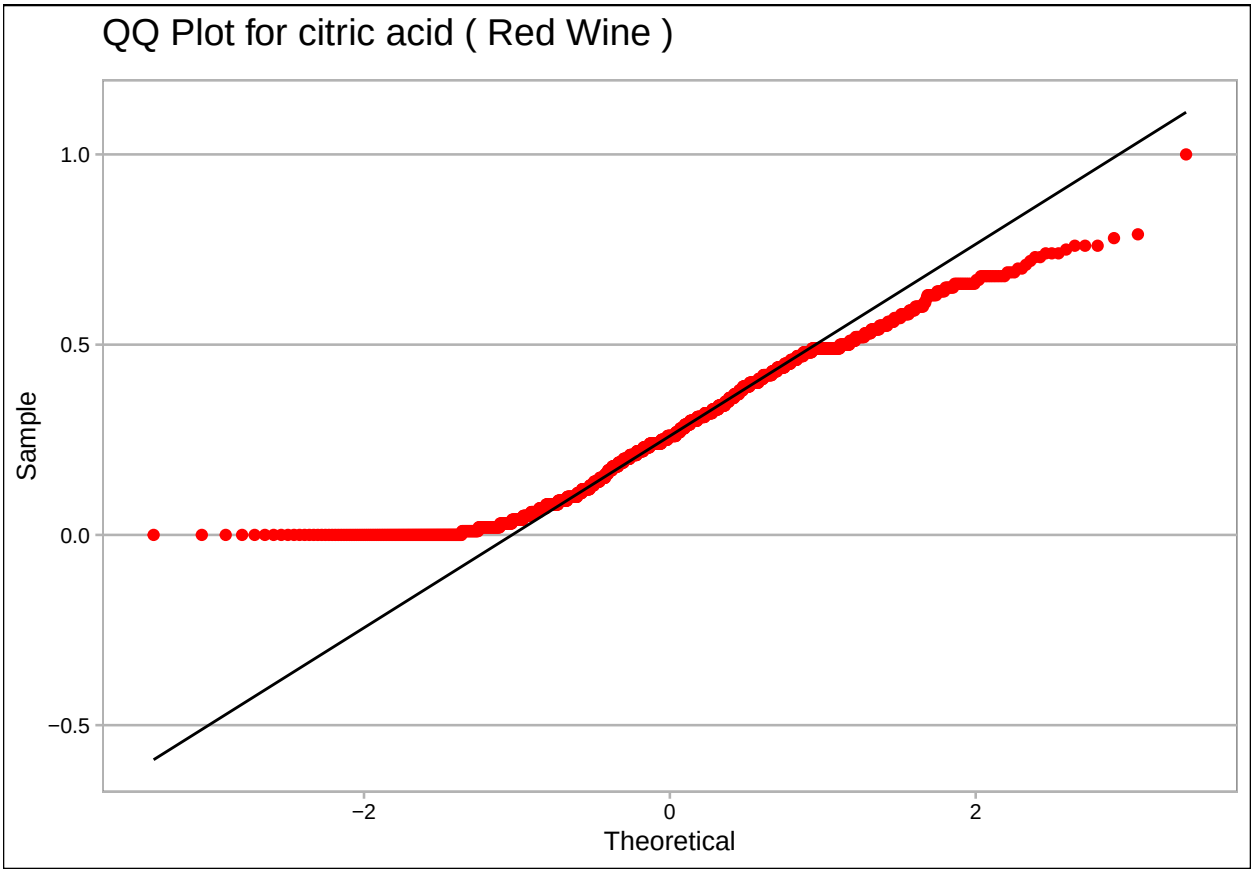

```
plot_boxplot(wine_data, 'citric acid')
```




```
plot_boxplot_quality(wine_data, 'citric acid')
```

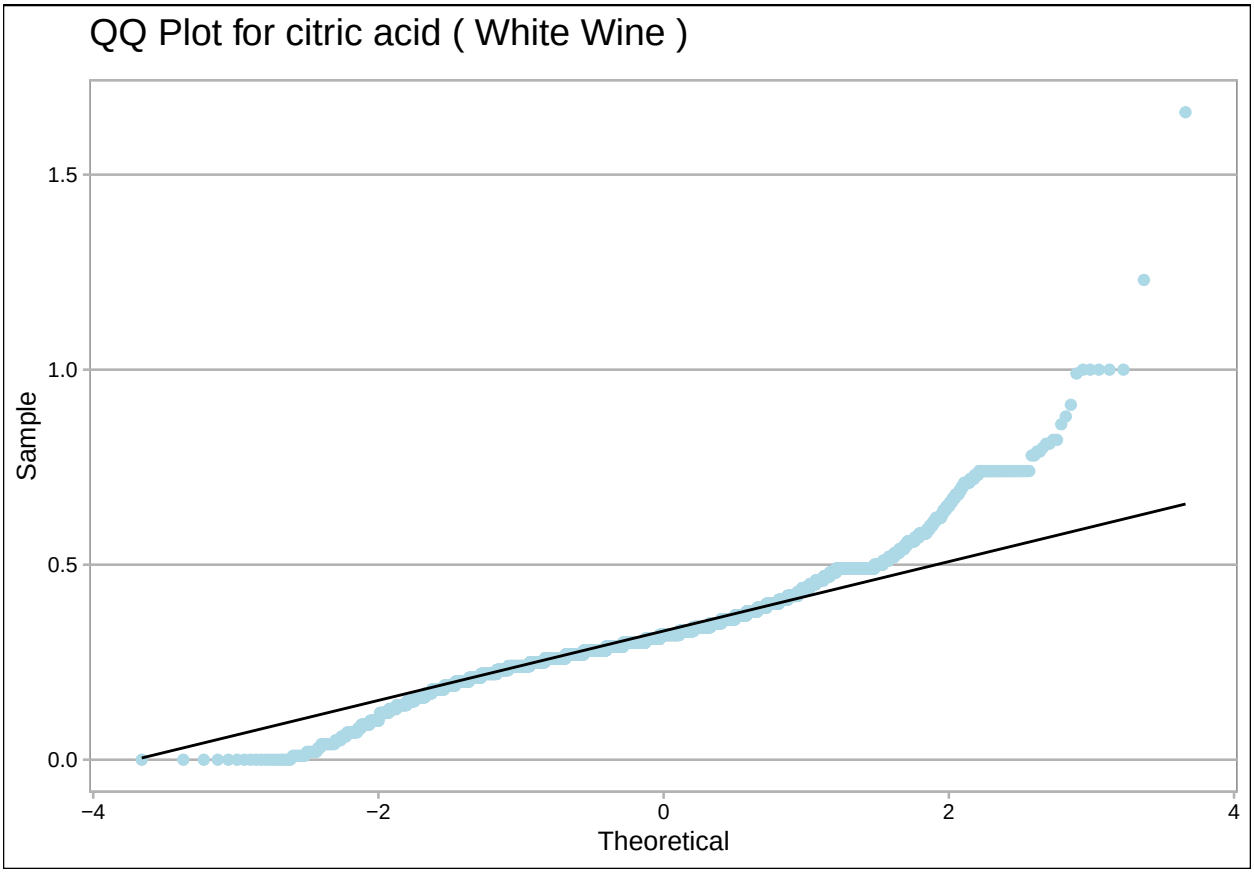



```
#performing normality check for red wine (citric acid)
normality_tester(red_wine, 'citric acid', 'red','Red Wine')
```



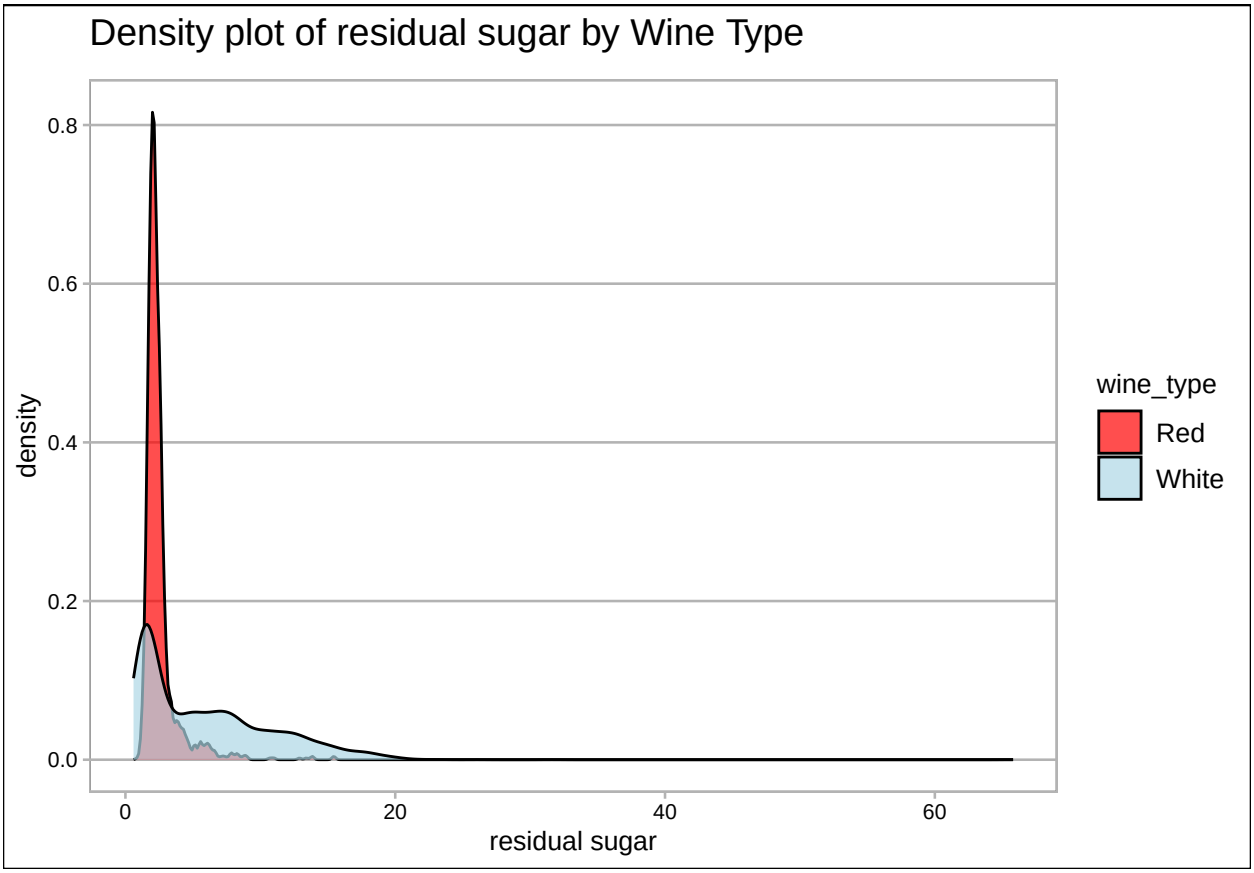

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.95552, p-value < 2.2e-16
```

```
#performing normality check for white wine (citric acid)
normality_tester(white_wine, 'citric acid', 'lightblue', "White Wine")
```

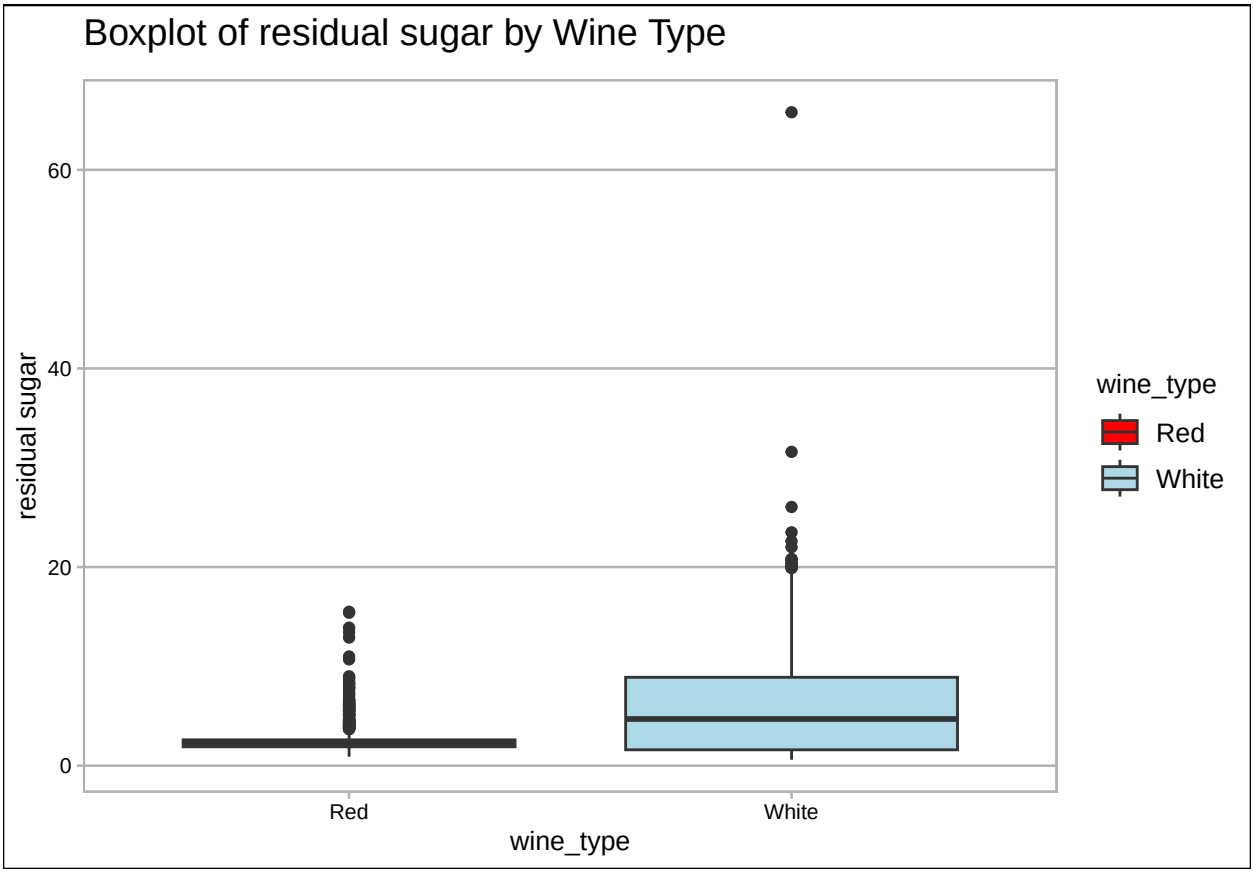



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.92054, p-value < 2.2e-16
```

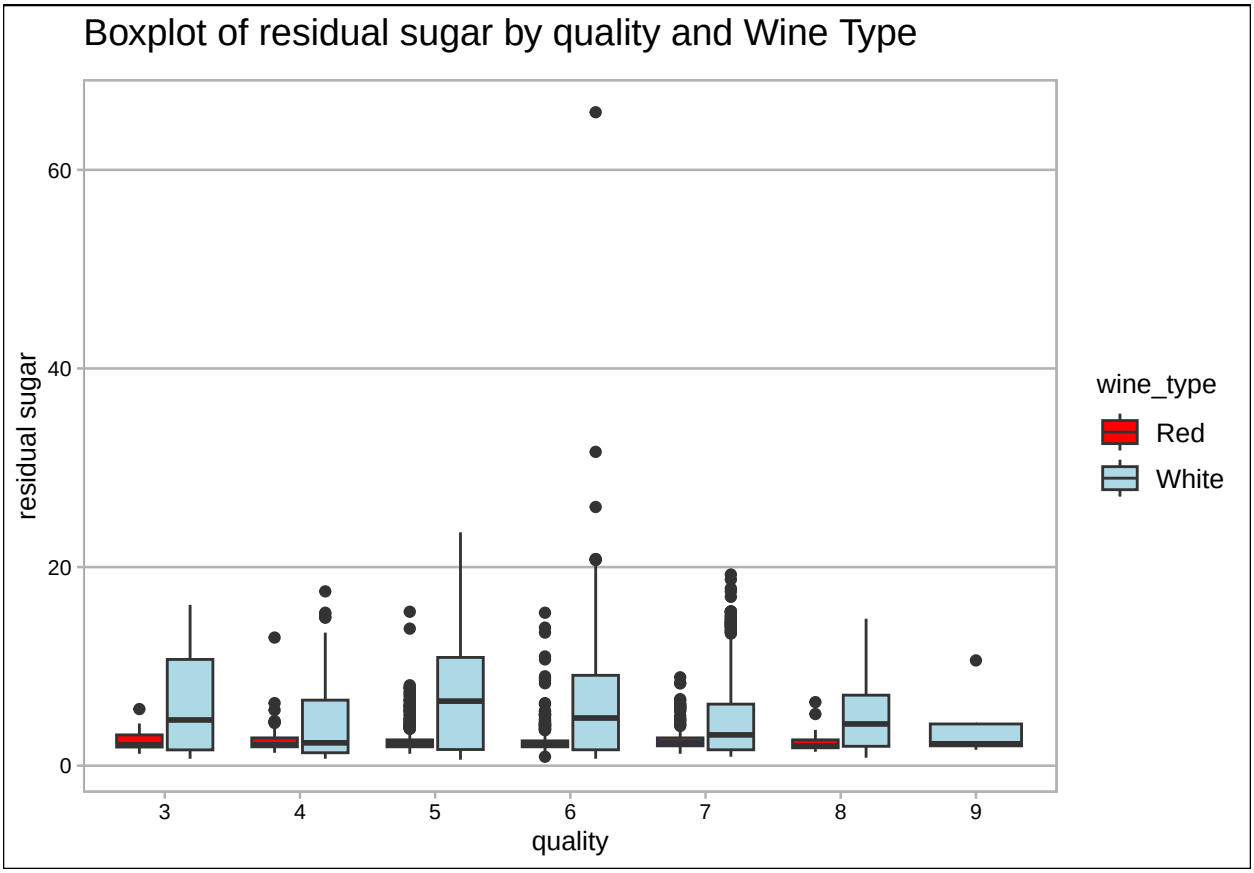
```
# residual sugar
plot_density_plot(wine_data, 'residual sugar')
```



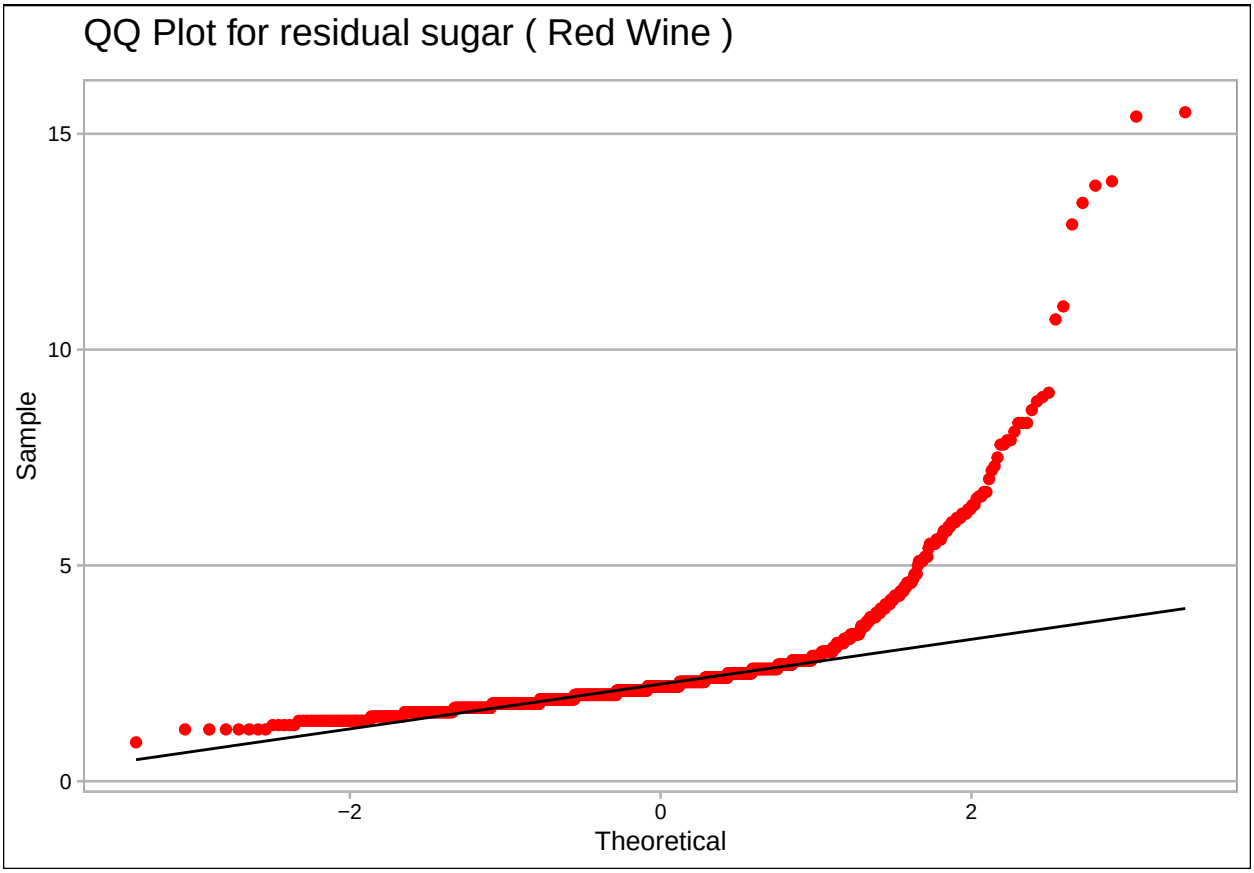

```
plot_boxplot(wine_data, 'residual sugar')
```




```
plot_boxplot_quality(wine_data, 'residual sugar')
```

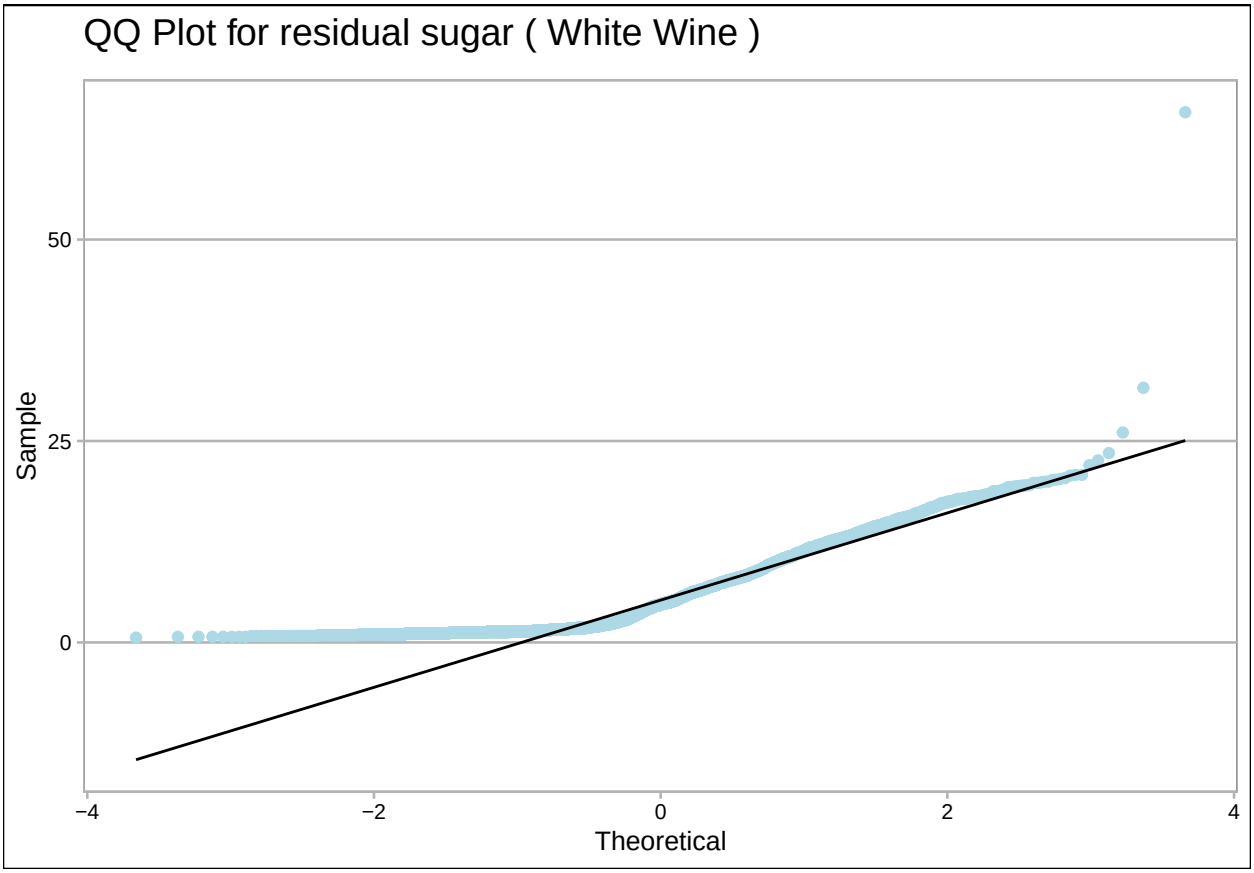



```
#performing normality check for red wine (residual sugar)
normality_tester(red_wine, 'residual sugar', 'red','Red Wine')
```



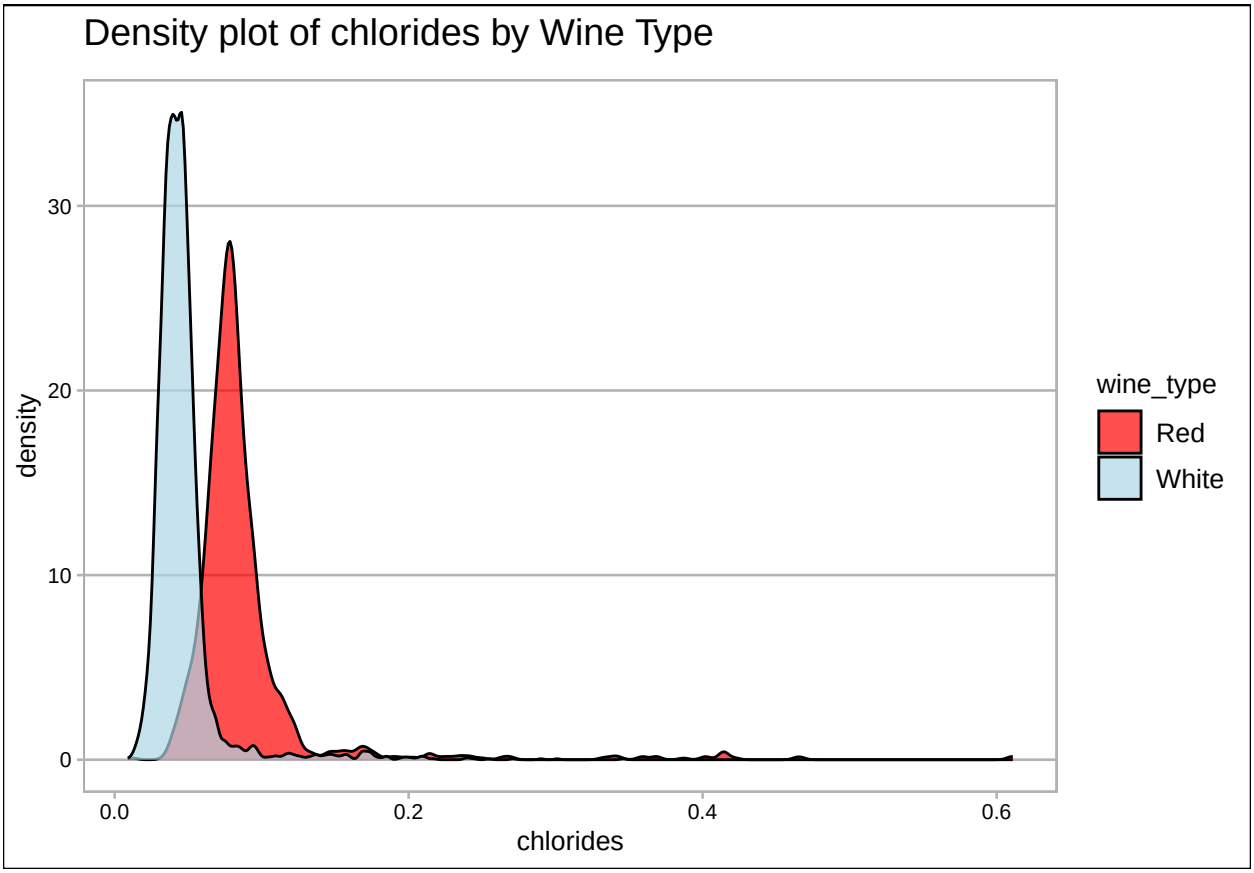

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.57673, p-value < 2.2e-16
```

```
#performing normality check for white wine (residual sugar)
normality_tester(white_wine, 'residual sugar', 'lightblue', "White Wine")
```

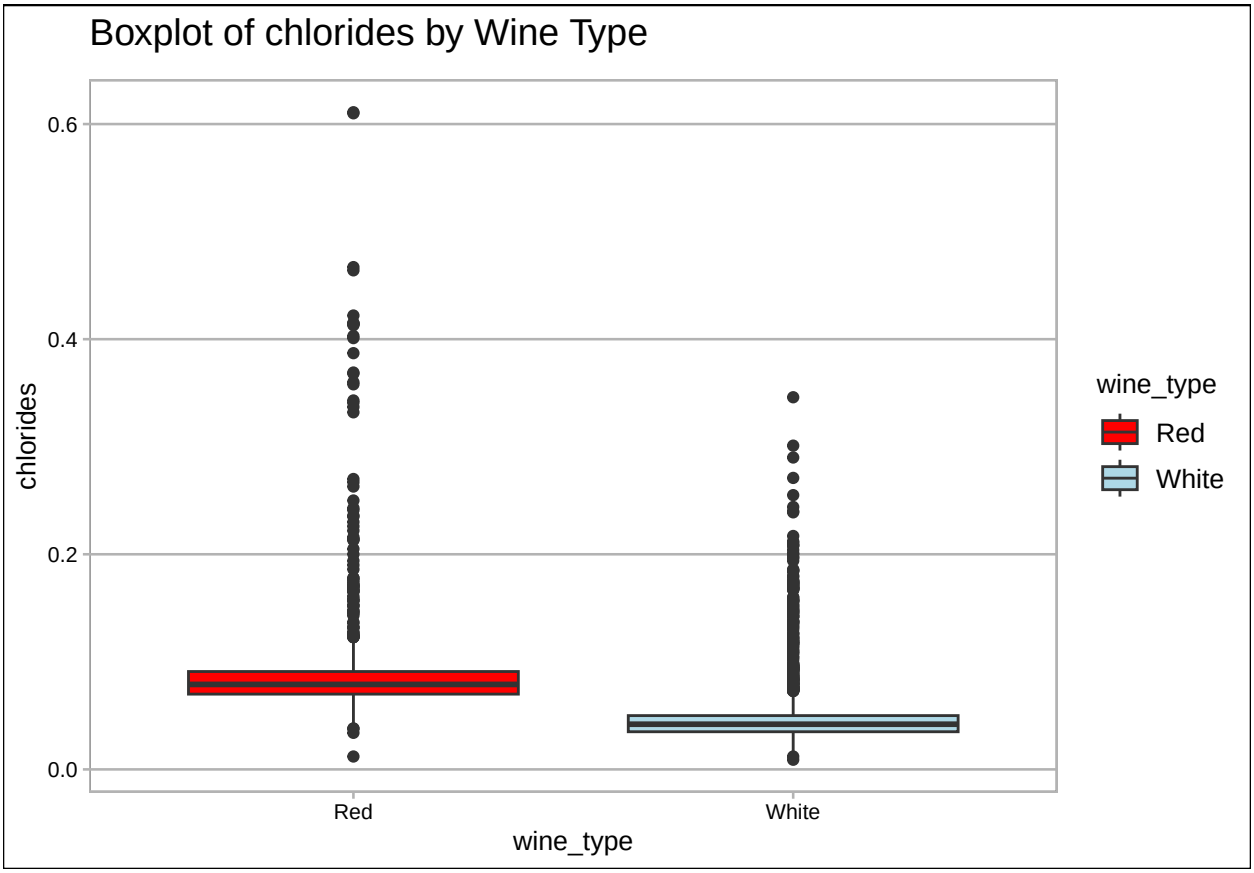



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.86654, p-value < 2.2e-16
```

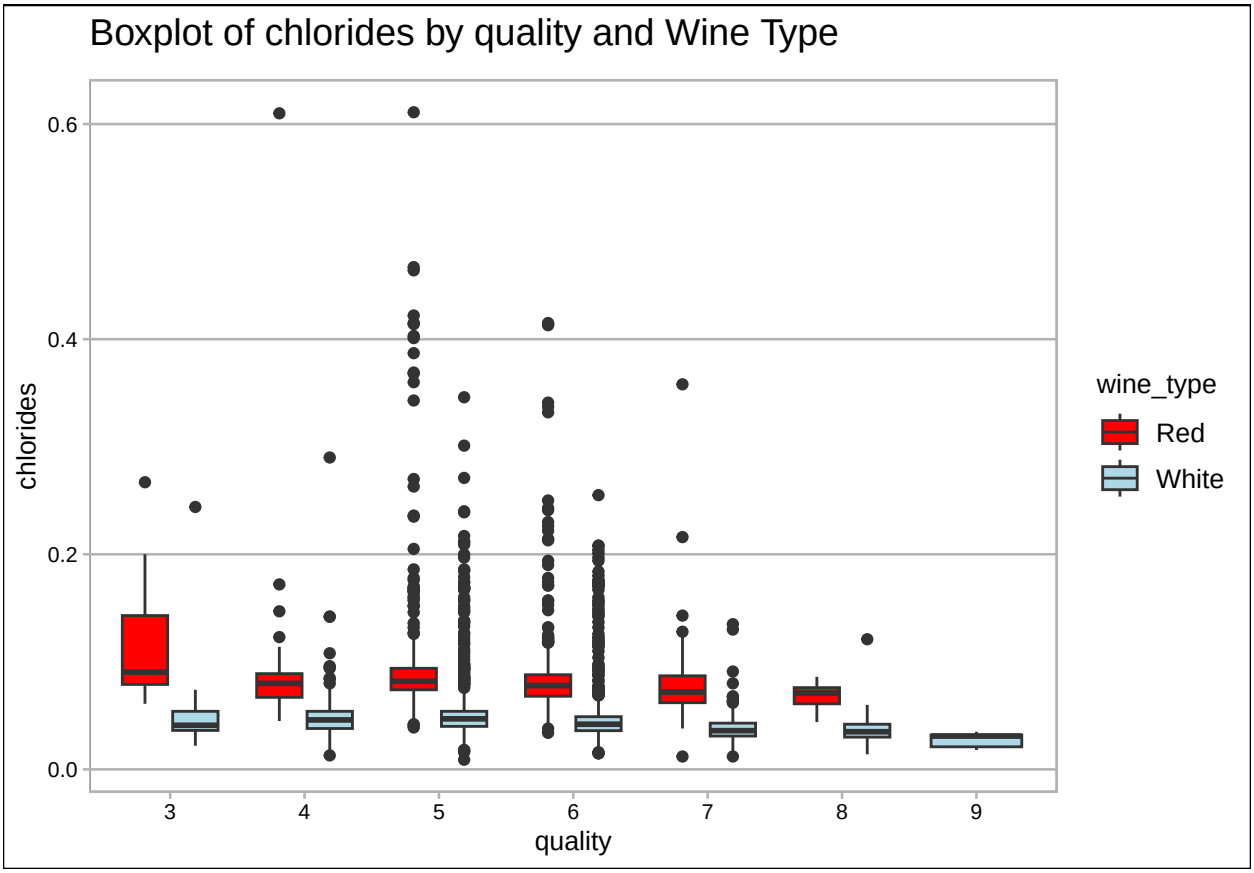
```
# chlorides
plot_density_plot(wine_data, 'chlorides')
```



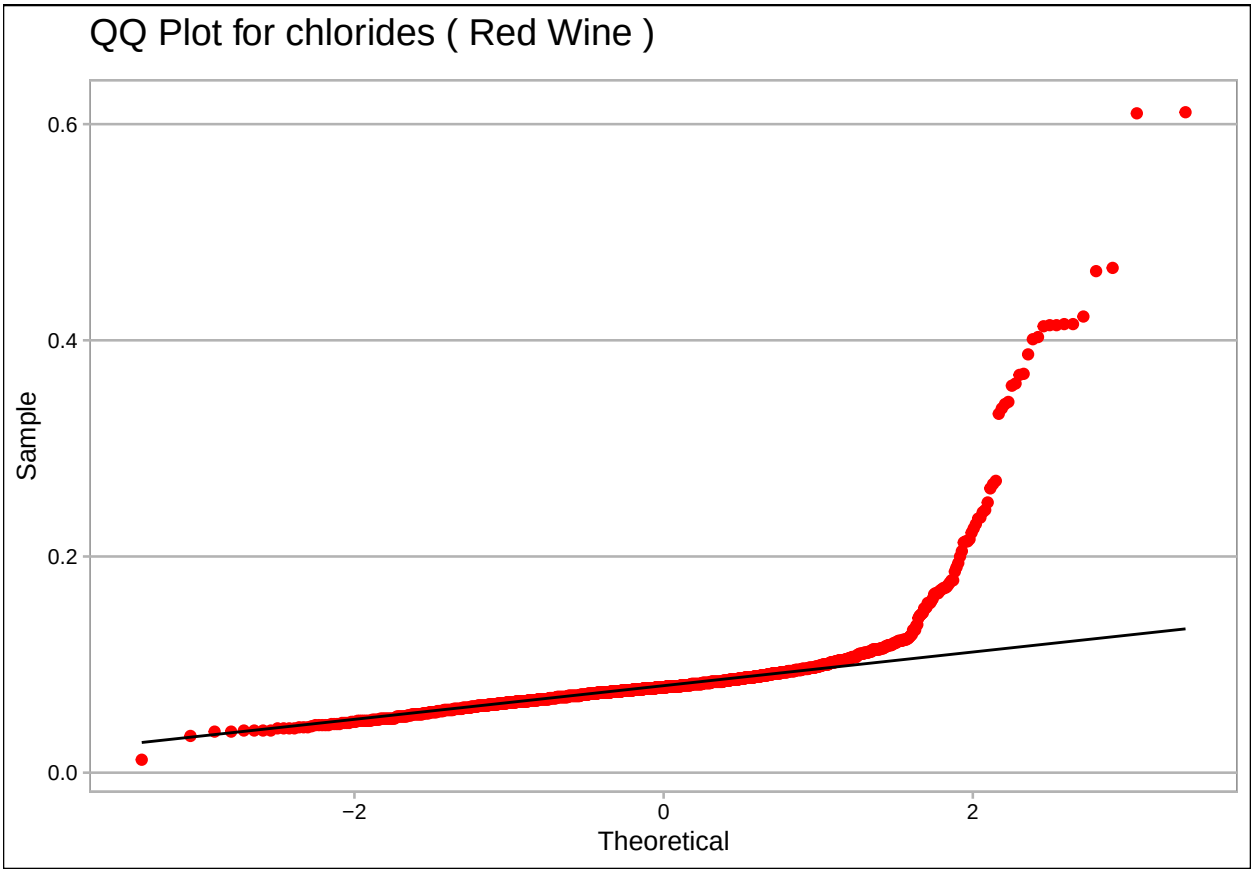

```
plot_boxplot(wine_data, 'chlorides')
```




```
plot_boxplot_quality(wine_data, 'chlorides')
```

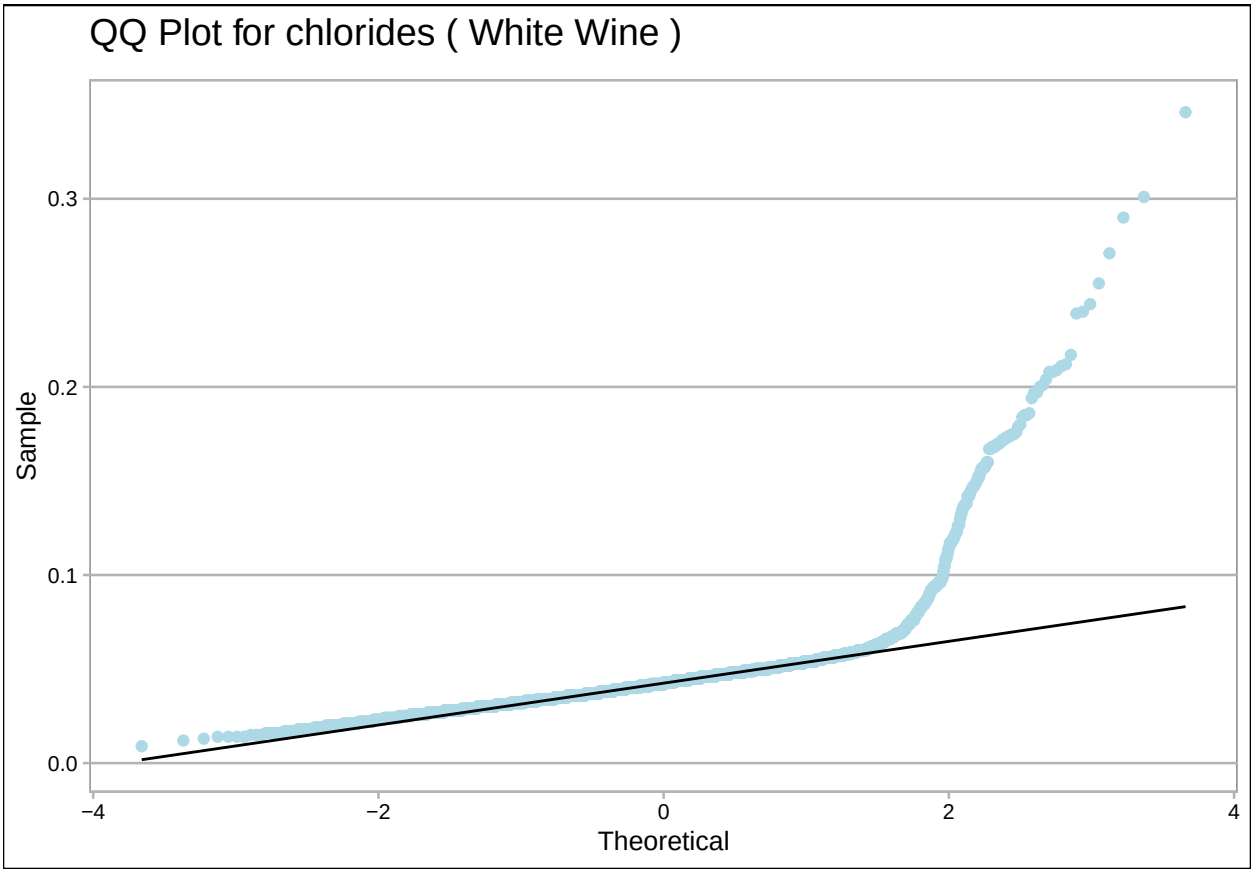



```
#performing normality check for red wine (chlorides)
normality_tester(red_wine, 'chlorides', 'red','Red Wine')
```



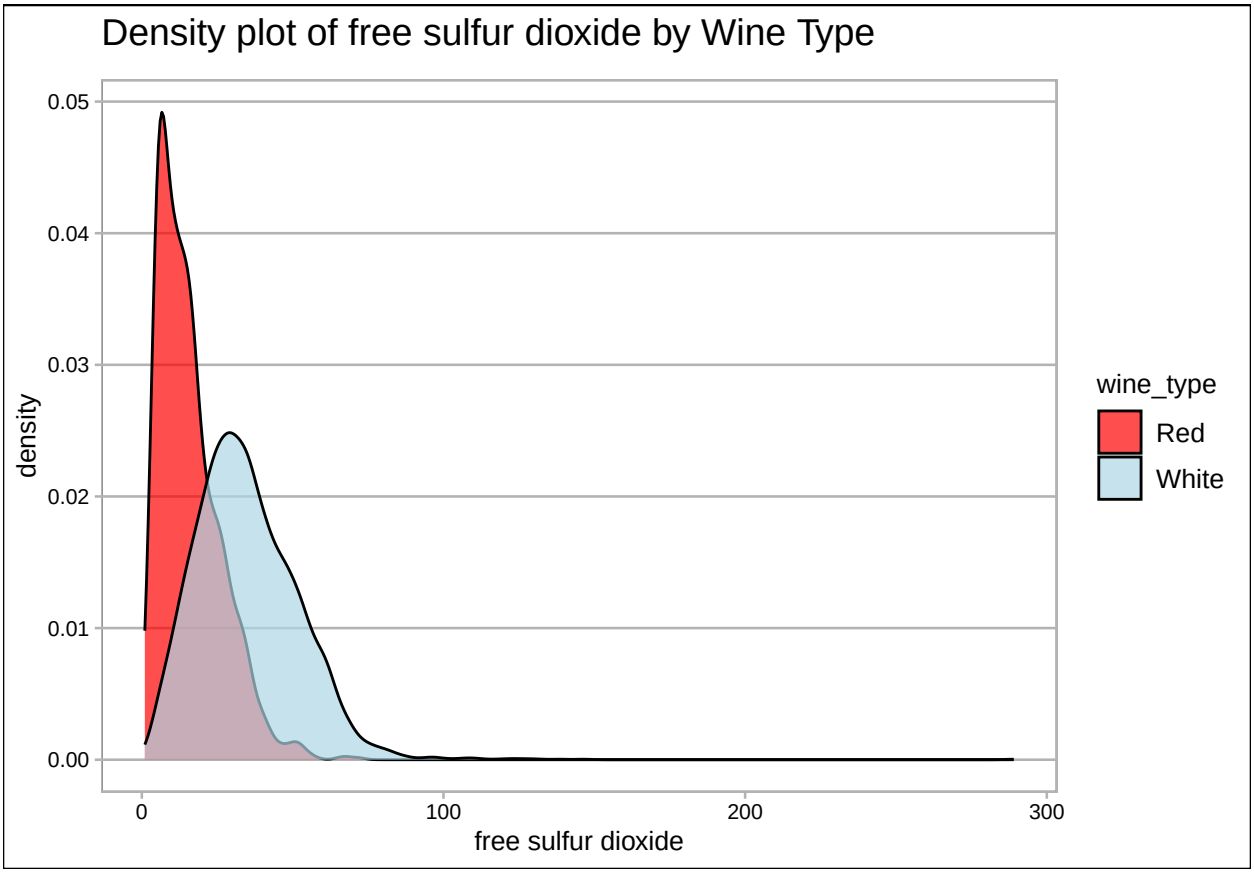

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.48448, p-value < 2.2e-16
```

```
#performing normality check for white wine (chlorides)
normality_tester(white_wine, 'chlorides', 'lightblue', "White Wine")
```

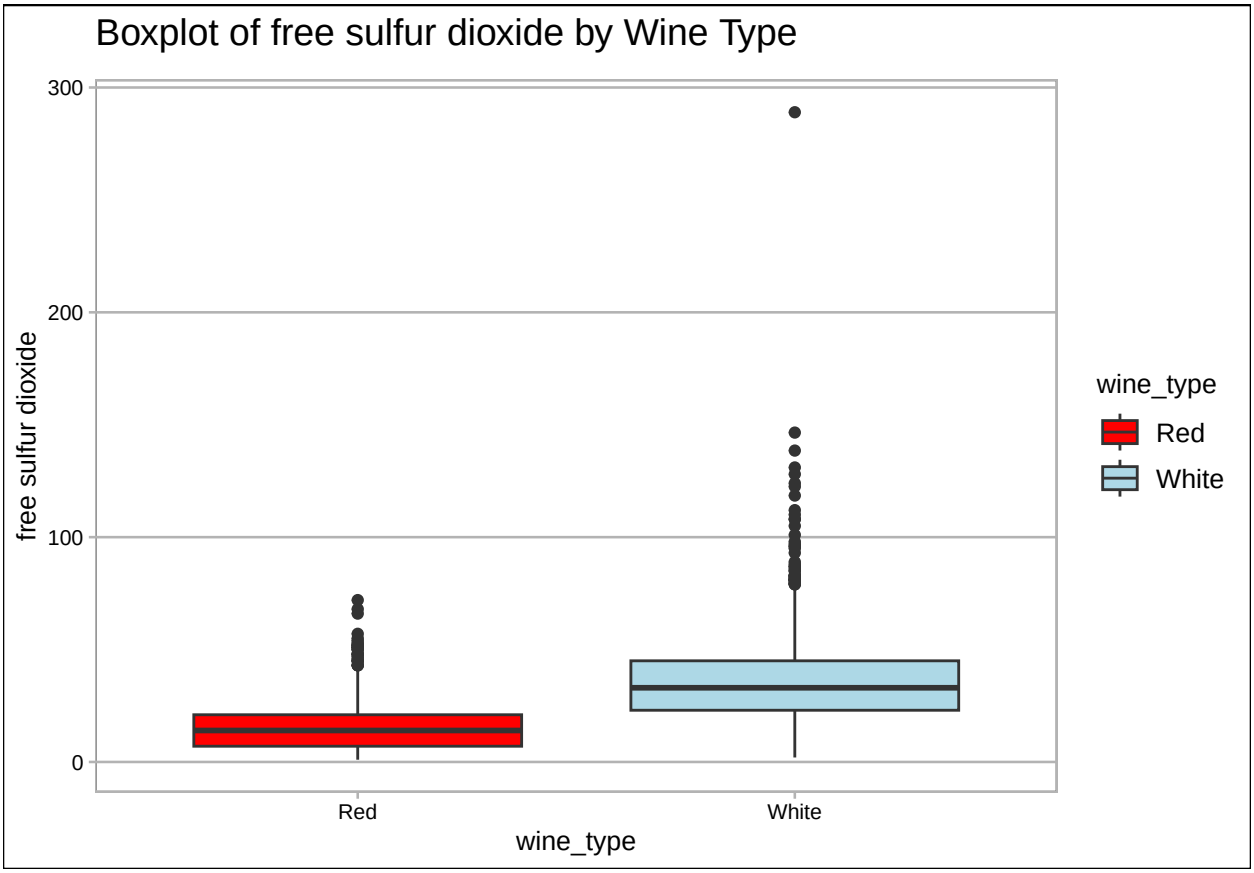



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.57789, p-value < 2.2e-16
```

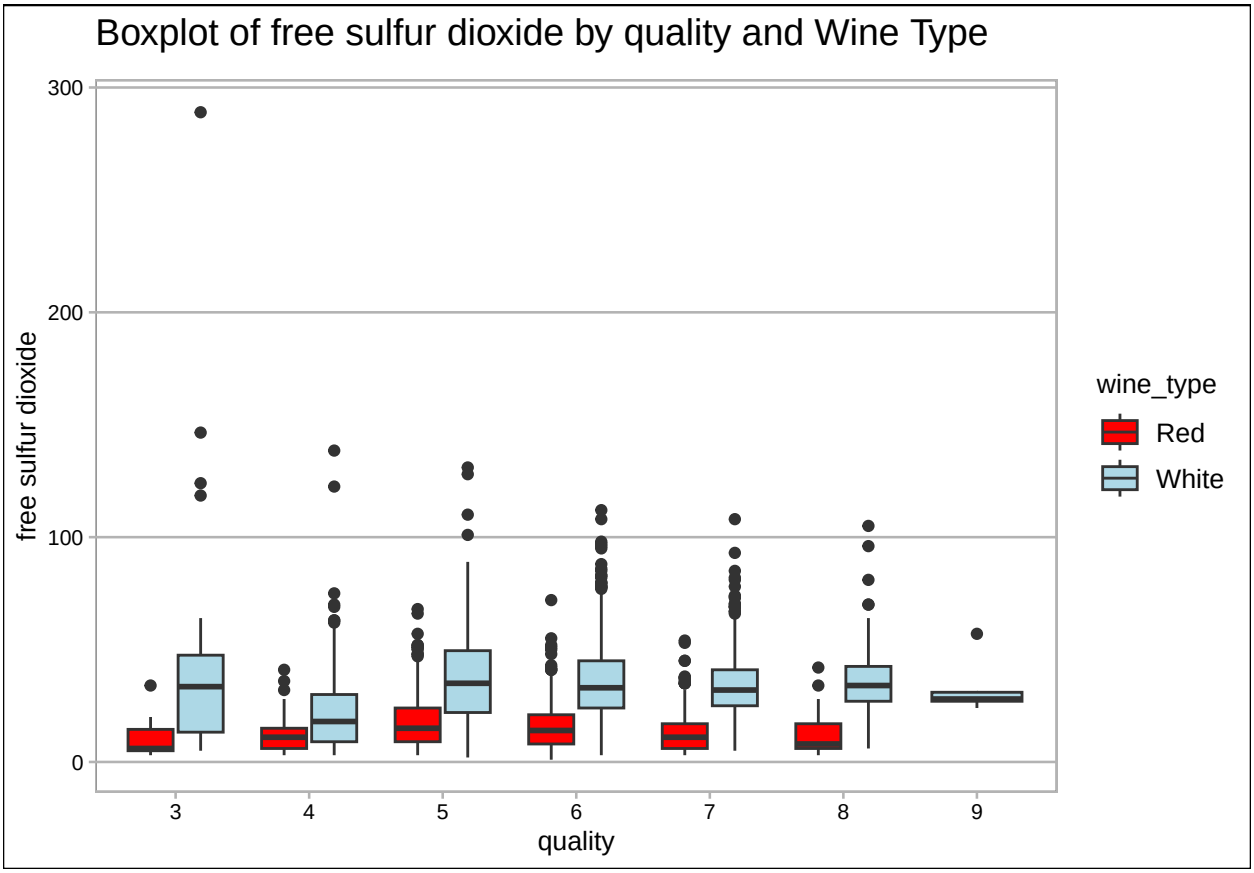
```
# free sulfur dioxide
plot_density_plot(wine_data, 'free sulfur dioxide')
```



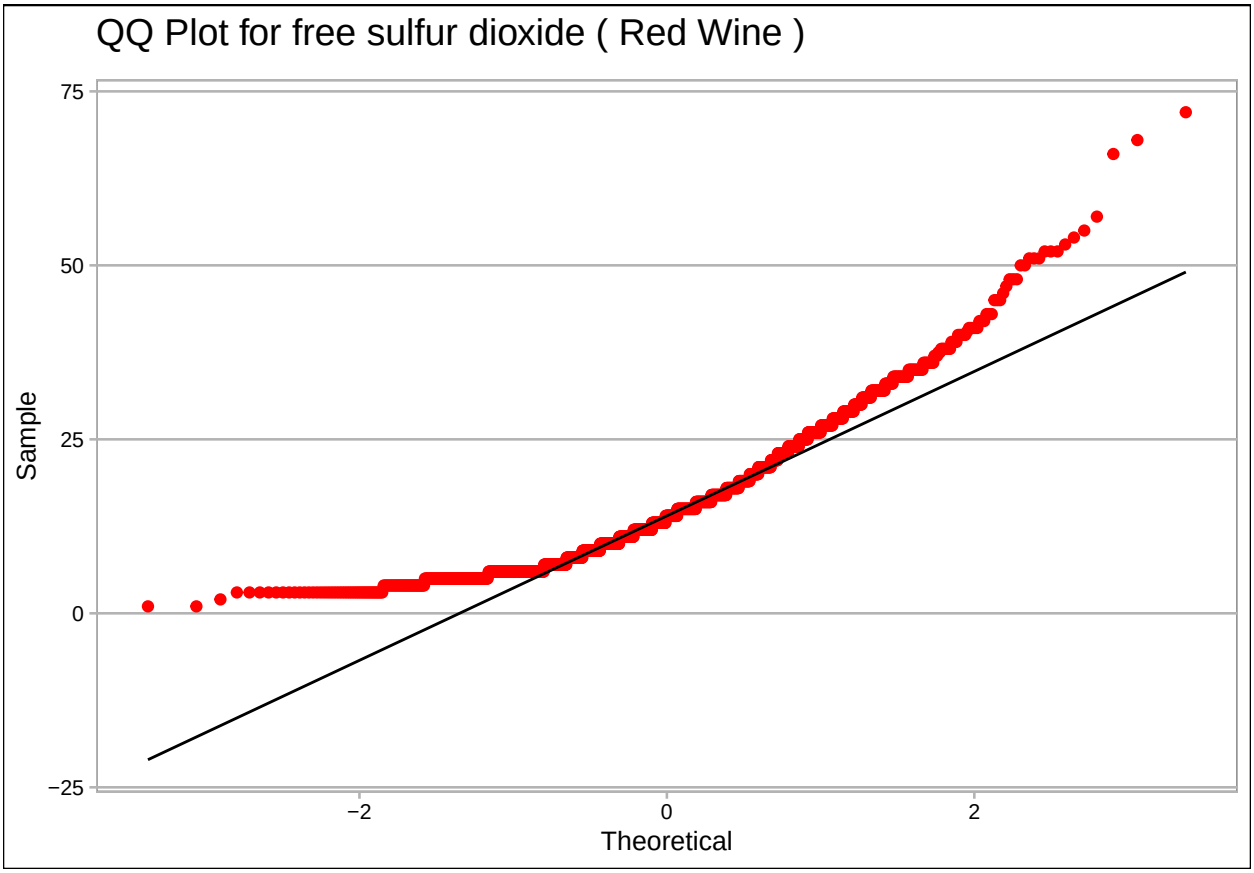

```
plot_boxplot(wine_data, 'free sulfur dioxide')
```




```
plot_boxplot_quality(wine_data, 'free sulfur dioxide')
```

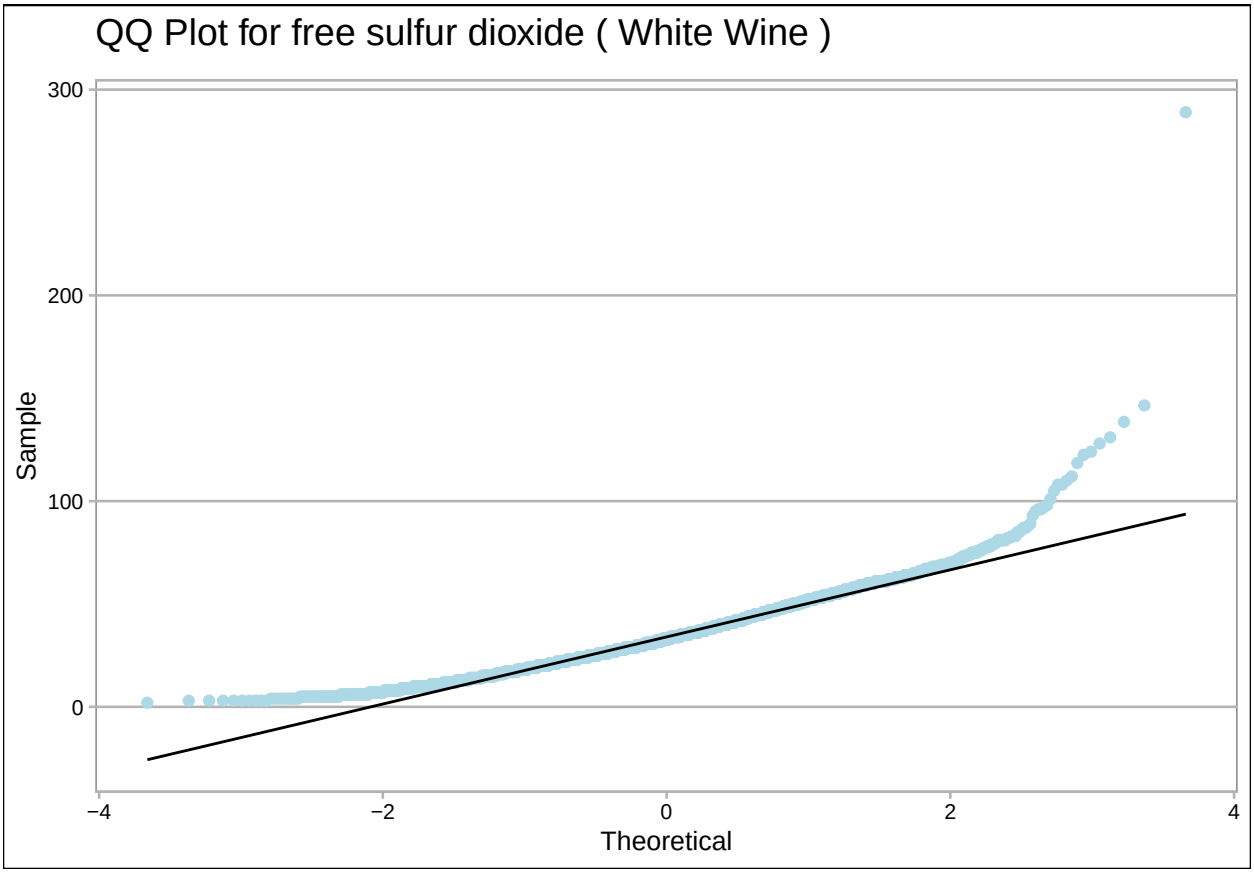



```
#performing normality check for red wine (free sulfur dioxide)
normality_tester(red_wine, 'free sulfur dioxide', 'red','Red Wine')
```



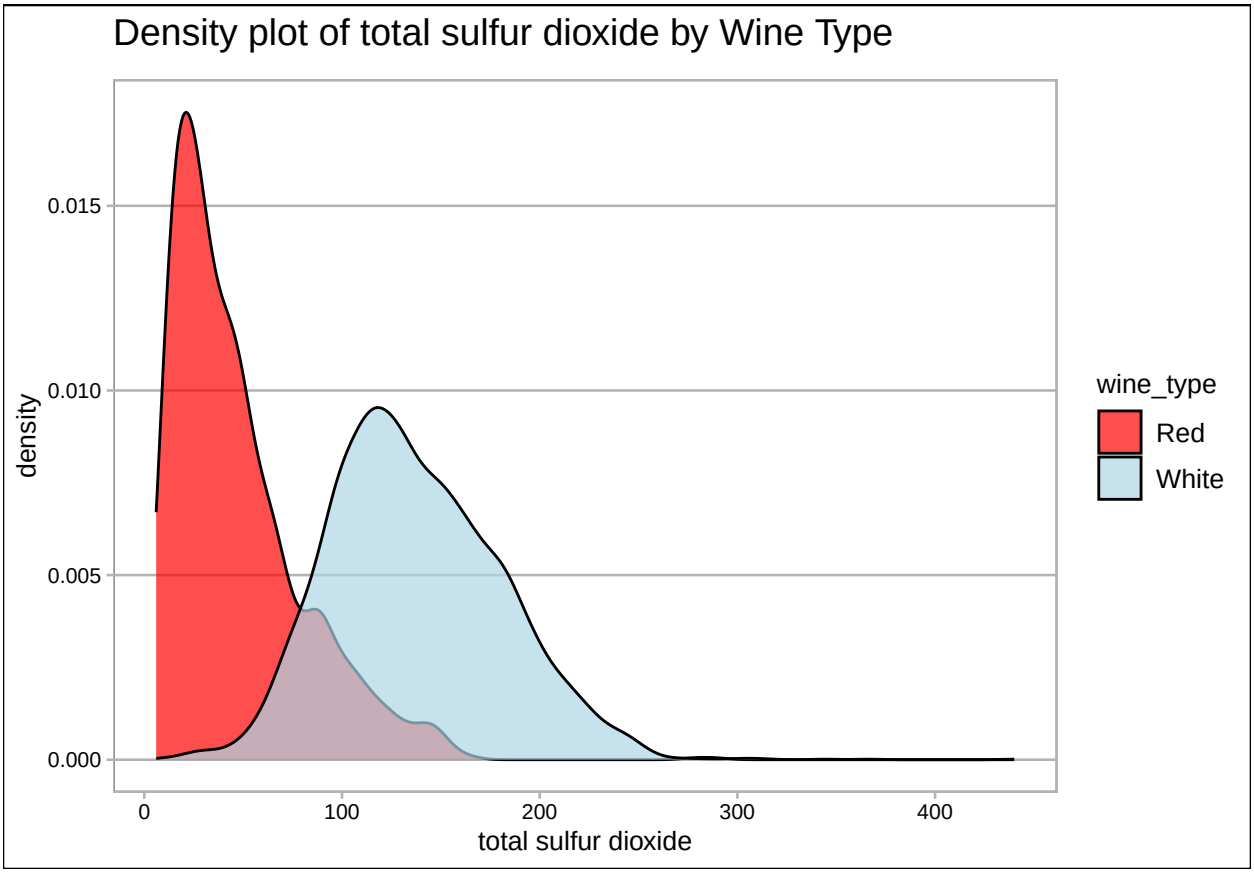

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.90323, p-value < 2.2e-16
```

```
#performing normality check for white wine (free sulfur dioxide)
normality_tester(white_wine, 'free sulfur dioxide', 'lightblue', "White Wine")
```

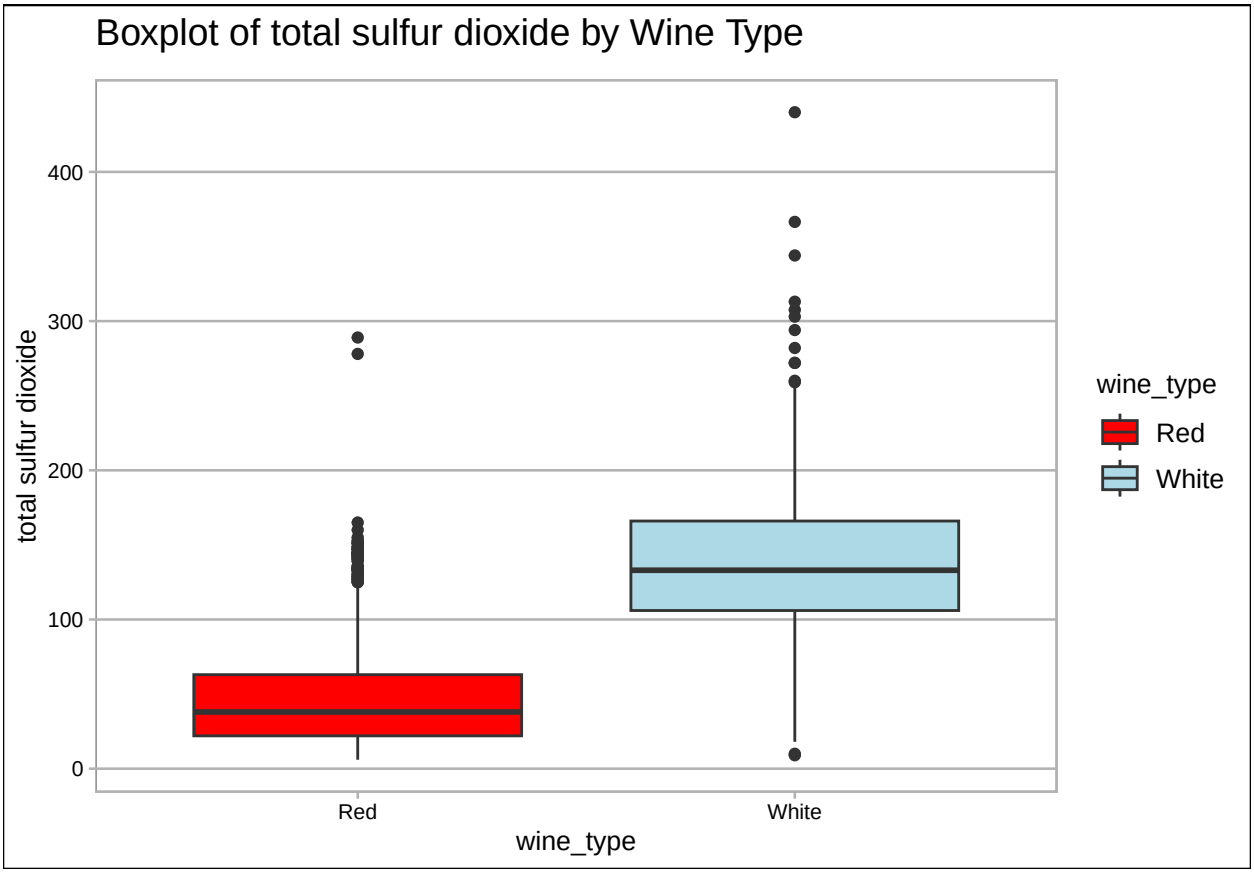



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.93339, p-value < 2.2e-16
```

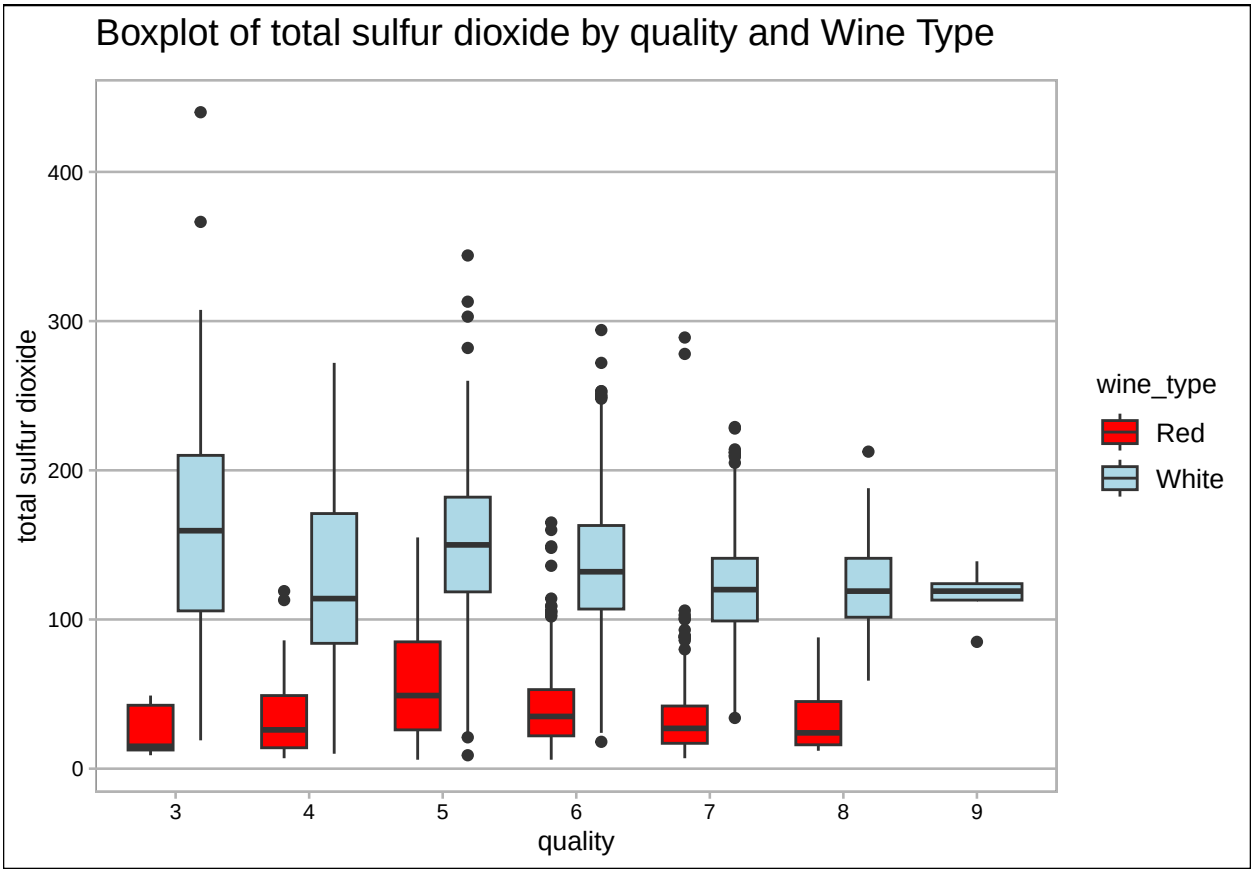
```
# total sulfur dioxide
plot_density_plot(wine_data, 'total sulfur dioxide')
```



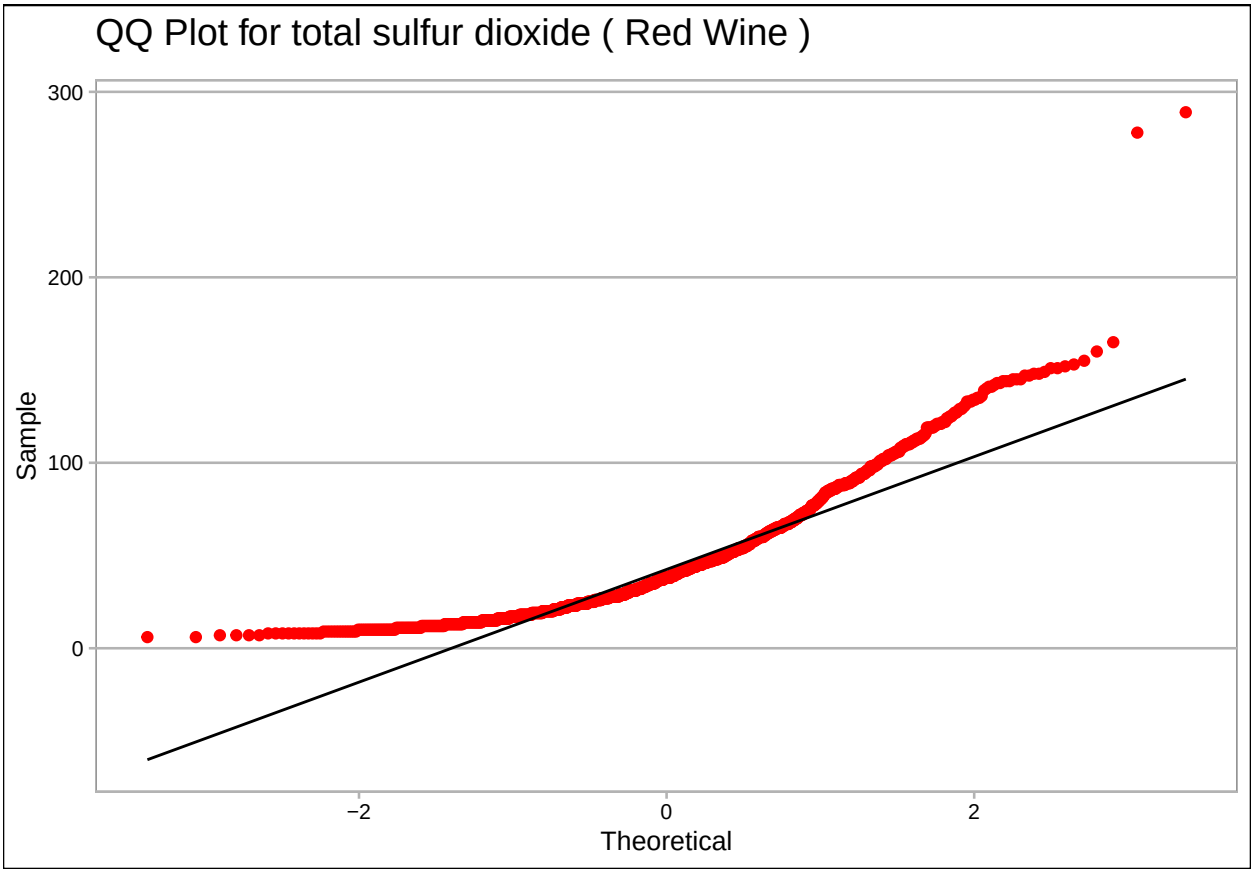

```
plot_boxplot(wine_data, 'total sulfur dioxide')
```




```
plot_boxplot_quality(wine_data, 'total sulfur dioxide')
```

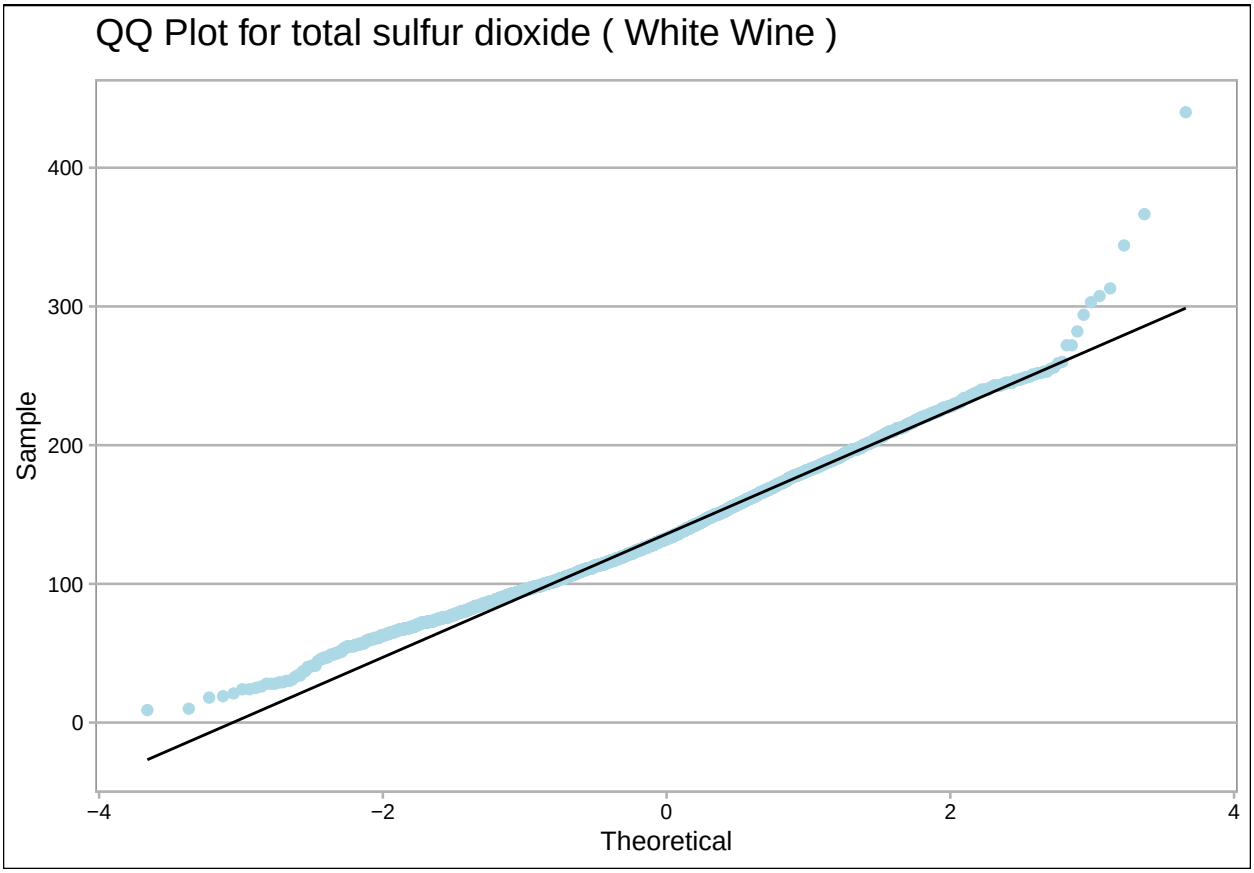



```
#performing normality check for red wine (total sulfur dioxide)
normality_tester(red_wine, 'total sulfur dioxide', 'red','Red Wine')
```



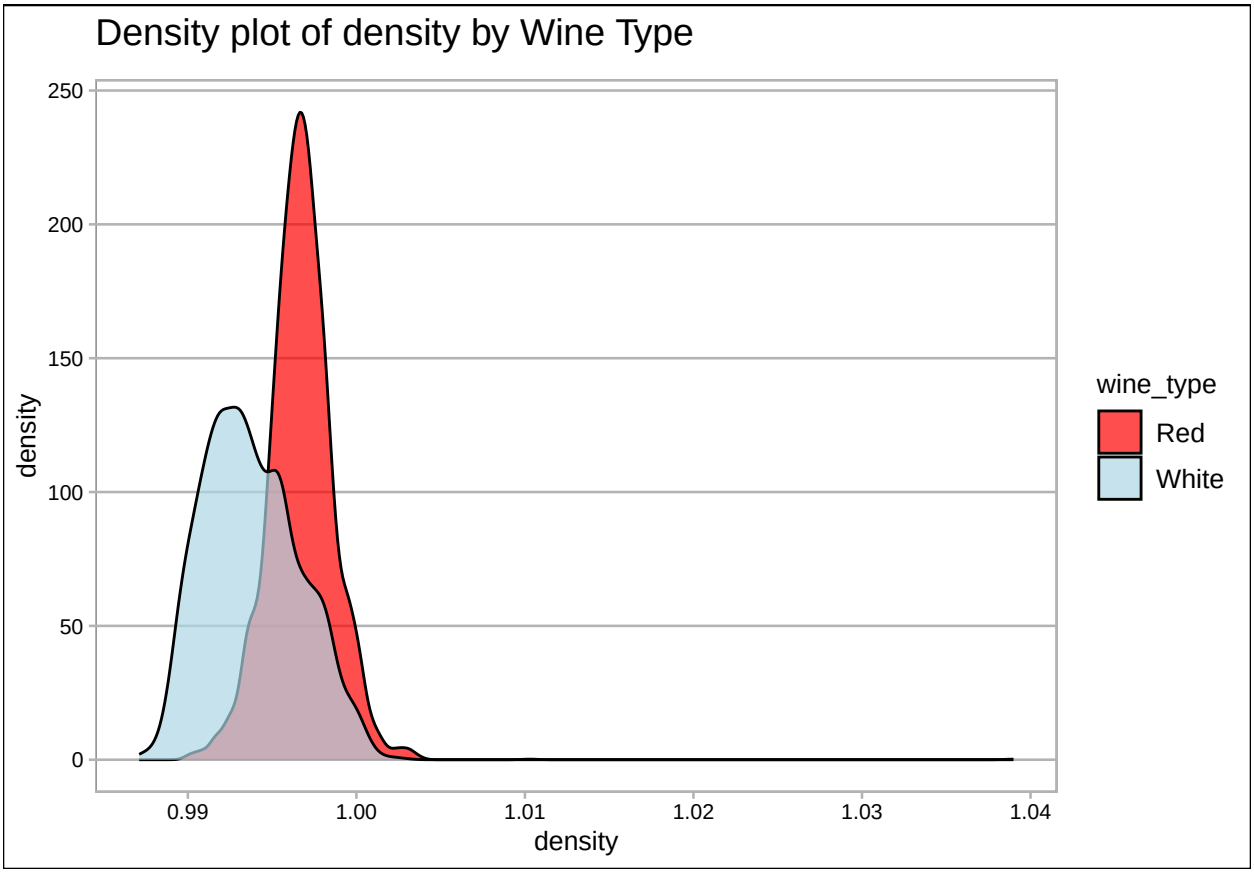

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.87169, p-value < 2.2e-16
```

```
#performing normality check for white wine (total sulfur dioxide)
normality_tester(white_wine, 'total sulfur dioxide', 'lightblue', "White Wine")
```

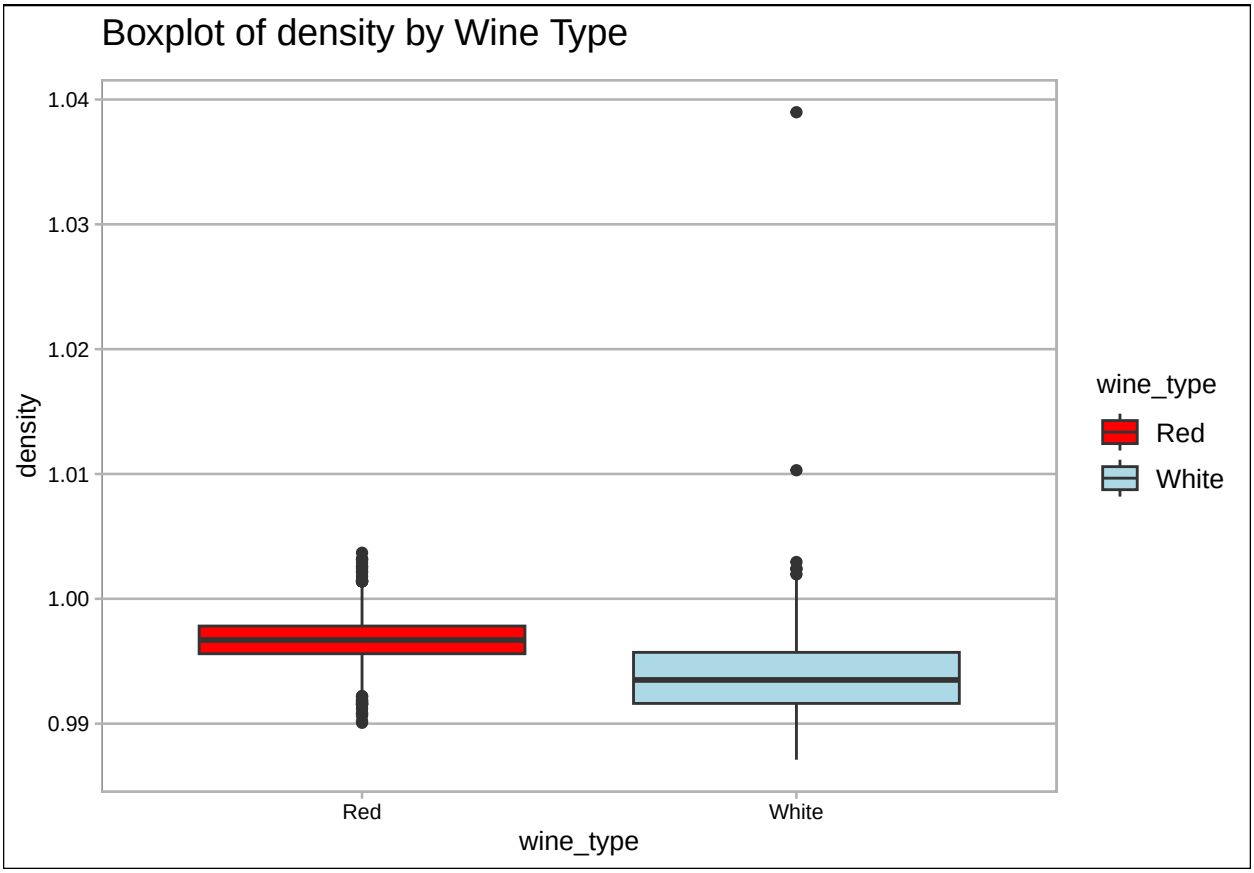



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.98638, p-value < 2.2e-16
```

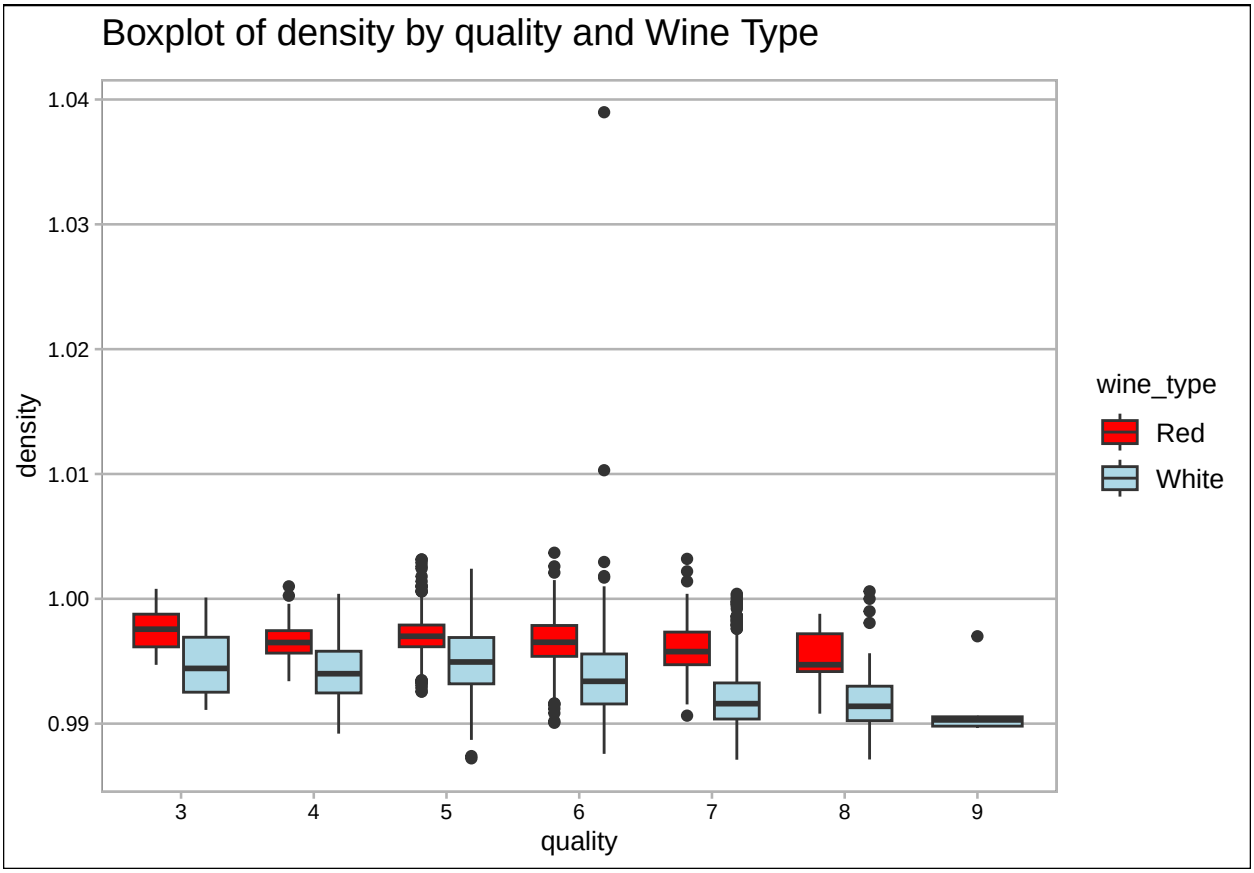
```
# density
plot_density_plot(wine_data, 'density')
```



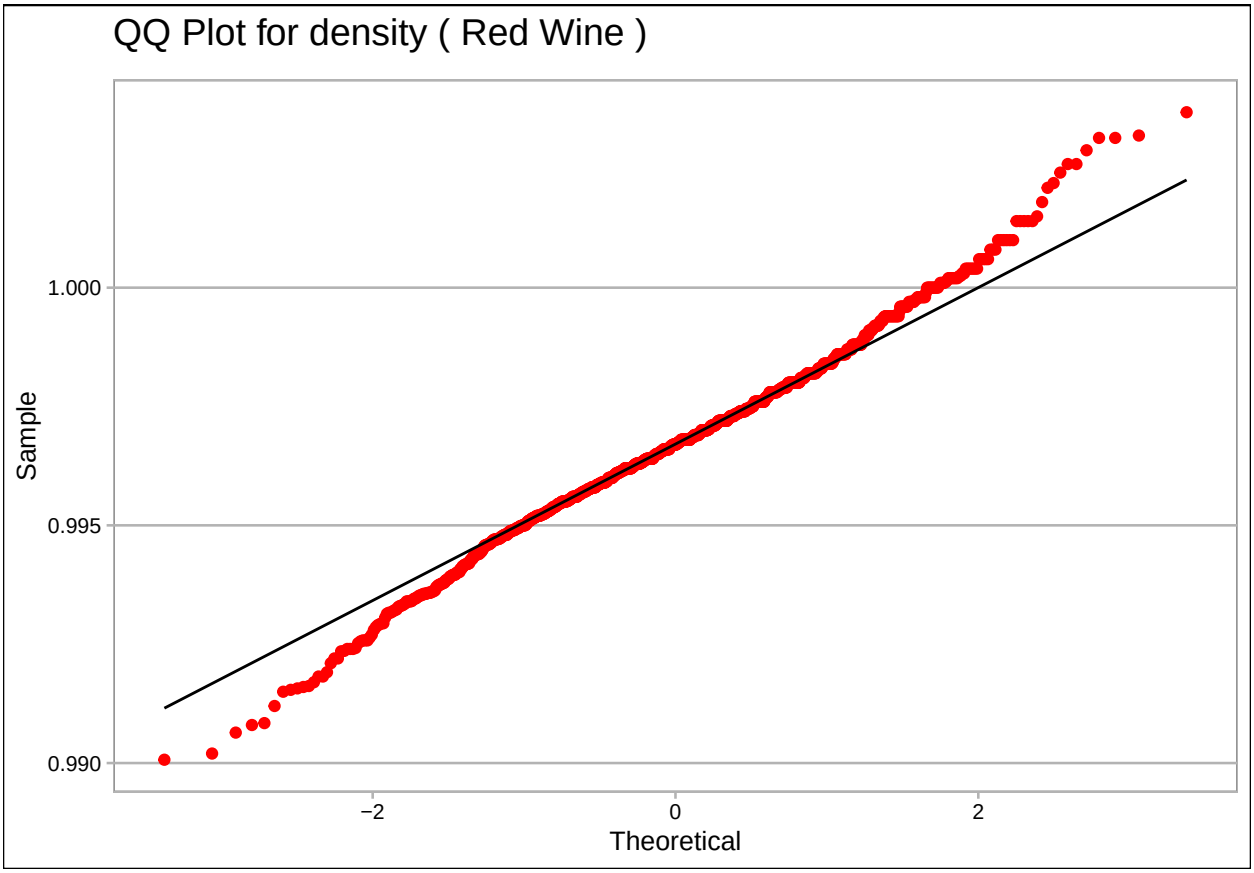

```
plot_boxplot(wine_data, 'density')
```




```
plot_boxplot_quality(wine_data, 'density')
```

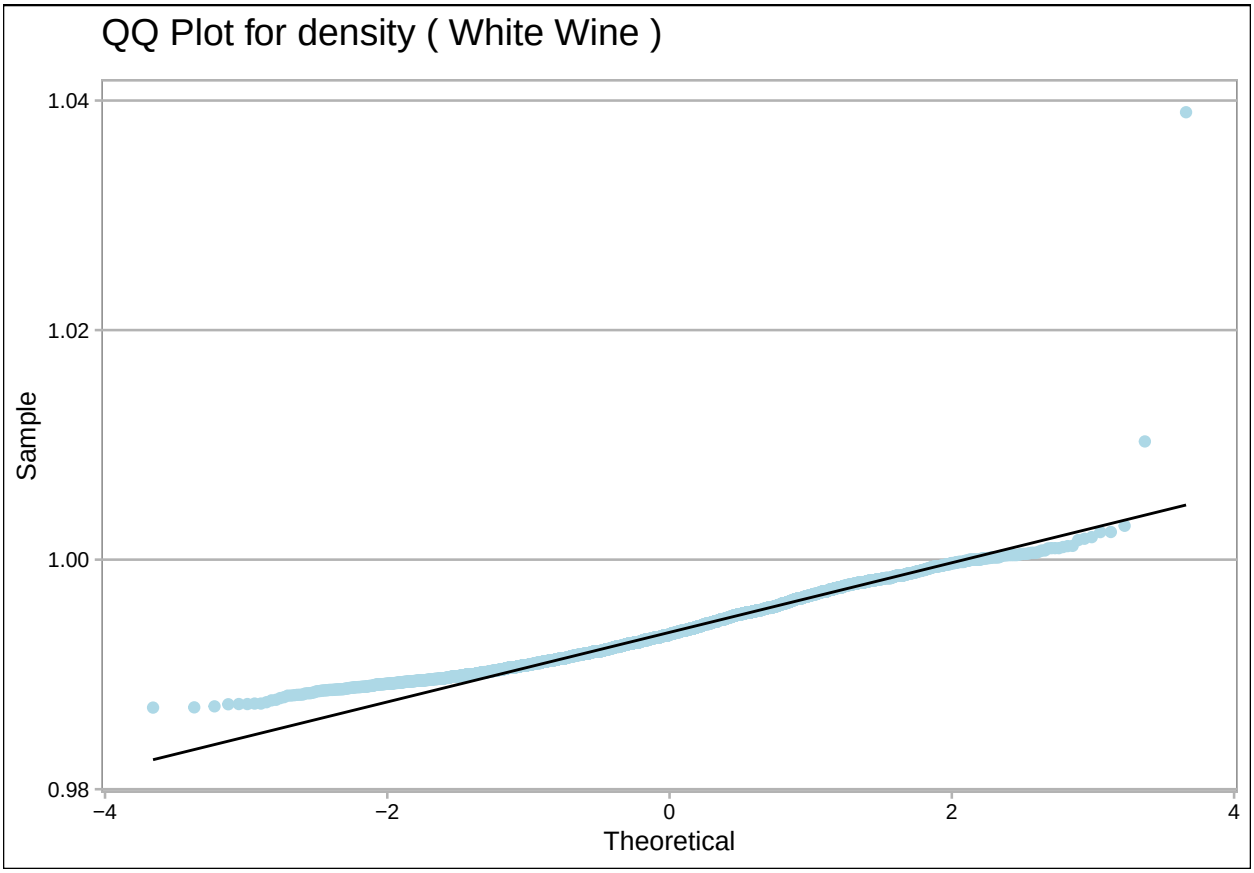



```
#performing normality check for red wine (density)
normality_tester(red_wine, 'density', 'red','Red Wine')
```



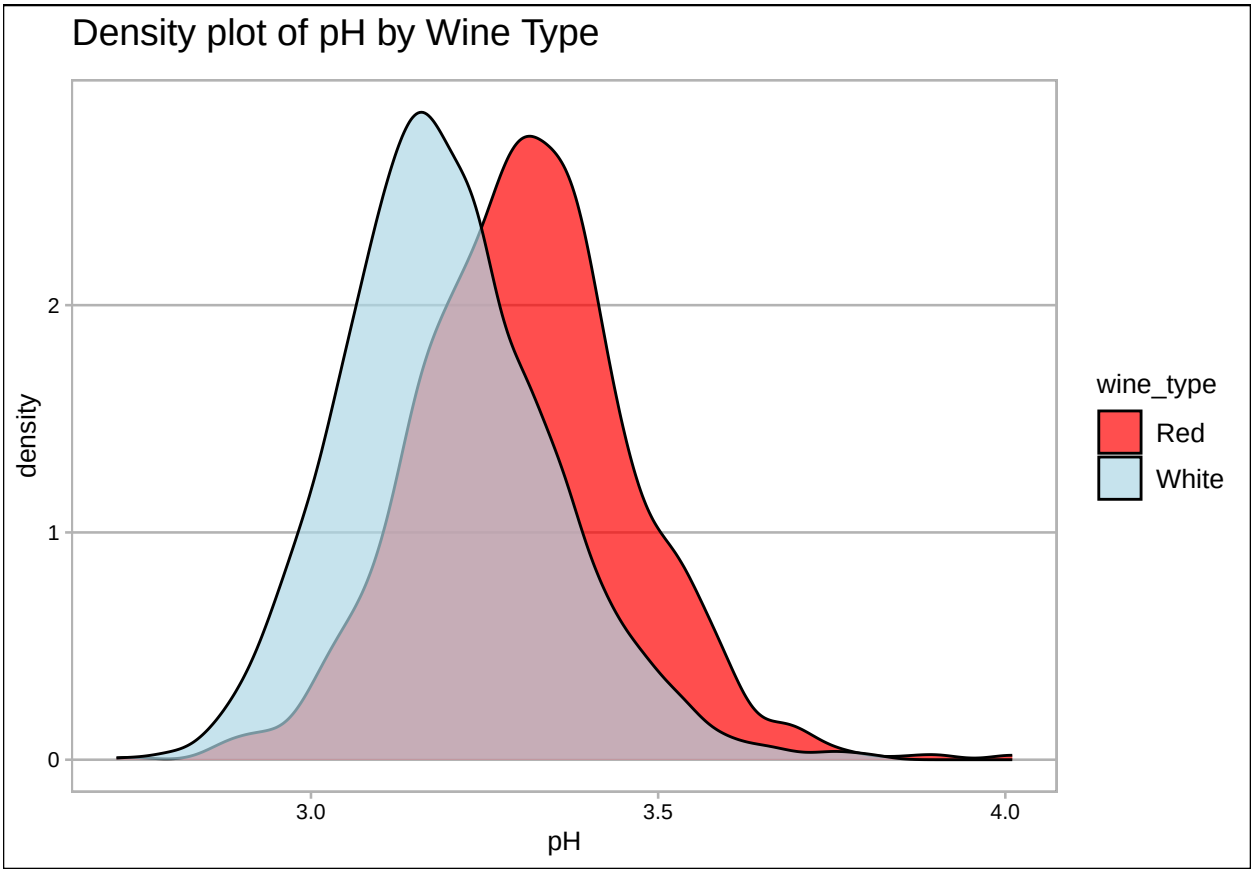

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.99239, p-value = 1.804e-06
```

```
#performing normality check for white wine (density)
normality_tester(white_wine, 'density', 'lightblue', "White Wine")
```

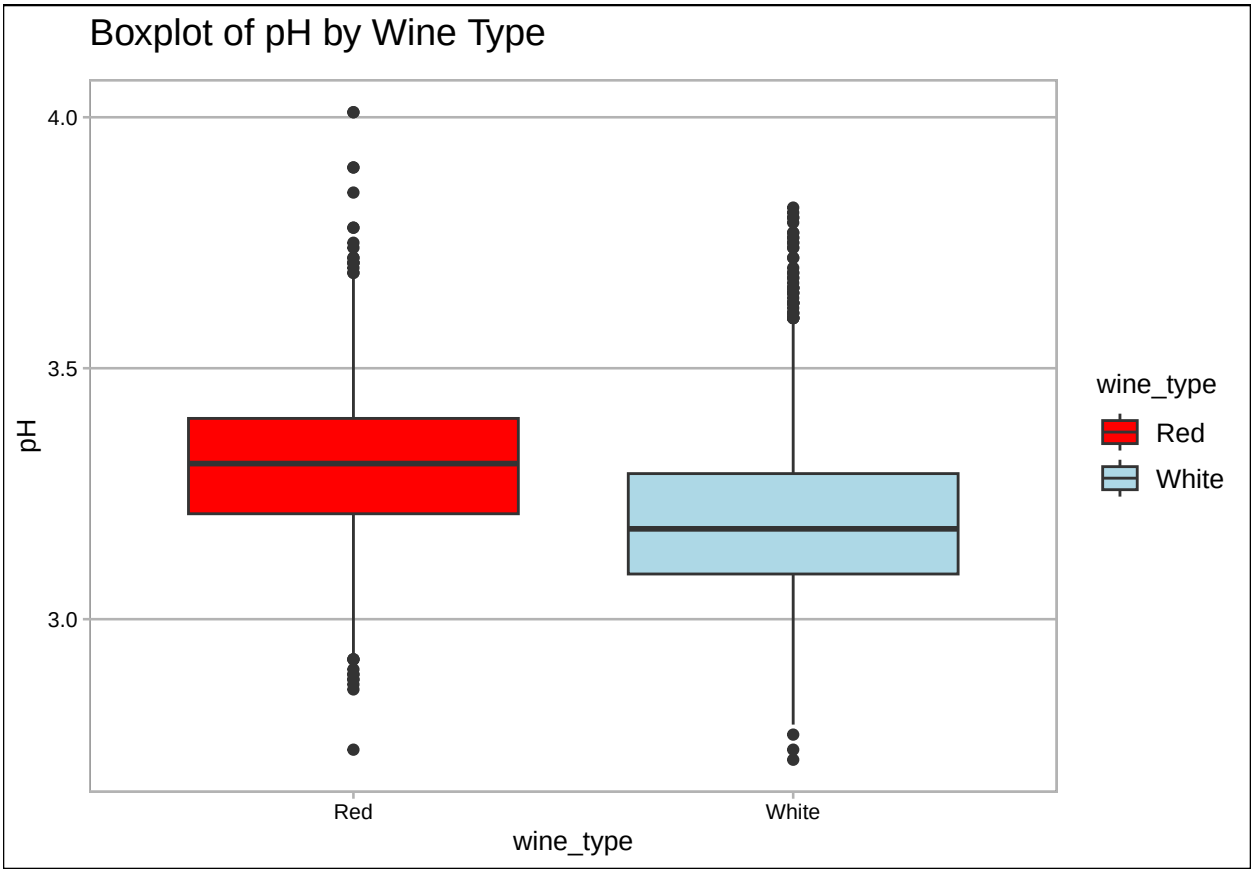



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.94739, p-value < 2.2e-16
```

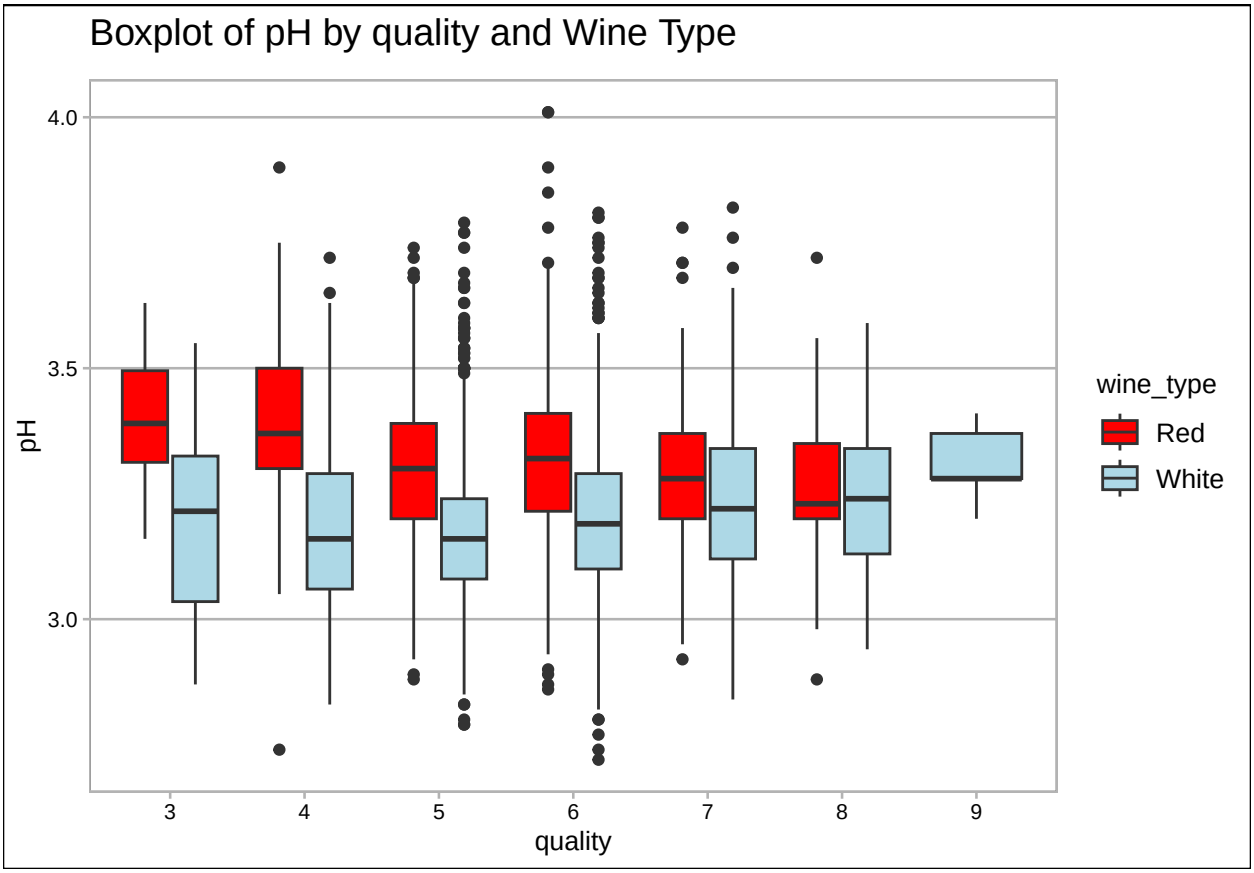
```
# pH
plot_density_plot(wine_data, 'pH')
```



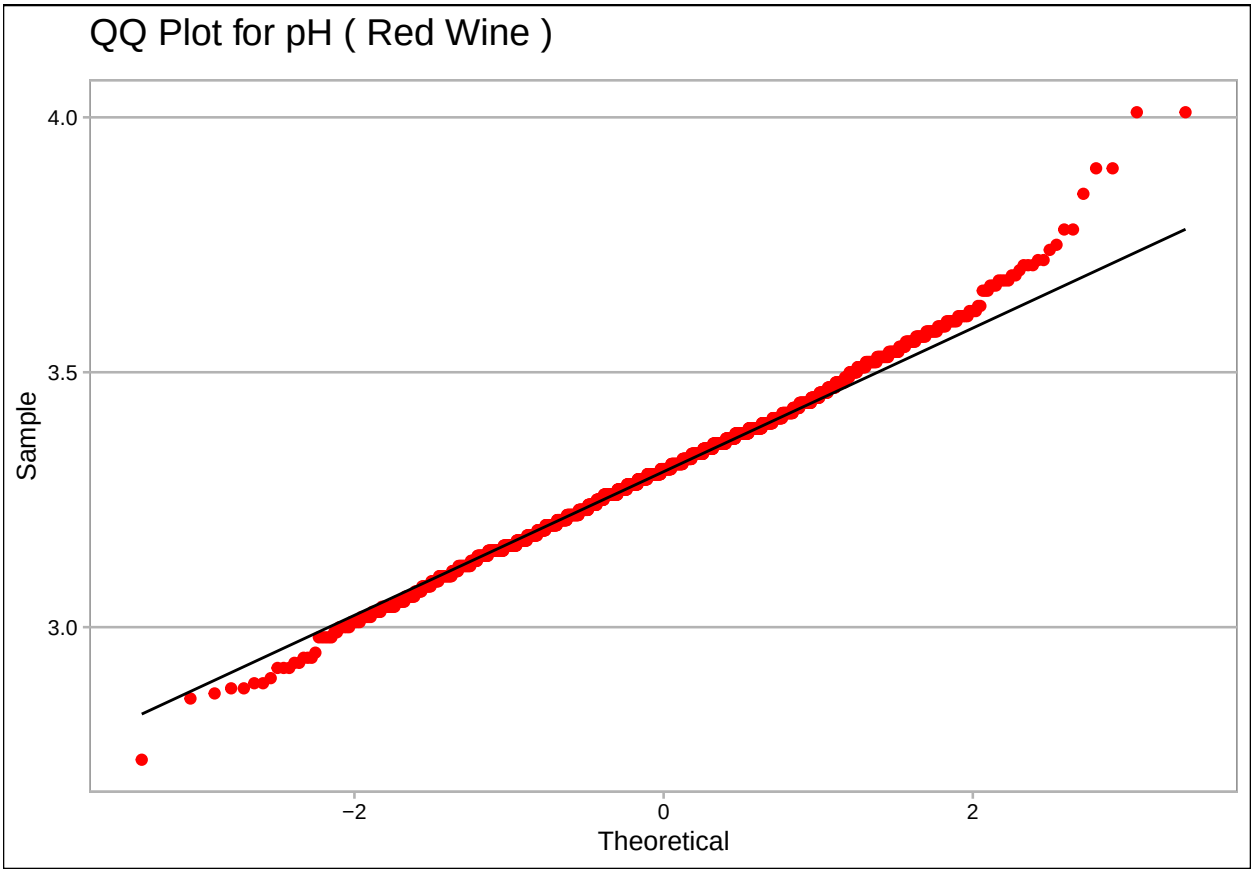

```
plot_boxplot(wine_data, 'pH')
```




```
plot_boxplot_quality(wine_data, 'pH')
```

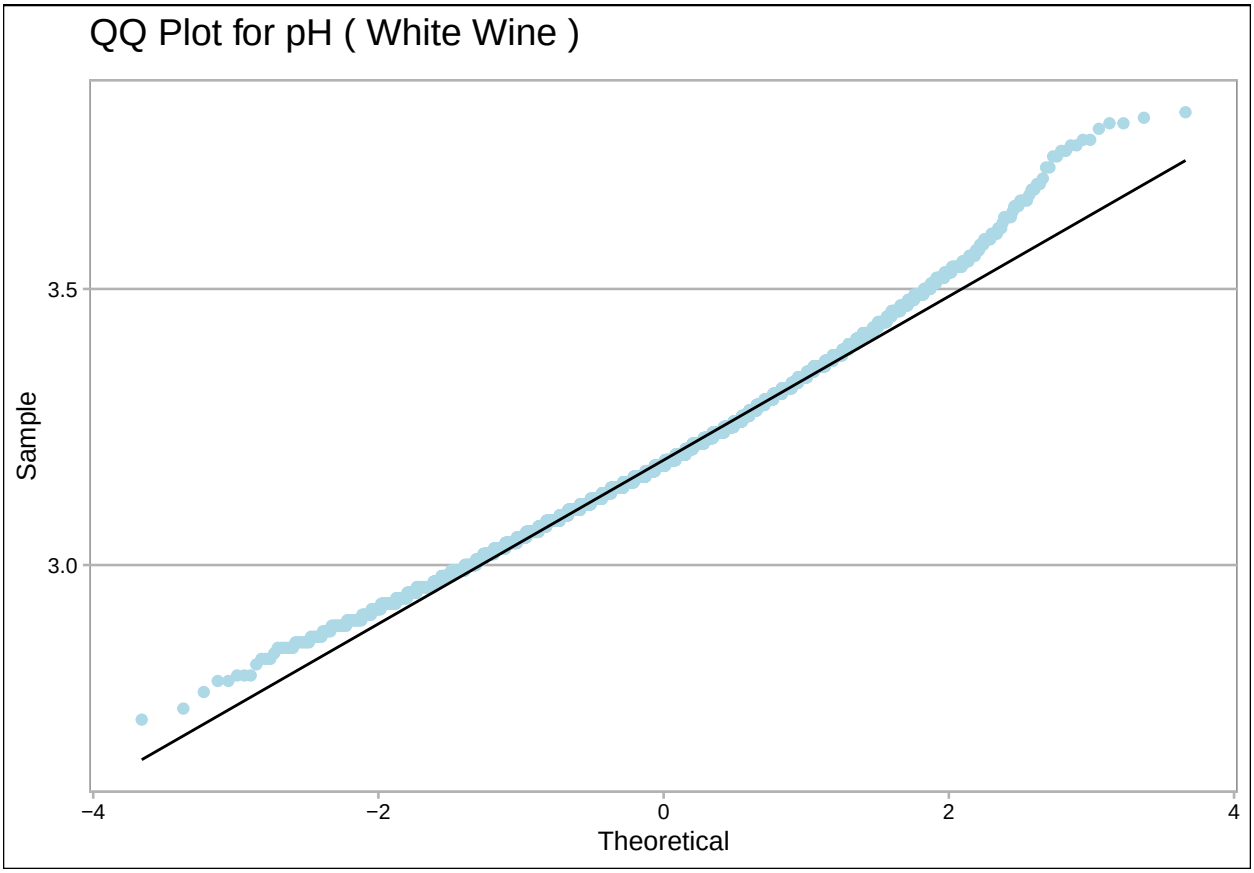



```
#performing normality check for red wine (pH)
normality_tester(red_wine, 'pH', 'red','Red Wine')
```



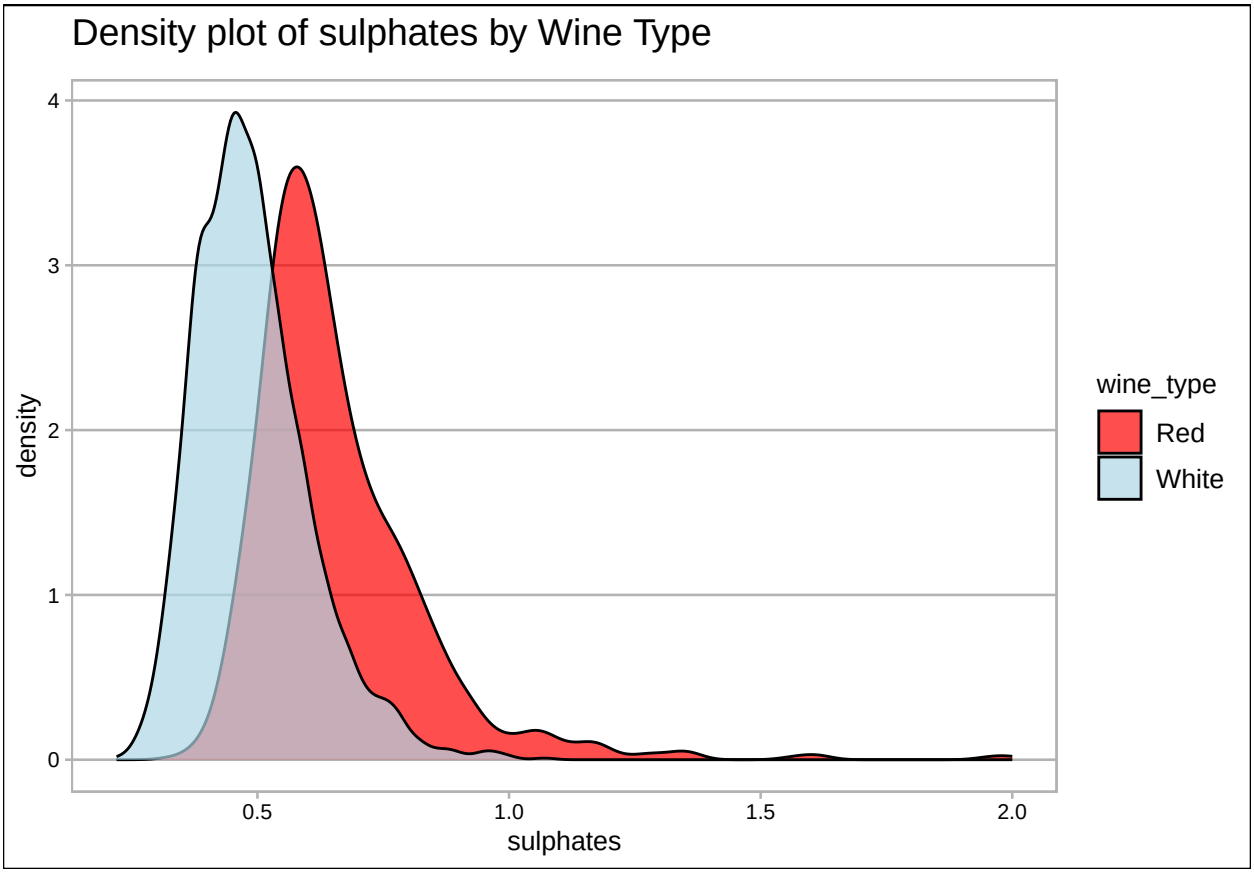

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.99272, p-value = 3.082e-06
```

```
#performing normality check for white wine (pH)
normality_tester(white_wine, 'pH', 'lightblue', "White Wine")
```

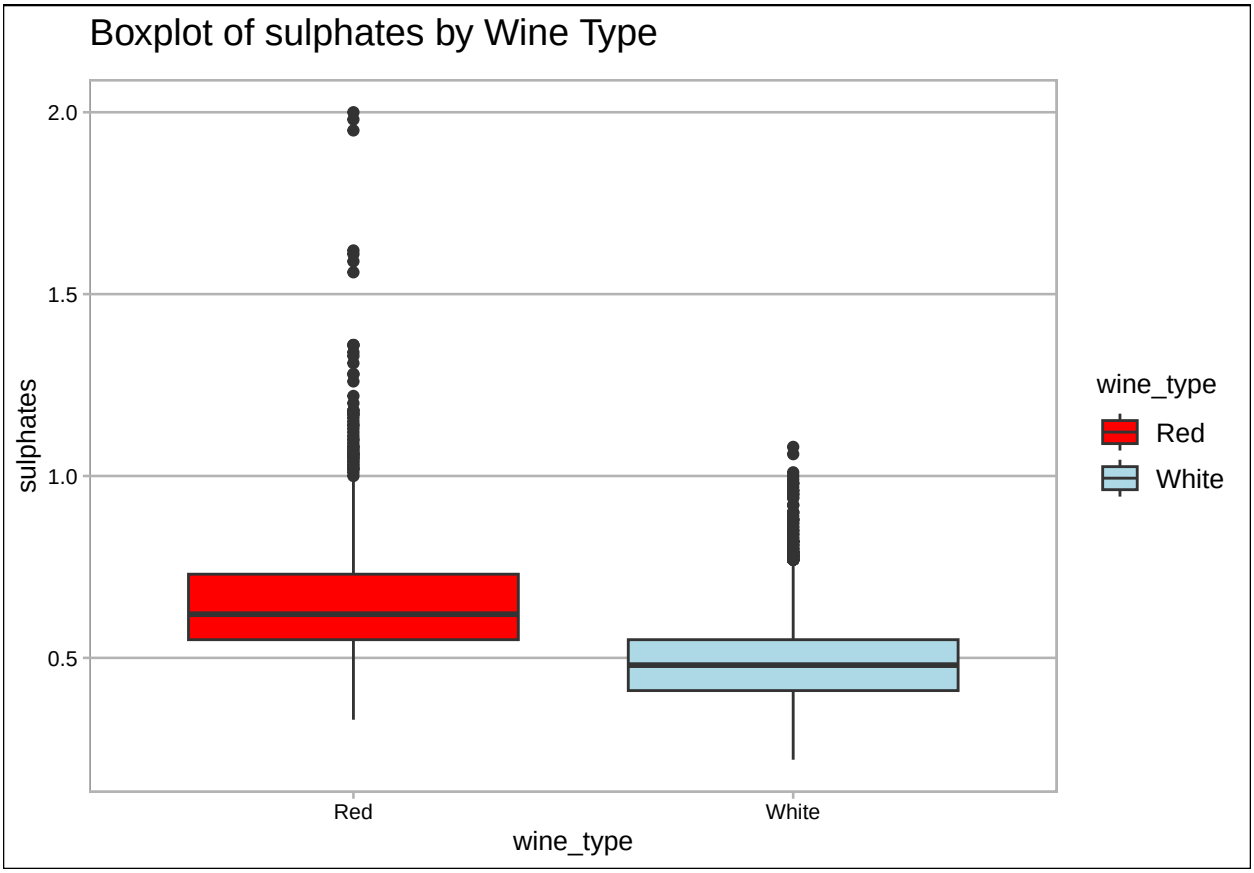



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.98817, p-value < 2.2e-16
```

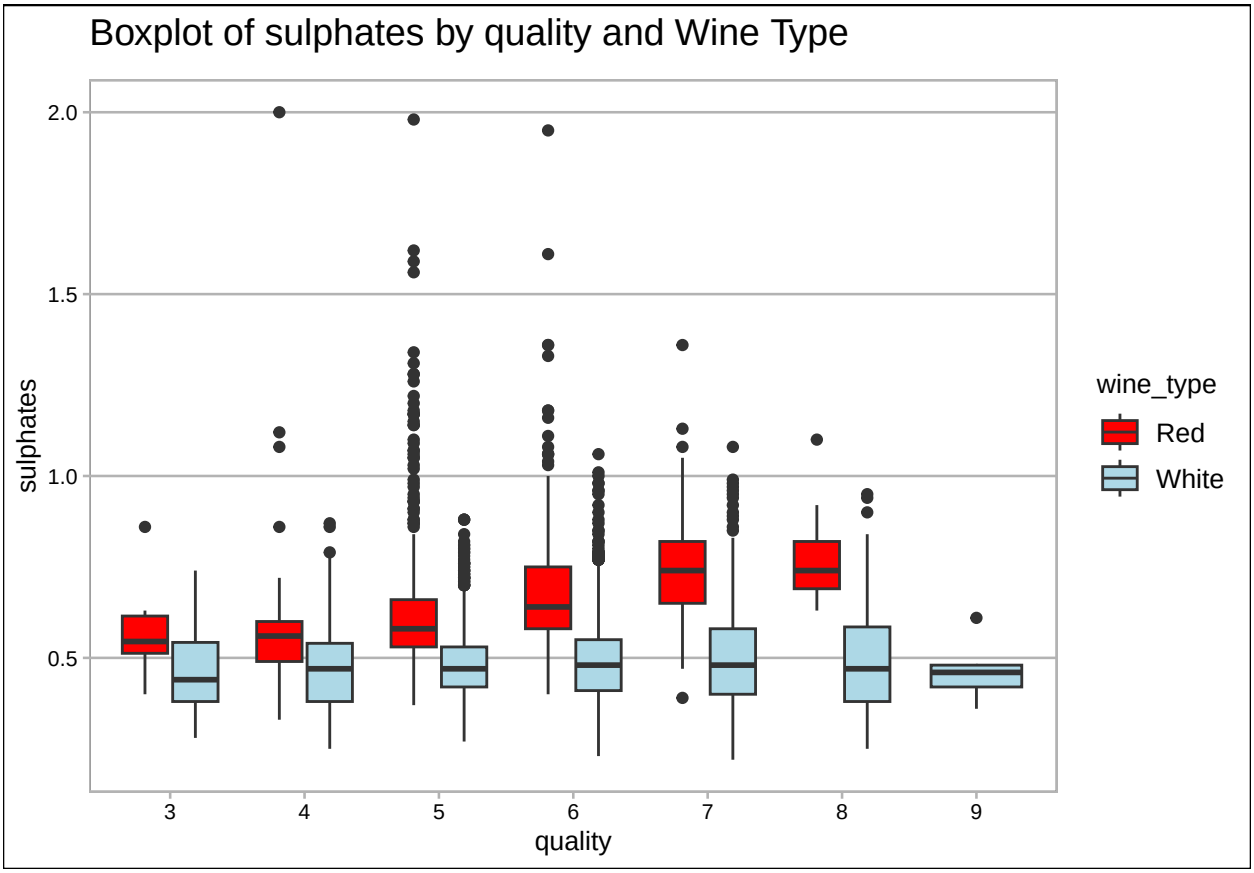
```
# sulphates
plot_density_plot(wine_data, 'sulphates')
```



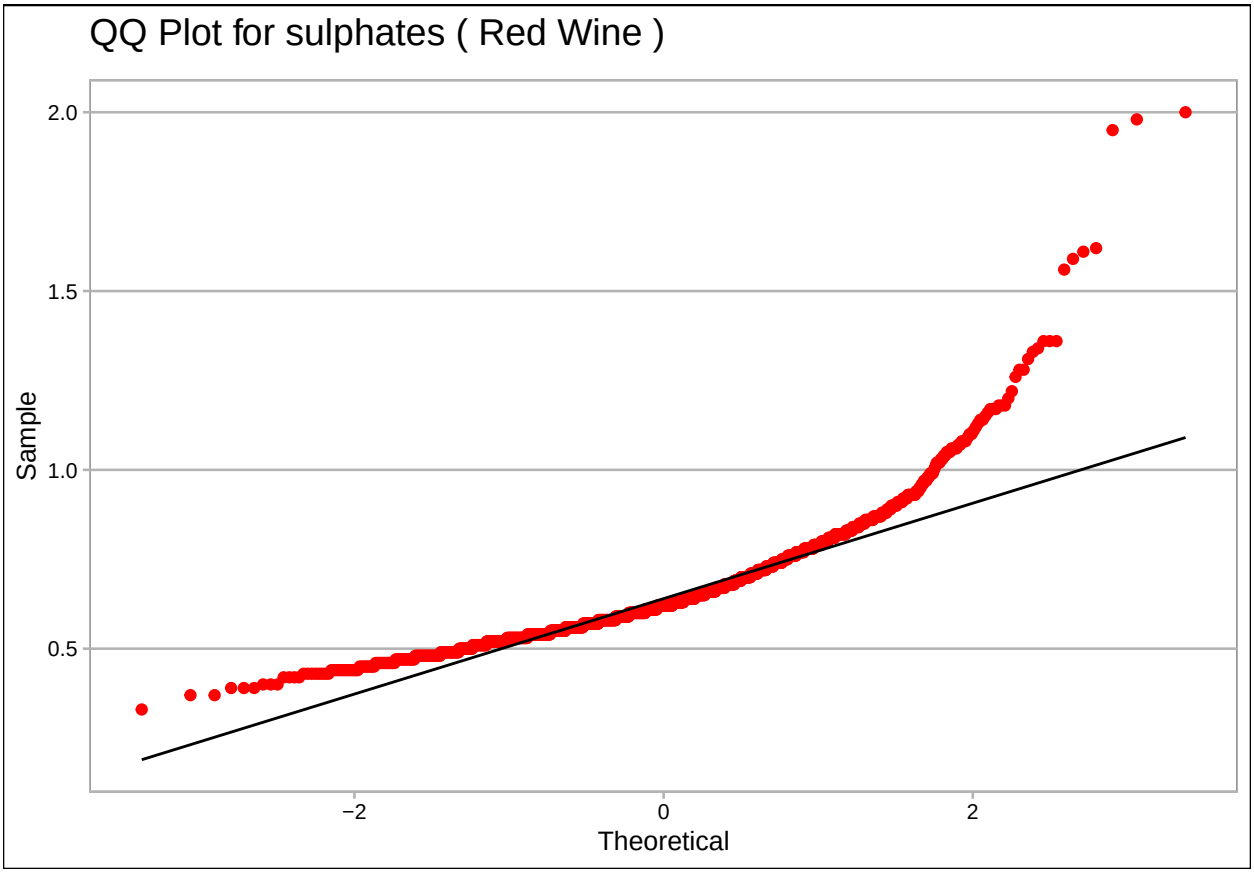

```
plot_boxplot(wine_data, 'sulphates')
```




```
plot_boxplot_quality(wine_data, 'sulphates')
```

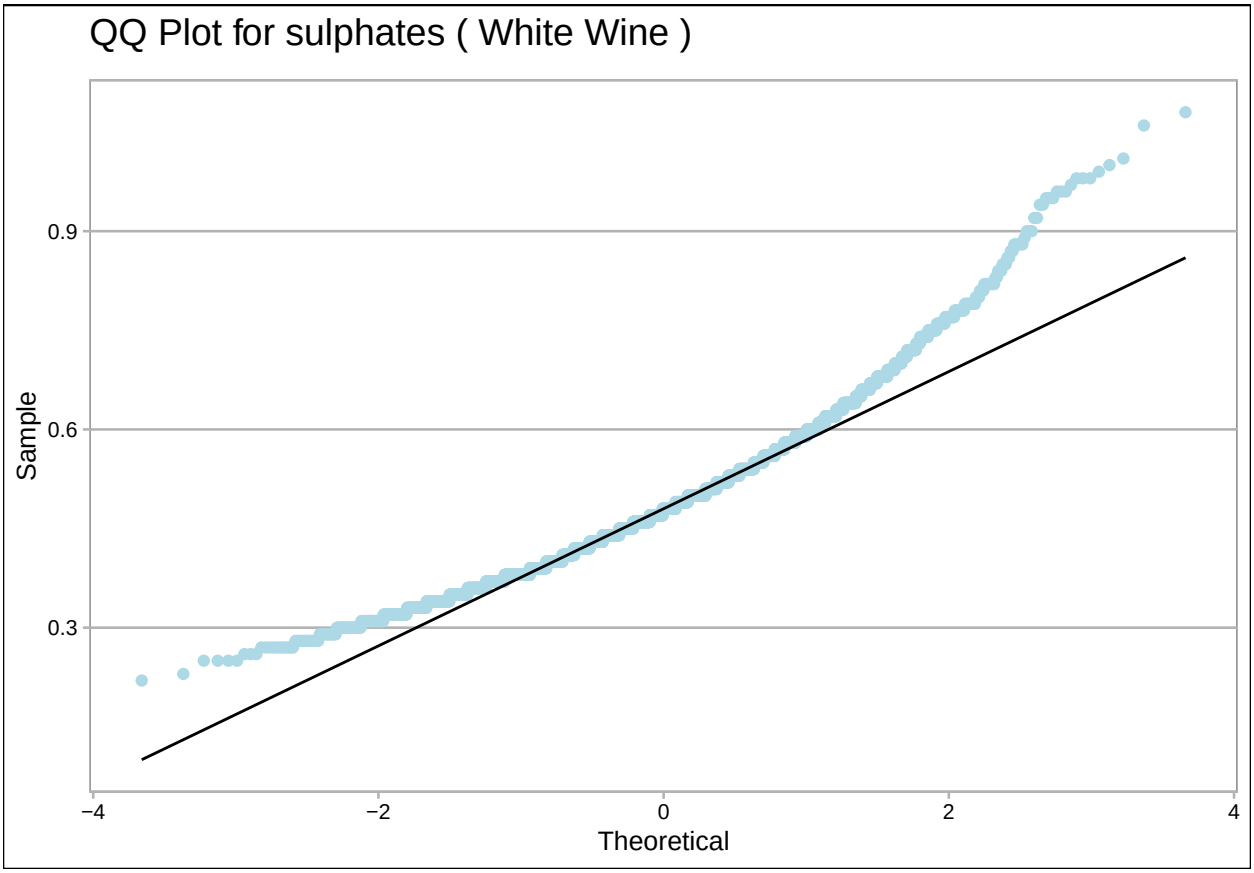



```
#performing normality check for red wine (sulphates)
normality_tester(red_wine, 'sulphates', 'red','Red Wine')
```



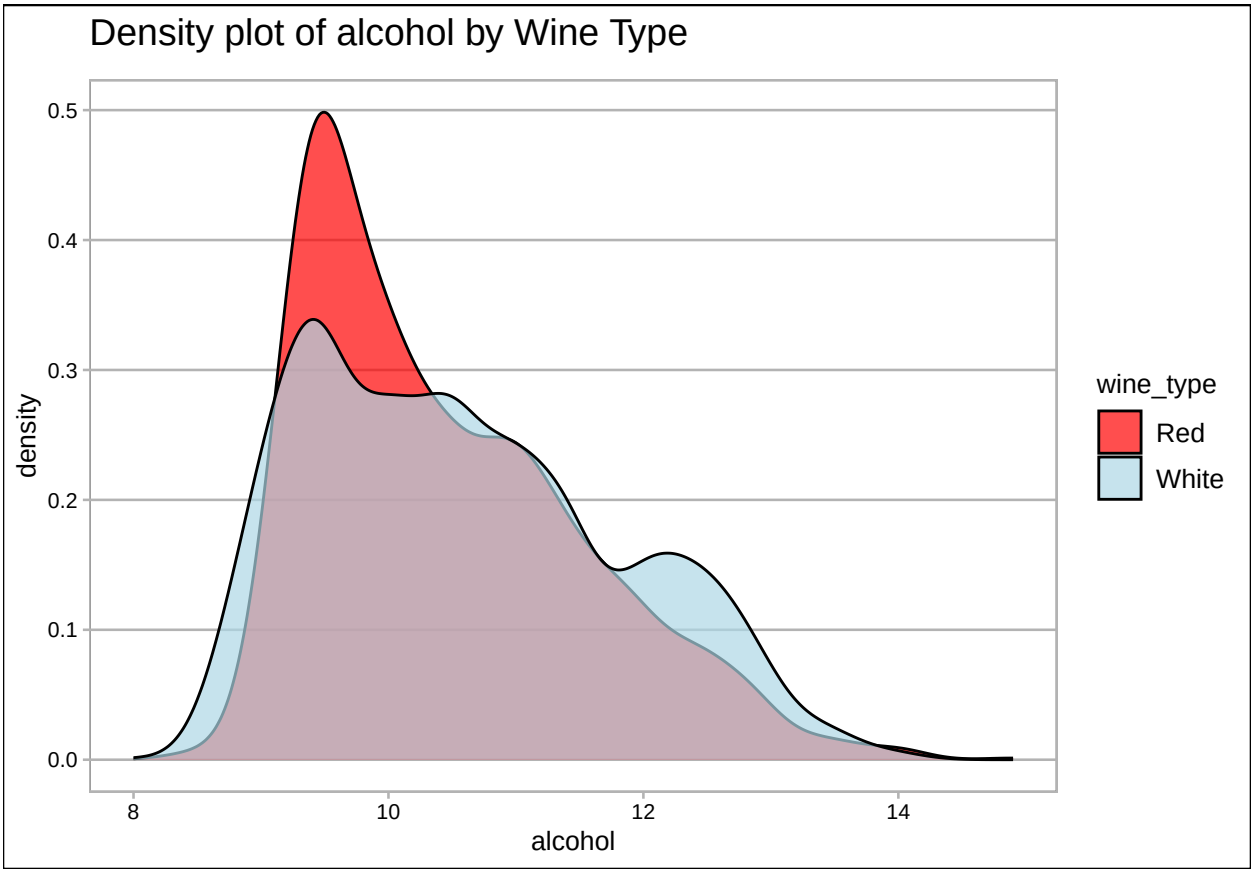

```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.83024, p-value < 2.2e-16
```

```
#performing normality check for white wine (sulphates)
normality_tester(white_wine, 'sulphates', 'lightblue', "White Wine")
```

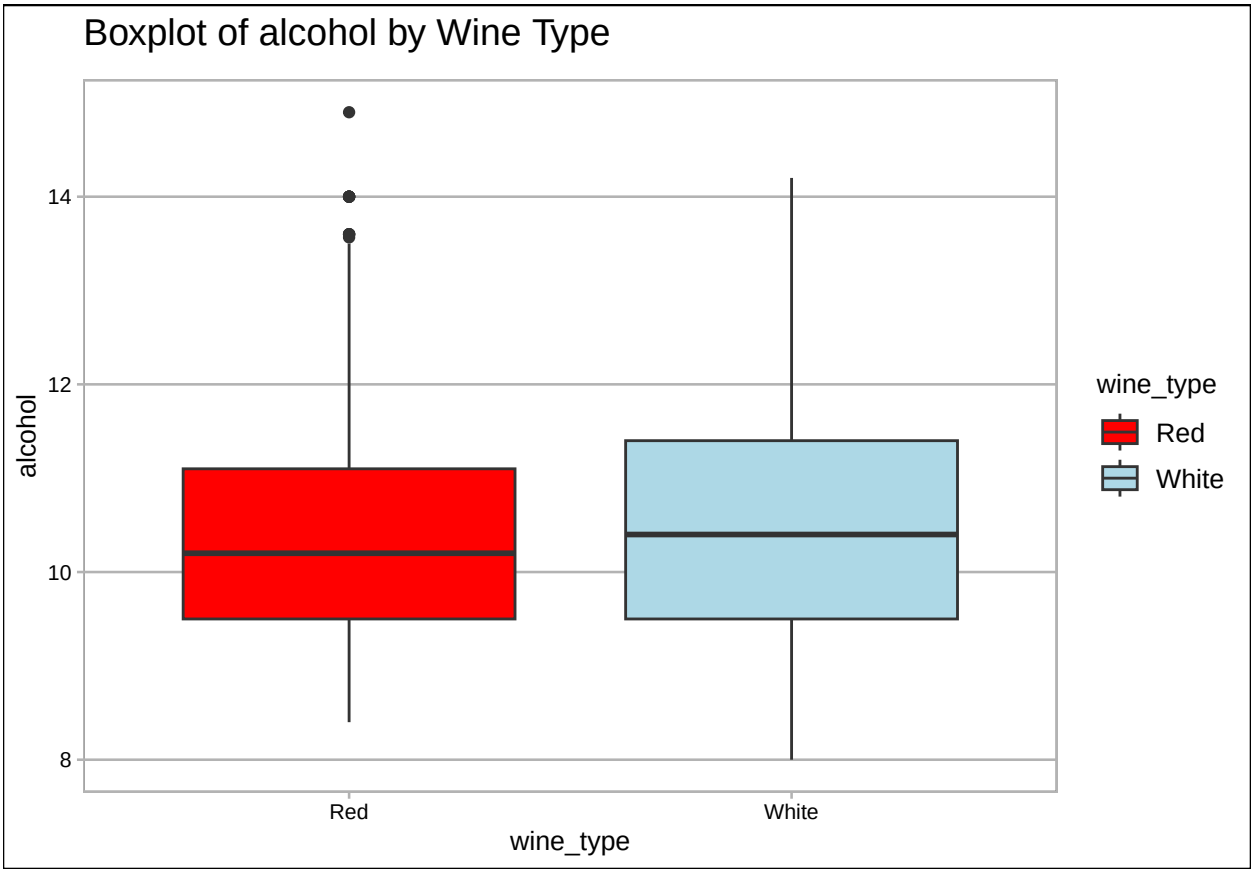



```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.95589, p-value < 2.2e-16
```

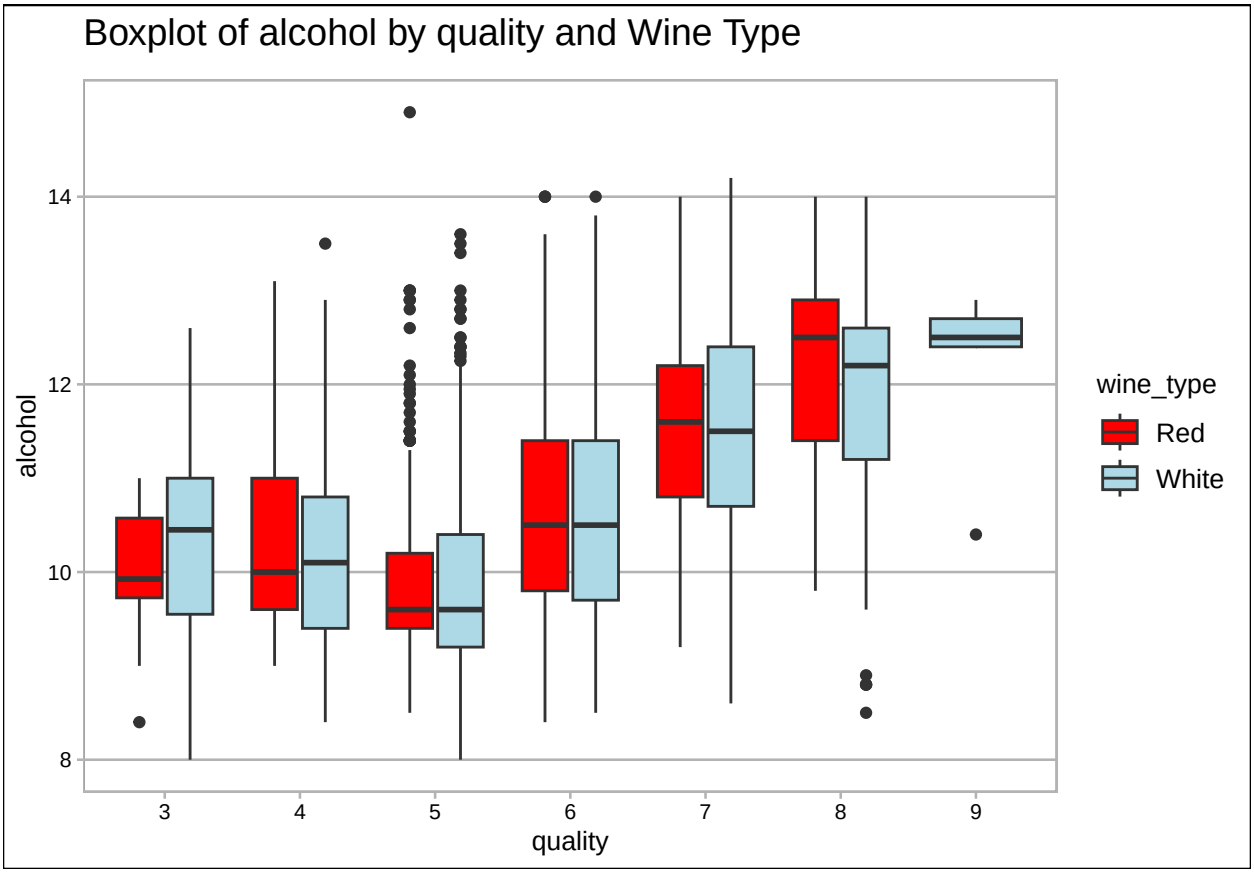
```
# alcohol
plot_density_plot(wine_data, 'alcohol')
```



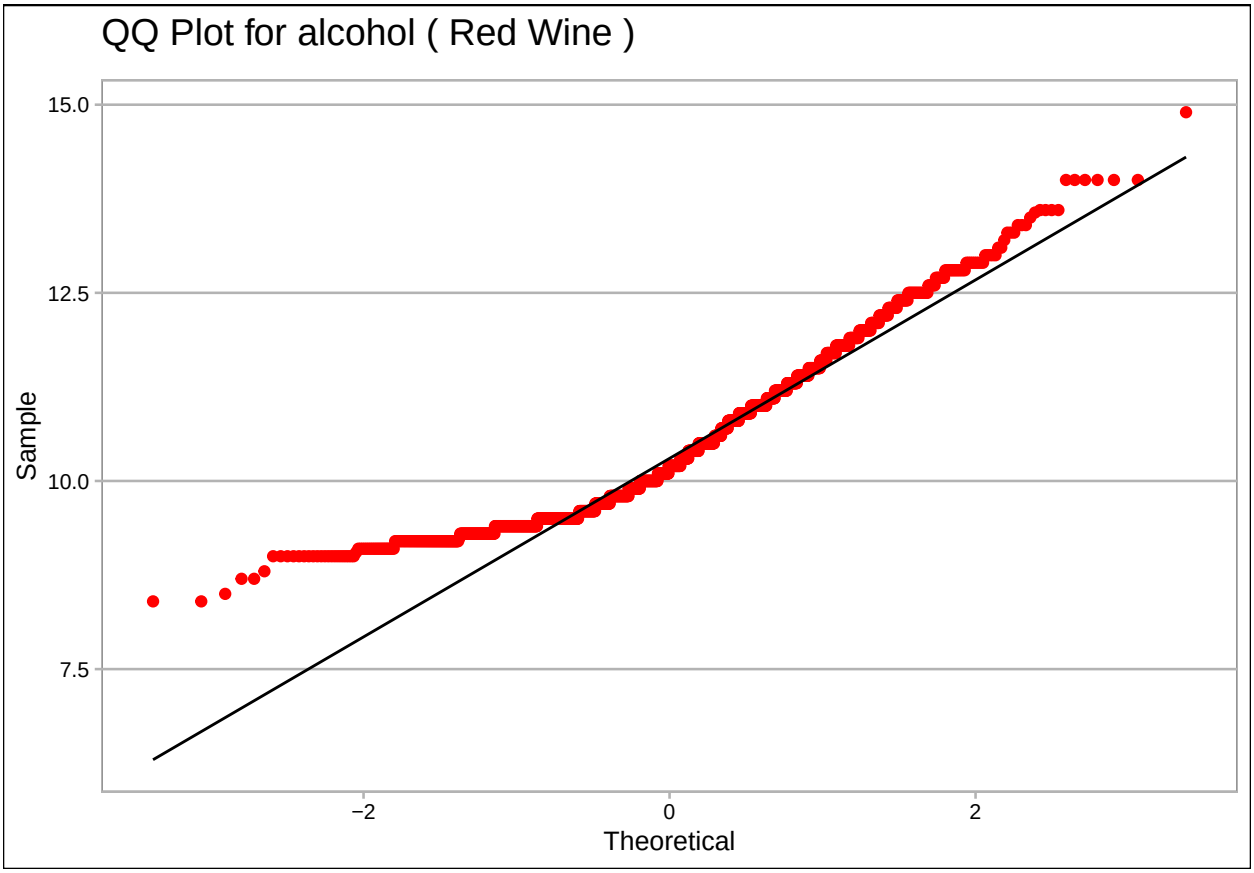

```
plot_boxplot(wine_data, 'alcohol')
```




```
plot_boxplot_quality(wine_data, 'alcohol')
```

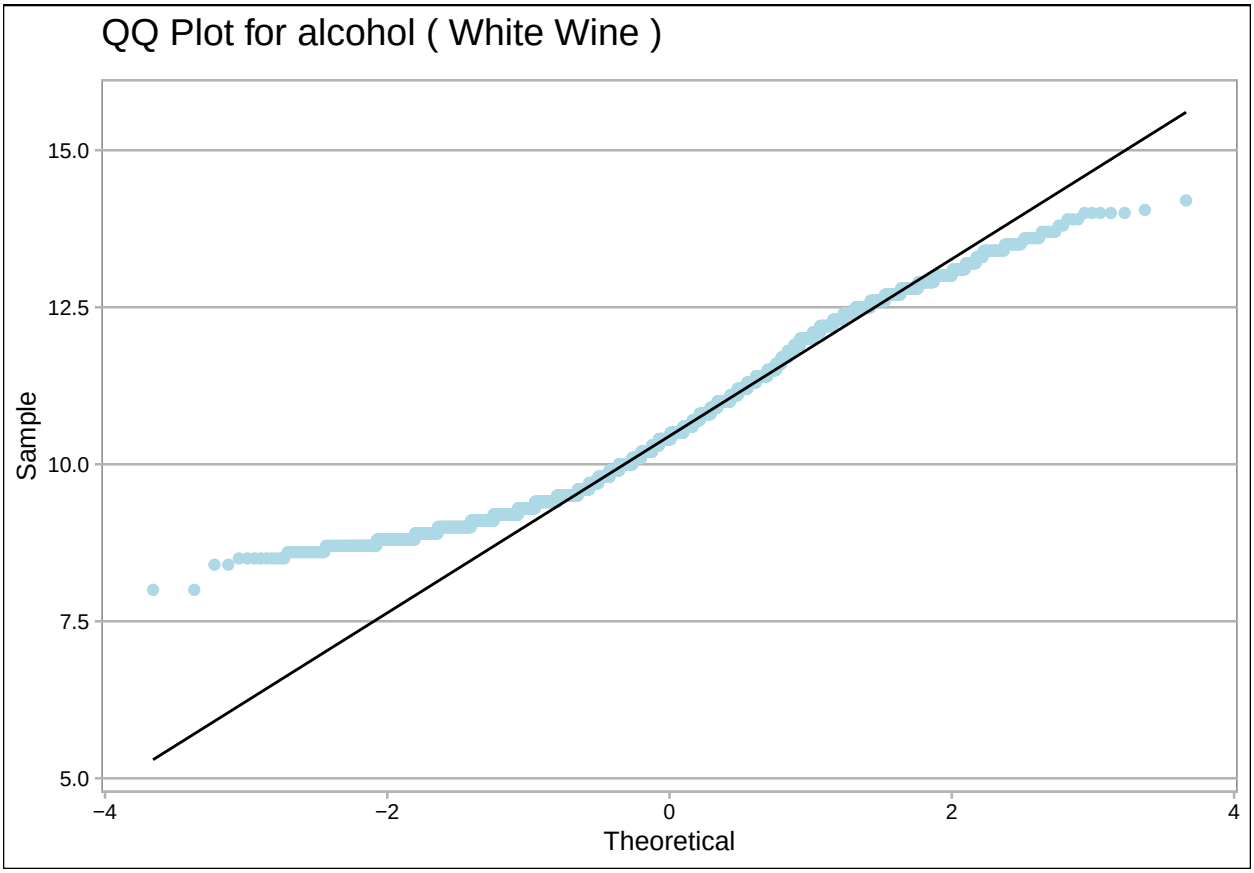



```
#performing normality check for red wine (alcohol)
normality_tester(red_wine, 'alcohol', 'red','Red Wine')
```




```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.9268, p-value < 2.2e-16
```

```
#performing normality check for white wine (alcohol)
normality_tester(white_wine, 'alcohol', 'lightblue', "White Wine")
```




```
##
##  Shapiro-Wilk normality test
##
## data:  column_data
## W = 0.96191, p-value < 2.2e-16
```

CORRELATION ANALYSIS

First we split the data into their various categories i.e continuous, ordinal and nominal categorical Quality is ordinal categorical. wine_type is nominal categorical. The rest are continuous.

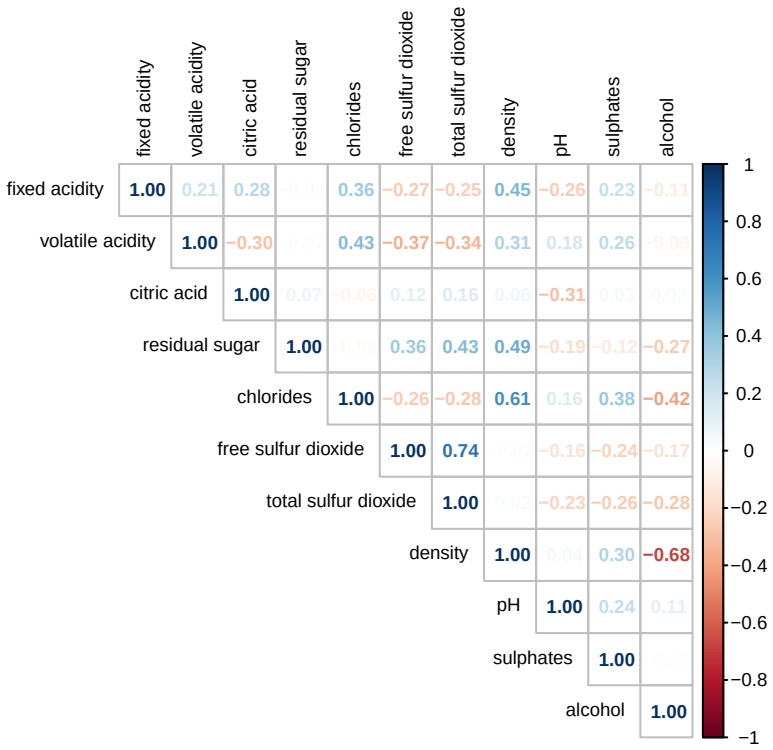
WINE DATA (RED AND WHITE WINES): CORRELATION MATRIX - MULTIPLE CONTINUOUS VARIABLES

```
#continuous columns in the wine_data data frame
wine_data_continuous <- wine_data %>%
  select(-quality,-wine_type)

# correlation matrix: Multiple continuous variables
# as seen in the normality tests, all the continuous variables are non-normal so we use Spearman's corr
wine_data_continuous_cor_matrix <- round(cor(wine_data_continuous, method = "spearman"), digit=2)
wine_data_continuous_cor_matrix
```

```
##                      fixed acidity volatile acidity citric acid residual sugar
## fixed acidity                 1.00             0.21        0.28          -0.03
## volatile acidity              0.21             1.00       -0.30          -0.02
```

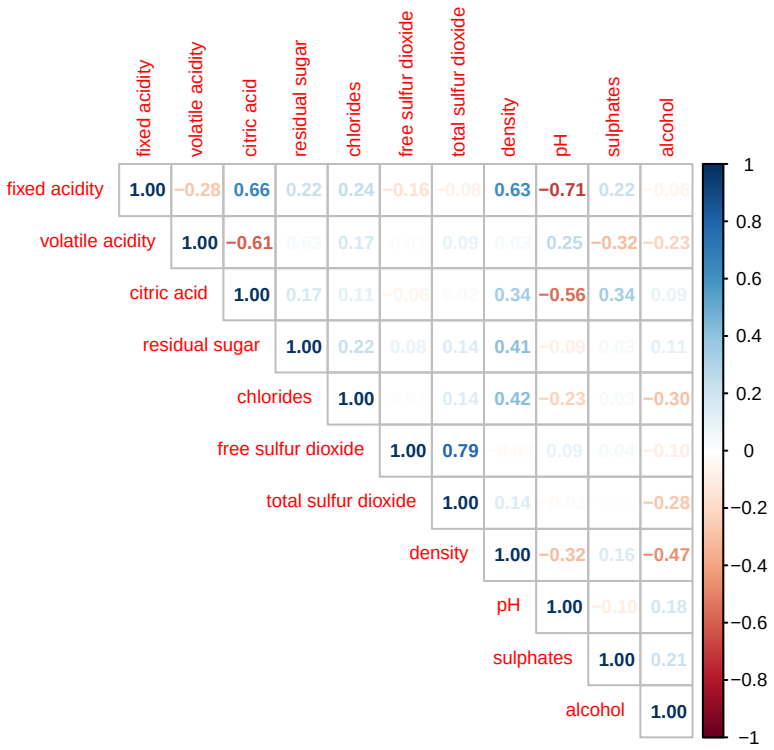
```
## citric acid                   0.28            -0.30        1.00           0.07
## residual sugar               -0.03            -0.02        0.07           1.00
## chlorides                     0.36             0.43       -0.06          -0.03
## free sulfur dioxide          -0.27            -0.37        0.12           0.36
## total sulfur dioxide         -0.25            -0.34        0.16           0.43
## density                       0.45             0.31        0.06           0.49
## pH                           -0.26             0.18       -0.31          -0.19
## sulphates                     0.23             0.26        0.03          -0.12
## alcohol                      -0.11            -0.06        0.02          -0.27
##                       chlorides free sulfur dioxide total sulfur dioxide density
## fixed acidity              0.36               -0.27                -0.25    0.45
## volatile acidity           0.43               -0.37                -0.34    0.31
## citric acid               -0.06                0.12                 0.16    0.06
## residual sugar            -0.03                0.36                 0.43    0.49
## chlorides                  1.00               -0.26                -0.28    0.61
## free sulfur dioxide       -0.26                1.00                 0.74   -0.02
## total sulfur dioxide      -0.28                0.74                 1.00    0.02
## density                    0.61               -0.02                 0.02    1.00
## pH                         0.16               -0.16                -0.23    0.04
## sulphates                  0.38               -0.24                -0.26    0.30
## alcohol                   -0.42               -0.17                -0.28   -0.68
##                           pH sulphates alcohol
## fixed acidity        -0.26      0.23   -0.11
## volatile acidity      0.18      0.26   -0.06
## citric acid          -0.31      0.03    0.02
## residual sugar       -0.19     -0.12   -0.27
## chlorides             0.16      0.38   -0.42
## free sulfur dioxide  -0.16     -0.24   -0.17
## total sulfur dioxide -0.23     -0.26   -0.28
## density               0.04      0.30   -0.68
## pH                    1.00      0.24    0.11
## sulphates             0.24      1.00   -0.02
## alcohol               0.11     -0.02    1.00
```

```r
# Visualizing the continuous correlation matrix
corrplot(wine_data_continuous_cor_matrix,
         method = "number",
         type = "upper",
         title = "Corrplot for Continous Variables (Wine data)",
         tl.col = "black",
         number.cex = 0.6,
         tl.cex = 0.6,
         cl.cex = 0.6,
         mar = c(1, 0, 2, 0))
```



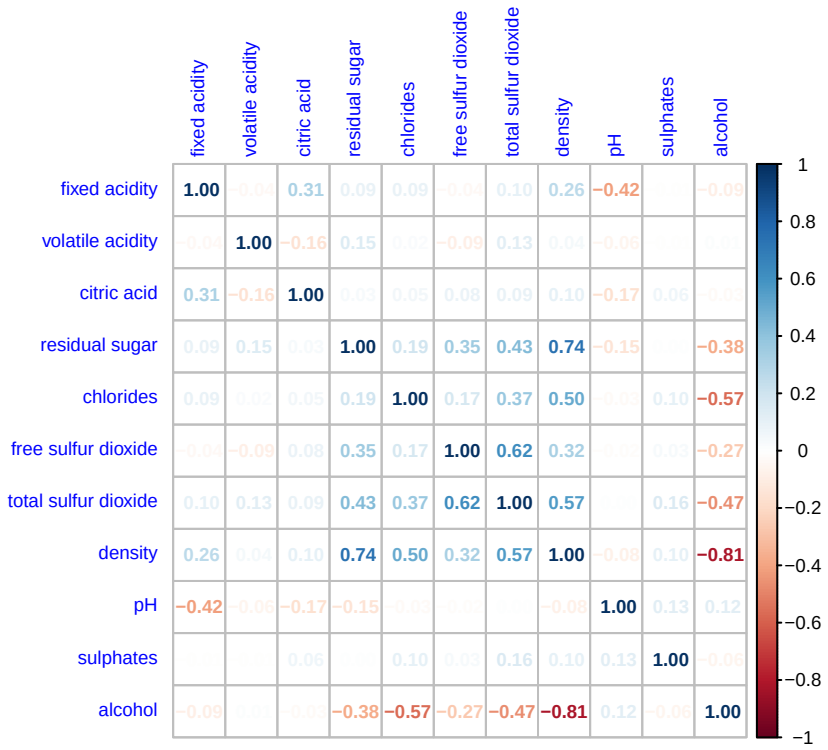
RED WINE: CORRELATION MATRIX - MULTIPLE CONTINUOUS VARIABLES

```
# correlation matrix: Multiple continuous variables
red_wine_continuous <- red_wine %>% select(-quality,-wine_type)
red_wine_continuous_cor_matrix <- round(cor(red_wine_continuous, method = "spearman"), digit=2)

corrplot(red_wine_continuous_cor_matrix,
         method = "number",
         type = "upper",
         title = "Corrplot for Continous Variables (Red Wine)",
         tl.col = "red",
         number.cex = 0.6,
         tl.cex = 0.6,
         cl.cex = 0.6,
         mar = c(1, 0, 2, 0))
```



WHITE WINE: CORRELATION MATRIX - MULTIPLE CONTINUOUS VARIABLES

```r
# correlation matrix: Multiple continuous variables
white_wine_continuous <- white_wine %>% select(-quality,-wine_type)
white_wine_continuous_cor_matrix <- round(cor(white_wine_continuous, method = "spearman"), digit=2)

corrplot(white_wine_continuous_cor_matrix,
         method = "number",
         type = "full",
         title = "Corrplot for Continous Variables (White Wine)",
         tl.col = "blue",
         number.cex = 0.6,
         tl.cex = 0.6,
         cl.cex = 0.6,
         mar = c(1, 0, 2, 0))
```




```r
# Creating a copy of the wine_data df to introduce wine_type as numeric.
wine_data_copy <- wine_data %>%
  mutate(wine_type_numeric = ifelse(wine_type == "Red", 0, 1))
```

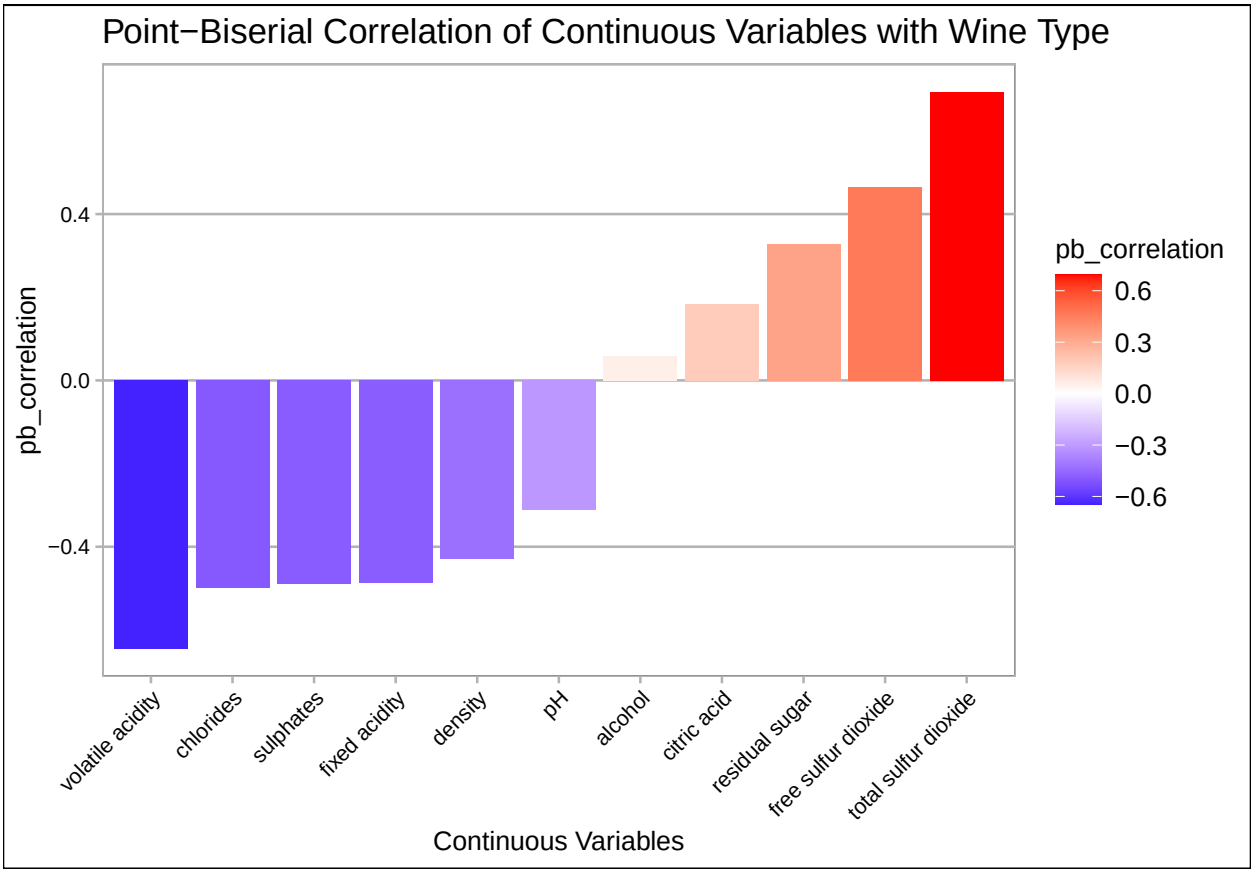
CORRELATION BETWEEN CONTINUOUS AND NOMINAL CATEGORICAL

```r
# Using Point-biserial correlation with wine type
point_biserial_cor <- sapply(names(wine_data_continuous), function(var) {
  cor.test(wine_data_copy[[var]], wine_data_copy$wine_type_numeric, method = "pearson")$estimate
})

# Converting to a data frame for better readability
pb_cor_df <- data.frame(
  Variable = names(wine_data_continuous),
  pb_correlation = point_biserial_cor
)
print(pb_cor_df)
```

```
##                                  Variable pb_correlation
## fixed acidity.cor             fixed acidity    -0.48625291
## volatile acidity.cor       volatile acidity    -0.64533485
## citric acid.cor                 citric acid     0.18375899
## residual sugar.cor           residual sugar     0.32869493
## chlorides.cor                     chlorides    -0.49951694
## free sulfur dioxide.cor  free sulfur dioxide     0.46532604
```

```
## total sulfur dioxide.cor total sulfur dioxide     0.69422873
## density.cor                           density    -0.42937707
## pH.cor                                     pH    -0.31091908
## sulphates.cor                       sulphates    -0.49036389
## alcohol.cor                           alcohol     0.05775617
```

```r
# Visualizing the point-biserial correlations between the continuous variables and wine type
ggplot(pb_cor_df, aes(x = reorder(Variable, pb_correlation),
                      y = pb_correlation,
                      fill = pb_correlation)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  labs(title = "Point-Biserial Correlation of Continuous Variables with Wine Type",
       x = "Continuous Variables") +
  theme_calc() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



CORREALTION BETWEEN CONTINUOUS AND ORDINAL CATEGORICAL VARAIBLES

```r
# Spearman correlation of continuous variables with quality
spearman_cor <- sapply(names(wine_data_continuous), function(var) {
  cor.test(wine_data_copy[[var]], as.numeric(wine_data_copy$quality), method = "spearman")$estimate
})
```

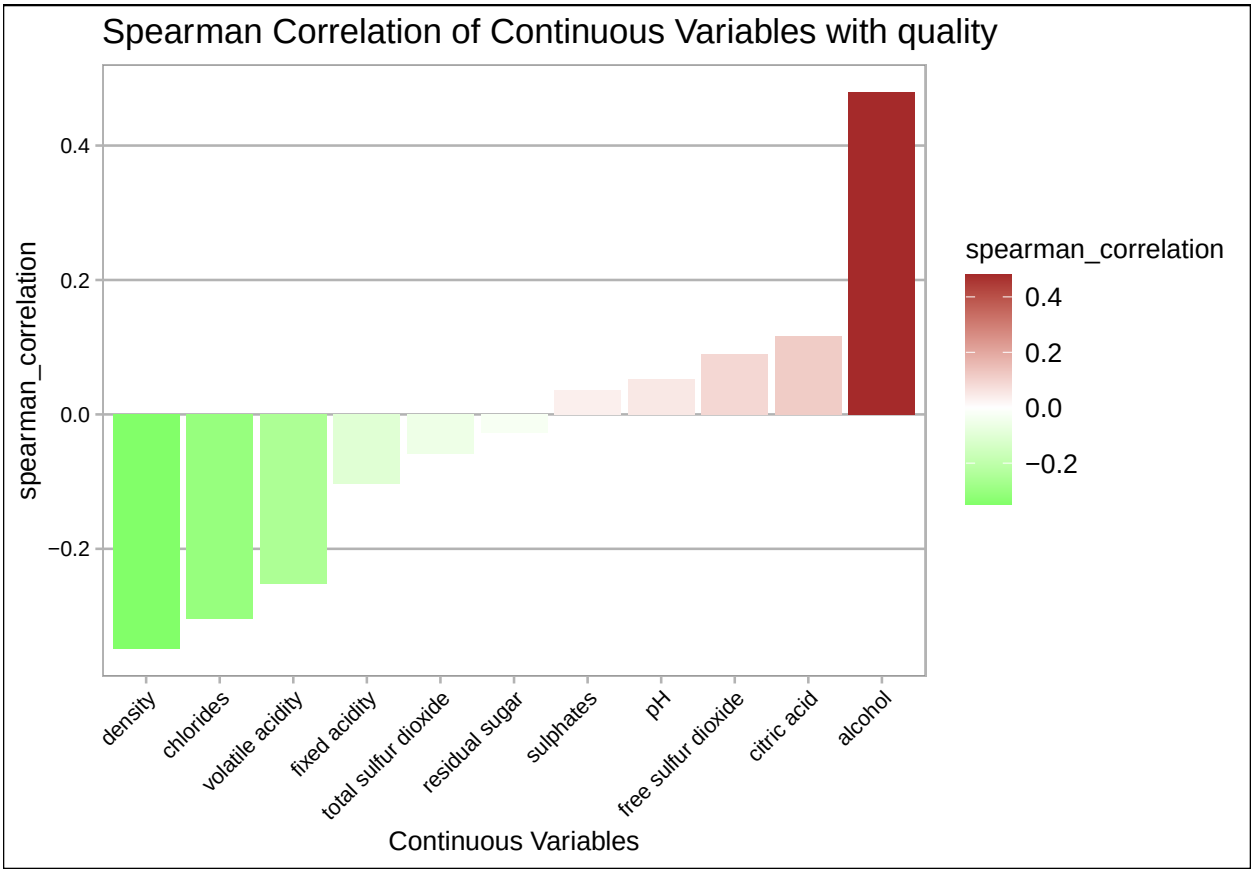
```
## Warning in cor.test.default(wine_data_copy[[var]],
```

```
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
## Warning in cor.test.default(wine_data_copy[[var]],
## as.numeric(wine_data_copy$quality), : Cannot compute exact p-value with ties
```

```
# Converting results to a data frame
spearman_cor_df <- data.frame(
  Variable = names(wine_data_continuous),
  spearman_correlation = spearman_cor
)
print(spearman_cor_df)
```

```
##                                   Variable spearman_correlation
## fixed acidity.rho            fixed acidity          -0.10402946
## volatile acidity.rho      volatile acidity          -0.25144983
## citric acid.rho                citric acid           0.11647748
## residual sugar.rho          residual sugar          -0.02816966
## chlorides.rho                    chlorides          -0.30387176
## free sulfur dioxide.rho   free sulfur dioxide        0.08997795
## total sulfur dioxide.rho total sulfur dioxide       -0.05822876
## density.rho                        density          -0.34901219
## pH.rho                                  pH           0.05279759
## sulphates.rho                    sulphates           0.03576869
## alcohol.rho                        alcohol           0.47978321
```

```
# Visualizing the spearman correlation between the continuous variables and quality
ggplot(spearman_cor_df, aes(x = reorder(Variable, spearman_correlation),
                              y = spearman_correlation,
                              fill = spearman_correlation)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient2(low = "green", mid = "white", high = "brown", midpoint = 0) +
  labs(title = "Spearman Correlation of Continuous Variables with quality",
       x = "Continuous Variables") +
  theme_calc() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



HYPOTHESIS TESTING. Since all the continuous variables have non-normal distributions, We will perform 3 transformations to try to make them approximately normal (log, sqrt, and cuberoot)

```
# continuous columns in red and white wine data frames
columns_red <- names(red_wine_continuous)
columns_white <- names(white_wine_continuous)
```

```
# function to log transform and perform Shapiro-Wilk test
log_shapiro <- function(data, columns) {
  p_values <- list()
  for (col in columns) {
      # log transforming
      log_transformed <- log(data[[col]])

      # Shapiro-Wilk test
      shapiro_test <- shapiro.test(log_transformed)
      p_values[[col]] <- shapiro_test$p.value
  }
  return(unlist(p_values))
}
```

```
# function to sqrt transform and perform Shapiro-Wilk test
sqrt_shapiro <- function(data, columns) {
  p_values <- list()
  for (col in columns) {
      # sqrt transforming
```

```
      sqrt_transformed <- sqrt(data[[col]])

      # Shapiro-Wilk test
      shapiro_test <- shapiro.test(sqrt_transformed)
      p_values[[col]] <- shapiro_test$p.value
  }
  return(unlist(p_values))
}
```

```
# function to cuberoot transform and perform Shapiro-Wilk test
cuberoot_shapiro <- function(data, columns) {
  p_values <- list()
  for (col in columns) {
      # cuberoot transforming
      sqrt_transformed <- (data[[col]])^(1/3)

      # Shapiro-Wilk test
      shapiro_test <- shapiro.test(sqrt_transformed)
      p_values[[col]] <- shapiro_test$p.value
  }
  return(unlist(p_values))
}
```

P-values for red wine after transforming the features

```
# Dataframe showing the P-Values of the transformed variables in the red wine
red_wine_shapiro_test <- data.frame(log= log_shapiro(red_wine,columns_red),
                                    sqrt= sqrt_shapiro(red_wine,columns_red),
                                    cuberoot = cuberoot_shapiro(red_wine,columns_red))
red_wine_shapiro_test
```

```
##                              log         sqrt     cuberoot
## fixed acidity        1.730292e-09 9.593182e-16 1.230796e-13
## volatile acidity     1.450131e-07 1.382114e-04 6.206505e-03
## citric acid                   NaN 3.469046e-22 5.893489e-31
## residual sugar       9.990759e-33 6.057998e-42 4.509124e-39
## chlorides            1.020107e-35 2.331442e-45 2.438503e-42
## free sulfur dioxide  1.418632e-11 1.461464e-15 1.359168e-11
## total sulfur dioxide 1.449619e-08 2.339152e-18 2.759950e-13
## density              1.892750e-06 1.850000e-06 1.864589e-06
## pH                   2.833297e-04 5.598931e-05 1.127282e-04
## sulphates            6.230880e-20 2.692217e-28 1.287389e-25
## alcohol              3.216595e-22 1.096492e-23 3.445611e-23
```

P-values for white wine after transforming the features

```
# Dataframe showing the P-Values of the transformed variables in the white wine
white_wine_shapiro_test <- data.frame(log= log_shapiro(white_wine,columns_white),
                                    sqrt= sqrt_shapiro(white_wine,columns_white),
                                    cuberoot = cuberoot_shapiro(white_wine,columns_white))
white_wine_shapiro_test
```

```
##                              log         sqrt     cuberoot
## fixed acidity        1.602488e-13 7.122192e-19 1.102804e-16
## volatile acidity     1.692680e-10 6.404266e-30 1.394124e-23
## citric acid                   NaN 4.216884e-40 2.732651e-51
## residual sugar       1.311379e-38 2.641676e-40 1.114089e-38
## chlorides            2.967257e-44 1.462811e-59 9.874785e-55
## free sulfur dioxide  2.679285e-33 5.108499e-14 7.290604e-16
## total sulfur dioxide 1.429462e-31 1.081904e-09 4.777707e-16
## density              6.805314e-35 3.492282e-35 4.364422e-35
## pH                   2.206415e-11 1.366406e-14 1.577266e-13
## sulphates            1.188562e-07 1.733875e-21 5.017367e-17
## alcohol              2.781279e-27 4.875834e-29 2.047624e-28
```

Since all the transformed variables failed to be normal based on the Shapiro_Wilk test. QQ plots might give us approximate normality.

```
# function to plot the log transformed QQ plot
log_qqplot <- function(data, column_name, color, wine_type) {
  qqplot <- ggplot(data, aes(sample = log(.data[[column_name]]))) +
    stat_qq(color = color) +
    stat_qq_line(color = "black") +
    labs(
      title = paste("QQ Plot for log", column_name, "(", wine_type, ")"),
      x = "Theoretical",
      y = "Sample"
    ) +
    theme_calc()+
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
  print(qqplot)
}
```

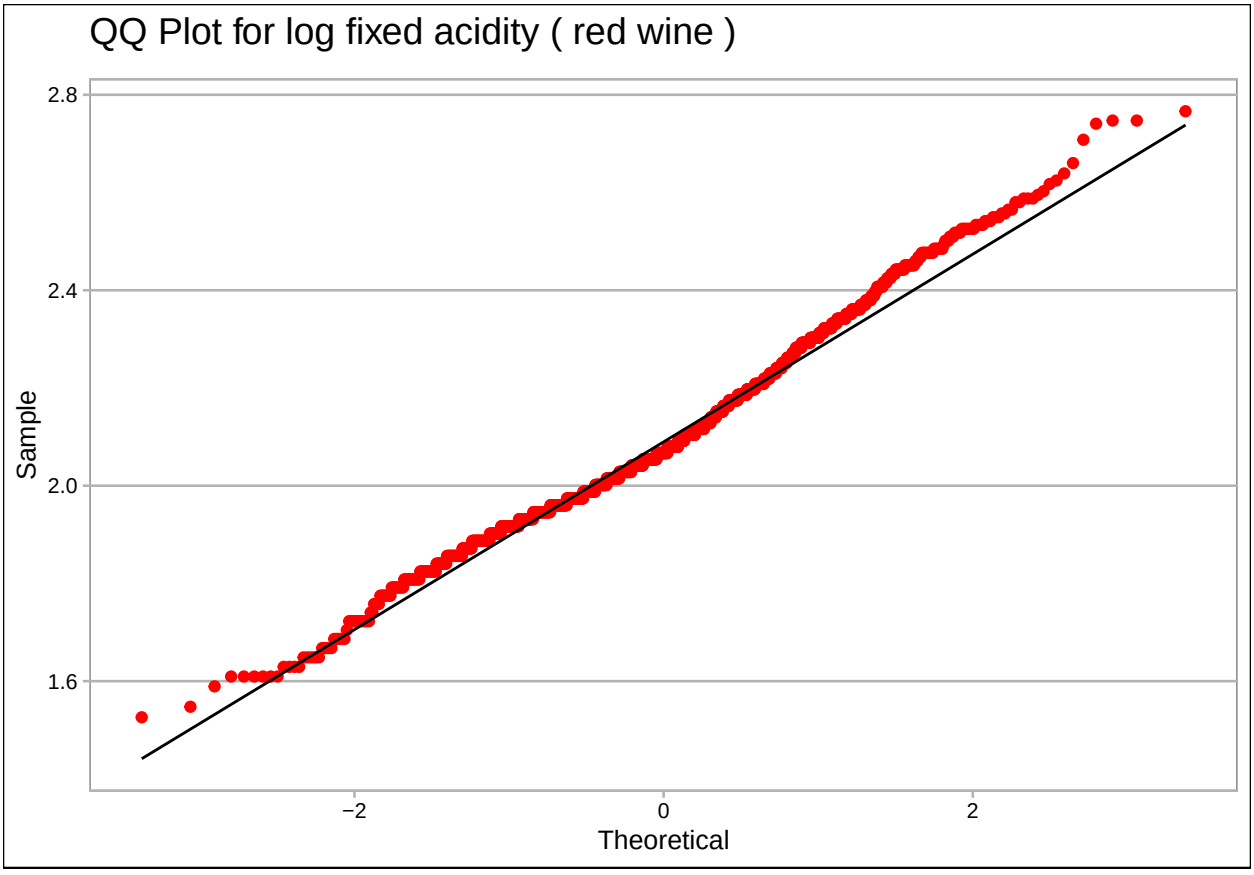
```
# function to plot the sqrt transformed QQ plot
sqrt_qqplot <- function(data, column_name, color, wine_type) {
  qqplot <- ggplot(data, aes(sample = sqrt(.data[[column_name]]))) +
    stat_qq(color = color) +
    stat_qq_line(color = "black") +
    labs(
      title = paste("QQ Plot for sqrt", column_name, "(", wine_type, ")"),
      x = "Theoretical",
      y = "Sample"
    ) +
    theme_calc()+
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
  print(qqplot)
}
```

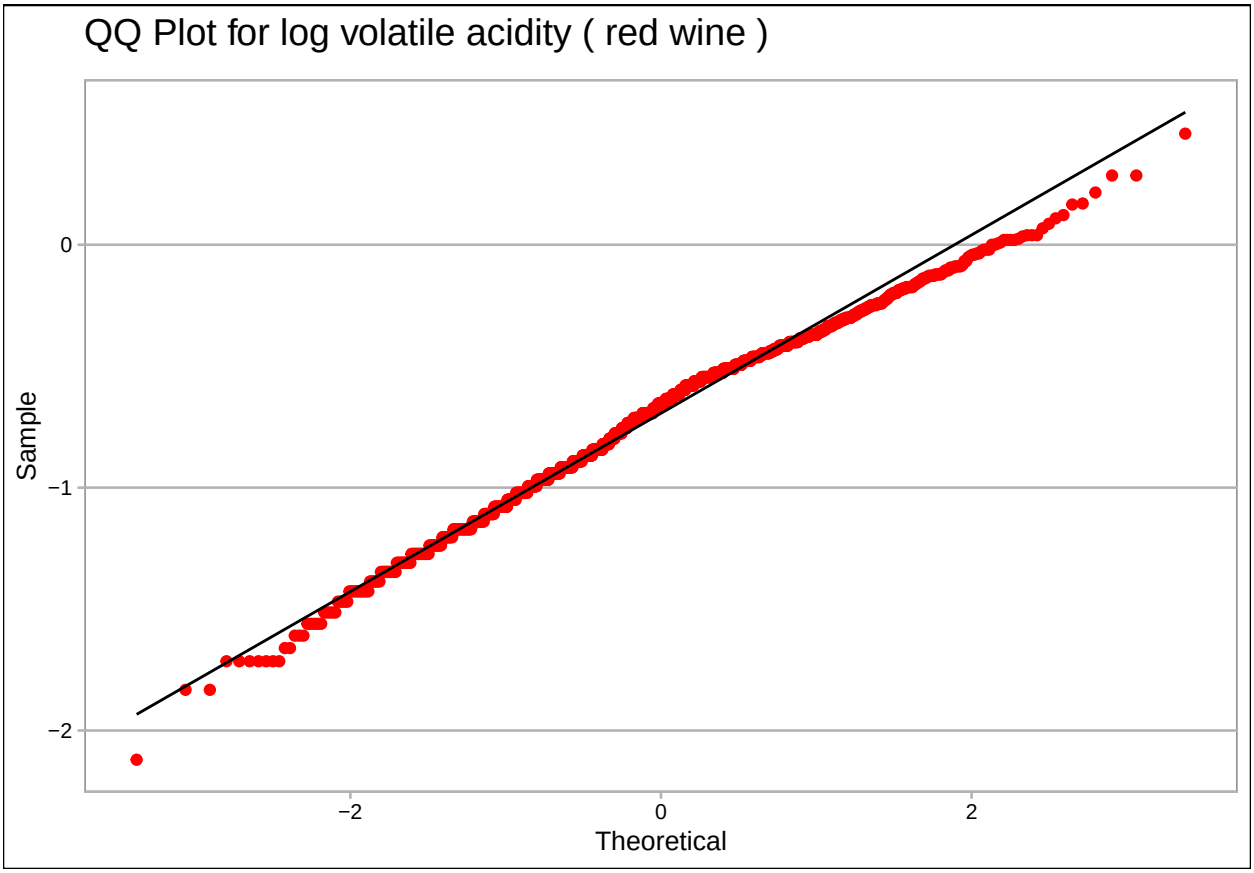
```
# function to plot the cuberoot transformed QQ plot
cuberoot_qqplot <- function(data, column_name, color, wine_type) {
  qqplot <- ggplot(data, aes(sample = (.data[[column_name]])^(1/3))) +
    stat_qq(color = color) +
    stat_qq_line(color = "black") +
    labs(
      title = paste("QQ Plot for cuberoot", column_name, "(", wine_type, ")"),
```

```
      x = "Theoretical",
      y = "Sample"
    ) +
    theme_calc()+
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
  print(qqplot)
}
```

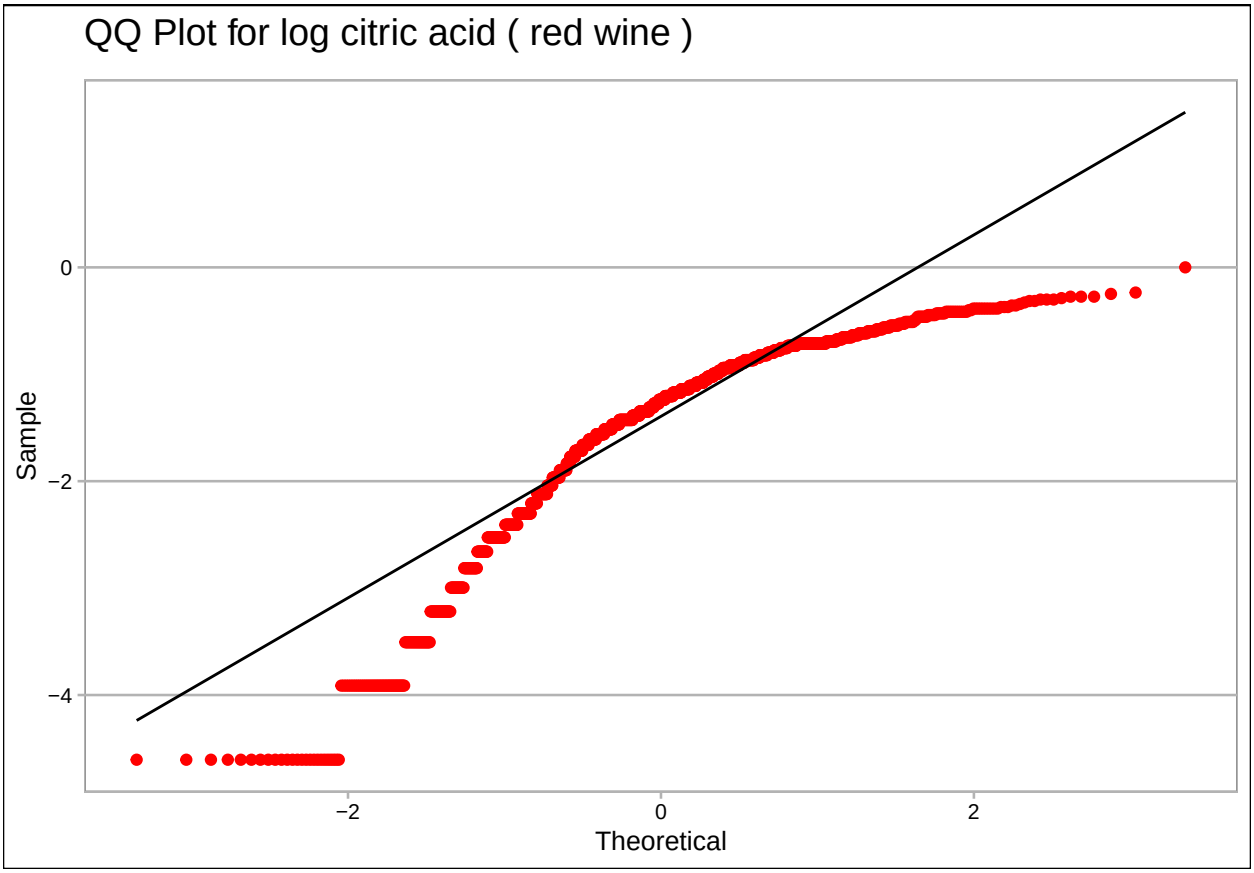
LOG TRANSFORMED QQPLOT FOR RED WINE

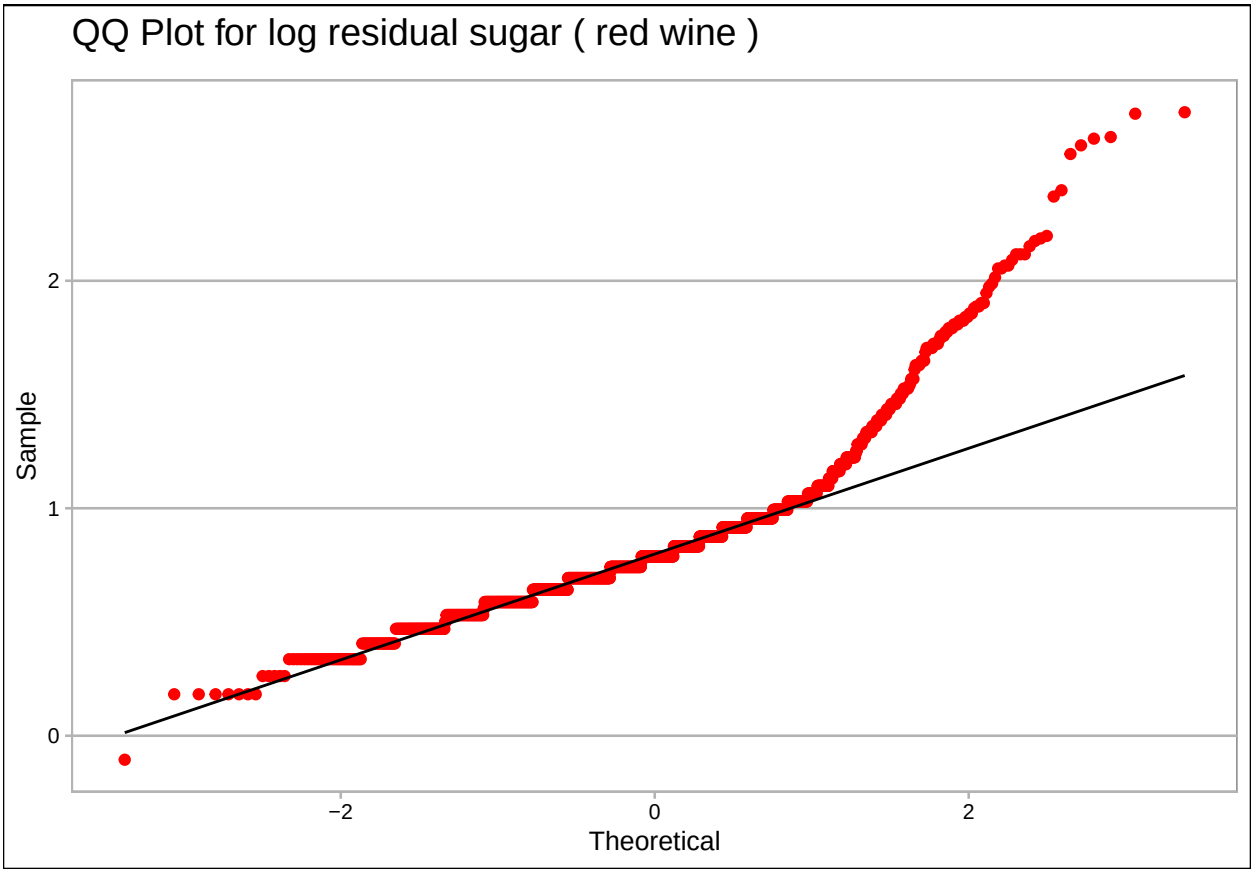
```
# Plotting the log transformed QQ plots for all the variables in Red wine
for (i in names(red_wine_continuous)) {
  log_qqplot(red_wine, i, 'red', 'red wine')
}
```



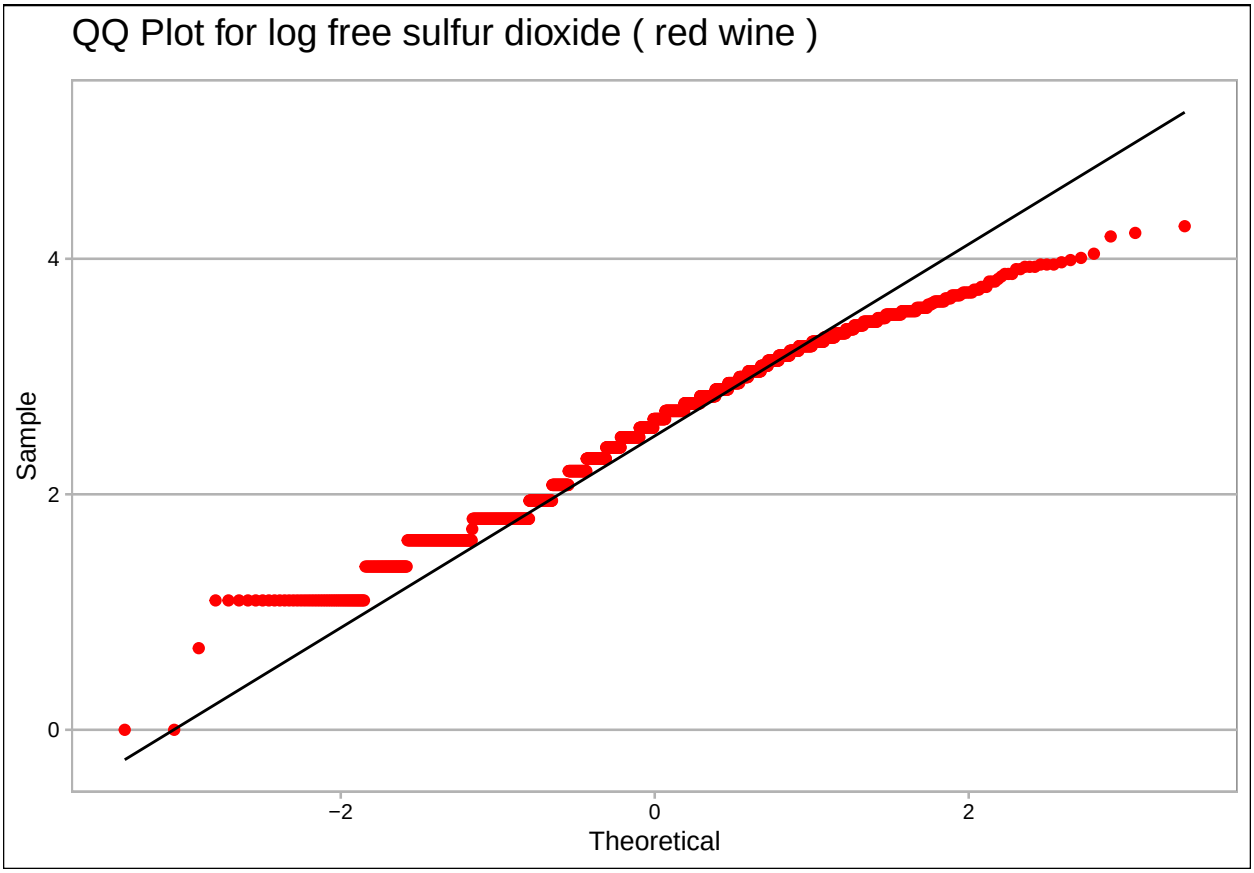


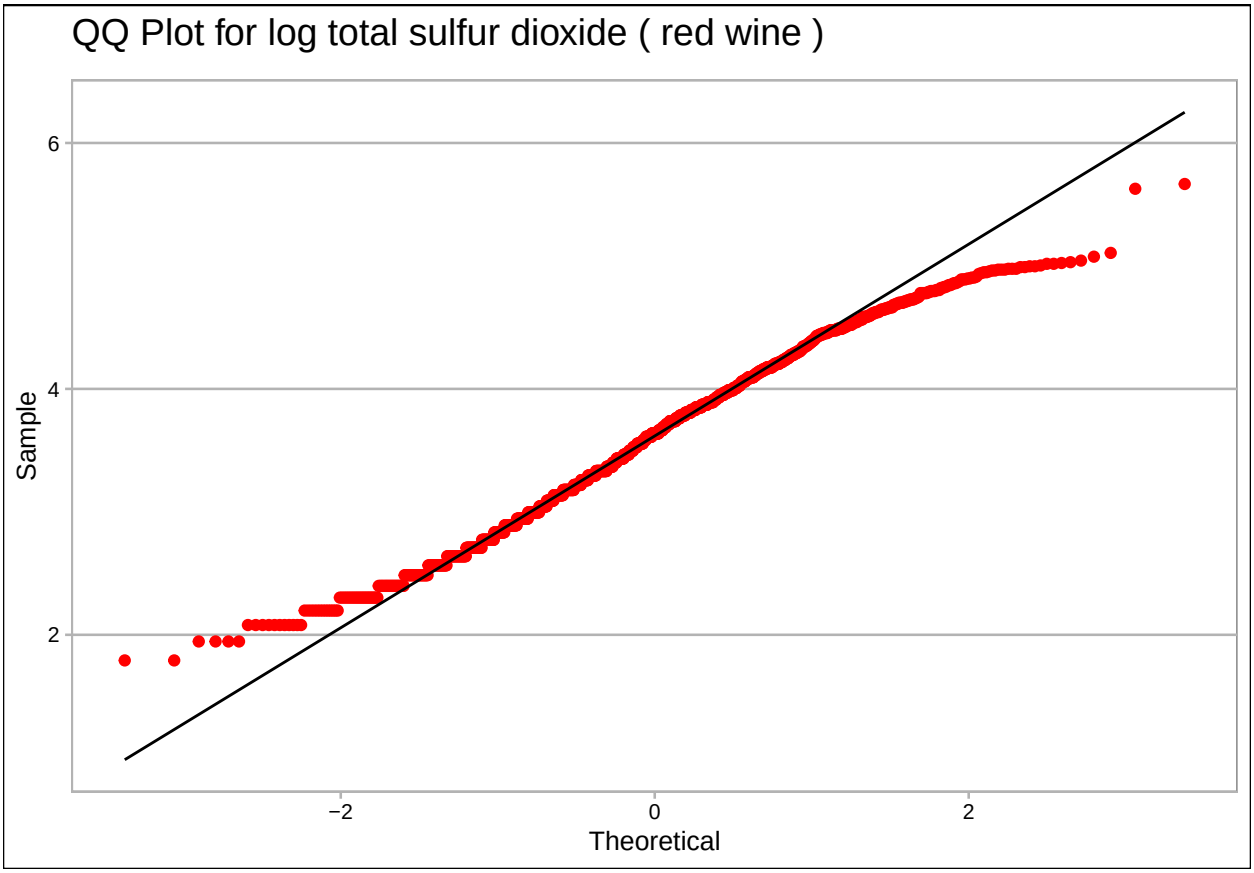

```
## Warning: Removed 118 rows containing non-finite outside the scale range
## (`stat_qq()`).

## Warning: Removed 118 rows containing non-finite outside the scale range
## (`stat_qq_line()`).
```

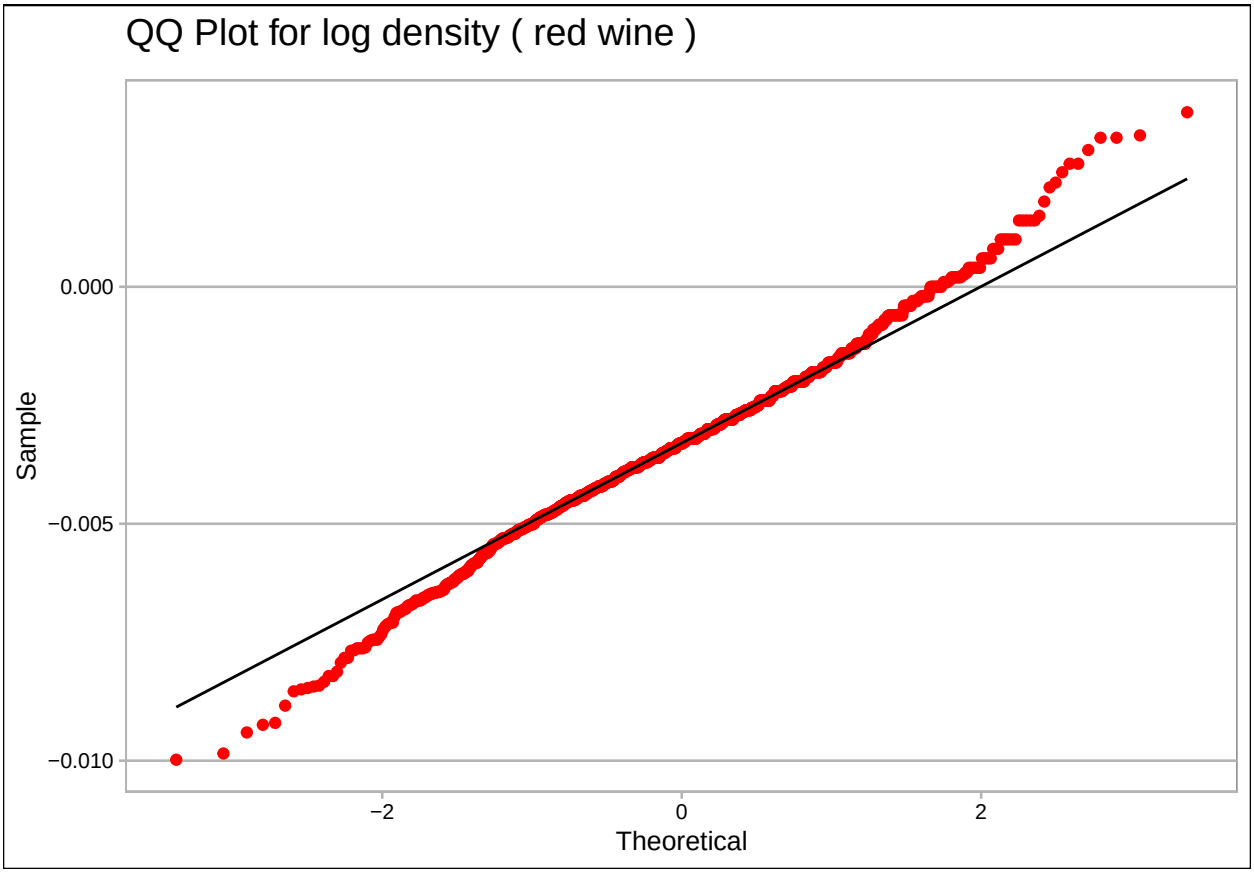

QQ Plot for log citric acid (red wine)
Sample
0
−2
−4
−2
0
2
Theoretical

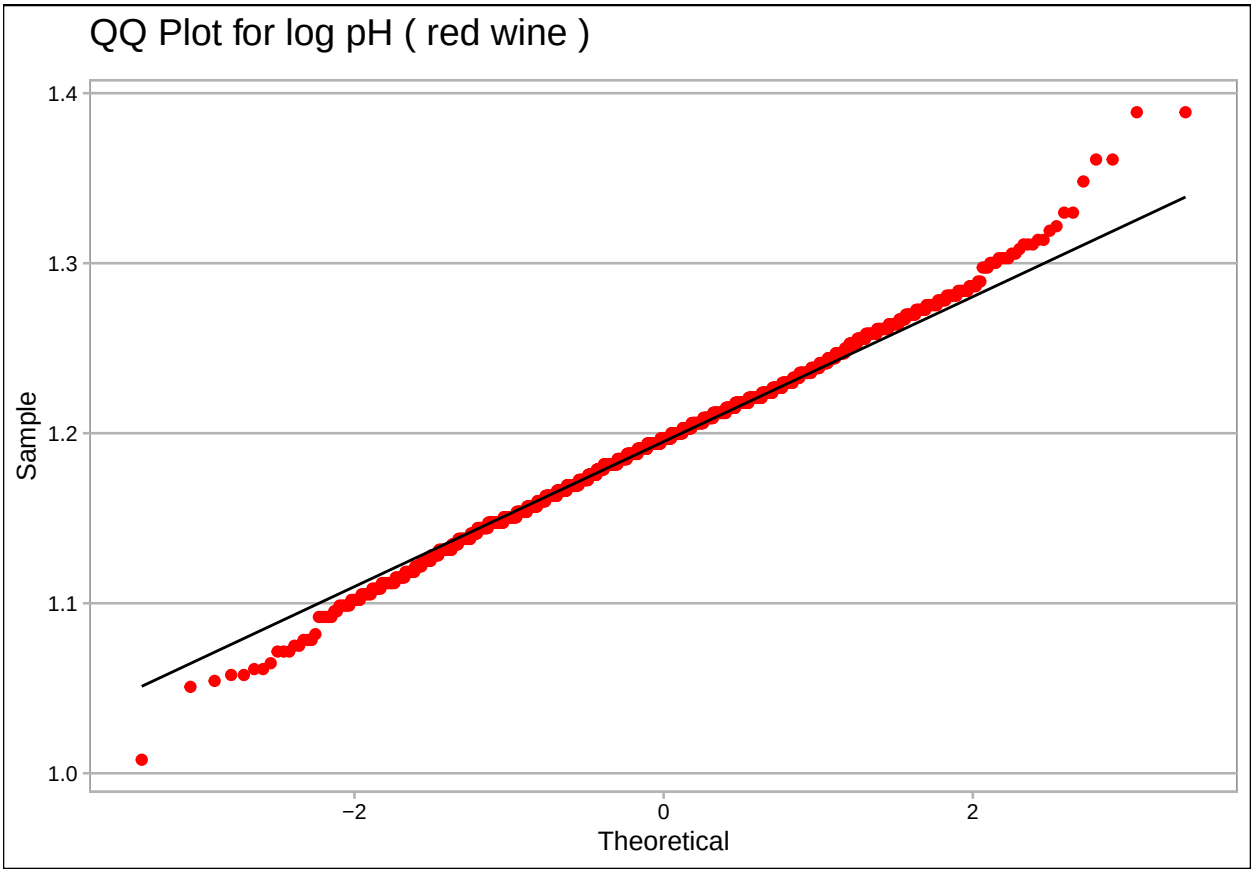

QQ Plot for log residual sugar (red wine)
Sample
2
1
0
−2
0
2
Theoretical

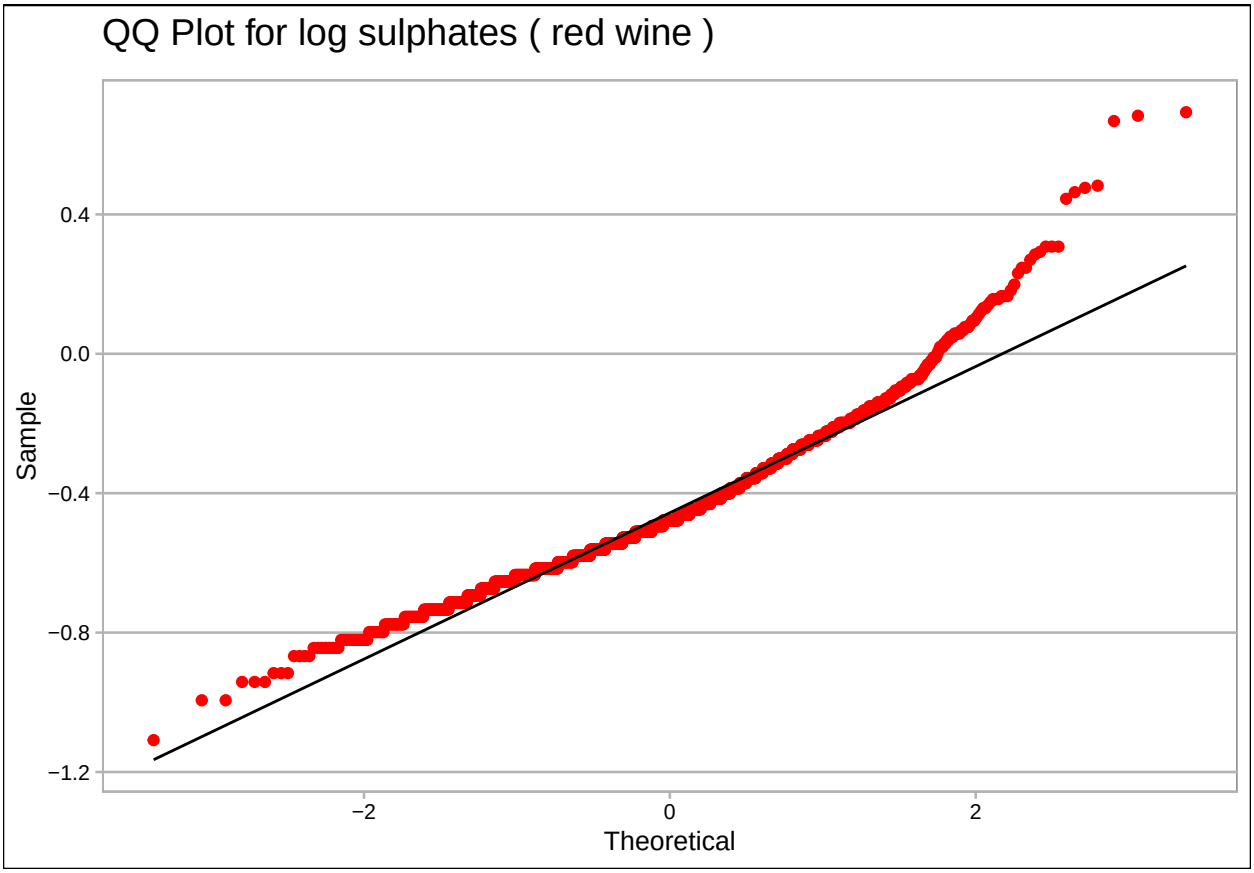

QQ Plot for log chlorides (red wine)
Sample
−1
−2
−3
−4
−2
0
2
Theoretical

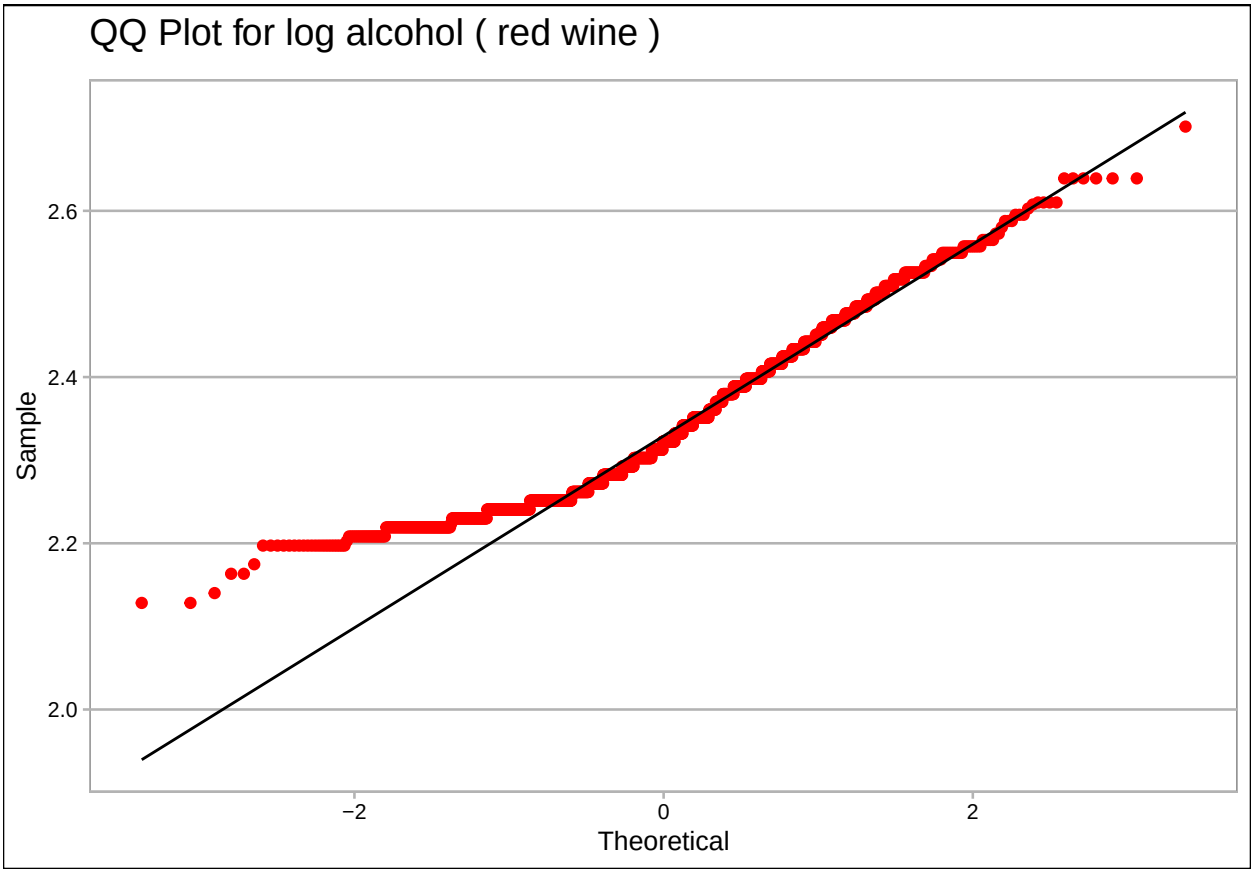

QQ Plot for log free sulfur dioxide (red wine)
Sample
4
2
0
−2
0
2
Theoretical


QQ Plot for log total sulfur dioxide (red wine)
Sample
6
4
2
−2
0
2
Theoretical


QQ Plot for log density (red wine)
0.000
−0.005
−0.010
Sample
−2
0
2
Theoretical


QQ Plot for log pH (red wine)
Sample
1.4
1.3
1.2
1.1
1.0
−2
0
2
Theoretical

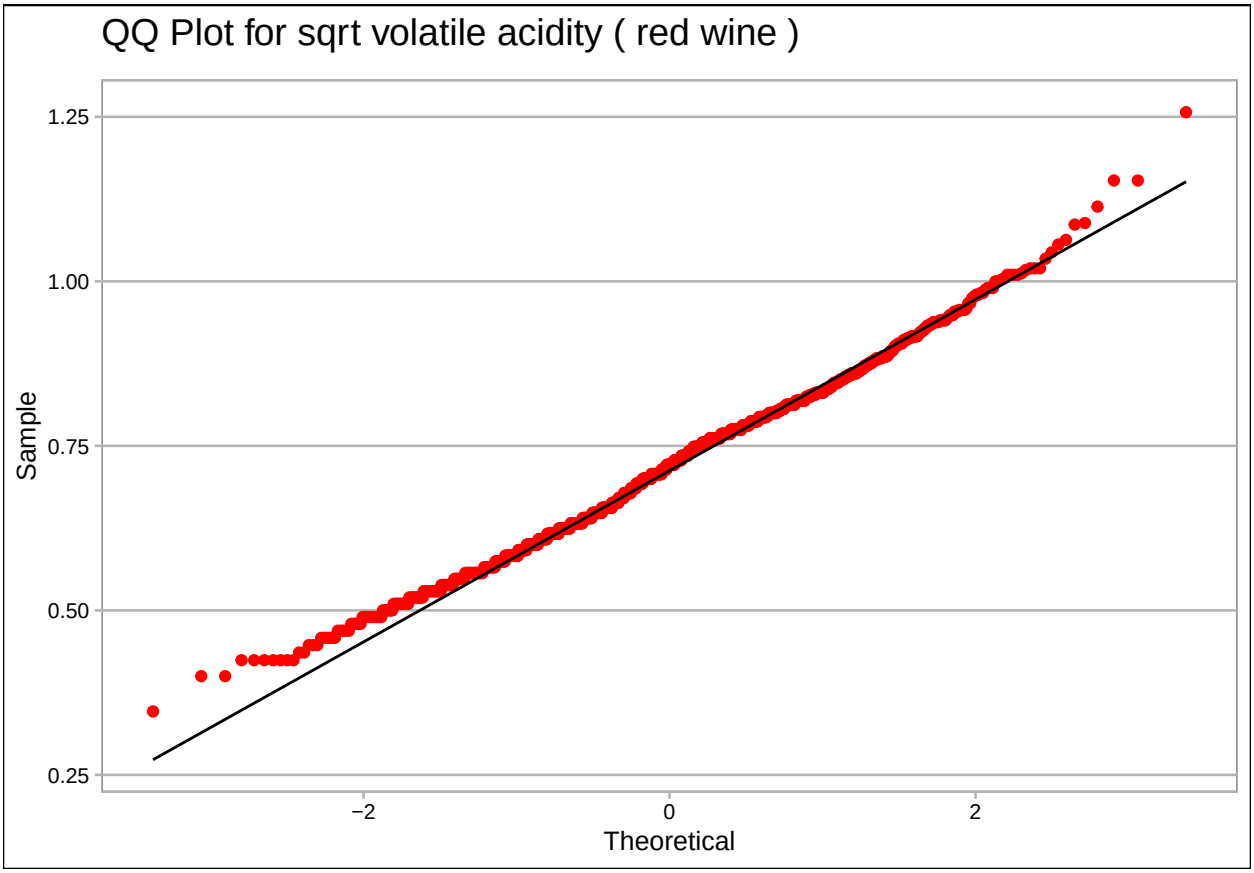

QQ Plot for log sulphates (red wine)
Sample
0.4
0.0
−0.4
−0.8
−1.2
−2
0
2
Theoretical

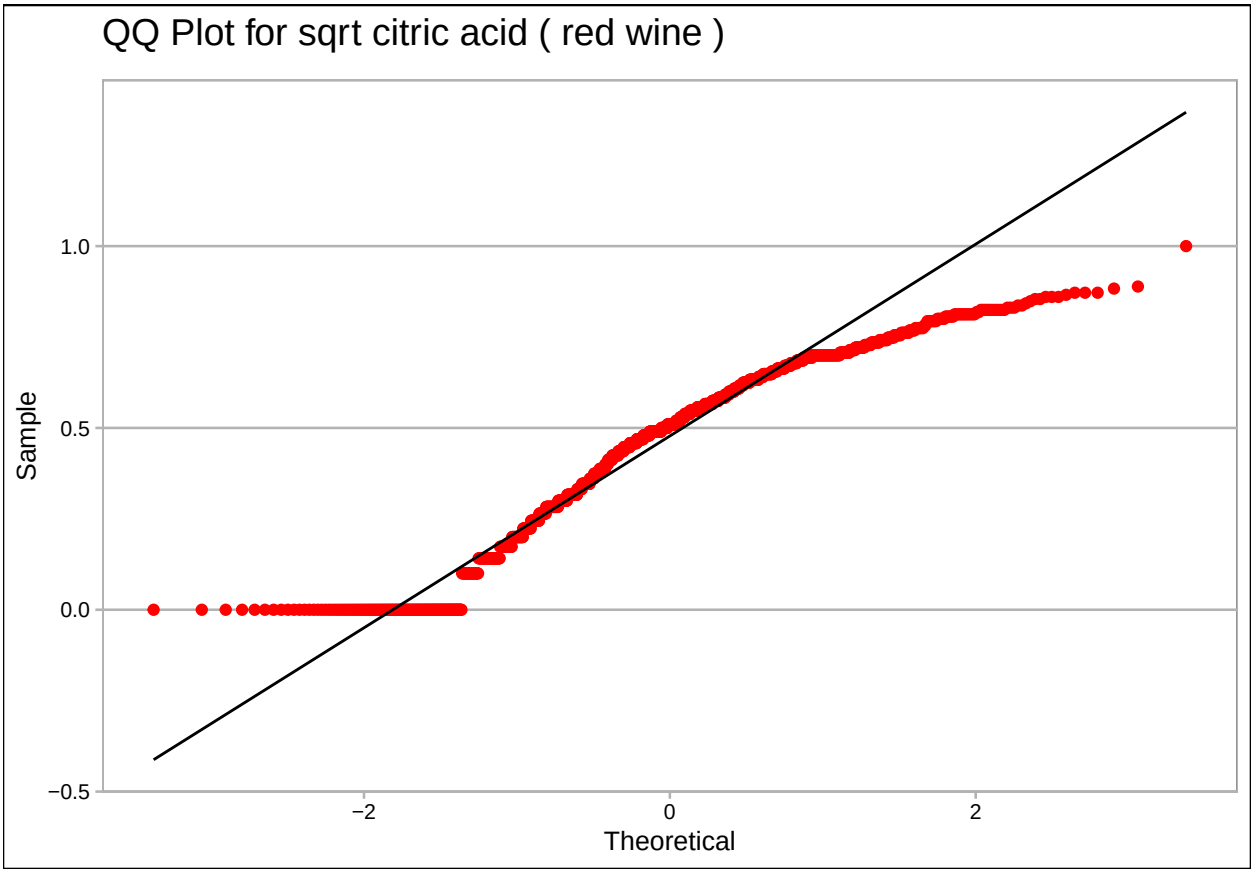


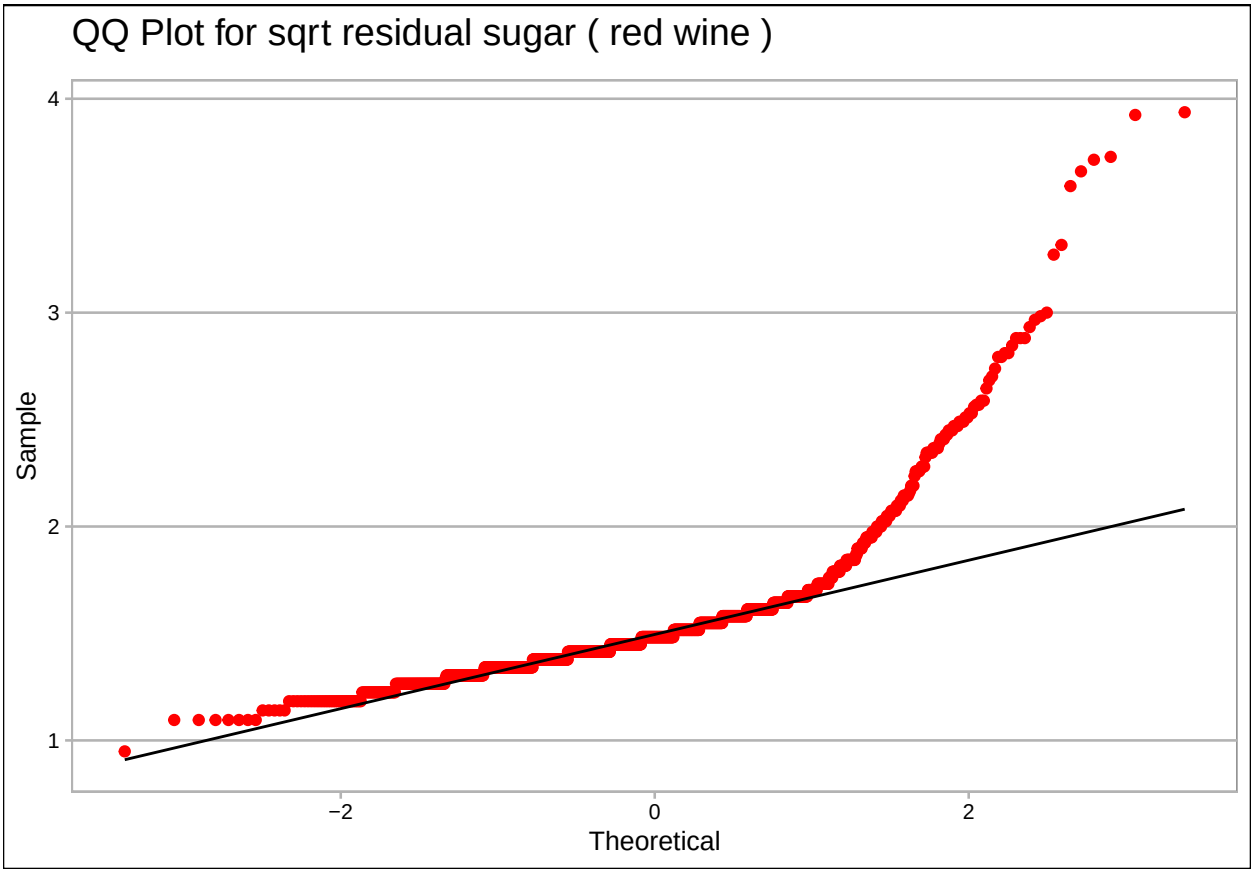
SQRT TRANSFORMED QQPLOT FOR RED WINE

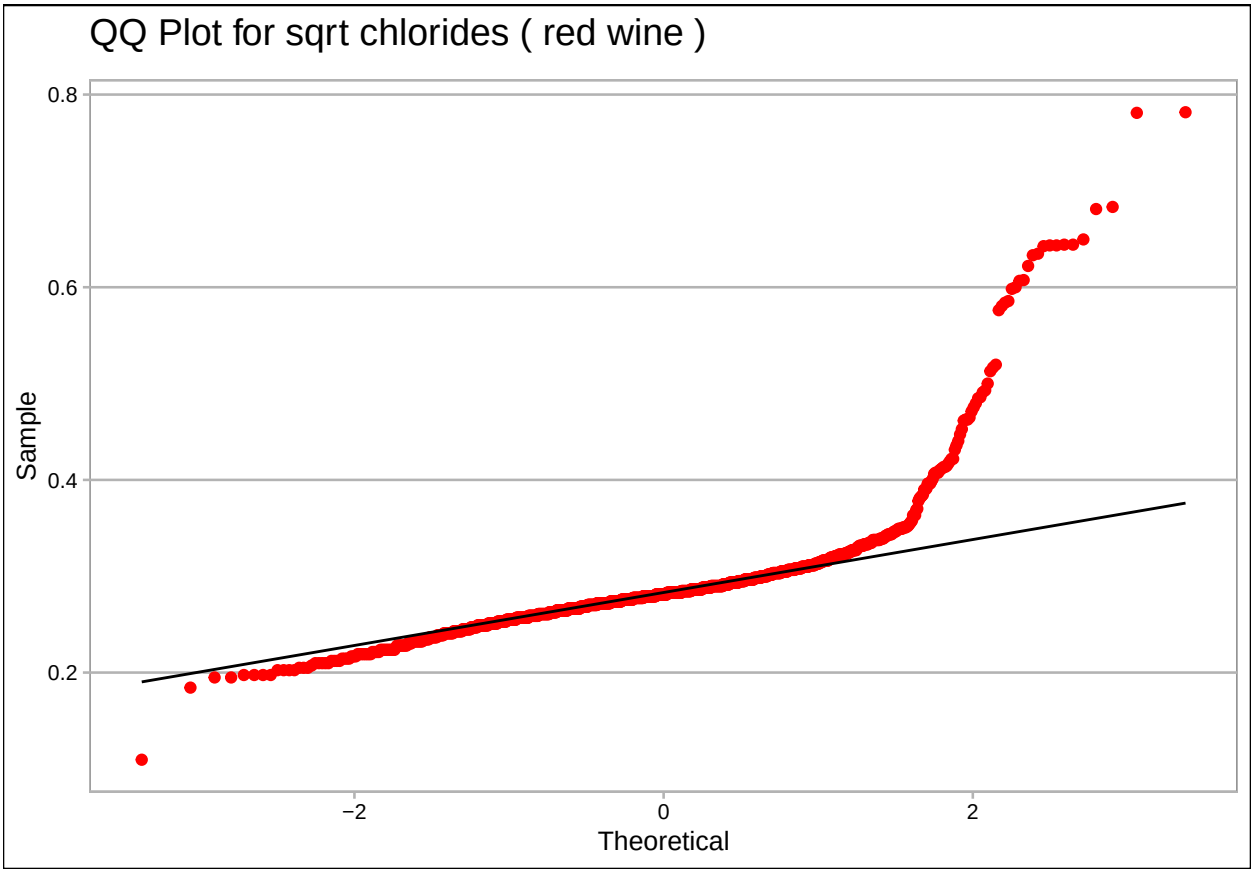
```
# Plotting the sqrt transformed QQ plots for all the variables in Red wine
for (i in names(red_wine_continuous)) {
  sqrt_qqplot(red_wine, i, 'red', 'red wine')
}
```


QQ Plot for sqrt fixed acidity (red wine)
4.0
3.5
3.0
2.5
2.0
Sample
−2
0
2
Theoretical

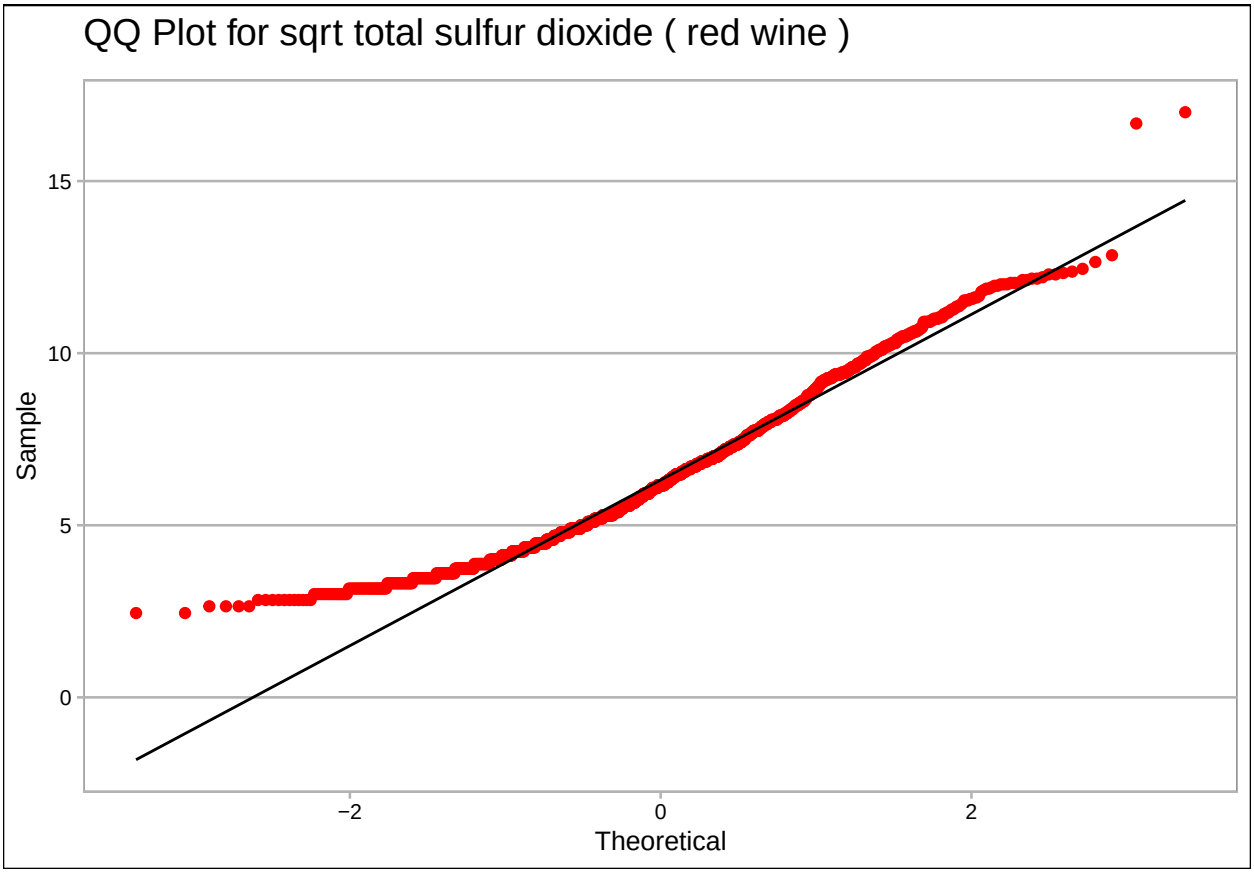

QQ Plot for sqrt volatile acidity (red wine)
1.25
1.00
0.75
0.50
0.25
Sample
−2
0
2
Theoretical

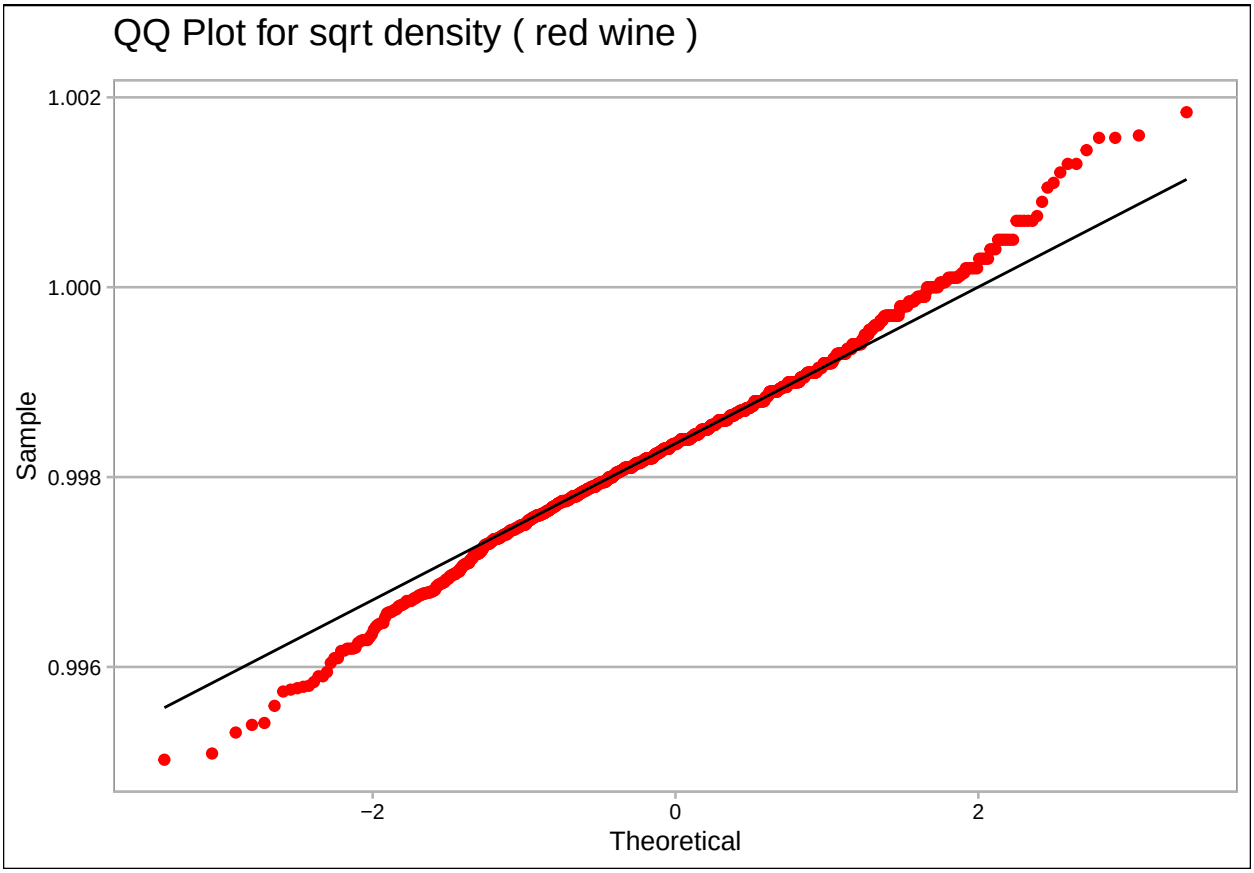

QQ Plot for sqrt citric acid (red wine)
Sample
1.0
0.5
0.0
−0.5
−2
0
2
Theoretical

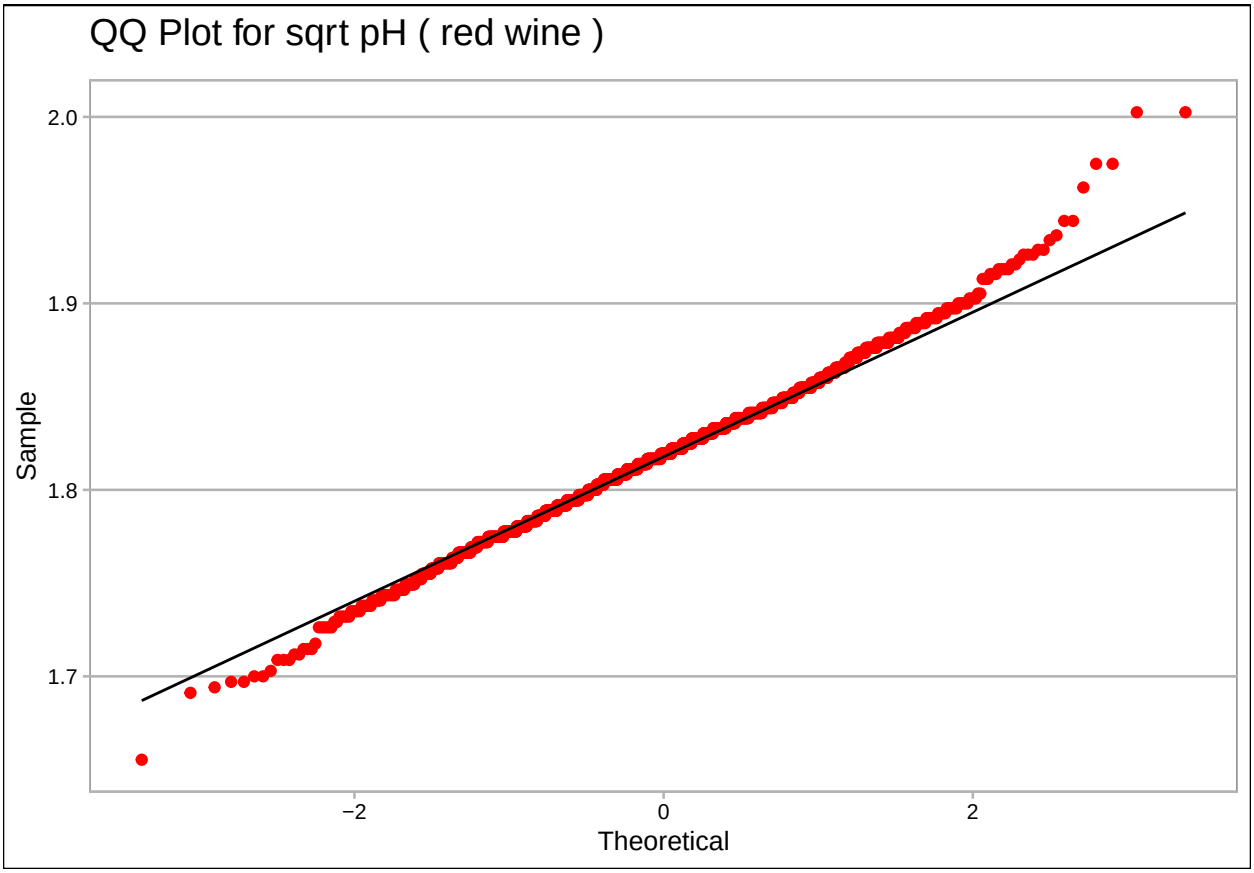

QQ Plot for sqrt residual sugar (red wine)
Sample
Theoretical
4
3
2
1
−2
0
2

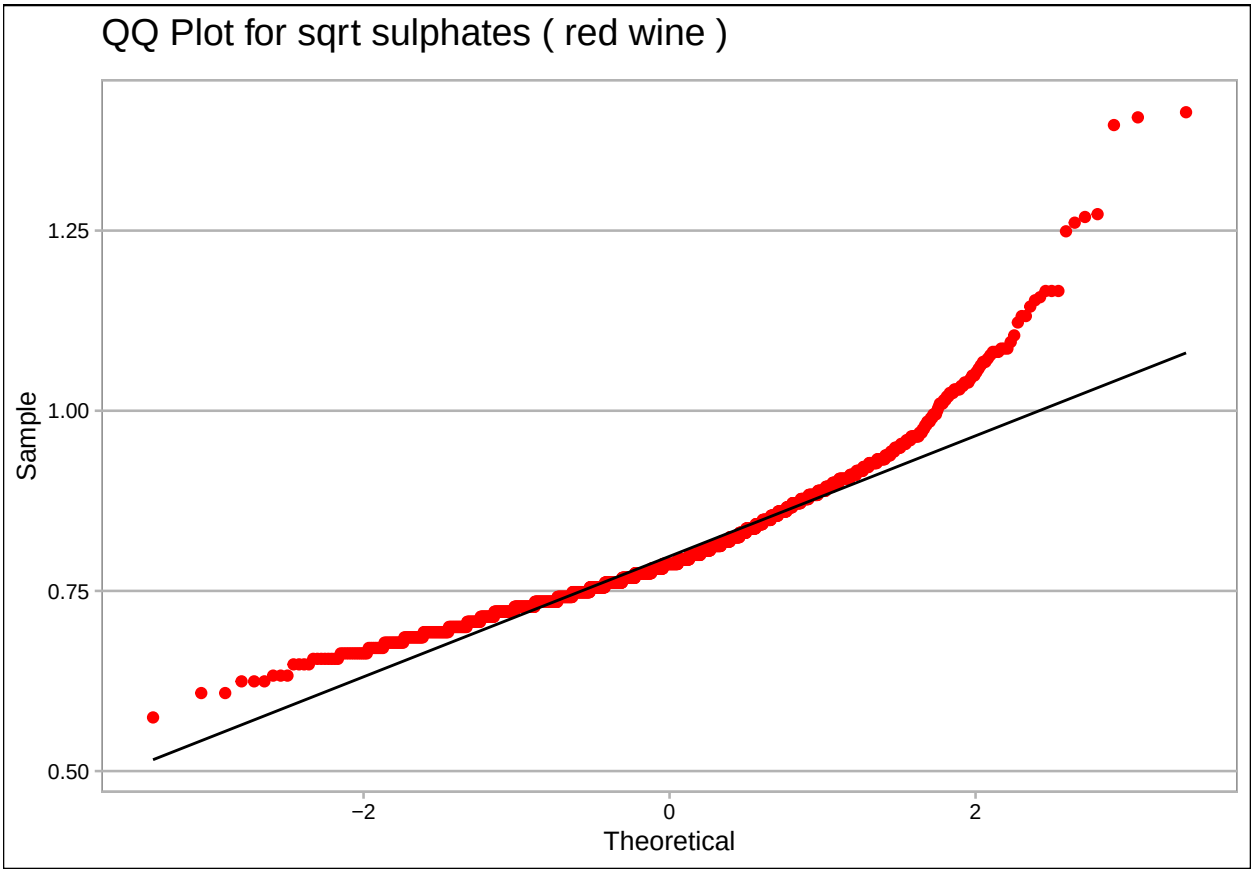

QQ Plot for sqrt chlorides (red wine)
0.8
0.6
0.4
0.2
Sample
−2
0
2
Theoretical

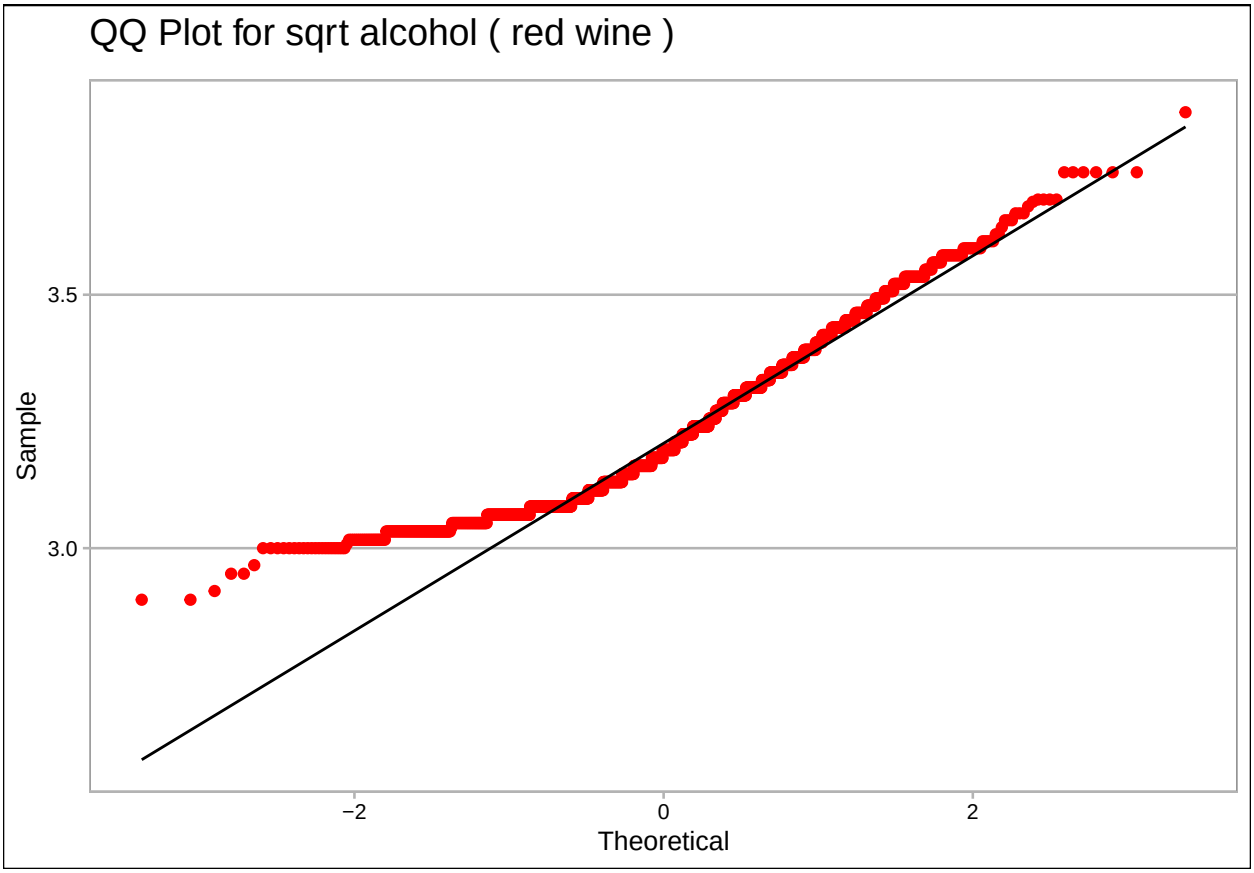

QQ Plot for sqrt free sulfur dioxide (red wine)
Sample
Theoretical
8
6
4
2
0
−2
0
2


QQ Plot for sqrt total sulfur dioxide (red wine)
Sample
15
10
5
0
−2
0
2
Theoretical


QQ Plot for sqrt density (red wine)
1.002
1.000
0.998
0.996
Sample
−2
0
2
Theoretical

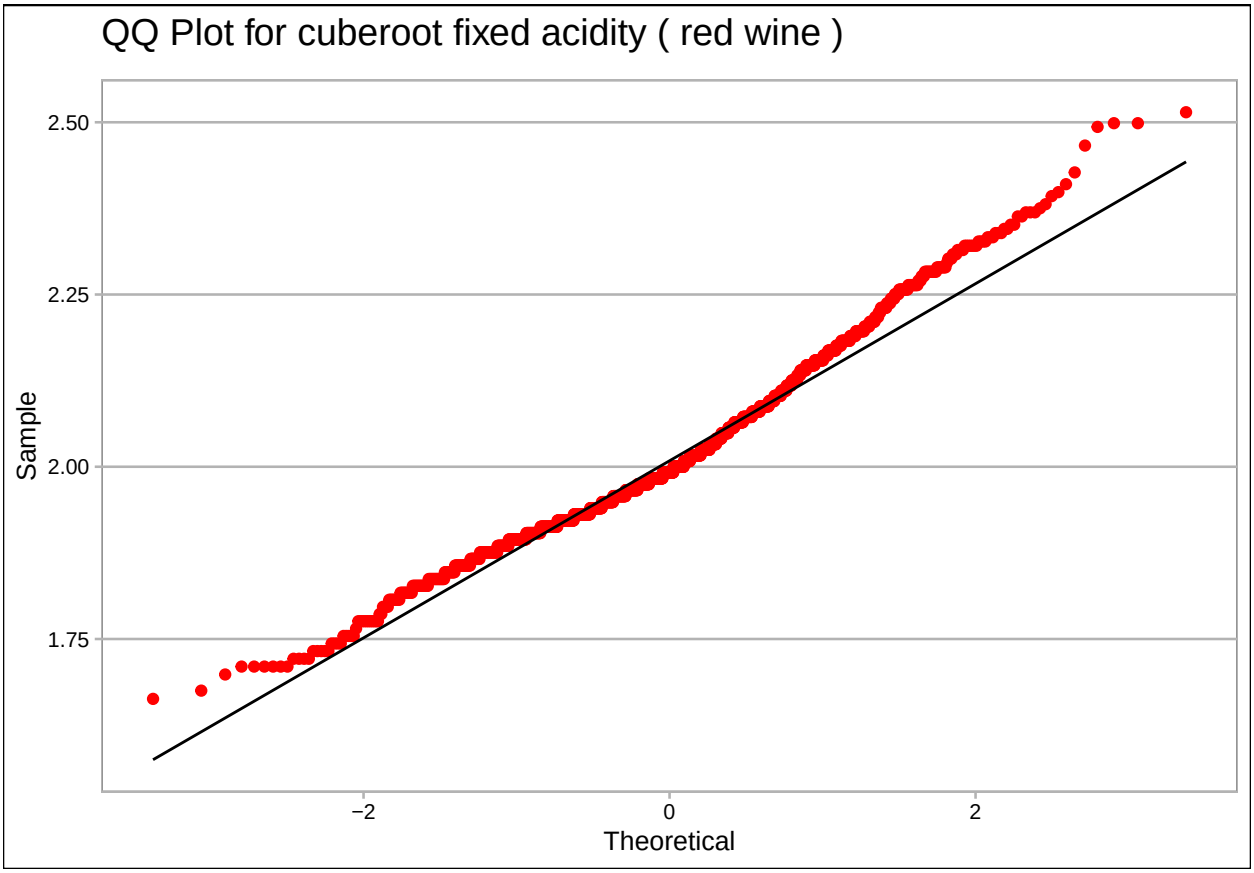

QQ Plot for sqrt pH (red wine)
2.0
1.9
1.8
1.7
Sample
−2
0
2
Theoretical

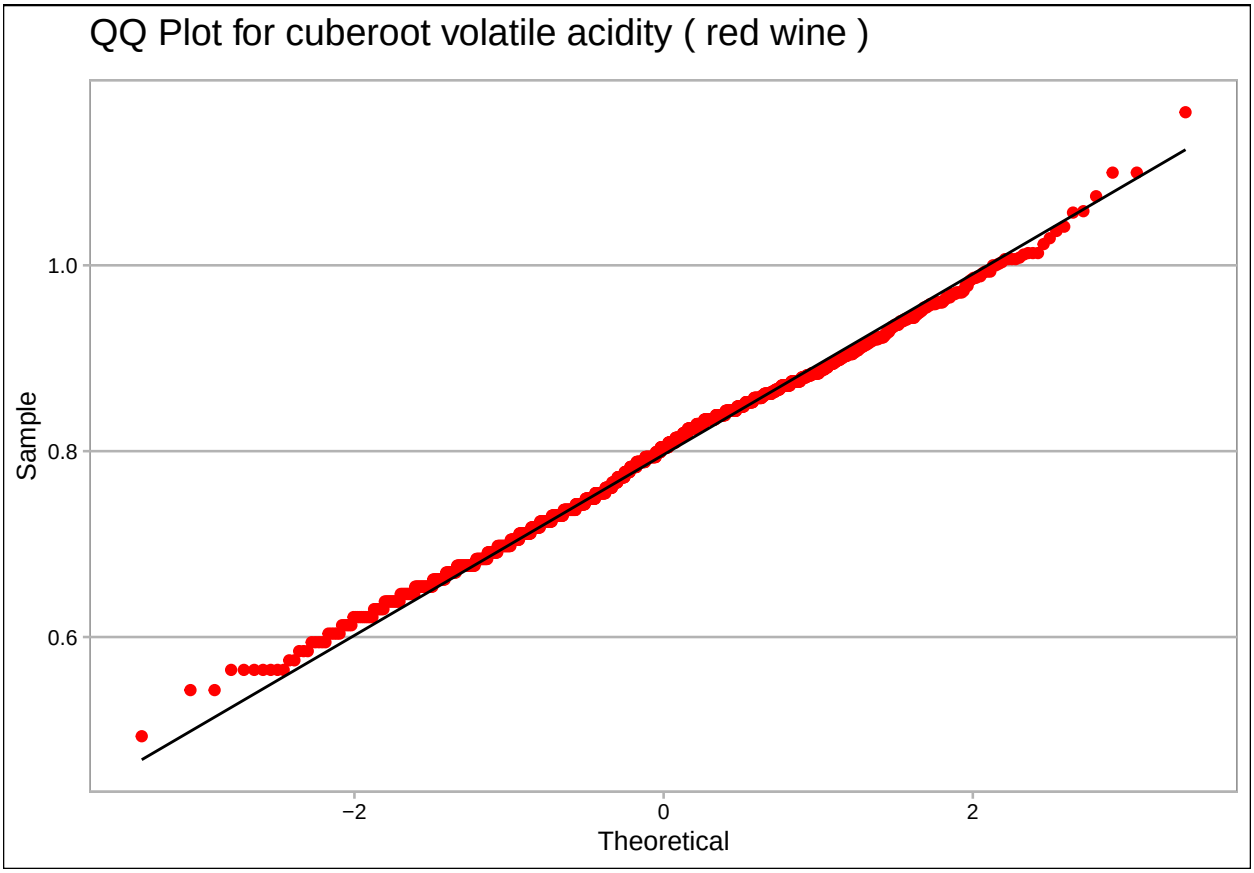

QQ Plot for sqrt sulphates (red wine)
Sample
1.25
1.00
0.75
0.50
−2
0
2
Theoretical

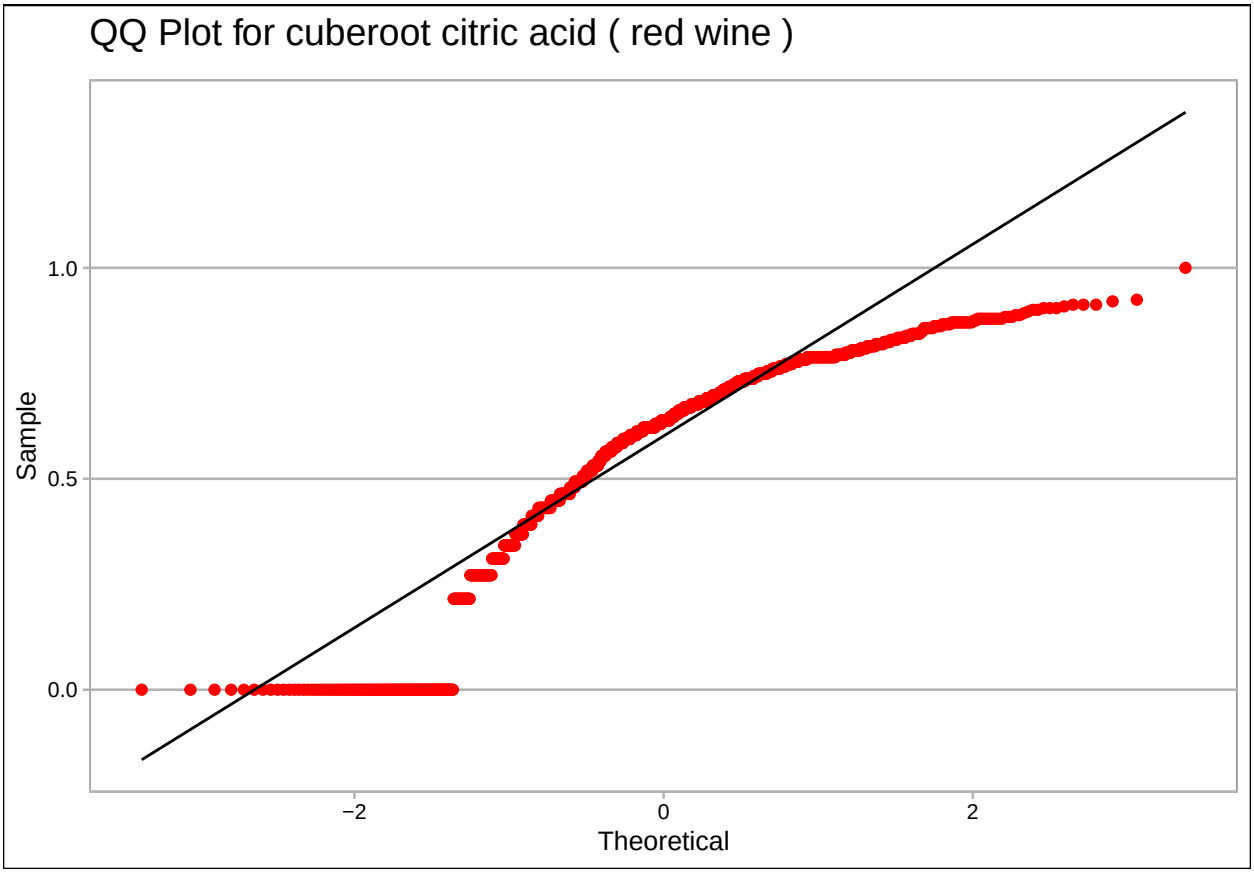


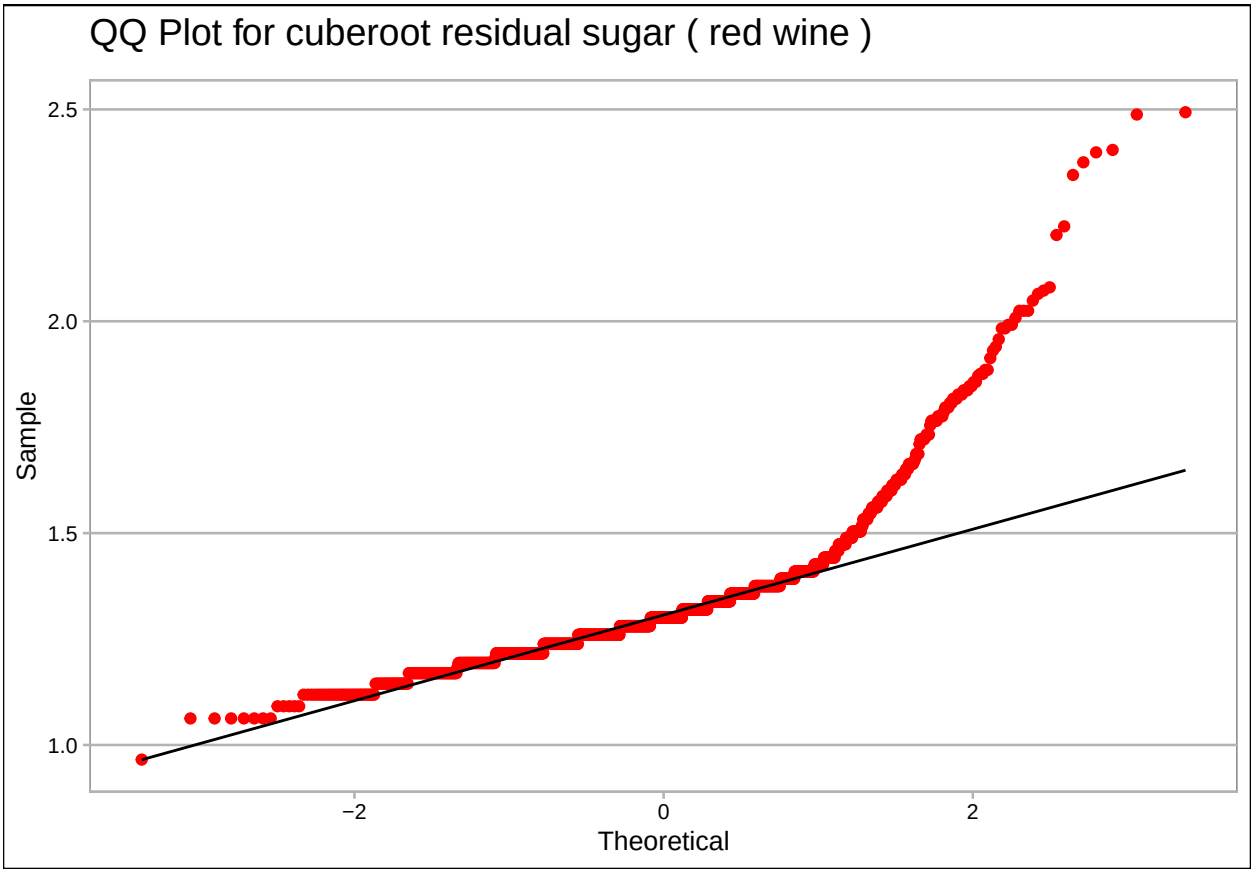
CUBEROOT TRANSFORMED QQPLOT FOR RED WINE

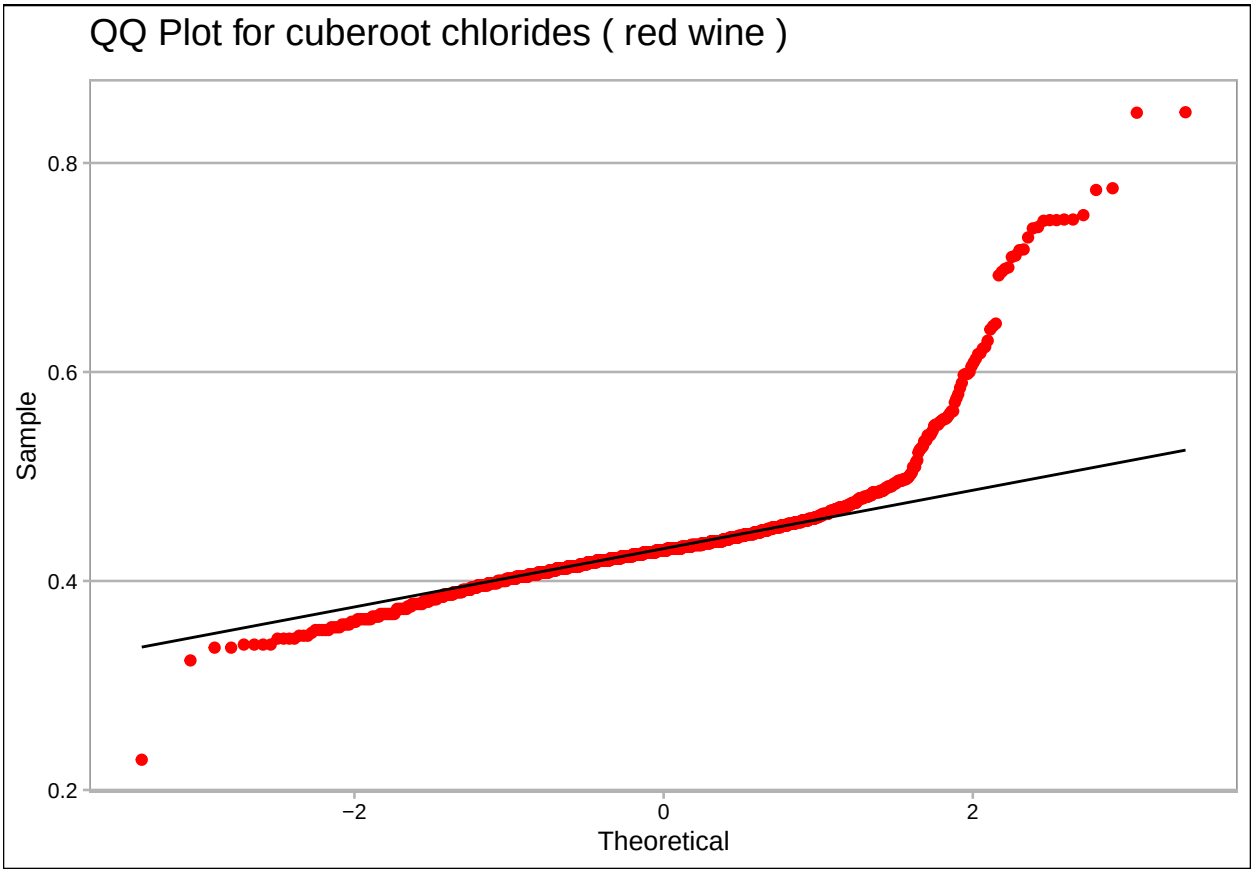
```
# Plotting the cuberoot transformed QQ plots for all the variables in Red wine
for (i in names(red_wine_continuous)) {
  cuberoot_qqplot(red_wine, i, 'red', 'red wine')
}
```

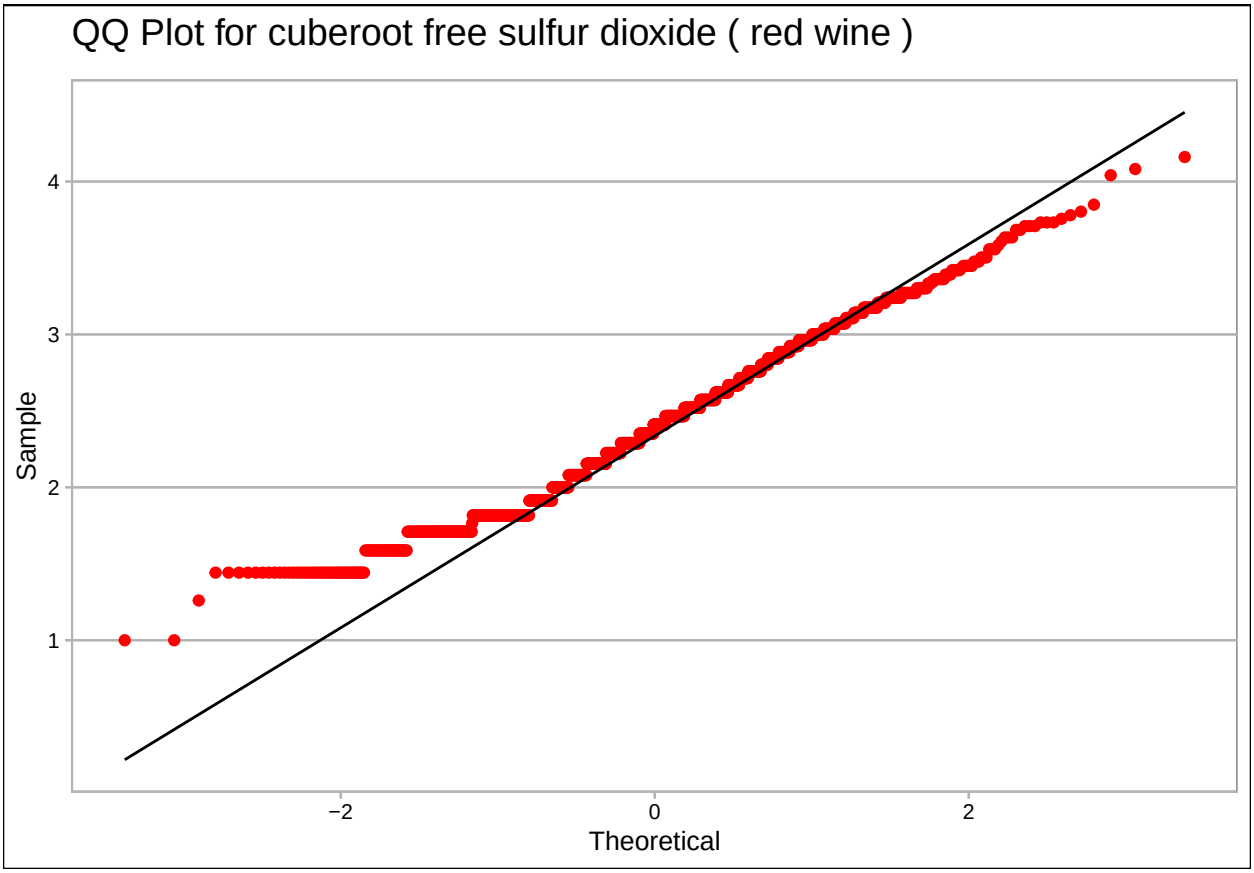

QQ Plot for cuberoot fixed acidity (red wine)
2.50
2.25
2.00
1.75
Sample
−2
0
2
Theoretical

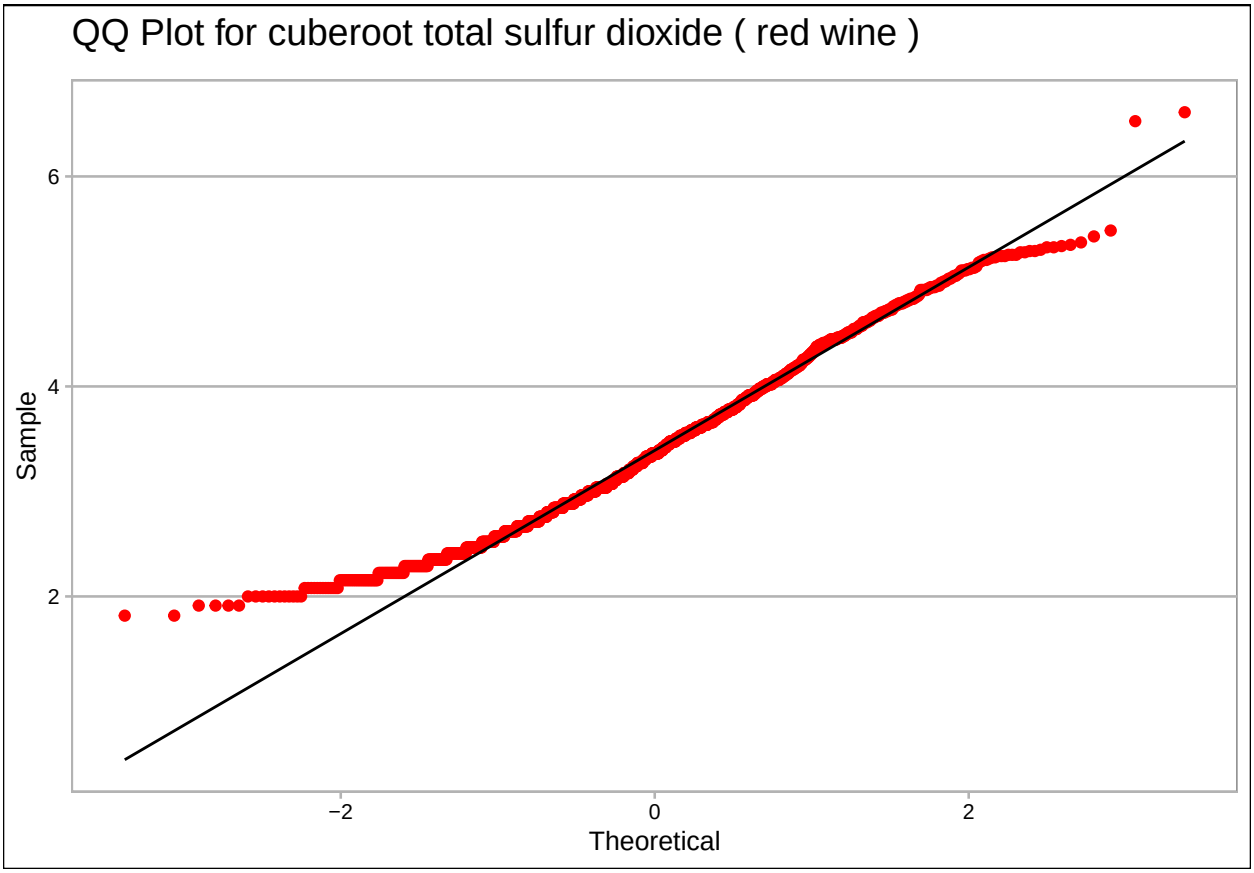

QQ Plot for cuberoot volatile acidity (red wine)
Sample
1.0
0.8
0.6
−2
0
2
Theoretical

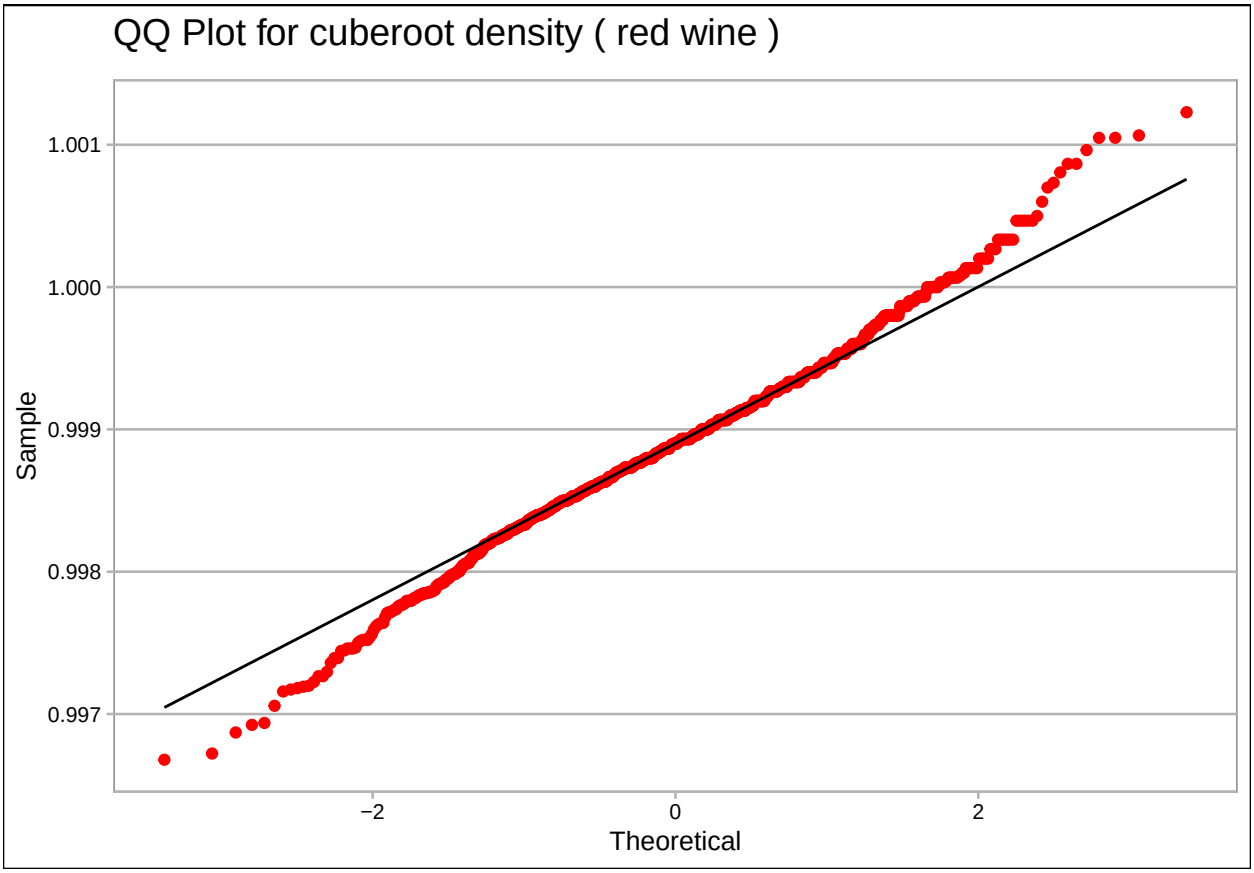

QQ Plot for cuberoot citric acid (red wine)
Sample
1.0
0.5
0.0
−2
0
2
Theoretical

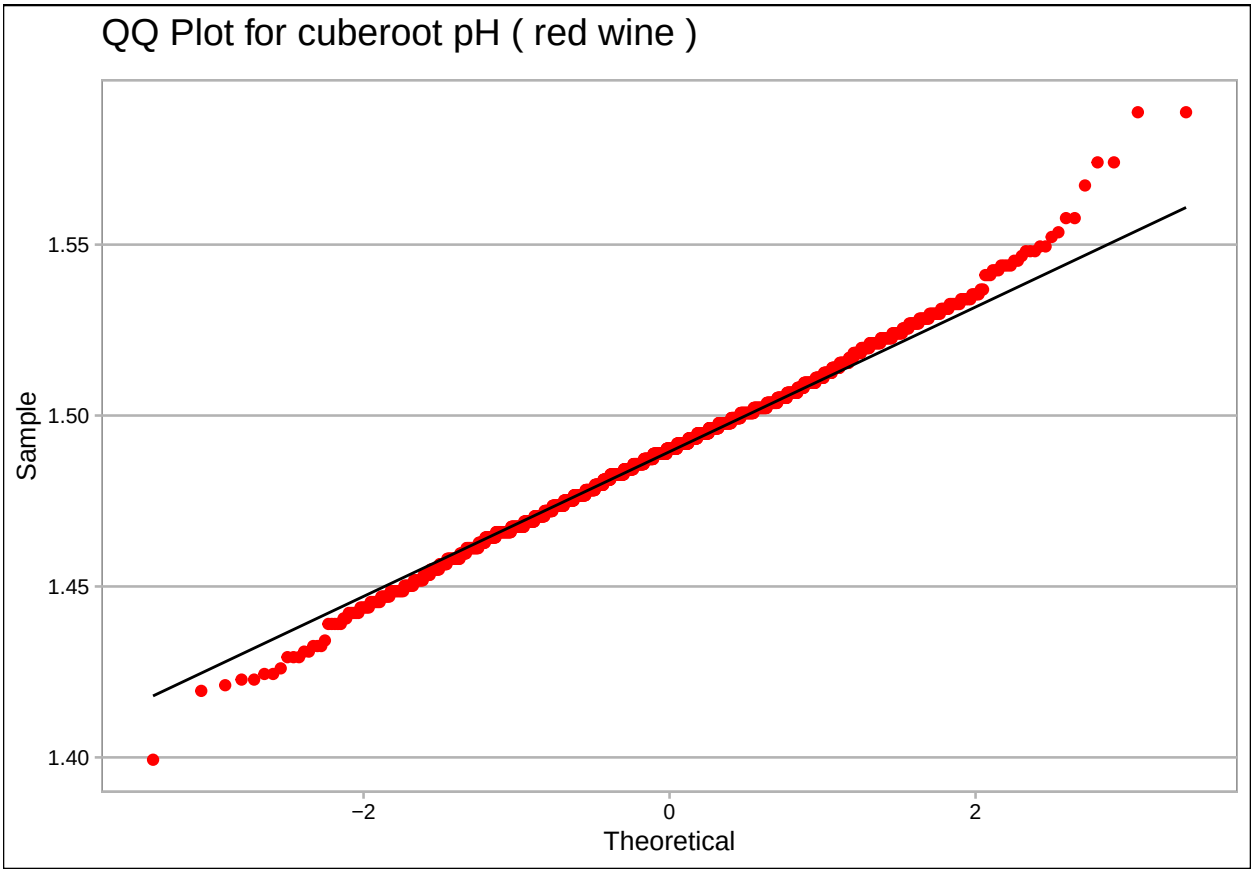

QQ Plot for cuberoot residual sugar (red wine)
2.5
2.0
1.5
1.0
Sample
−2
0
2
Theoretical

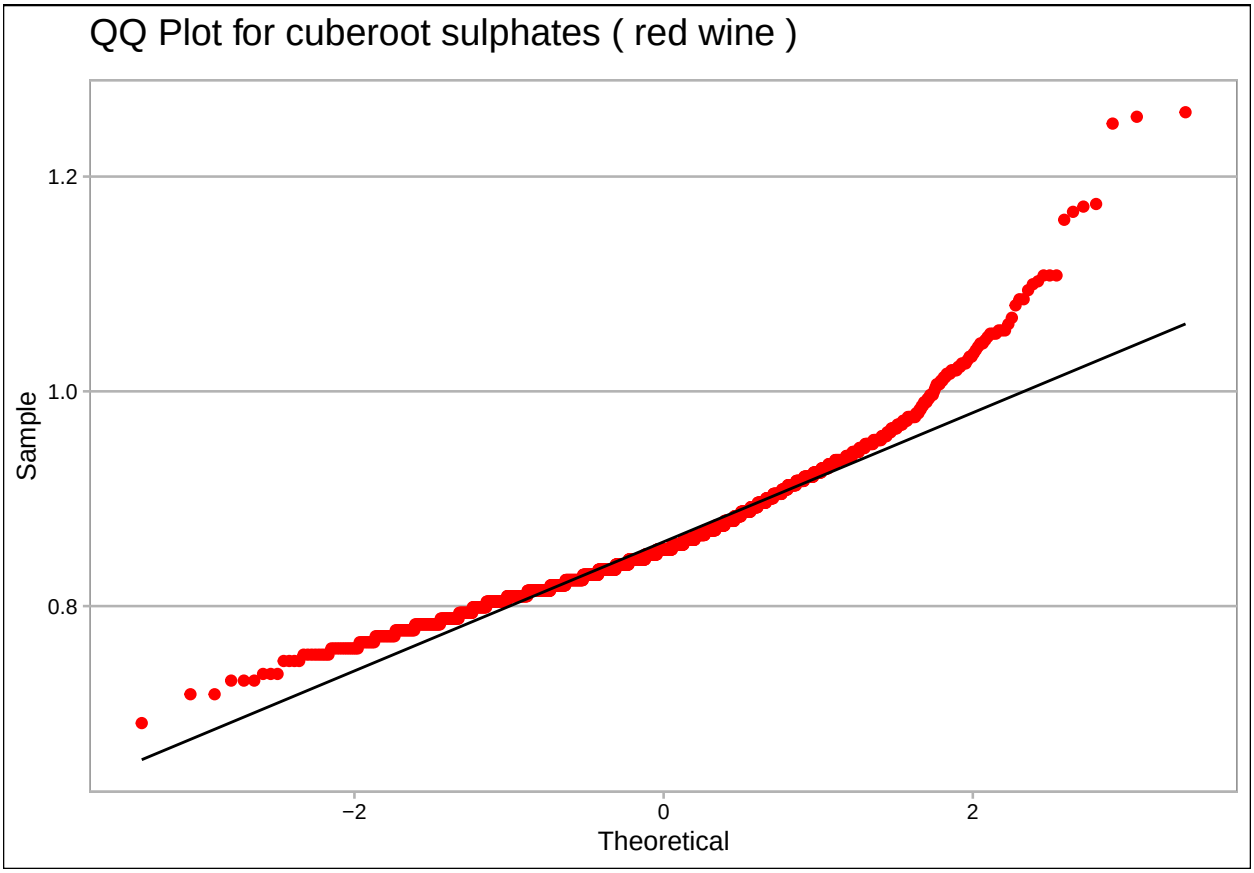

QQ Plot for cuberoot chlorides (red wine)
Sample
0.8
0.6
0.4
0.2
−2
0
2
Theoretical

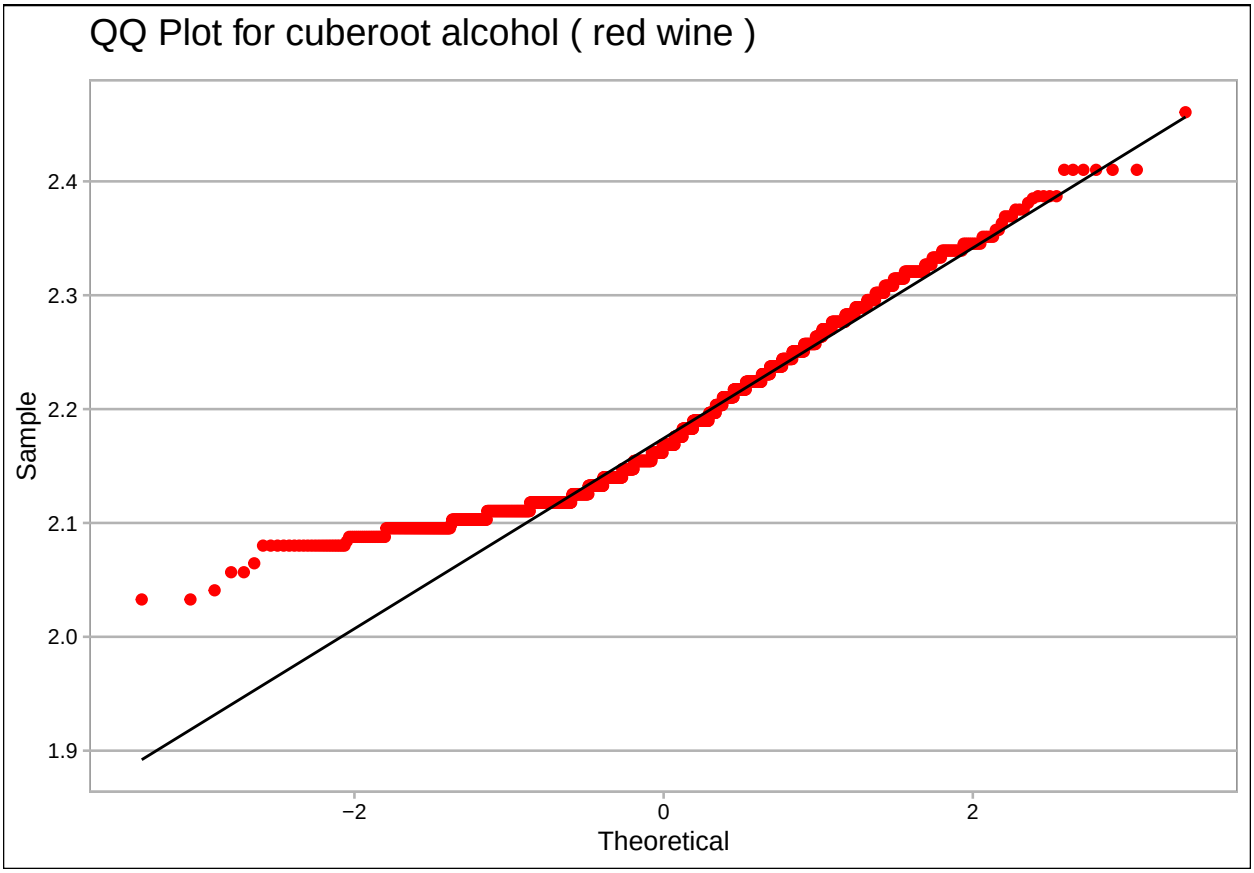

QQ Plot for cuberoot free sulfur dioxide (red wine)
Sample
Theoretical
4
3
2
1
−2
0
2


QQ Plot for cuberoot total sulfur dioxide (red wine)
Sample
6
4
2
−2
0
2
Theoretical


QQ Plot for cuberoot density (red wine)
1.001
1.000
0.999
0.998
0.997
Sample
−2
0
2
Theoretical

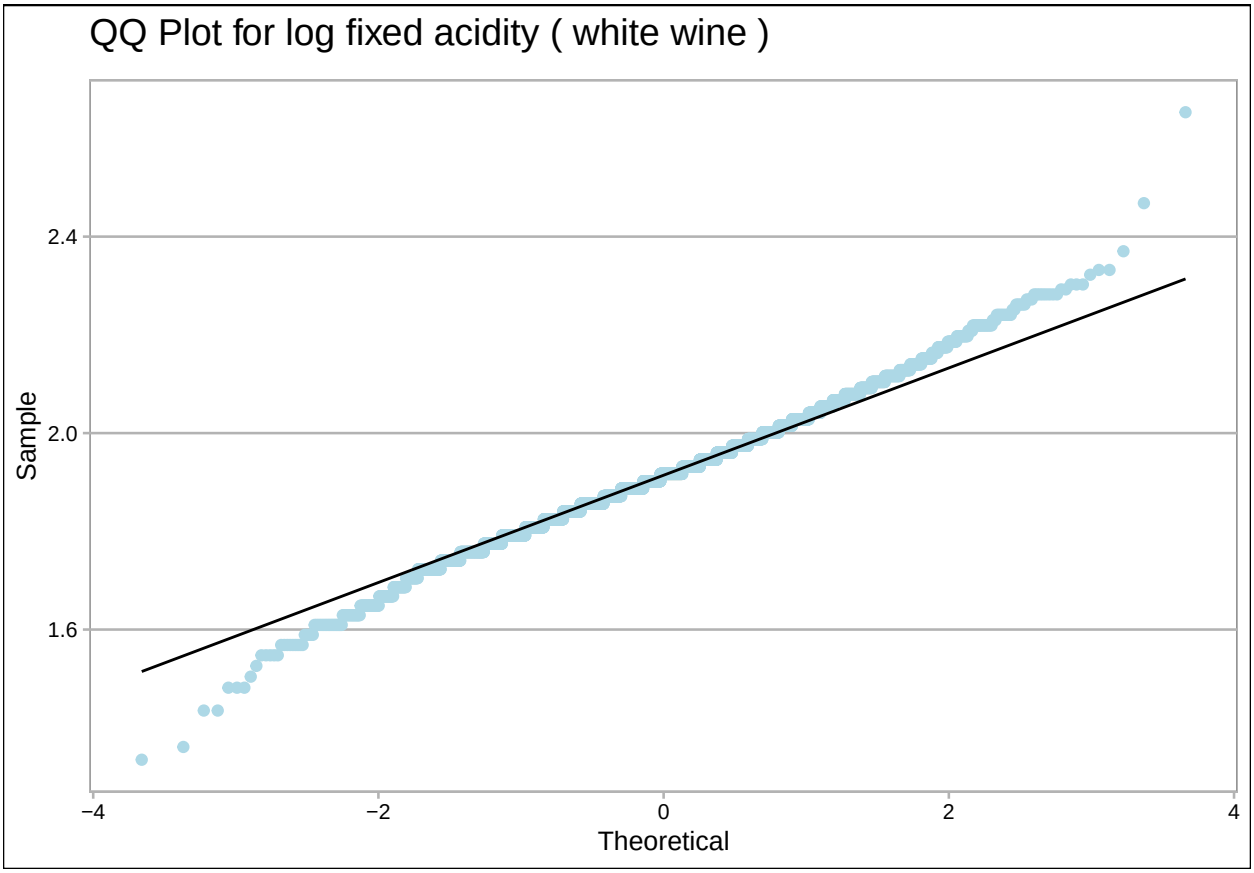

QQ Plot for cuberoot pH (red wine)
Sample
1.55
1.50
1.45
1.40
−2
0
2
Theoretical

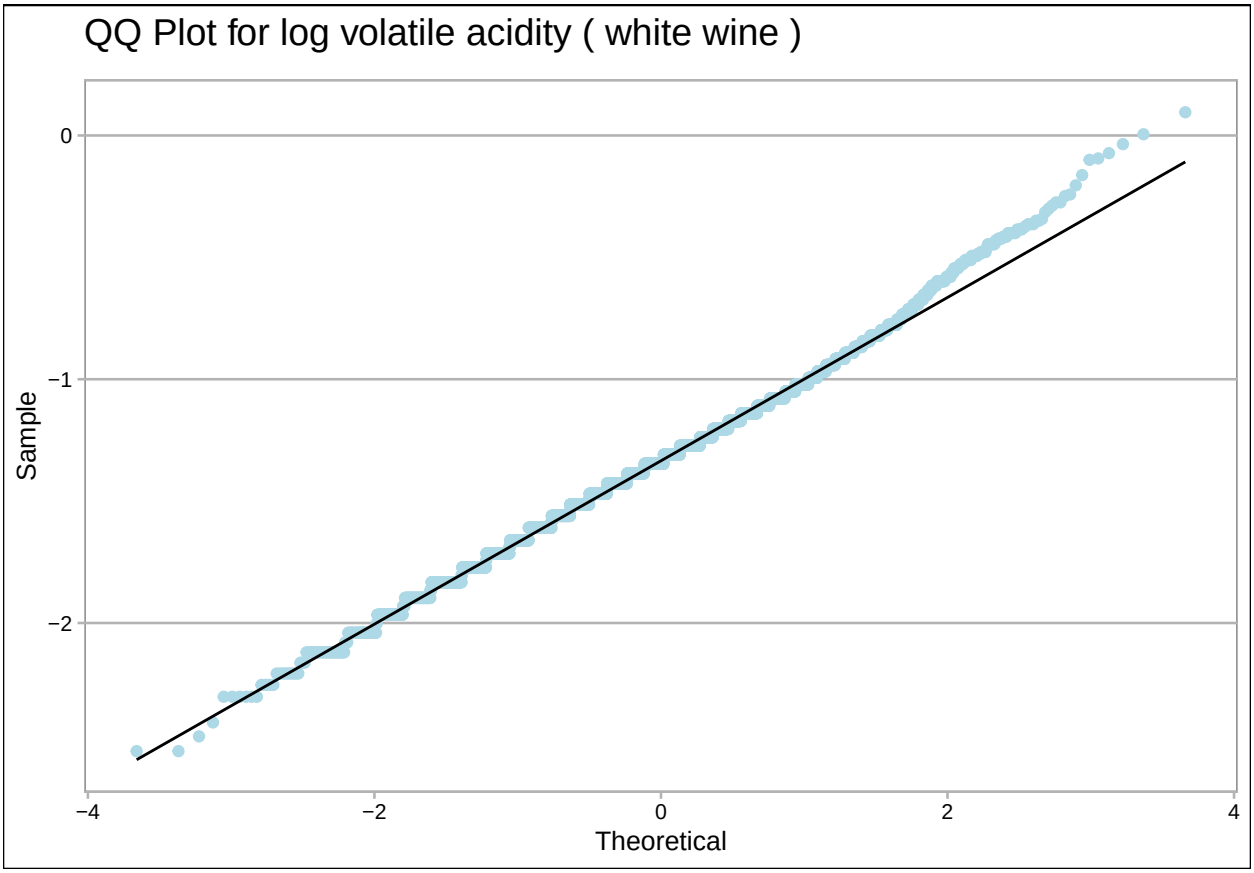

QQ Plot for cuberoot sulphates (red wine)
Sample
1.2
1.0
0.8
−2
0
2
Theoretical



LOG TRANSFORMED QQPLOT FOR WHITE WINE

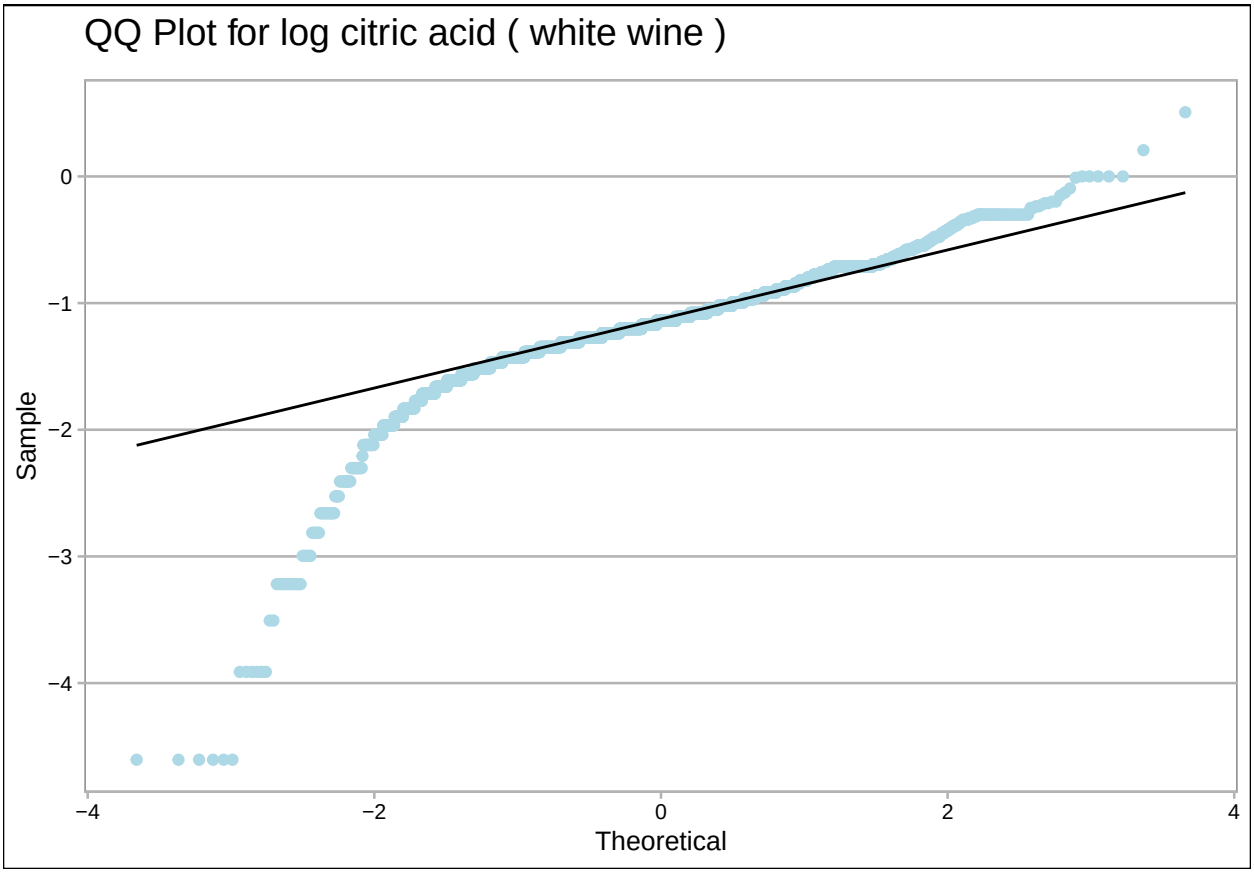
```
# Plotting the log transformed QQ plots for all the variables in White wine
for (i in names(white_wine_continuous)) {
  log_qqplot(white_wine, i, 'lightblue', 'white wine')
}
```

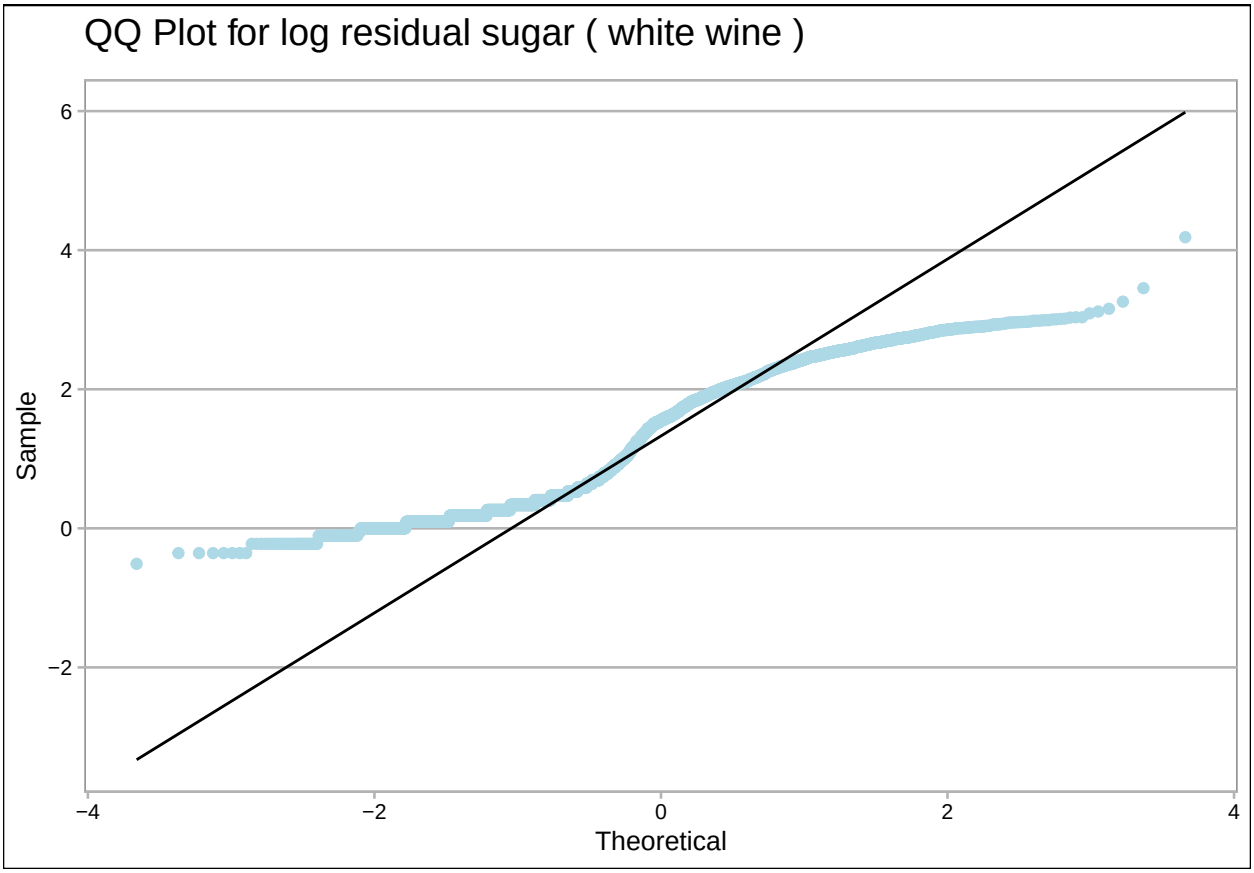

QQ Plot for log fixed acidity (white wine)
Sample
2.4
2.0
1.6
−4
−2
0
2
4
Theoretical

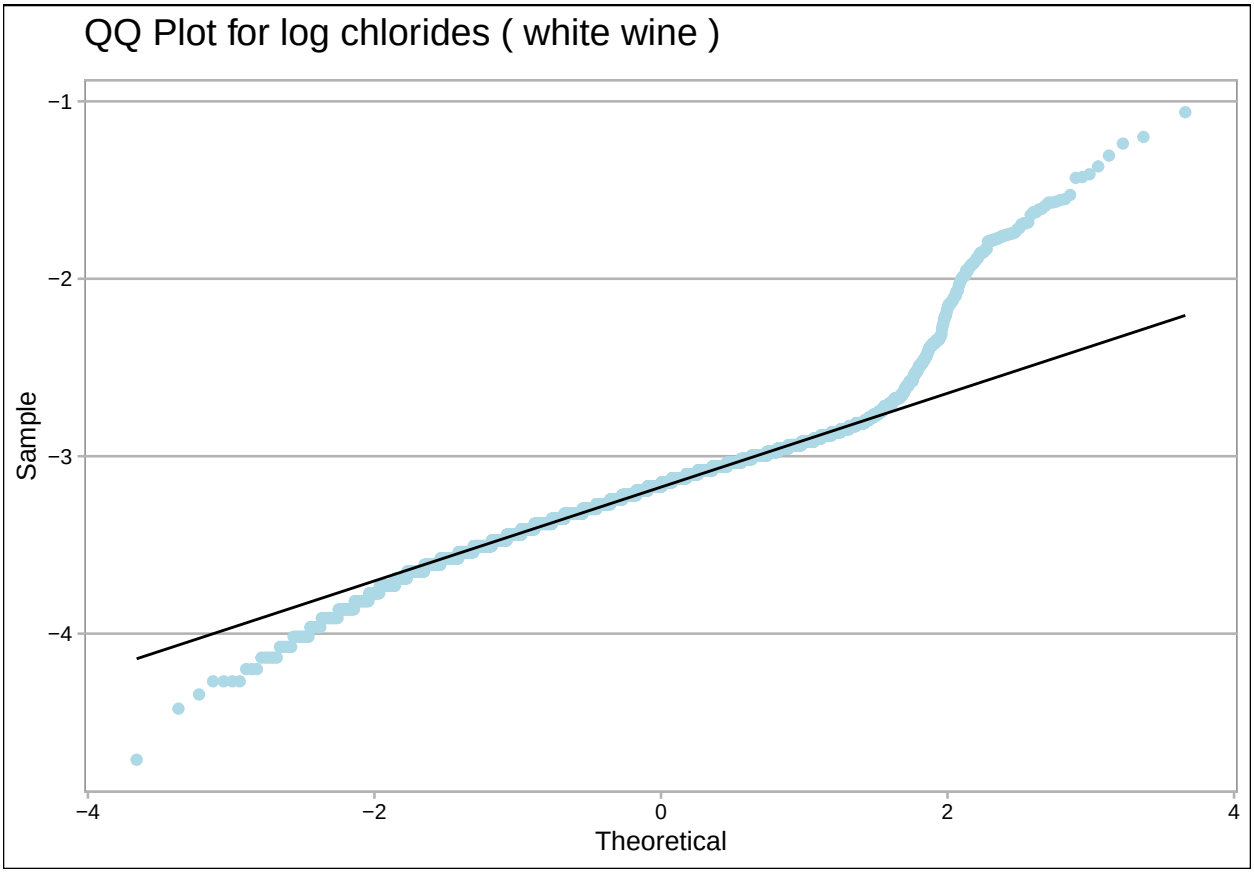


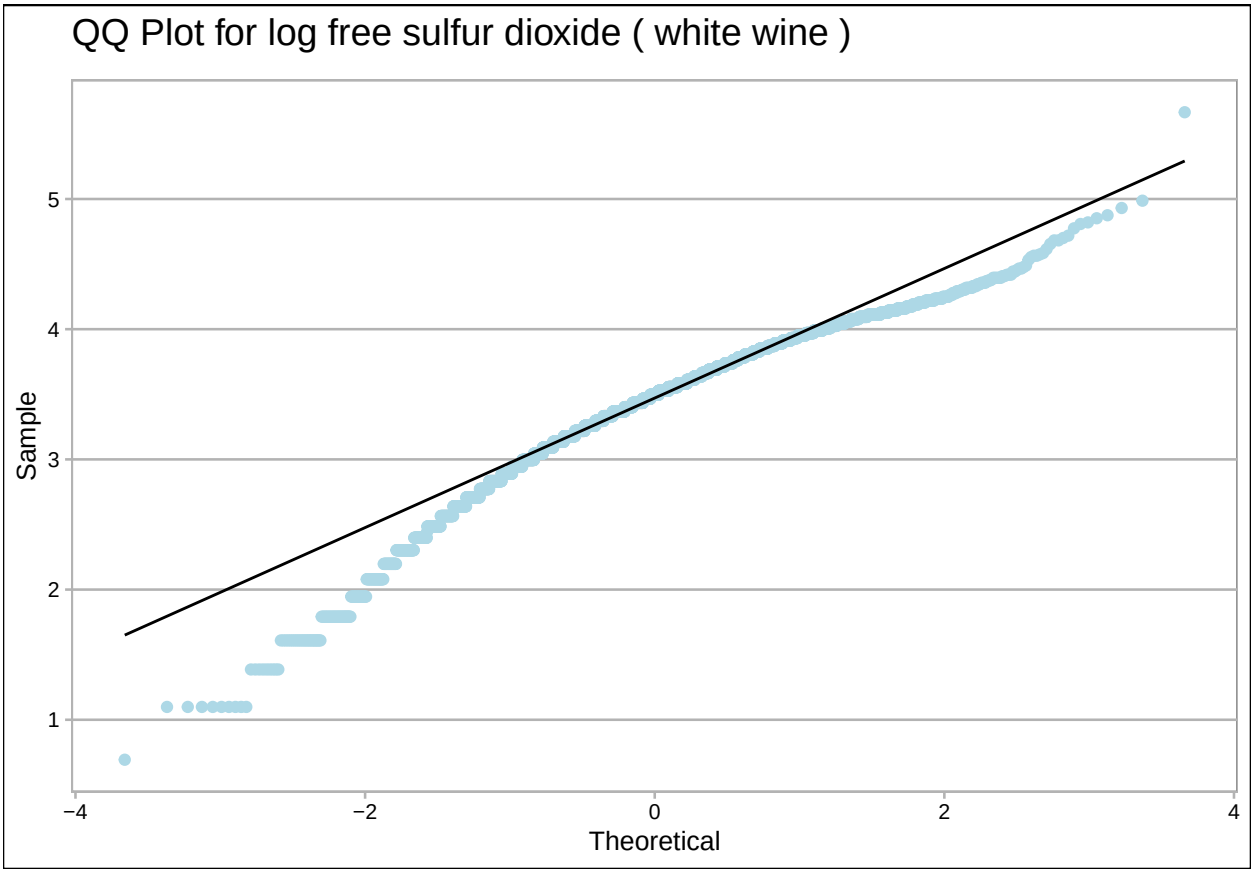

```
## Warning: Removed 18 rows containing non-finite outside the scale range
## (`stat_qq()`).
```

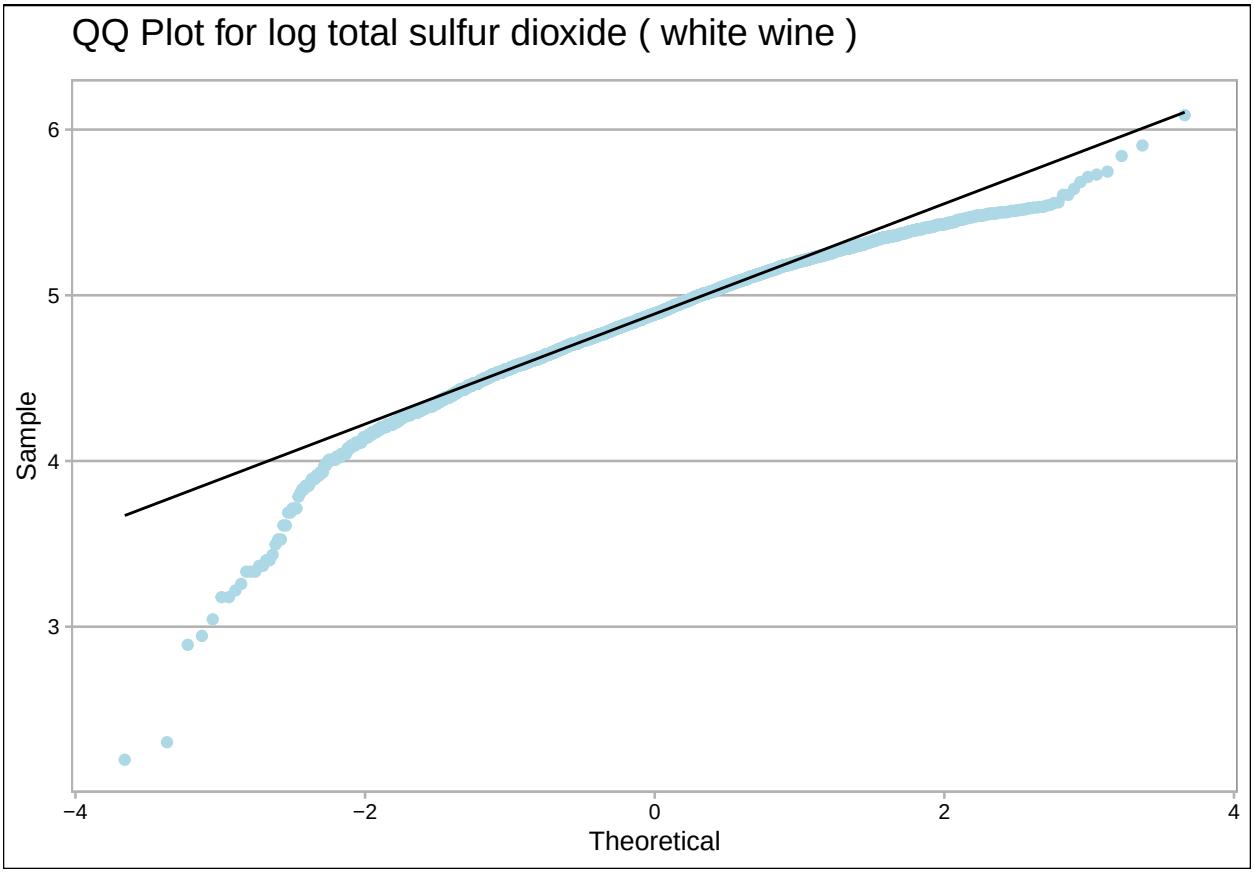
```
## Warning: Removed 18 rows containing non-finite outside the scale range
## (`stat_qq_line()`).
```

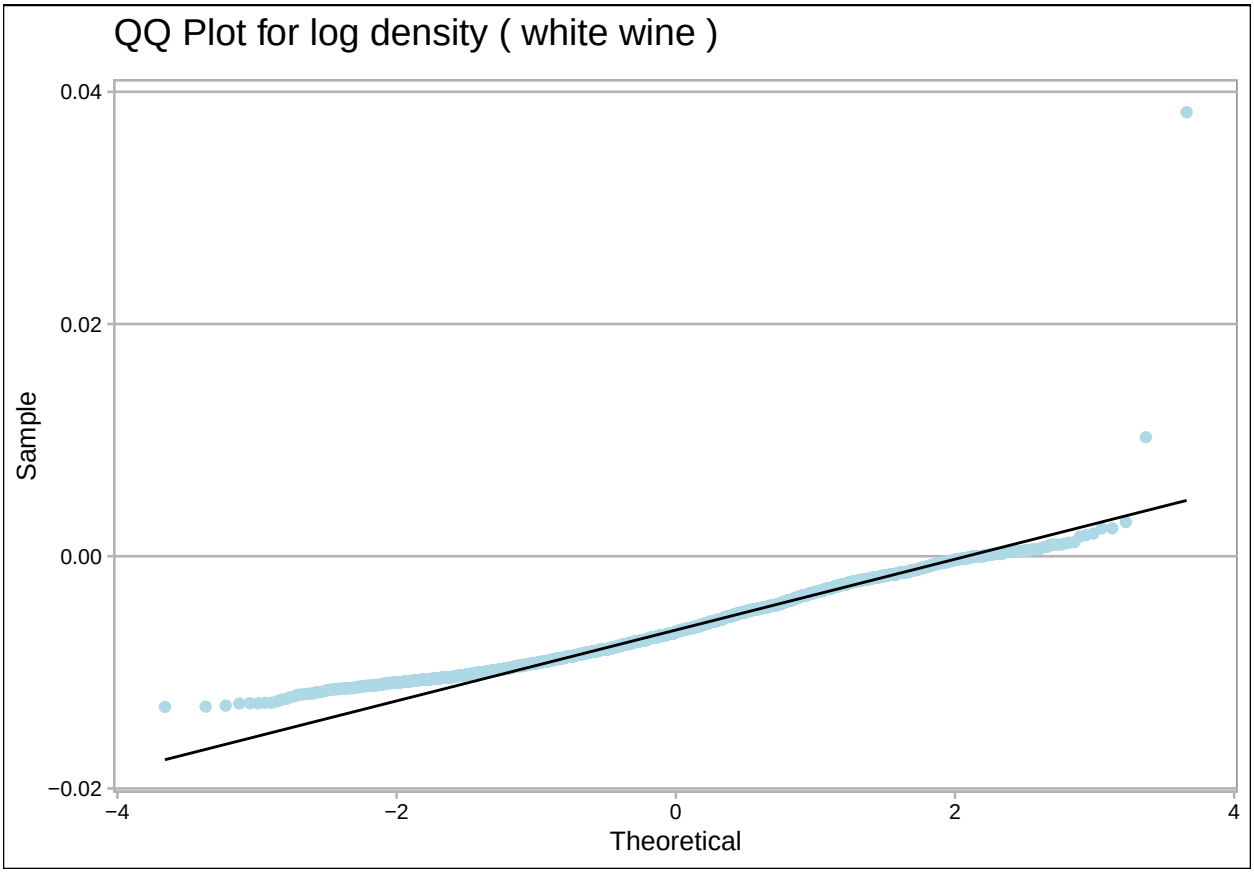

QQ Plot for log citric acid (white wine)
Sample
Theoretical
0
−1
−2
−3
−4
−4
−2
0
2
4

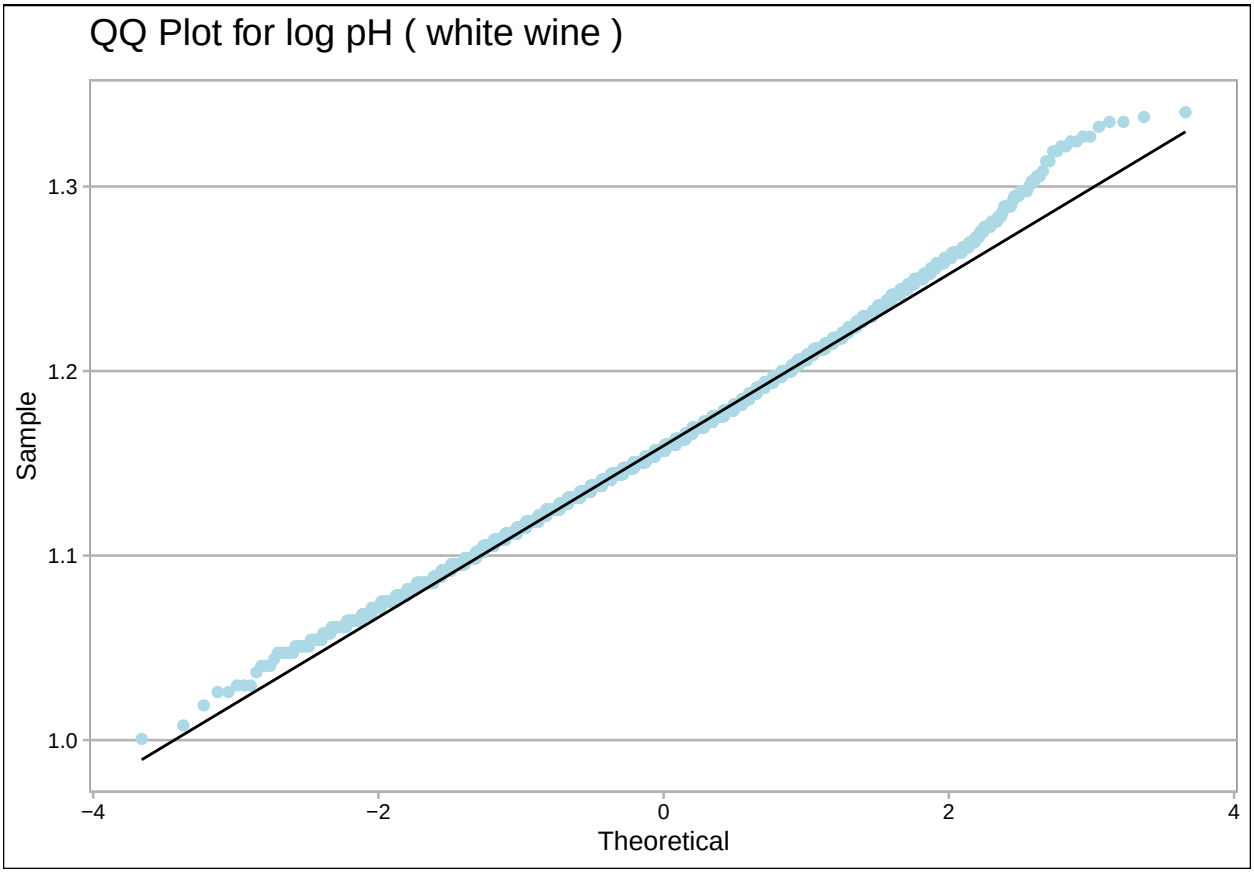

QQ Plot for log residual sugar (white wine)
Sample
Theoretical
6
4
2
0
−2
−4
−2
0
2
4

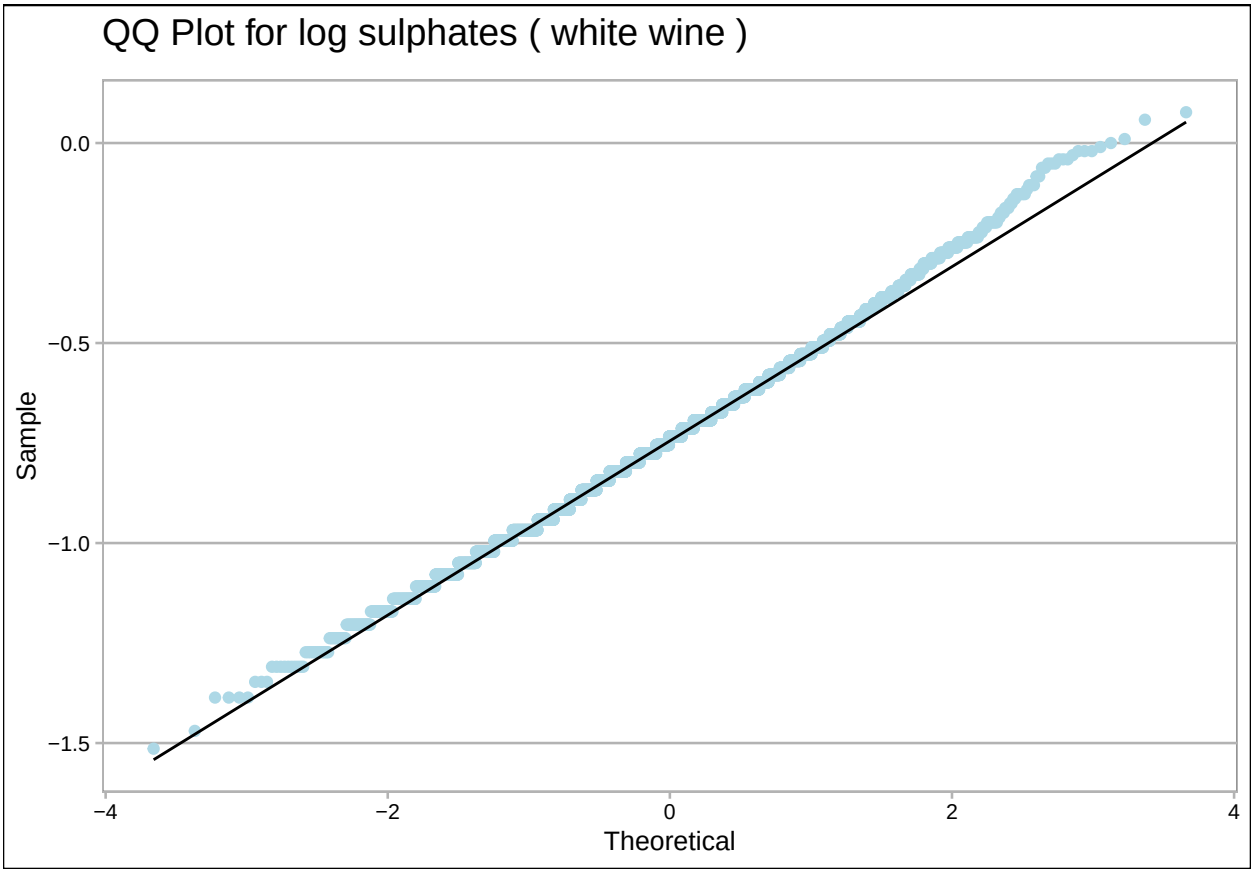

QQ Plot for log chlorides (white wine)
Sample
−1
−2
−3
−4
−4
−2
0
2
4
Theoretical

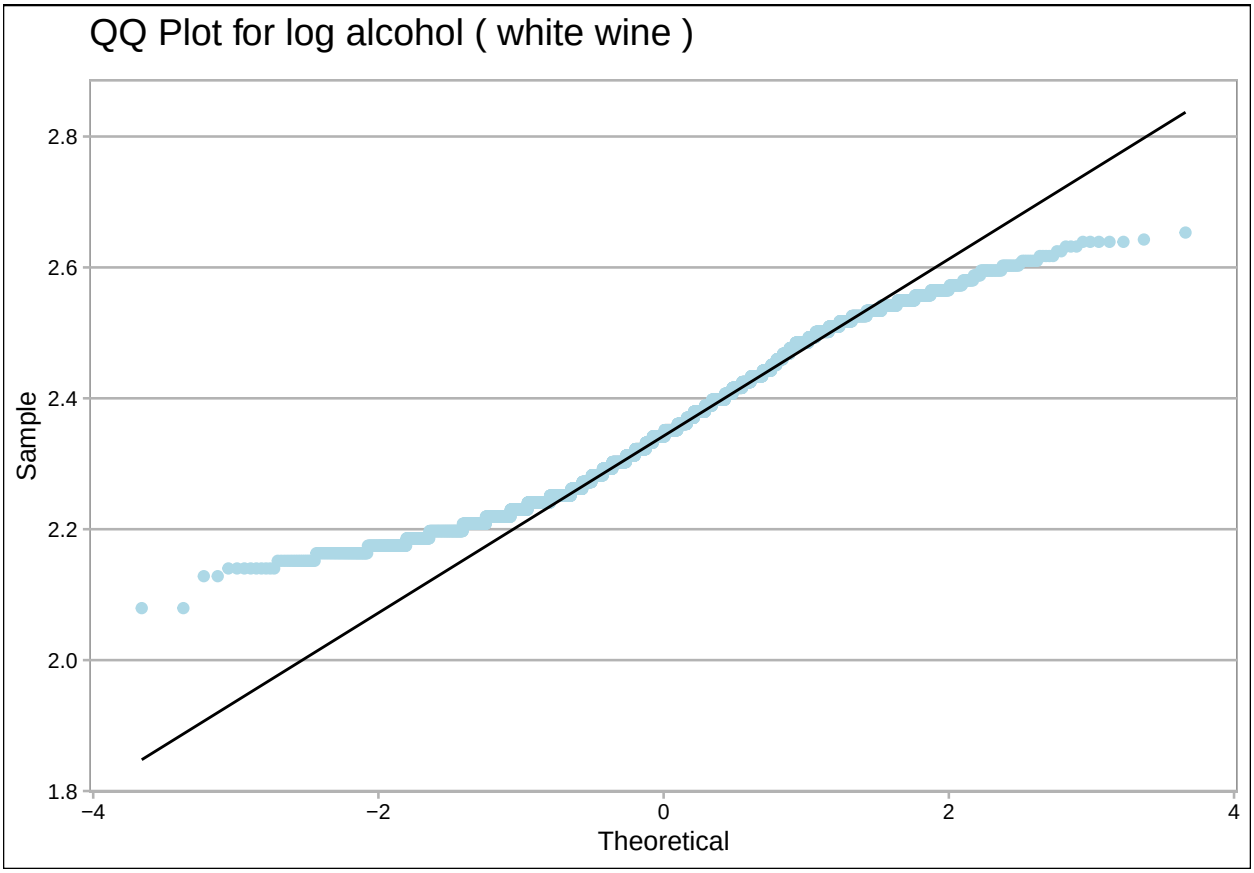

QQ Plot for log free sulfur dioxide (white wine)
Sample
Theoretical
5
4
3
2
1
−4
−2
0
2
4


QQ Plot for log total sulfur dioxide (white wine)
Sample
6
5
4
3
−4
−2
0
2
4
Theoretical


QQ Plot for log density (white wine)
0.04
0.02
0.00
−0.02
Sample
−4
−2
0
2
4
Theoretical

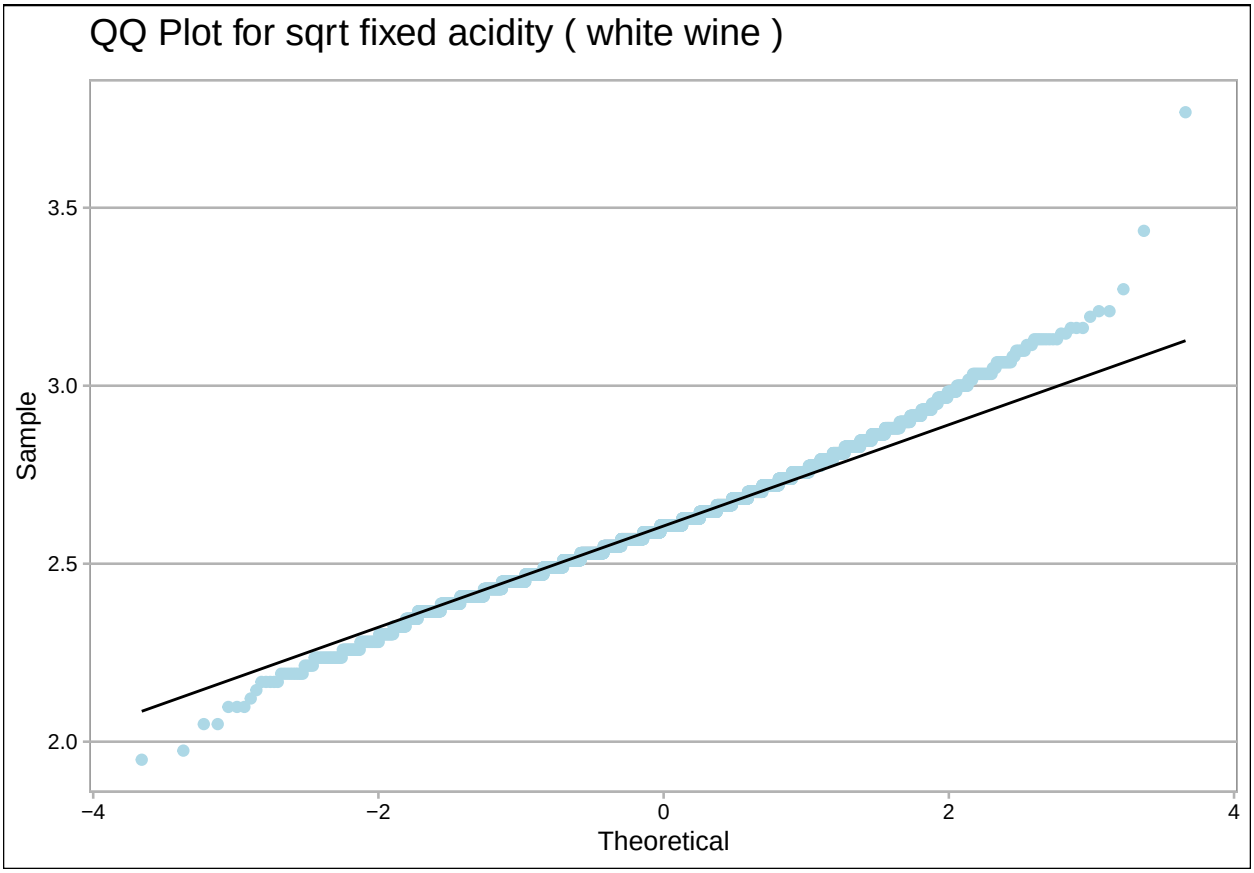

QQ Plot for log pH (white wine)
Sample
1.3
1.2
1.1
1.0
−4
−2
0
2
4
Theoretical

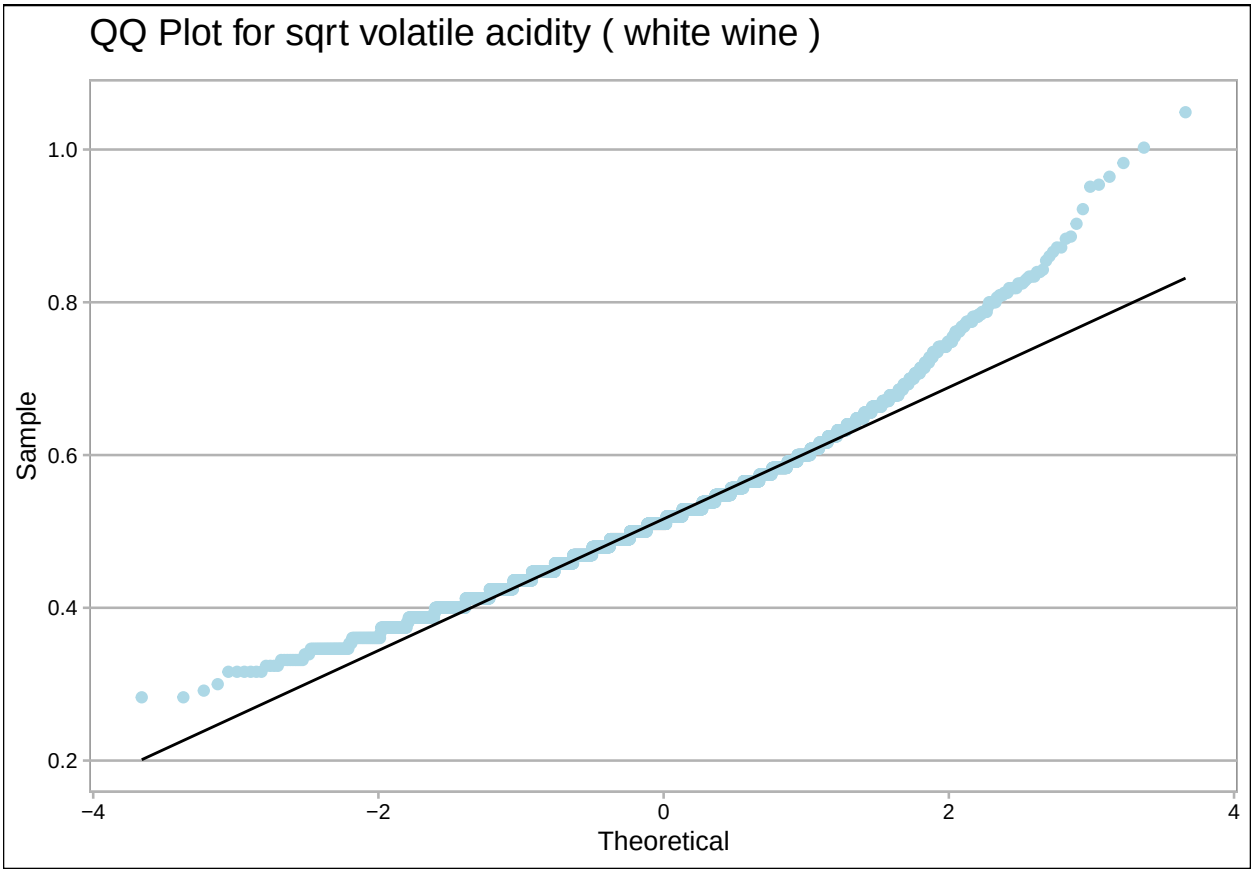

QQ Plot for log sulphates (white wine)
0.0
−0.5
−1.0
−1.5
Sample
−4
−2
0
2
4
Theoretical

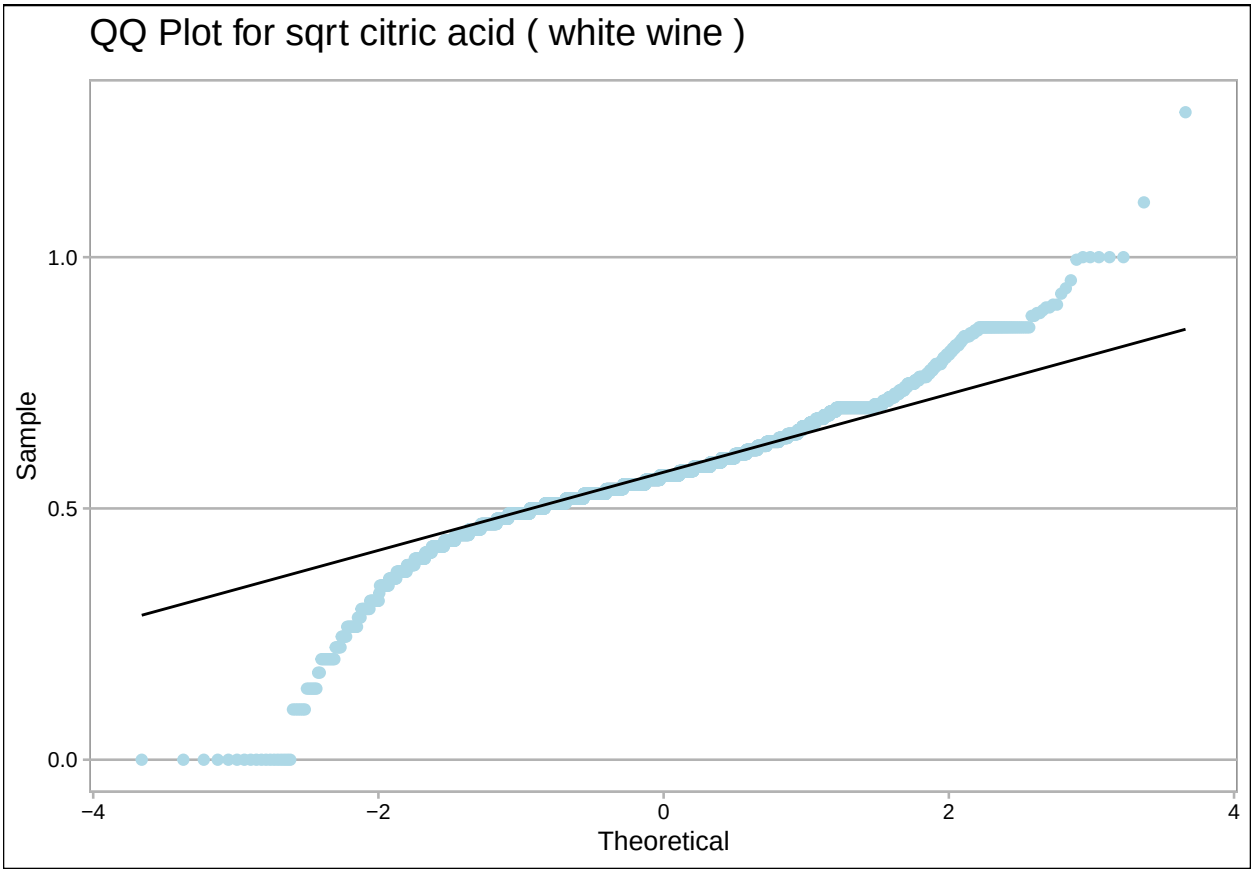


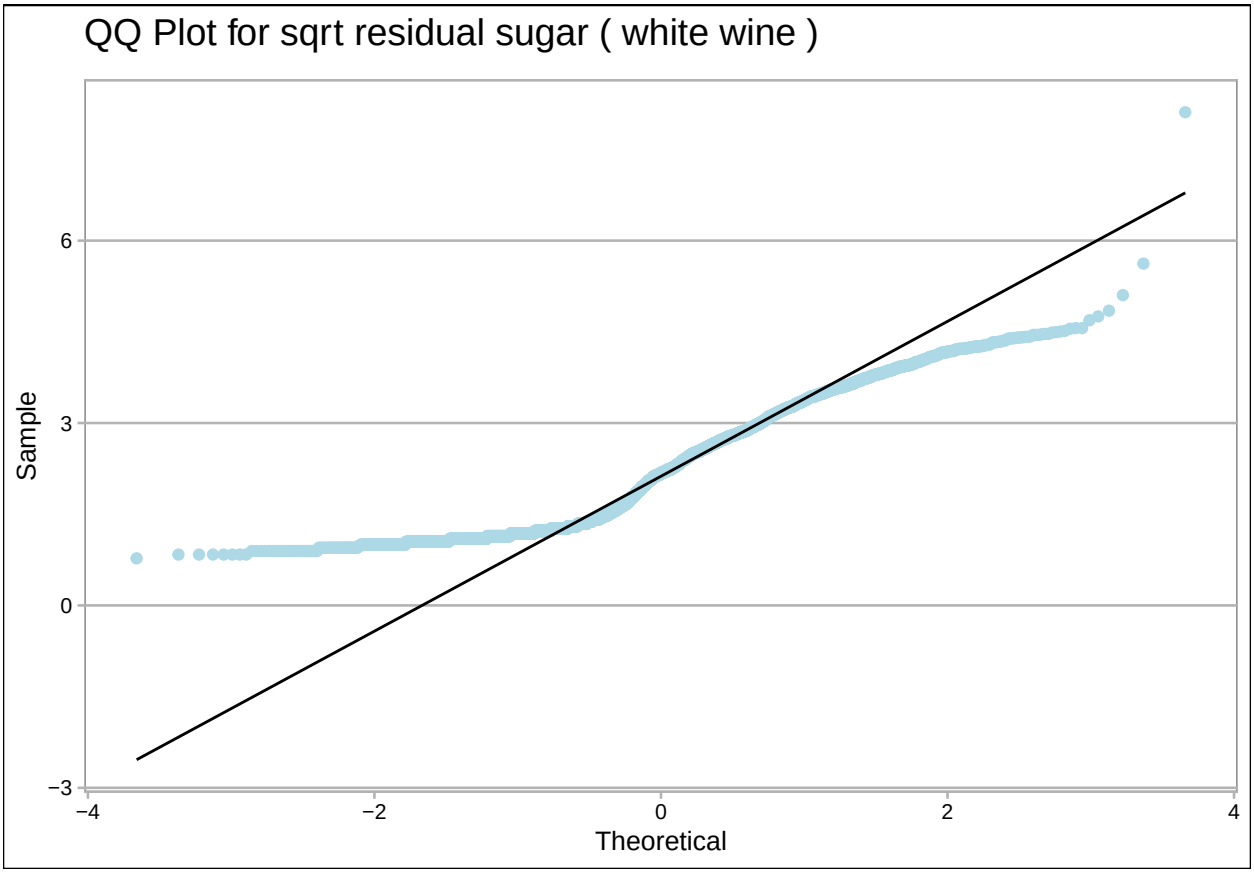
SQRT TRANSFORMED QQPLOT FOR WHITE WINE

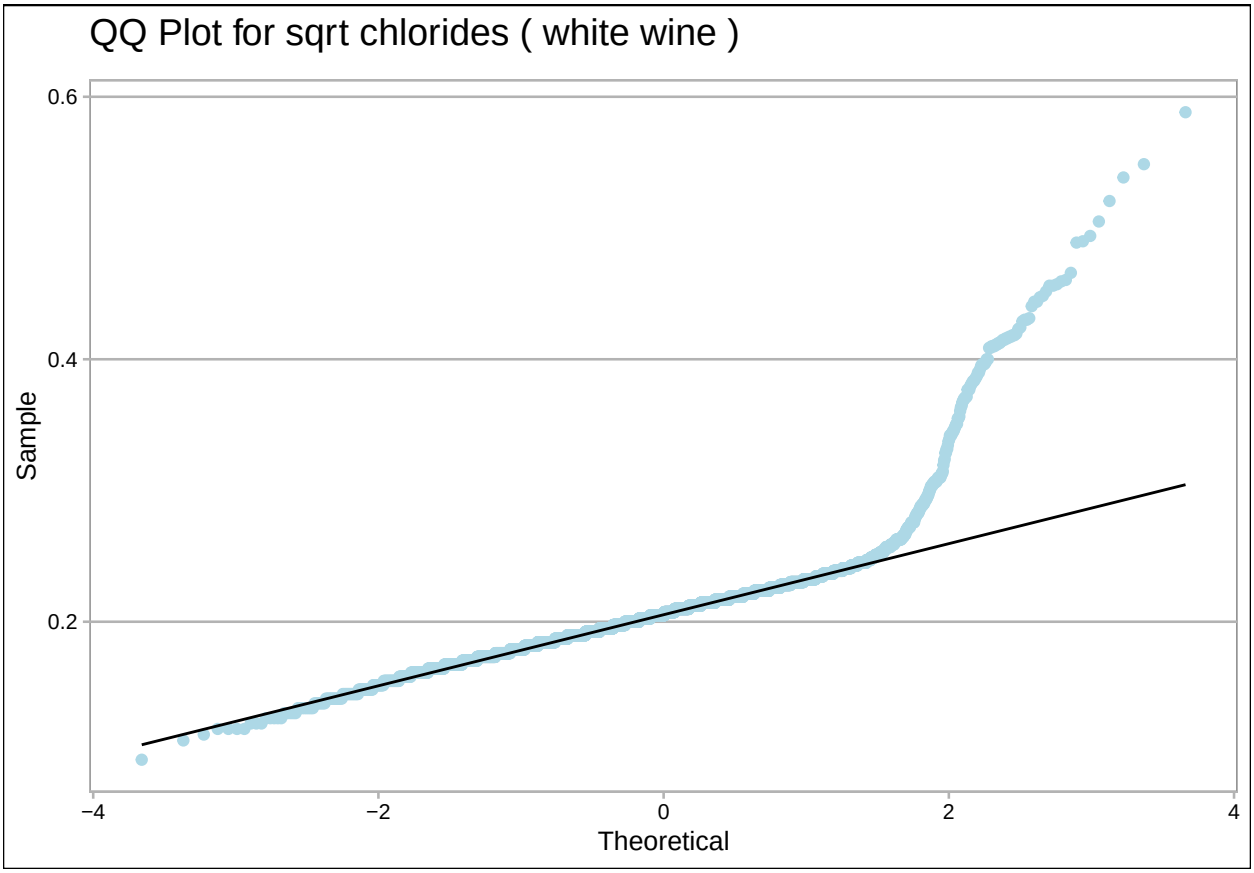
```
# Plotting the sqrt transformed QQ plots for all the variables in White wine
for (i in names(white_wine_continuous)) {
  sqrt_qqplot(white_wine, i, 'lightblue', 'white wine')
}
```

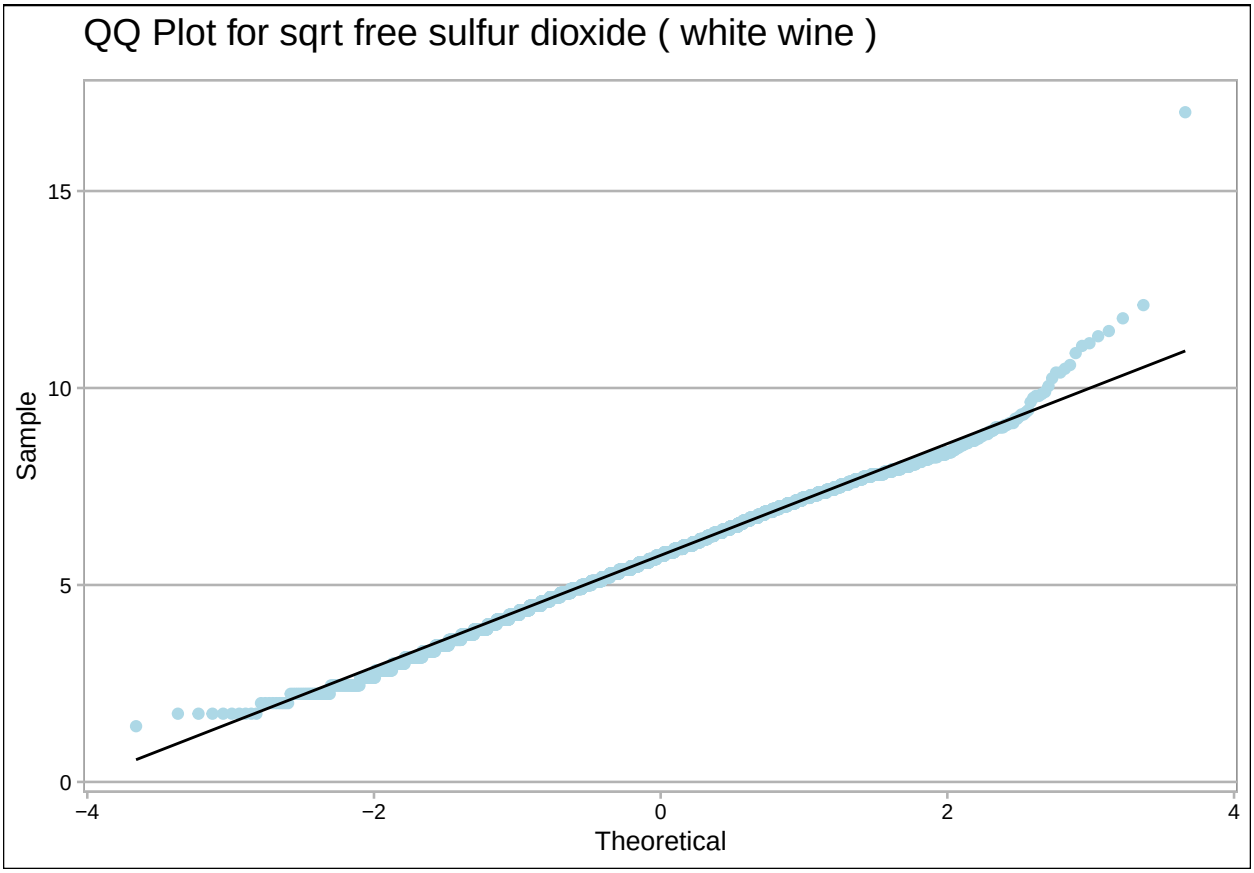

QQ Plot for sqrt fixed acidity (white wine)
Sample
3.5
3.0
2.5
2.0
−4
−2
0
2
4
Theoretical

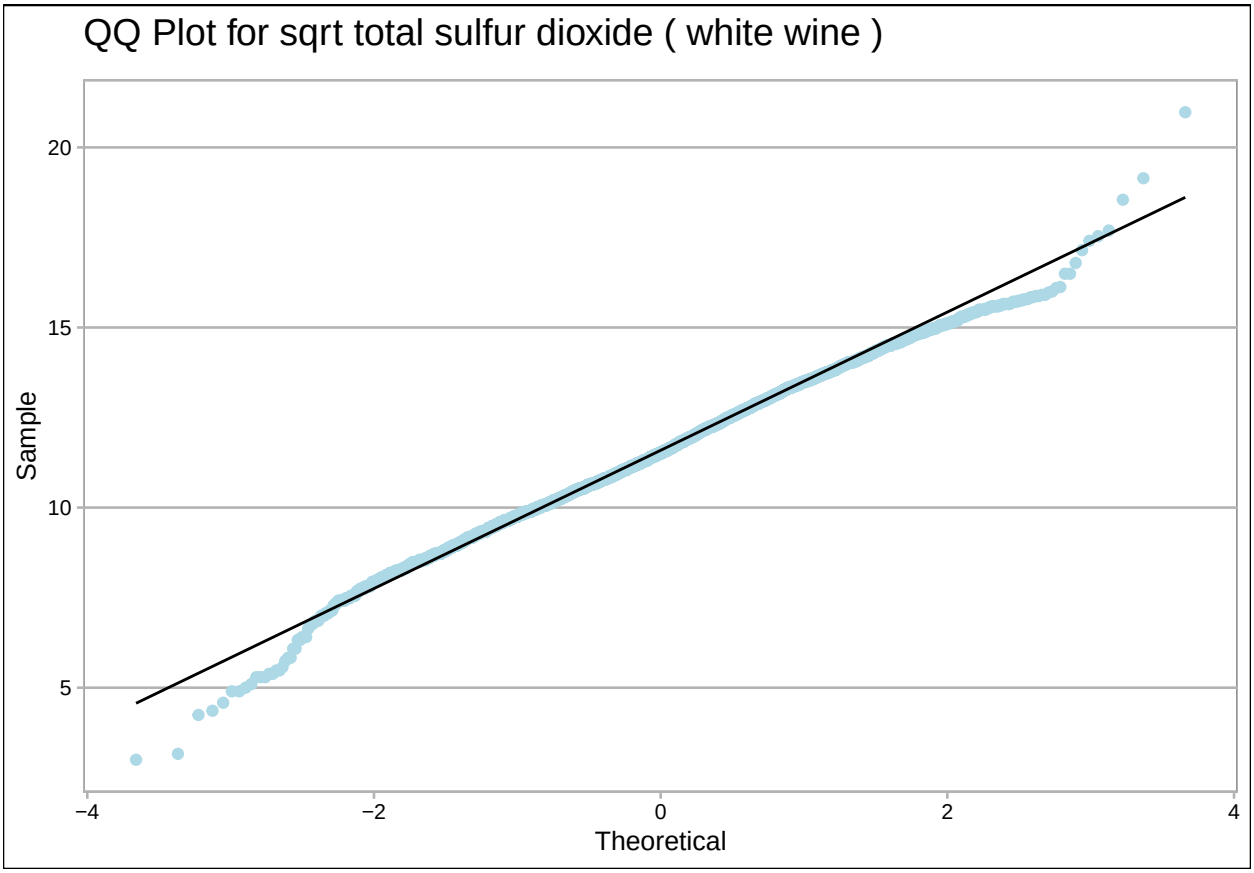

QQ Plot for sqrt volatile acidity (white wine)
Sample
1.0
0.8
0.6
0.4
0.2
−4
−2
0
2
4
Theoretical

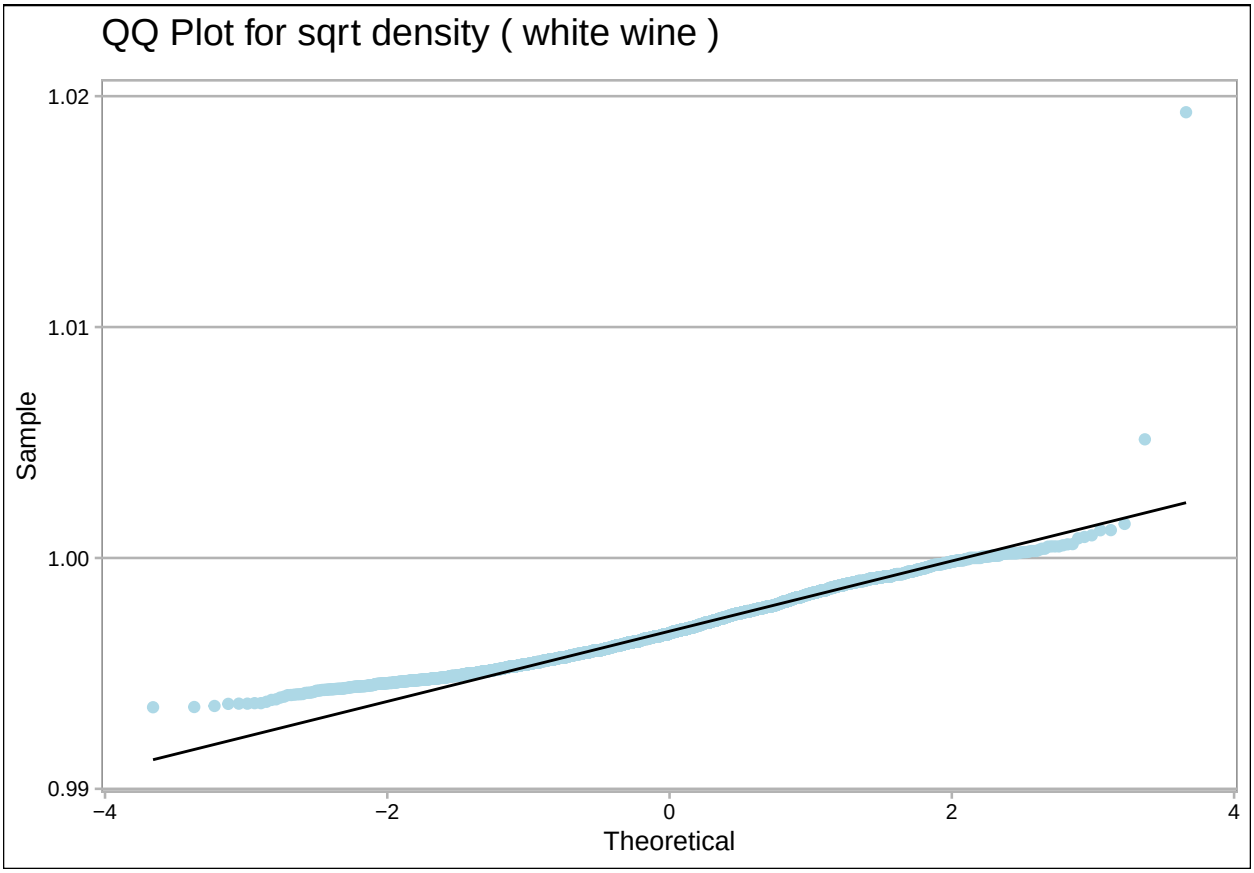

QQ Plot for sqrt citric acid (white wine)
Sample
1.0
0.5
0.0
−4
−2
0
2
4
Theoretical

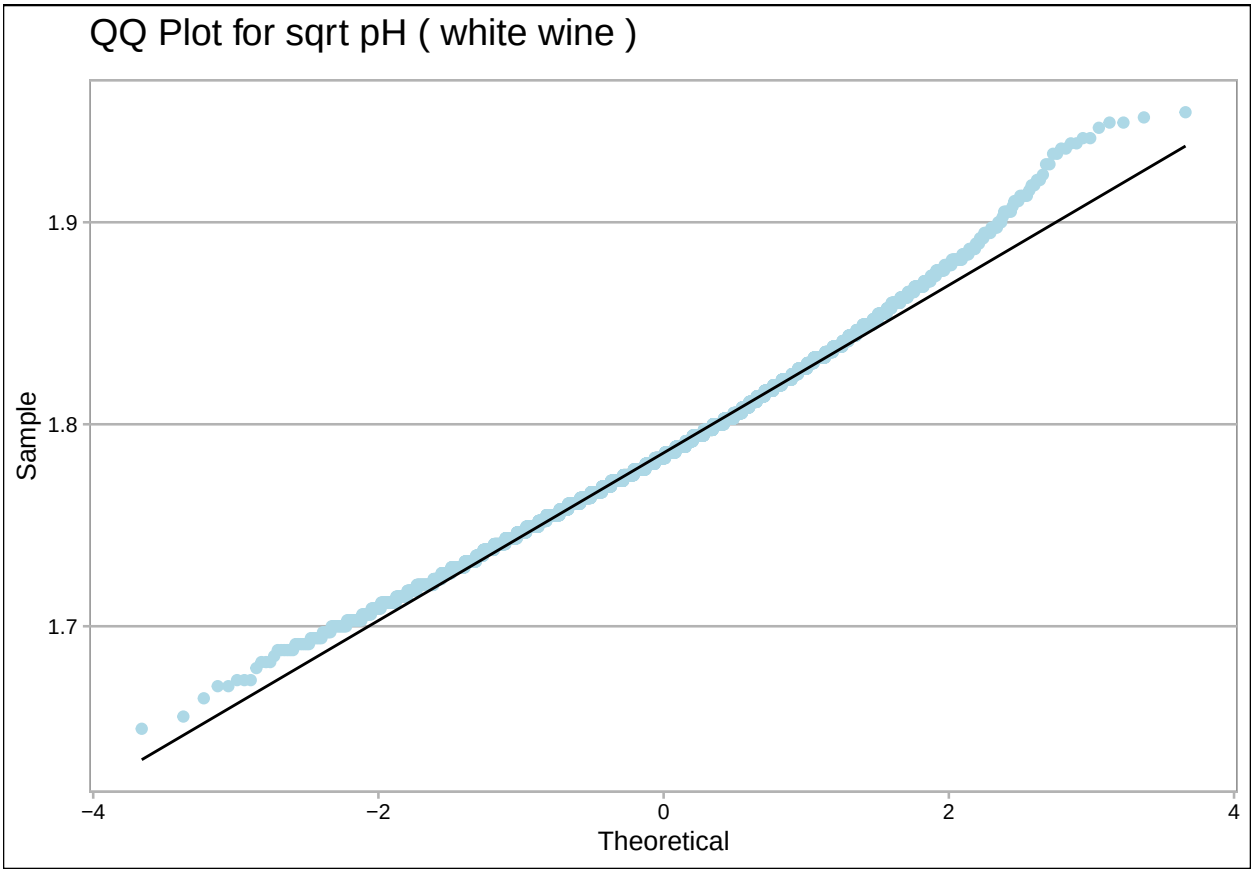

QQ Plot for sqrt residual sugar (white wine)
Sample
6
3
0
−3
−4
−2
0
2
4
Theoretical

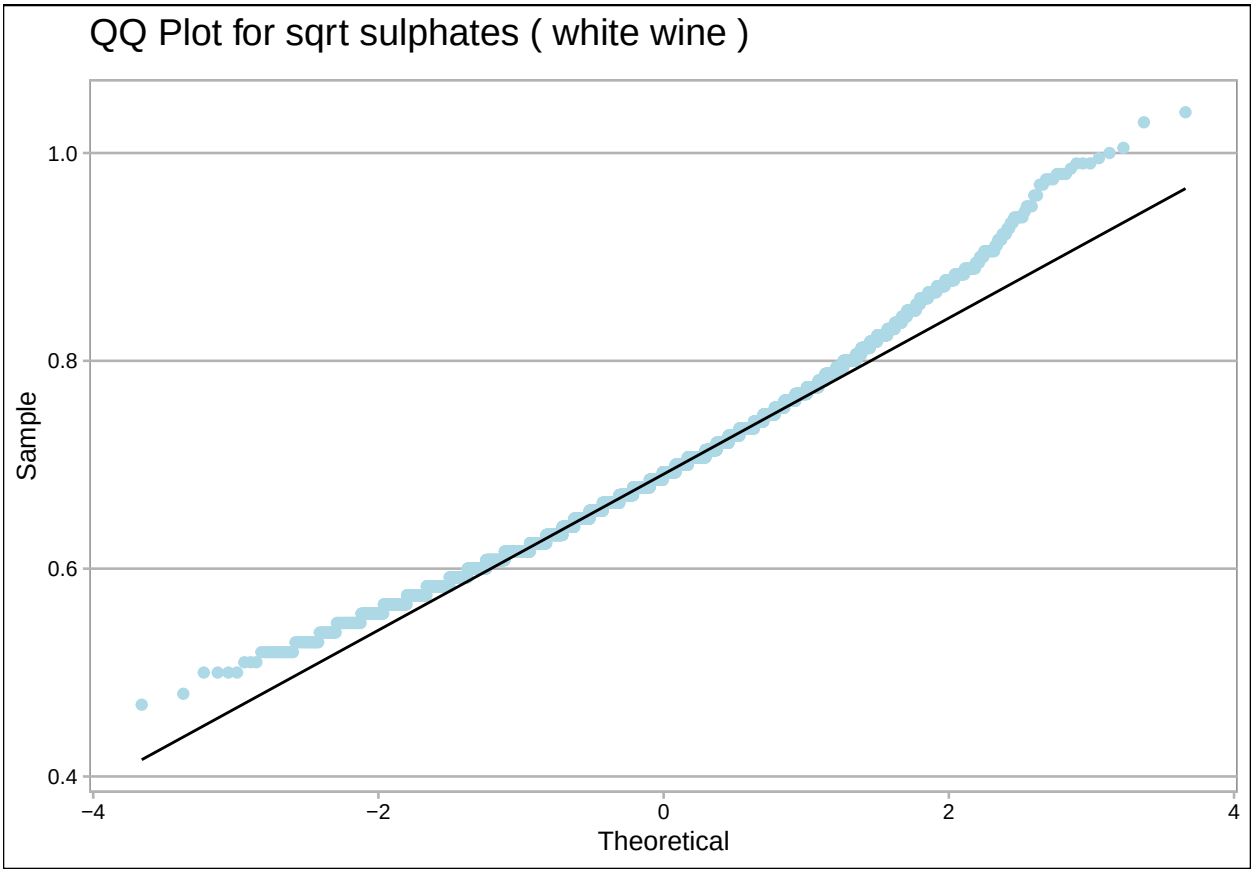

QQ Plot for sqrt chlorides (white wine)
0.6
0.4
0.2
Sample
−4
−2
0
2
4
Theoretical

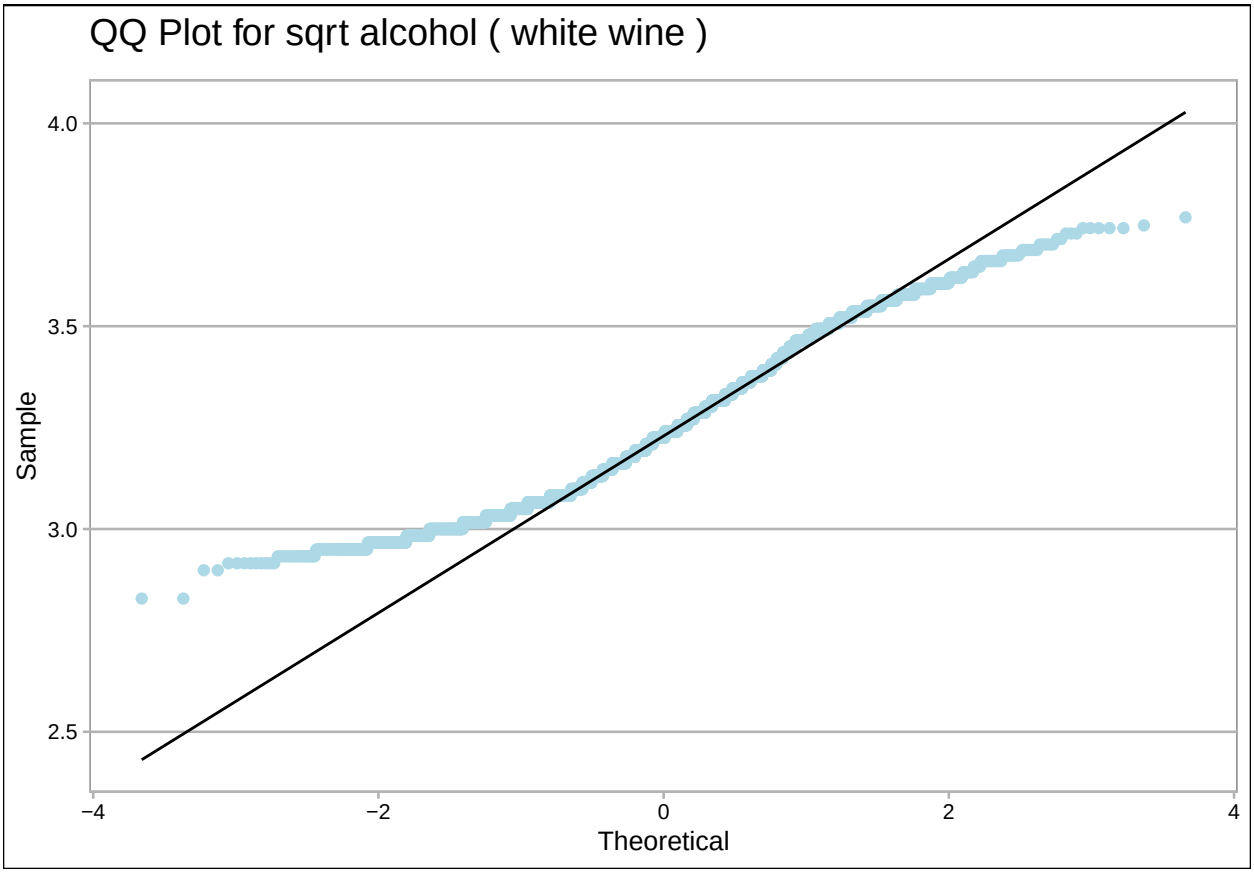

QQ Plot for sqrt free sulfur dioxide (white wine)
Sample
15
10
5
0
−4
−2
0
2
4
Theoretical


QQ Plot for sqrt total sulfur dioxide (white wine)
Sample
20
15
10
5
−4
−2
0
2
4
Theoretical


QQ Plot for sqrt density (white wine)
1.02
1.01
1.00
0.99
Sample
−4
−2
0
2
4
Theoretical

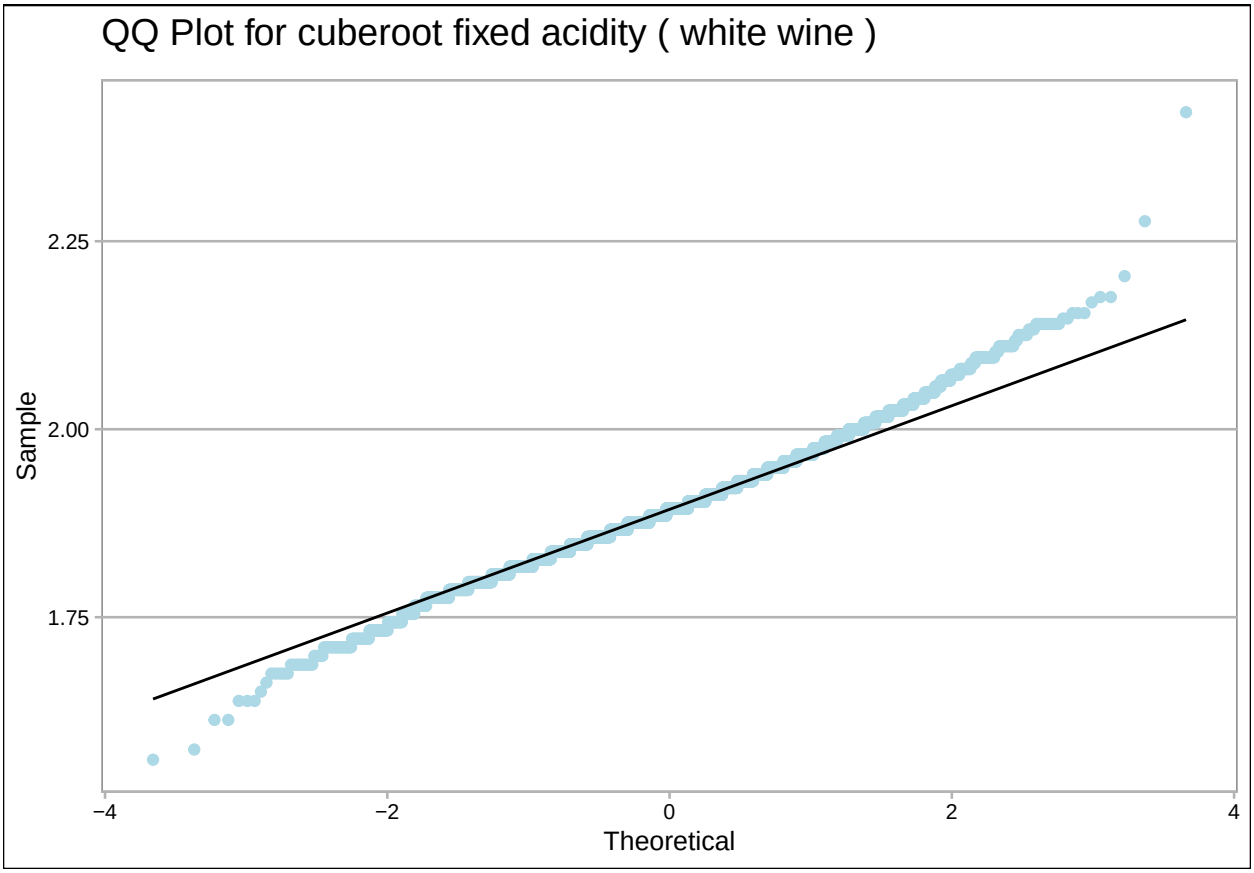

QQ Plot for sqrt pH (white wine)
Sample
1.9
1.8
1.7
−4
−2
0
2
4
Theoretical

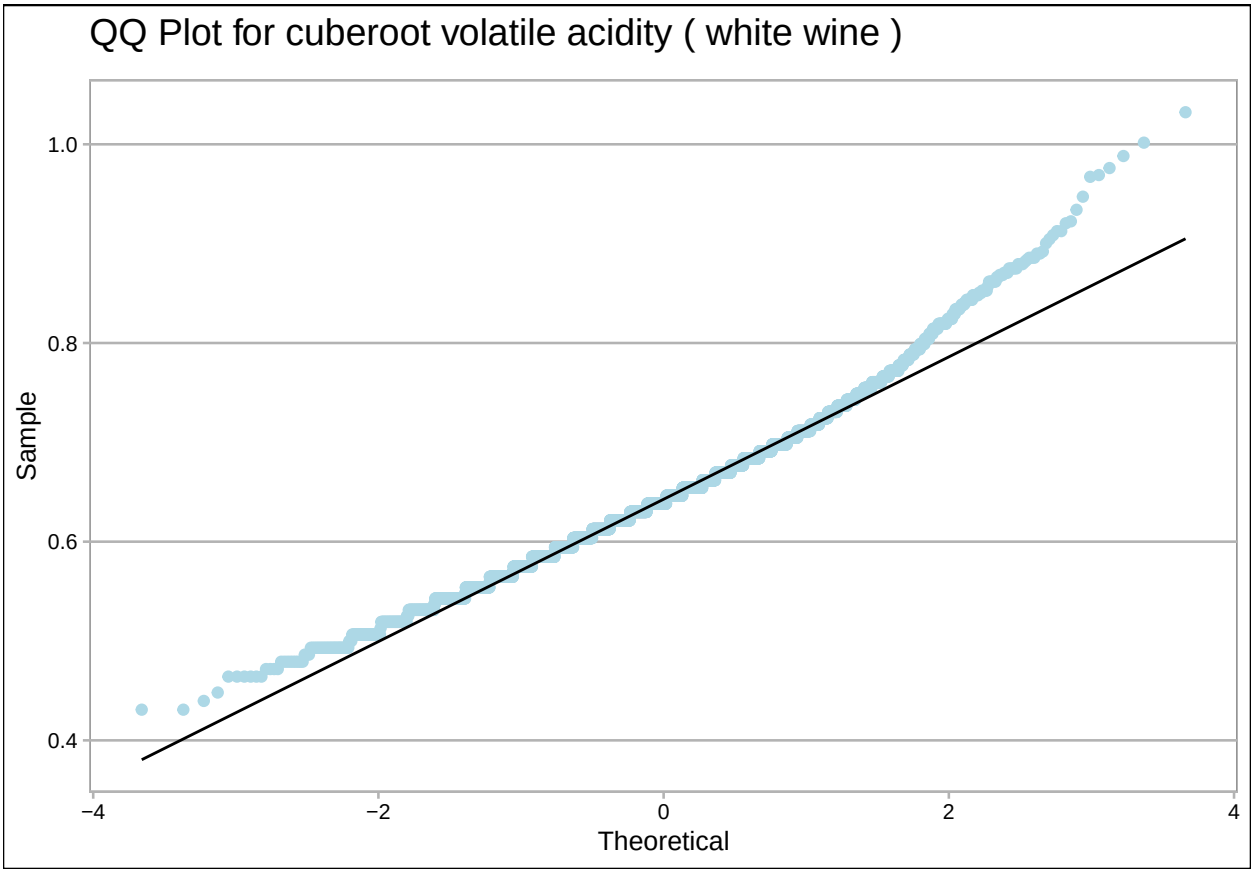

QQ Plot for sqrt sulphates (white wine)
Sample
1.0
0.8
0.6
0.4
−4
−2
0
2
4
Theoretical

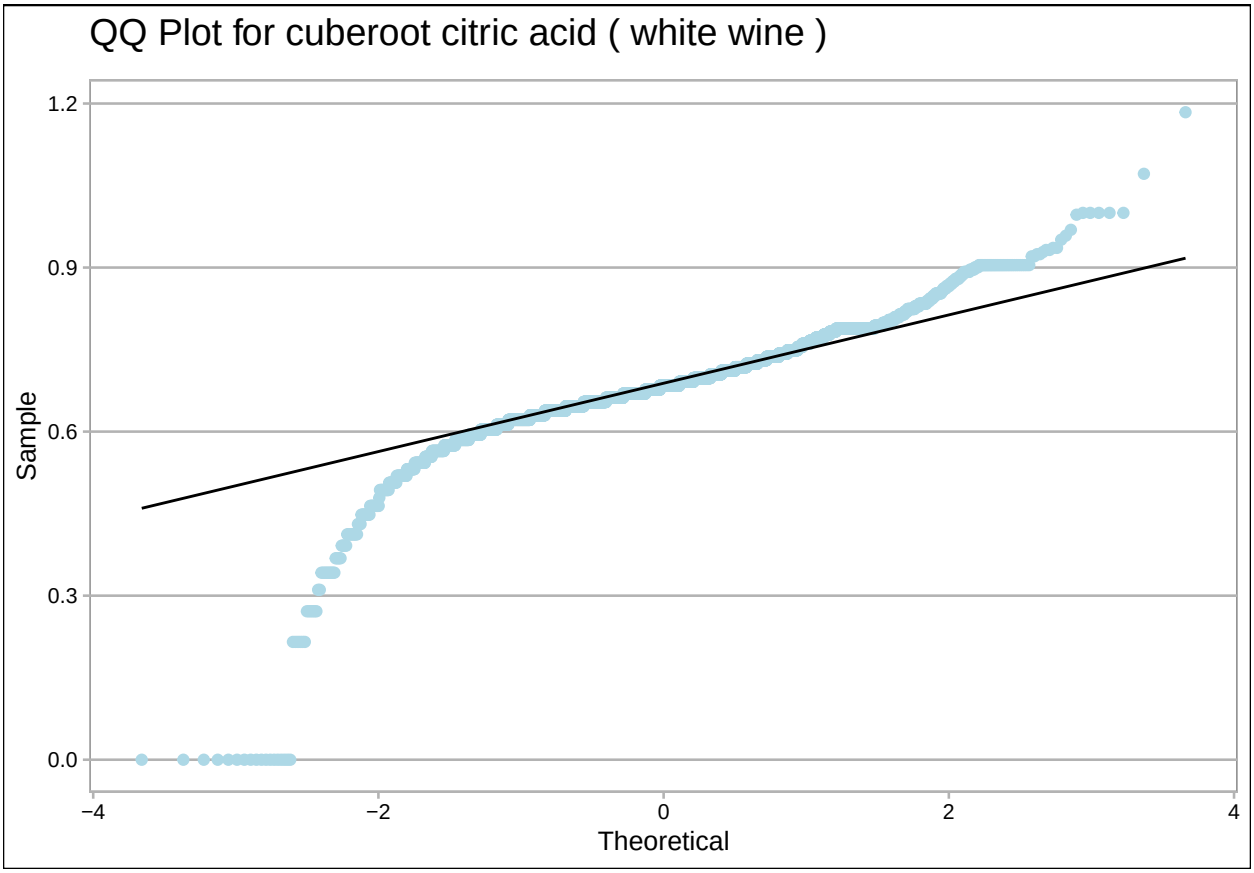


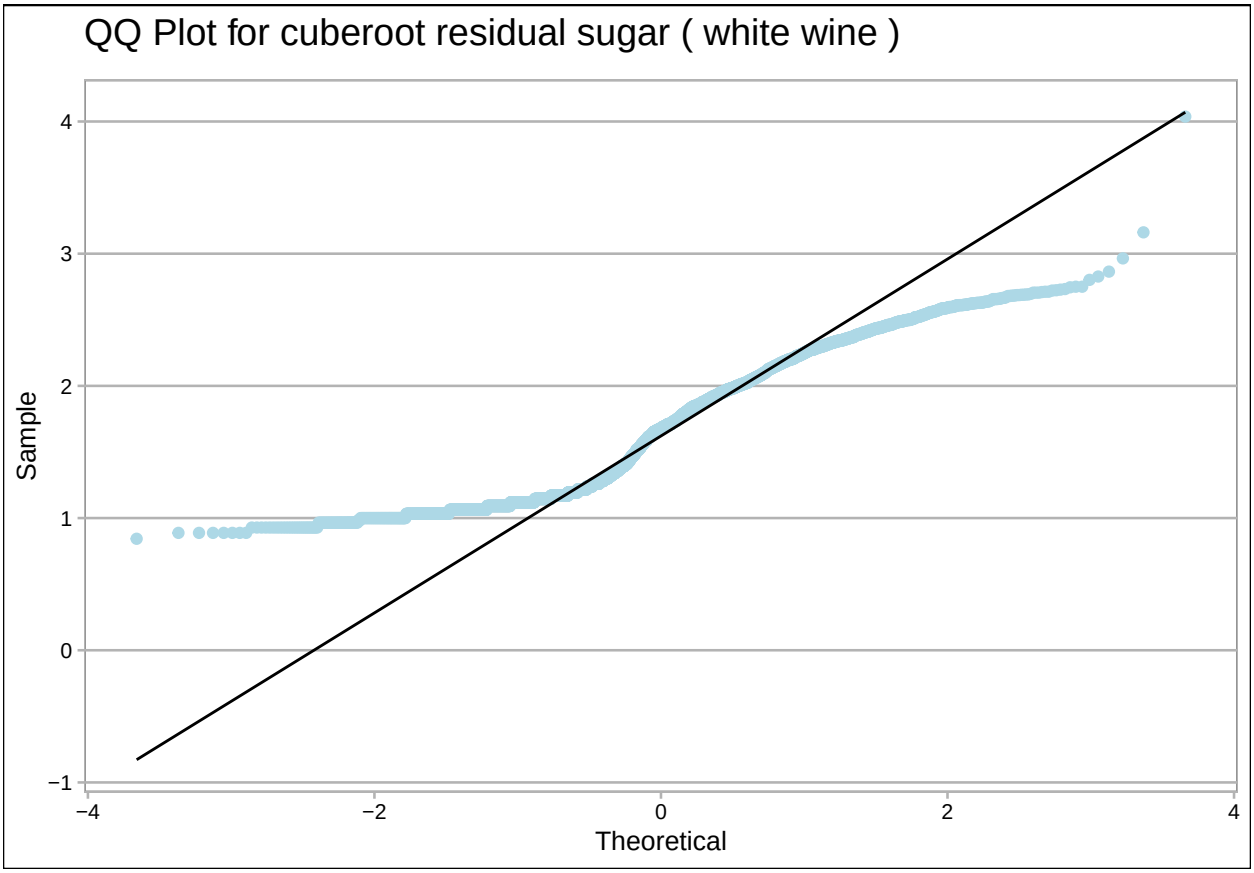
CUBEROOT TRANSFORMED QQPLOT FOR WHITE WINE

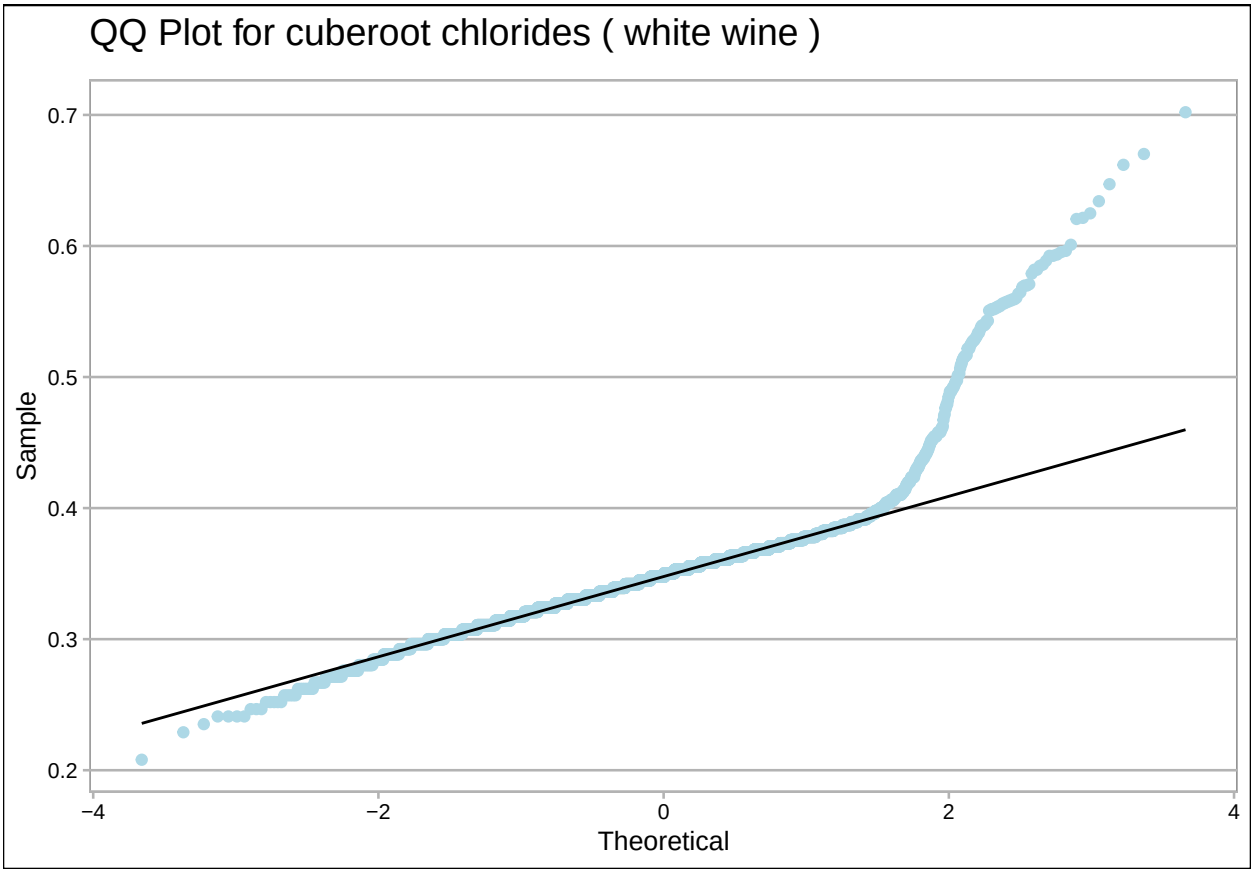
```
# Plotting the cuberoot transformed QQ plots for all the variables in White wine
for (i in names(white_wine_continuous)) {
  cuberoot_qqplot(white_wine, i, 'lightblue', 'white wine')
}
```

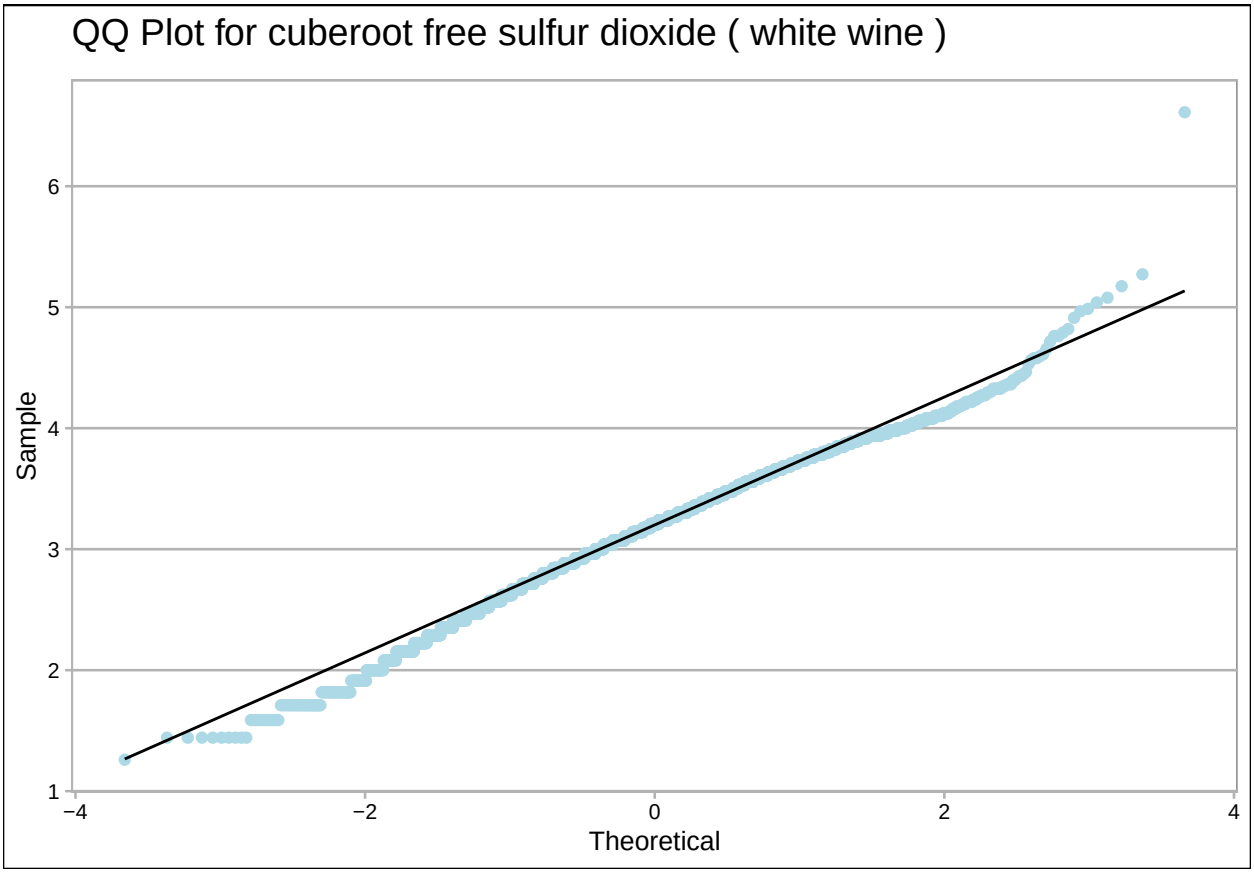

QQ Plot for cuberoot fixed acidity (white wine)
Sample
2.25
2.00
1.75
−4
−2
0
2
4
Theoretical

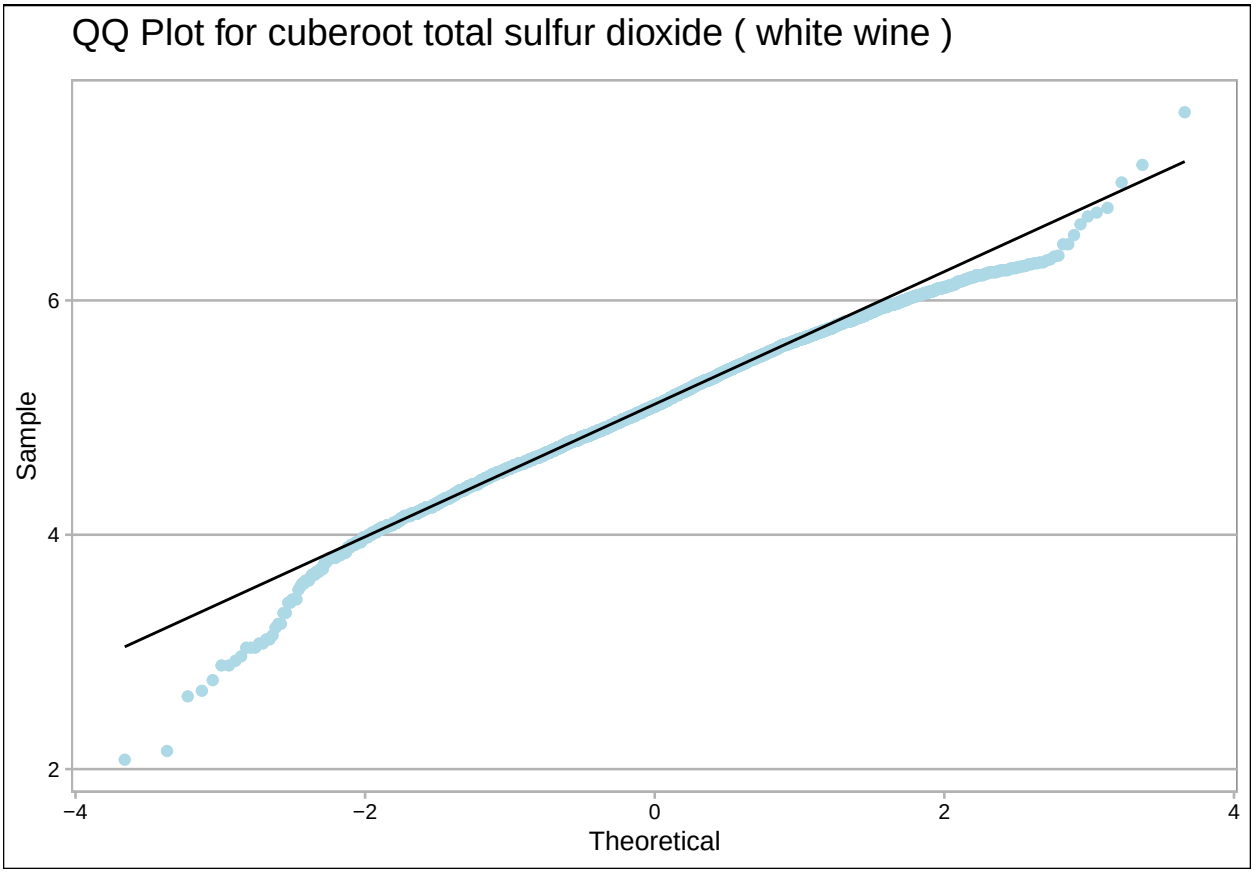

QQ Plot for cuberoot volatile acidity (white wine)
Sample
1.0
0.8
0.6
0.4
−4
−2
0
2
4
Theoretical

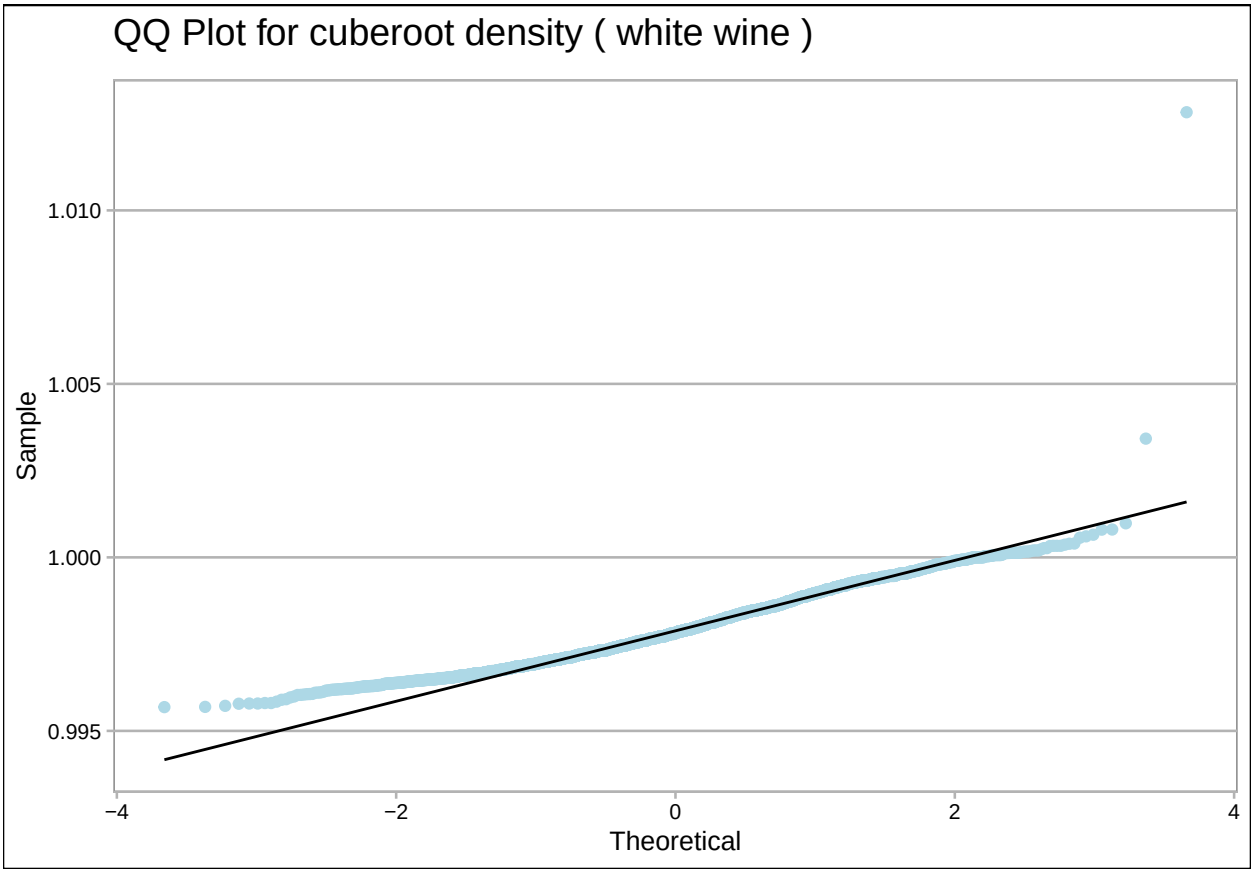

QQ Plot for cuberoot citric acid (white wine)
1.2
0.9
0.6
0.3
0.0
Sample
−4
−2
0
2
4
Theoretical

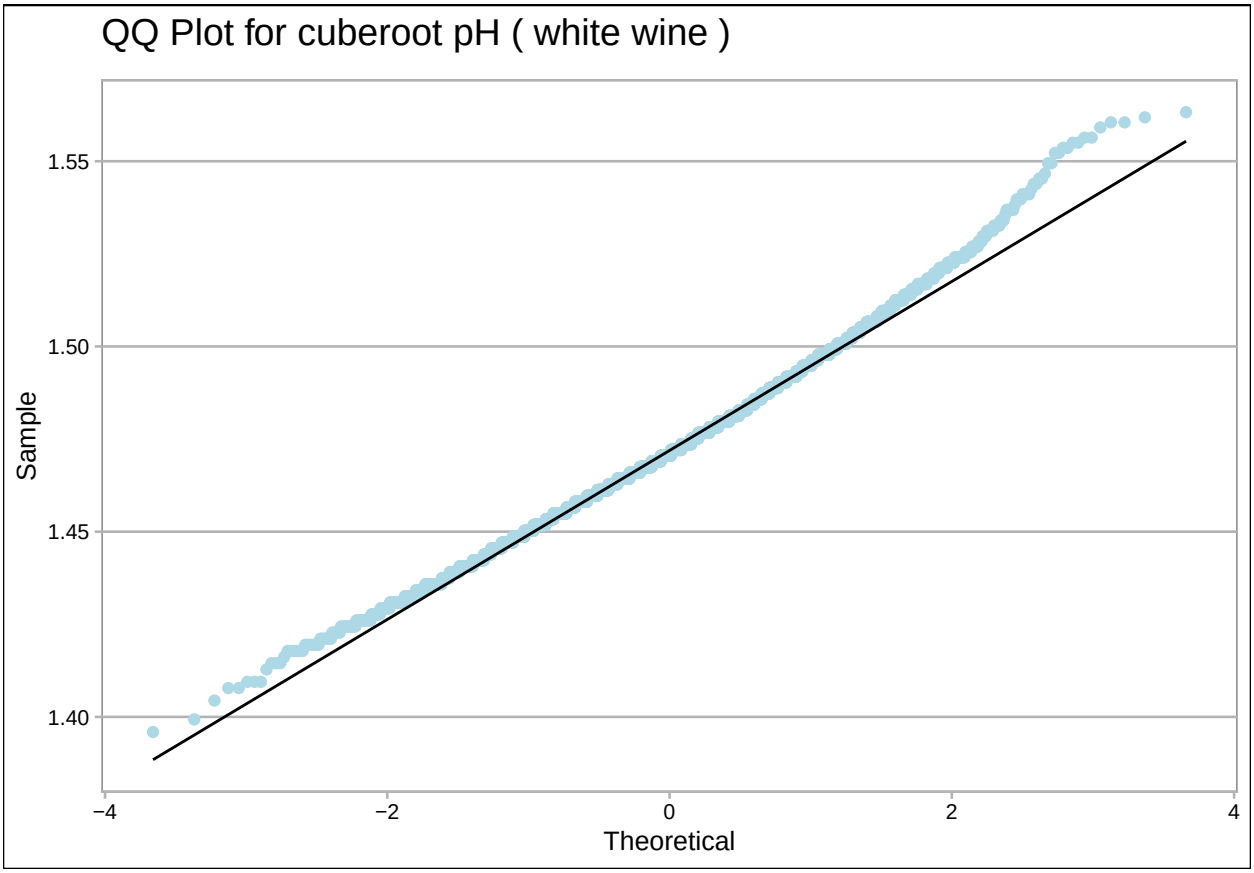

QQ Plot for cuberoot residual sugar (white wine)
Sample
Theoretical
4
3
2
1
0
−1
−4
−2
0
2
4

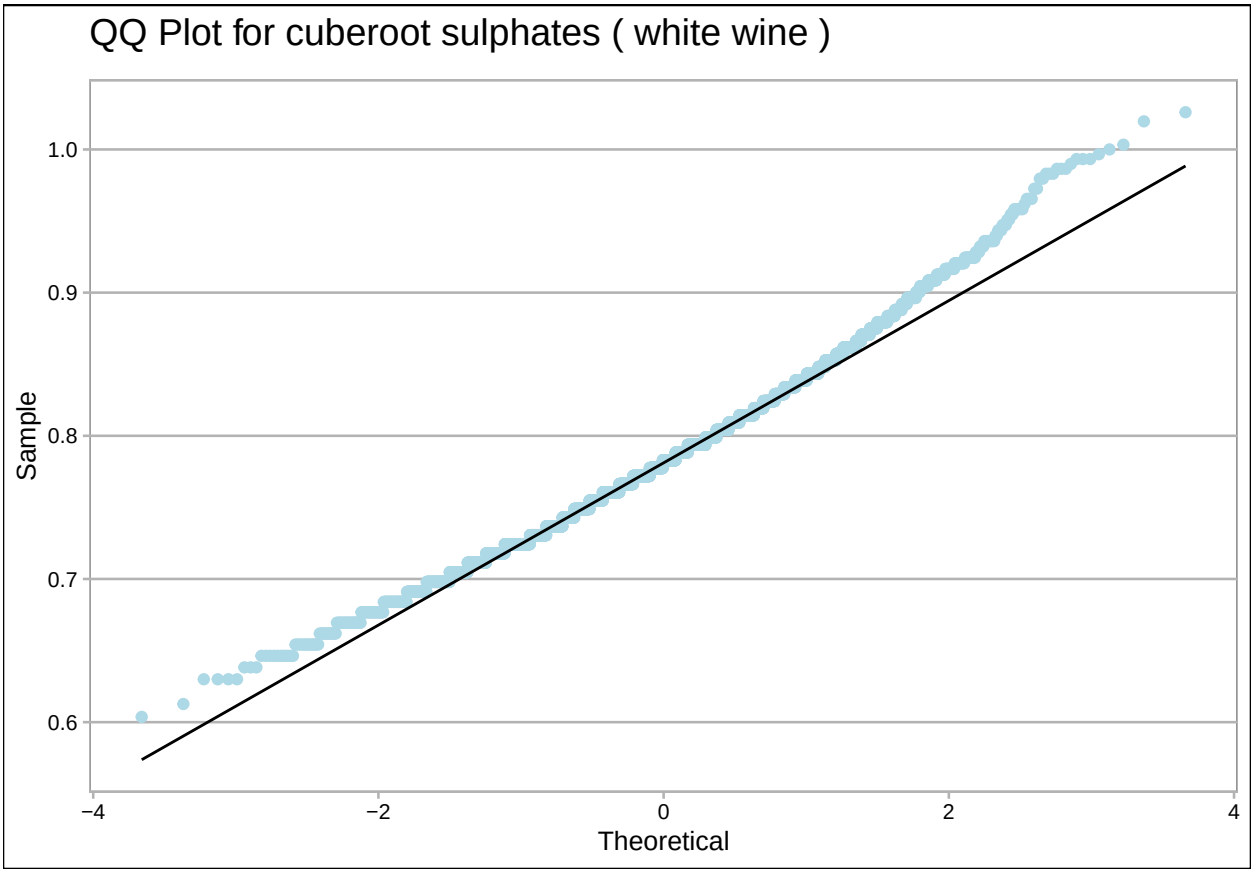

QQ Plot for cuberoot chlorides (white wine)
Sample
Theoretical
0.7
0.6
0.5
0.4
0.3
0.2
−4
−2
0
2
4

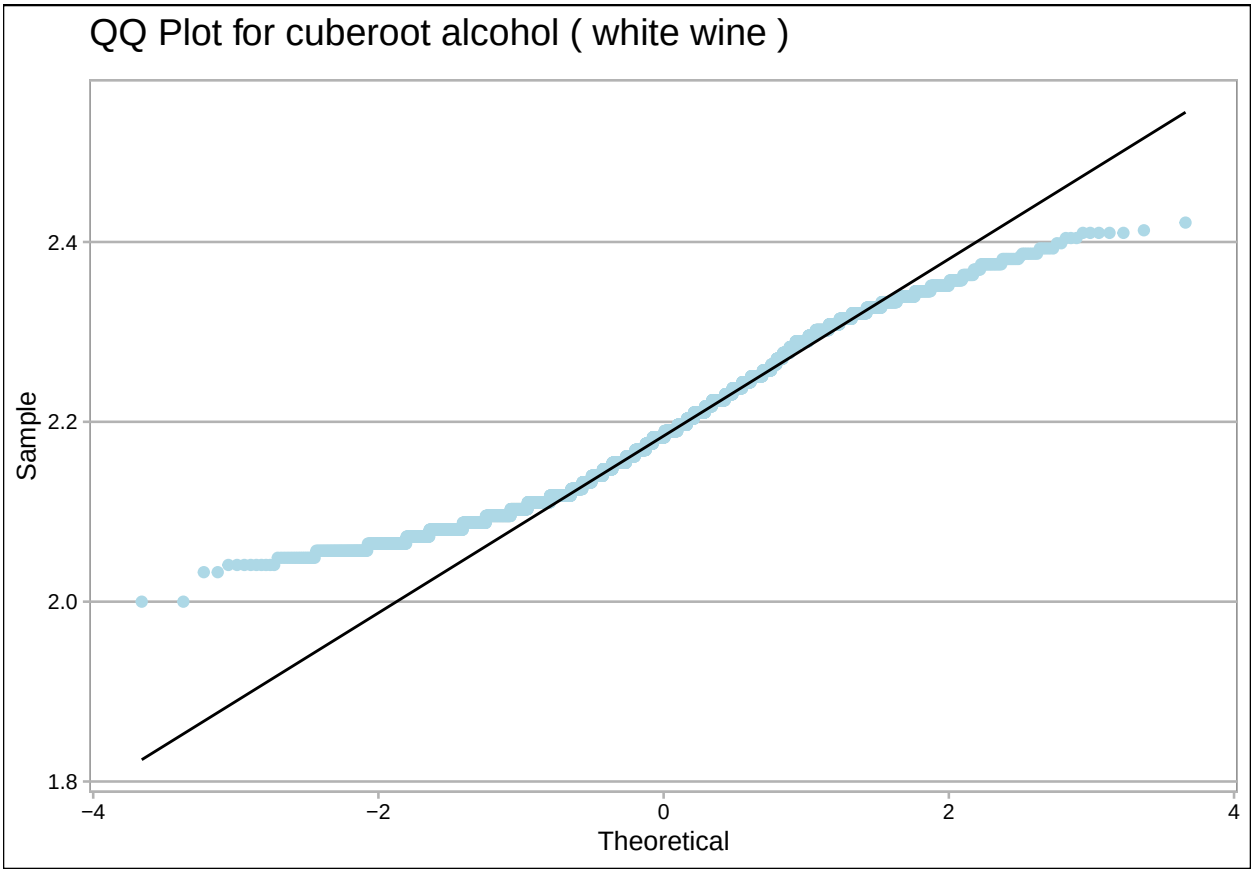

QQ Plot for cuberoot free sulfur dioxide (white wine)
Sample
Theoretical
1
2
3
4
5
6
−4
−2
0
2
4


QQ Plot for cuberoot total sulfur dioxide (white wine)
Sample
6
4
2
−4
−2
0
2
4
Theoretical


QQ Plot for cuberoot density (white wine)
Sample
1.010
1.005
1.000
0.995
−4
−2
0
2
4
Theoretical


QQ Plot for cuberoot pH (white wine)
Sample
1.55
1.50
1.45
1.40
−4
−2
0
2
4
Theoretical


QQ Plot for cuberoot sulphates (white wine)
Sample
1.0
0.9
0.8
0.7
0.6
−4
−2
0
2
4
Theoretical



Hypotheses Testing 1

Tests if there is an association between the wine type and quality

Since wine_type is nominal categorical and quality is ordinal

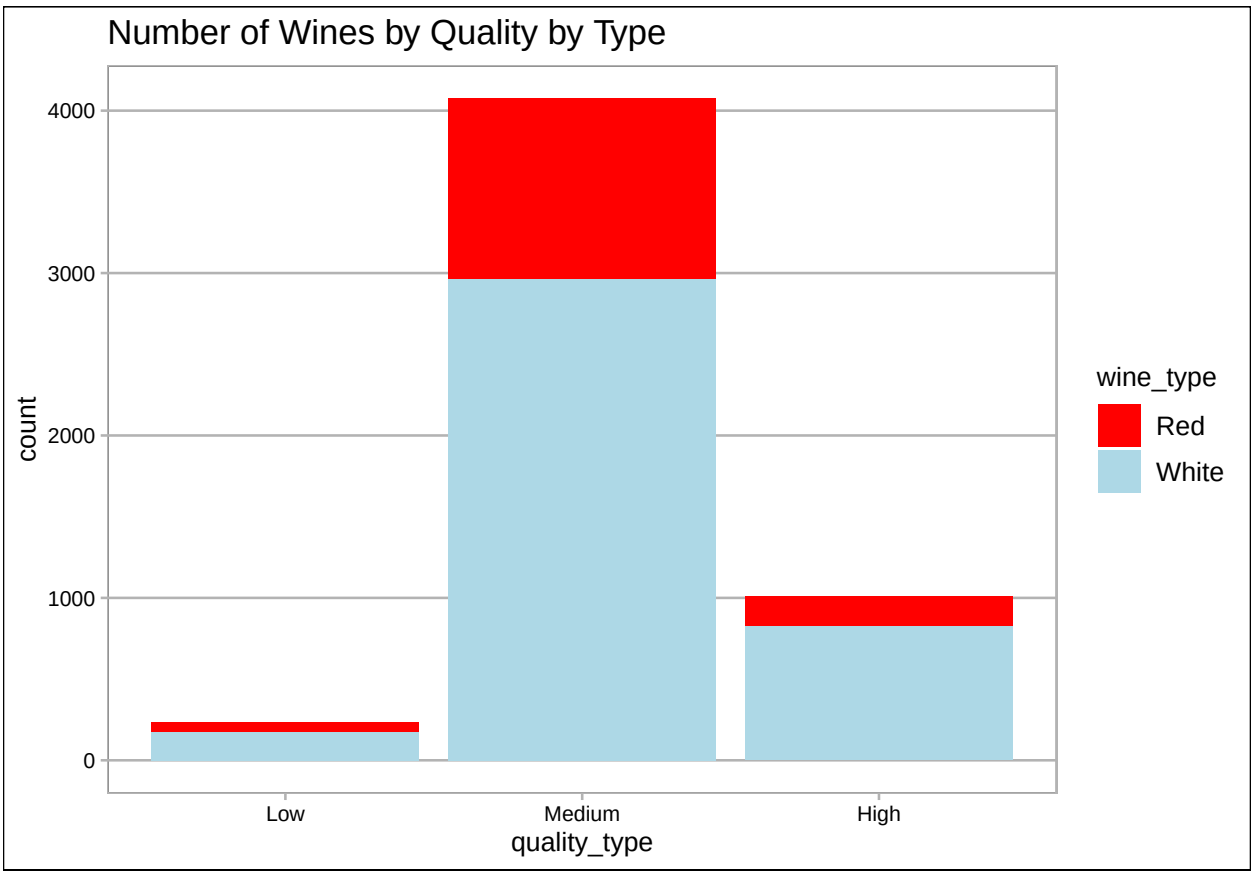
Chi-Square Test of Independence

Hypotheses: Null Hypothesis (H):There is no association between wine type and quality Alternative Hypothesis (H): Wine type and wine quality are associated

```
# Creating quality type to reduce the perform Chi-squared test
wine_data_copy <- wine_data_copy %>%
  mutate(quality_type = case_when(
    quality %in% c(3, 4) ~ "Low",
    quality %in% c(5,6) ~ "Medium",
    quality %in% c(7,8, 9) ~ "High"
  ))
wine_data_copy$quality_type  <- factor(wine_data_copy$quality_type ,
                                       levels = c("Low", "Medium", "High"),
                                       ordered = TRUE)
table(wine_data_copy$wine_type,wine_data_copy$quality_type)
```

```
##
##          Low Medium High
##   Red     63   1112  184
##   White  173   2963  825
```

```r
# Plotting number of wines by Quality and type
ggplot(wine_data_copy,aes(x = quality_type, fill = wine_type)) +
  geom_bar() +
  labs(title = "Number of Wines by Quality by Type") +
  scale_fill_manual(values = c("Red" = "red", "White" = "lightblue")) +  theme_calc()
```




```r
# Chi-squared test of independence.
chisq_test <- chisq.test(wine_data_copy$wine_type, wine_data_copy$quality_type)
chisq_test
```

```
##
##  Pearson's Chi-squared test
##
## data:  wine_data_copy$wine_type and wine_data_copy$quality_type
## X-squared = 35.017, df = 2, p-value = 2.49e-08
```

```r
chisq_test$residuals
```

```
##                         wine_data_copy$quality_type
## wine_data_copy$wine_type         Low     Medium       High
##                    Red    0.3494824  2.2017336 -4.5937106
##                    White -0.2047070 -1.2896507  2.6907351
```

Since the p-value is much less than 0.05, we reject the null hypothesis. This means there is a statistically significant association between wine_type and quality_type.

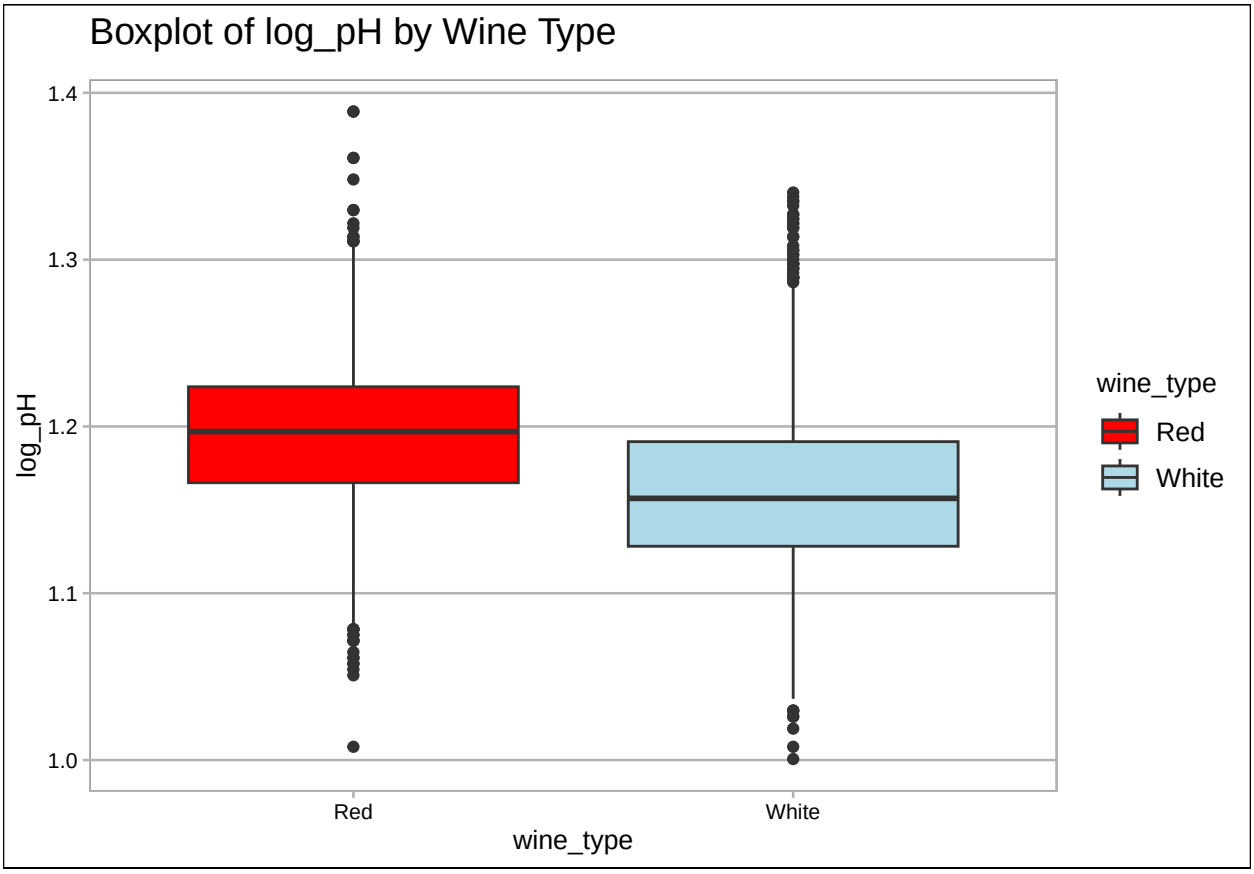
High-quality wines: Fewer for red wine, more for white wine. Medium-quality wines: Slightly more for red wine, slightly fewer for white wine.

HYPOTHESIS TEST 2.

Testing pH vs Wine Type (Two Sample T-Test)

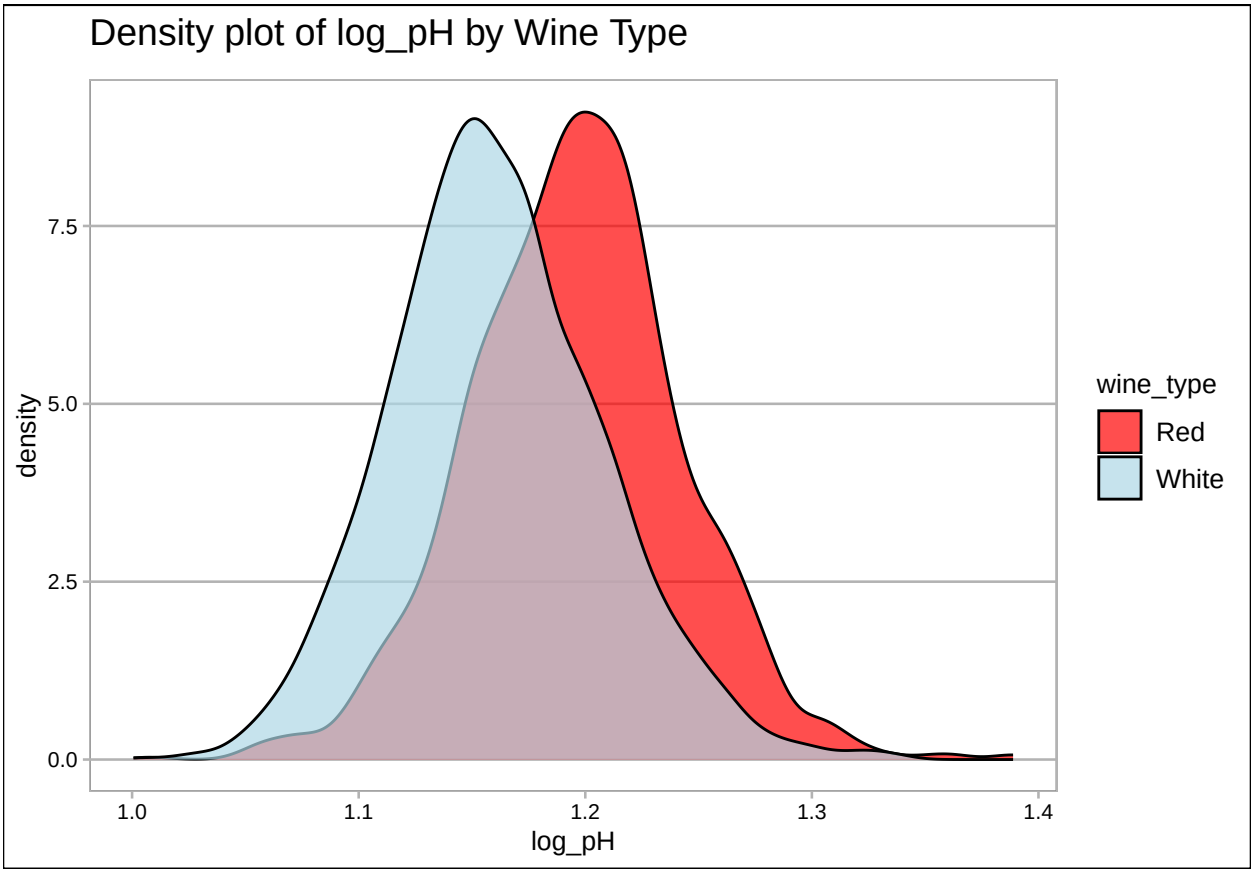
Null Hypothesis (H) : There is no significant difference in the mean pH between the two wine types. Alternative Hypothesis (H):There is significant difference in the mean pH between the two wine types

```
# Visualizing the distribution of log_pH based on wine type.
wine_data %>%
  mutate(log_pH = log(pH)) %>%
  ggplot(aes(x = wine_type, y = log_pH, fill = wine_type)) +
    geom_boxplot() +
    labs(title = paste("Boxplot of log_pH by Wine Type")) +
    scale_fill_manual(values = c("Red" = "red", "White" = "lightblue")) +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```




```
wine_data %>%
  mutate(log_pH = log(pH)) %>%
  ggplot(aes(x = log_pH, fill = wine_type)) +
  geom_density(alpha = 0.7) +
    labs(title = paste("Density plot of log_pH by Wine Type")) +
    scale_fill_manual(values = c("Red" = "red", "White" = "lightblue")) +
```

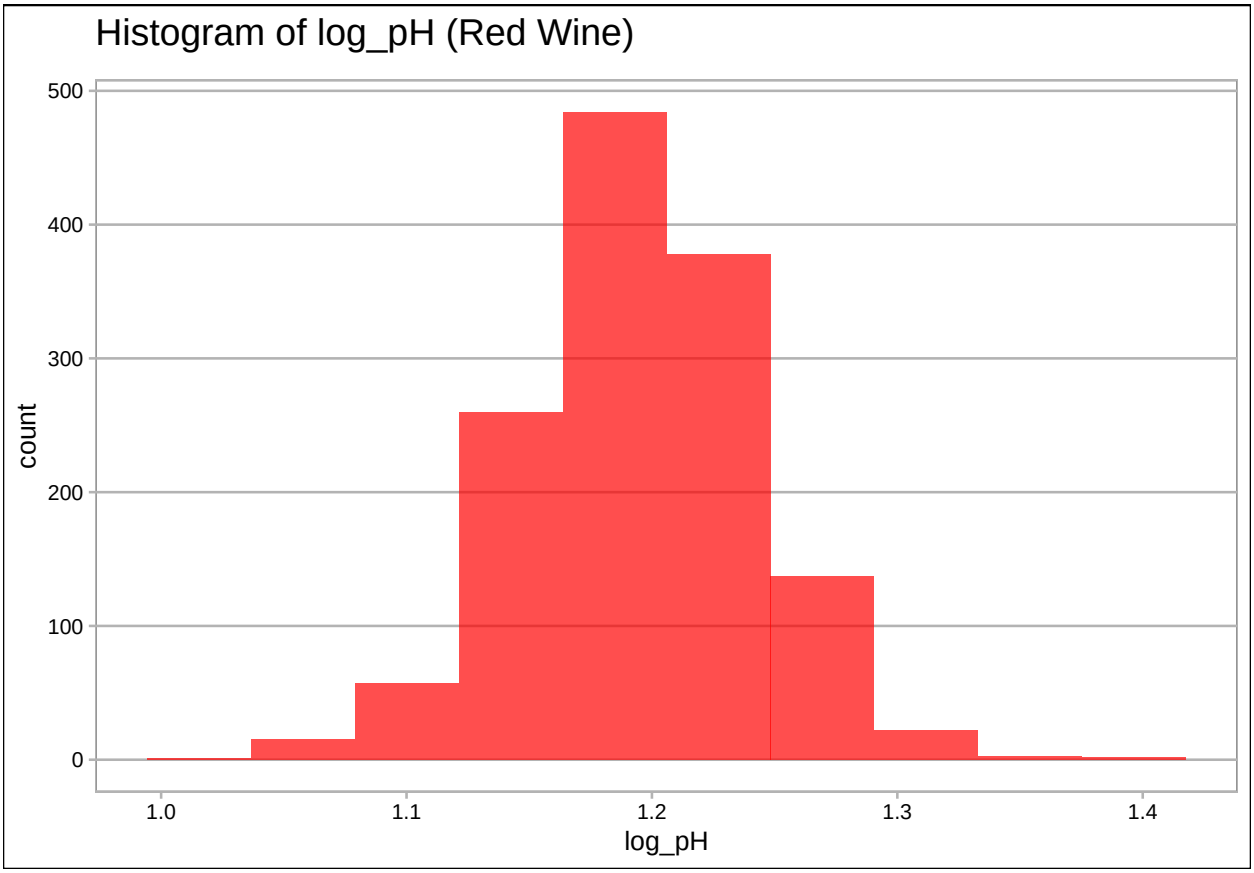
```
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```



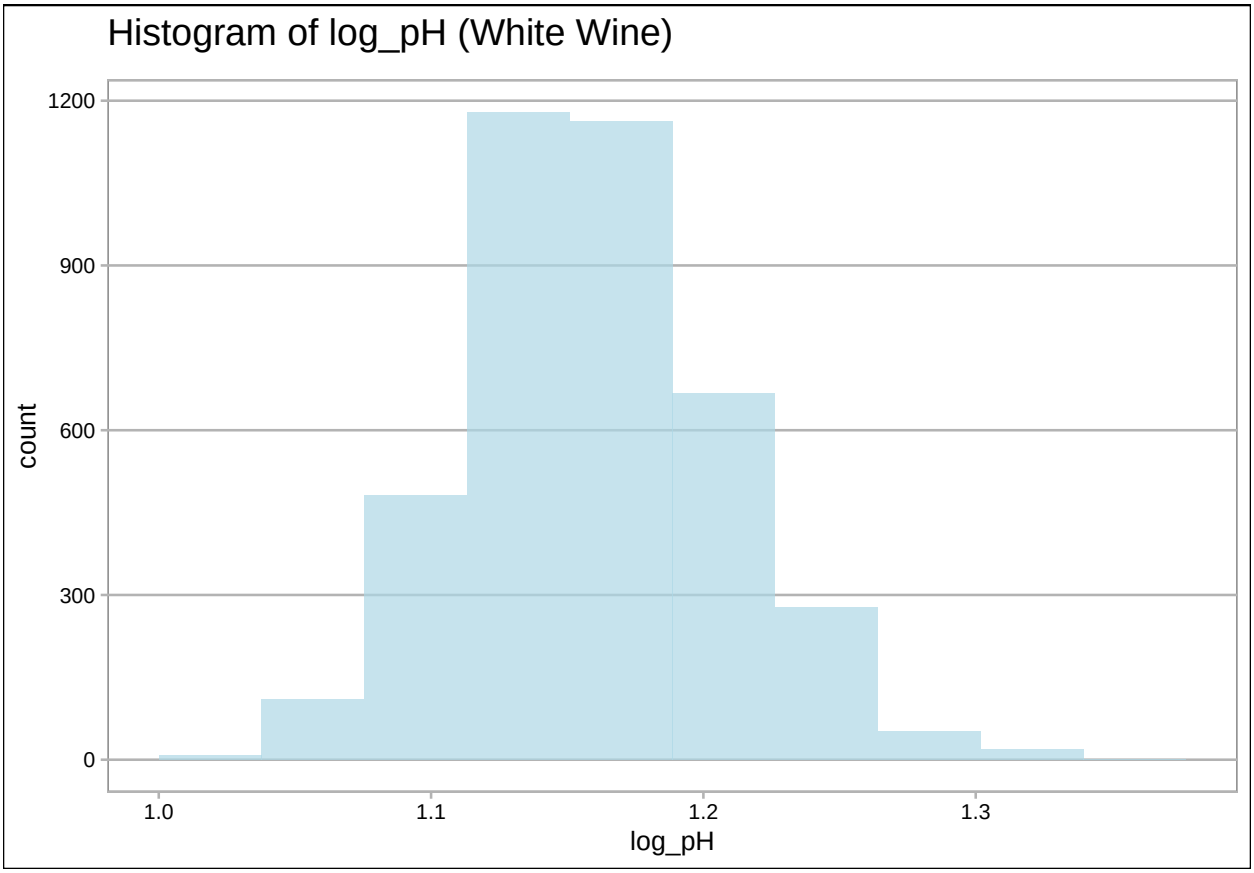
ASSUMPTIONS

Normality check

```
# plotting histograms to check normality distribution of log_pH
red_wine %>%
  mutate(log_pH = log(pH)) %>%
  ggplot(aes(x = log_pH)) +
    geom_histogram(alpha = 0.7, bins =10,fill = "red") +
    labs(title ="Histogram of log_pH (Red Wine)") +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```




```
white_wine %>%
  mutate(log_pH = log(pH)) %>%
  ggplot(aes(x = log_pH)) +
    geom_histogram(alpha = 0.7, bins =10, fill = "lightblue") +
    labs(title ="Histogram of log_pH (White Wine)") +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```

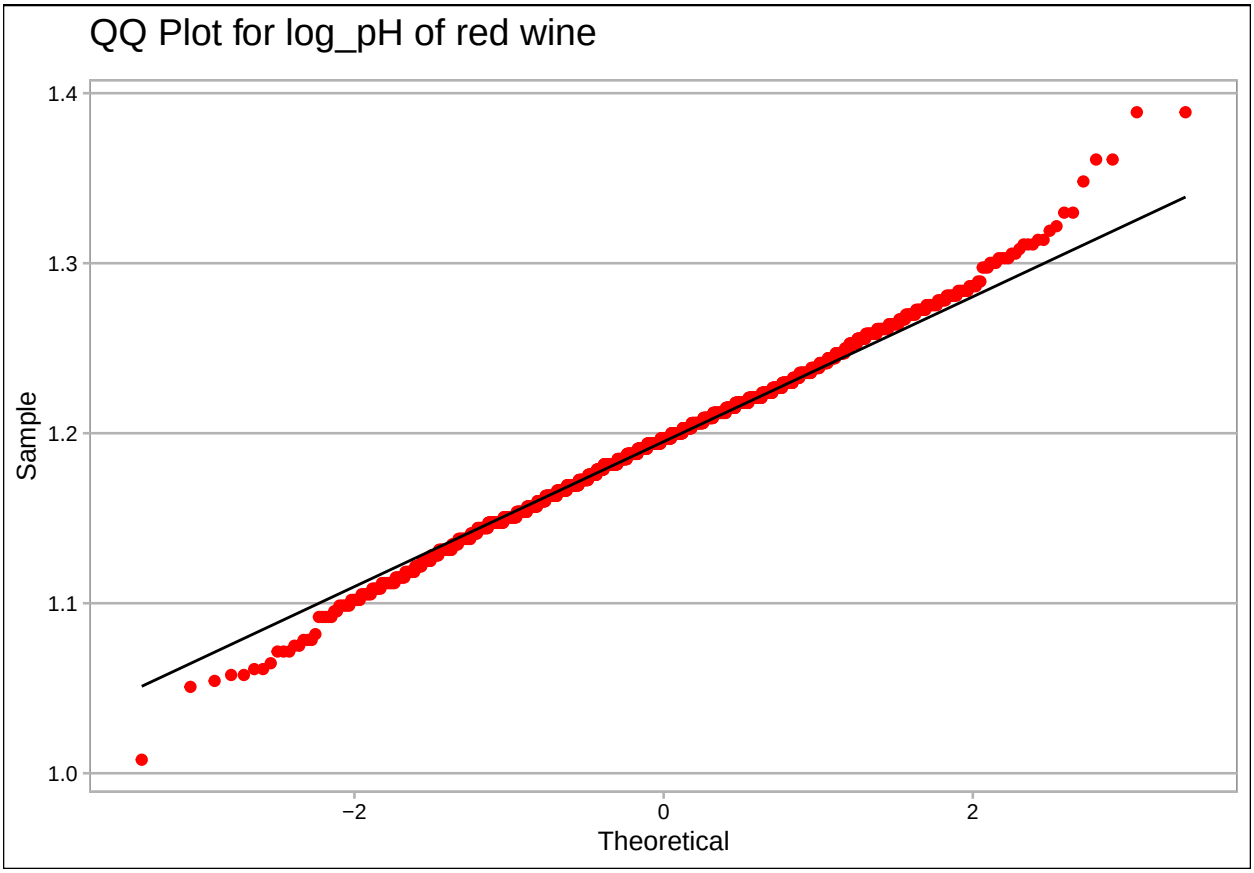



```
# Shapiro-Wilk test for normality
wine_data %>%
  group_by(wine_type) %>%
  mutate(log_pH = log(pH)) %>%
  summarise(p_value = shapiro.test(log_pH)$p.value)
```

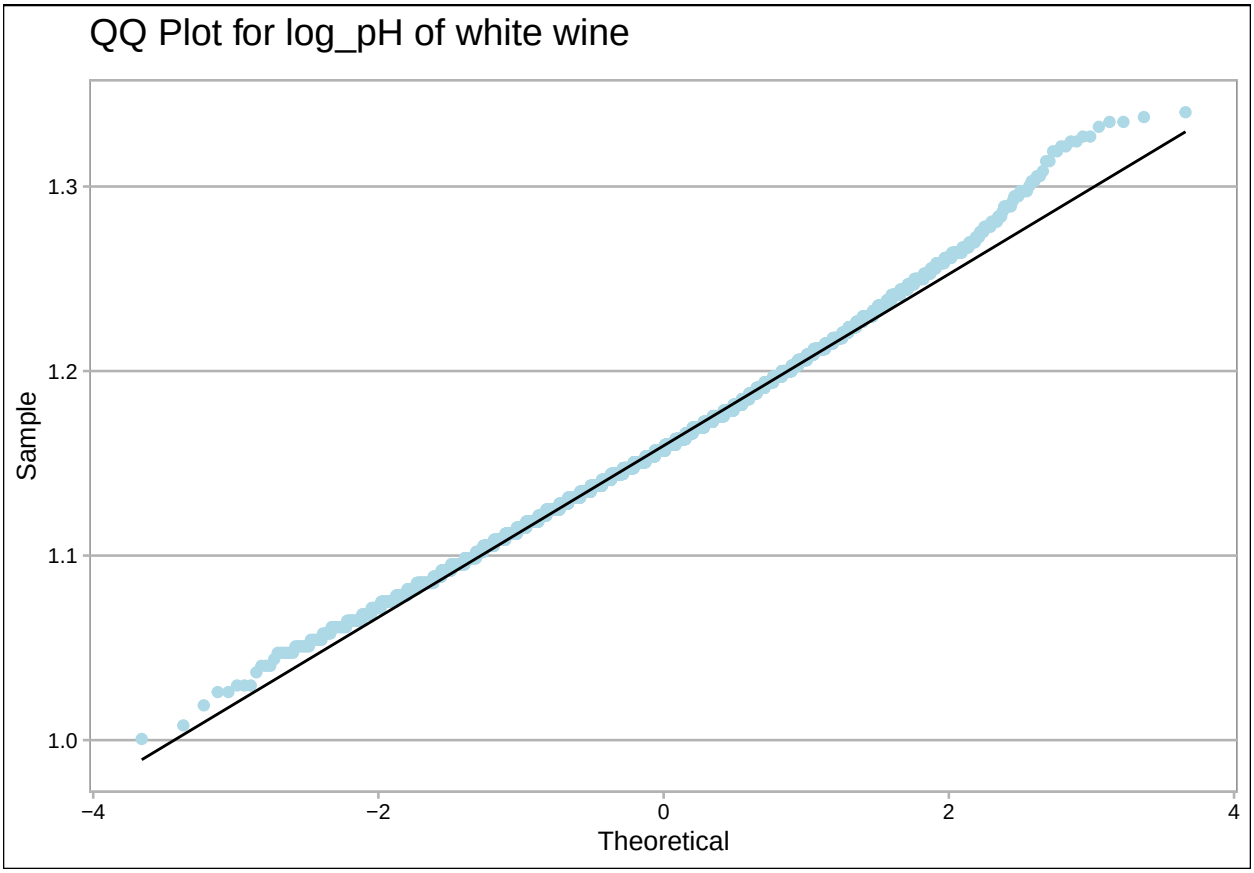
```
## # A tibble: 2 x 2
##   wine_type  p_value
##   <fct>        <dbl>
## 1 Red       2.83e- 4
## 2 White     2.21e-11
```

P<0.05, w ereject null hyposthesis of normality

```
# QQ plot check for normality
red_wine %>%
  mutate(log_pH = log(pH)) %>%
  ggplot(aes(sample = log_pH)) +
    stat_qq(color ='red') +
    stat_qq_line(color = "black") +
    labs(
      title = paste("QQ Plot for log_pH of red wine"),
      x = "Theoretical",
      y = "Sample") +
    theme_calc()+
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```




```
white_wine %>%
  mutate(log_pH = log(pH)) %>%
  ggplot(aes(sample = log_pH)) +
    stat_qq(color = 'lightblue') +
    stat_qq_line(color = "black") +
    labs(
      title = paste("QQ Plot for log_pH of white wine"),
      x = "Theoretical",
      y = "Sample") +
    theme_calc()+
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```



QQ plots appear approximately normal. We can carry on with a parametric test.

```
# Test for homogeneity of variance
wine_data %>%
  mutate(log_pH = log(pH)) %>%
  bartlett.test(log_pH ~ wine_type, data = .)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  log_pH by wine_type
## Bartlett's K-squared = 0.089408, df = 1, p-value = 0.7649
```

P>0,05, fail to reject null hypothesis, variance is equal.

```
# Two sample t-test (two tailed)
wine_data %>%
  mutate(log_pH = log(pH)) %>%
  t.test(pH ~ wine_type, data = ., var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  pH by wine_type
## t = 23.856, df = 5318, p-value < 2.2e-16
```

```
## alternative hypothesis: true difference in means between group Red and group White is not equal to 0
## 95 percent confidence interval:
##  0.1049333 0.1237235
## sample estimates:
##   mean in group Red mean in group White
##            3.309787            3.195458
```

In this case the p-value is less than 0.05, therefore the Null Hypothesis is rejected and Alternative Hypothesis is accepted.

Hypothesis Test 3

chlorides vs. Wine Type Hypotheses: Null Hypothesis (H) The distribution of chlorides is identical in red and white wines. Alternative Hypothesis (H):The distribution of chlorides is not identical in red and white wines.

```r
#normality checks
# Red wine
shapiro.test(red_wine$chlorides)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  red_wine$chlorides
## W = 0.48448, p-value < 2.2e-16
```

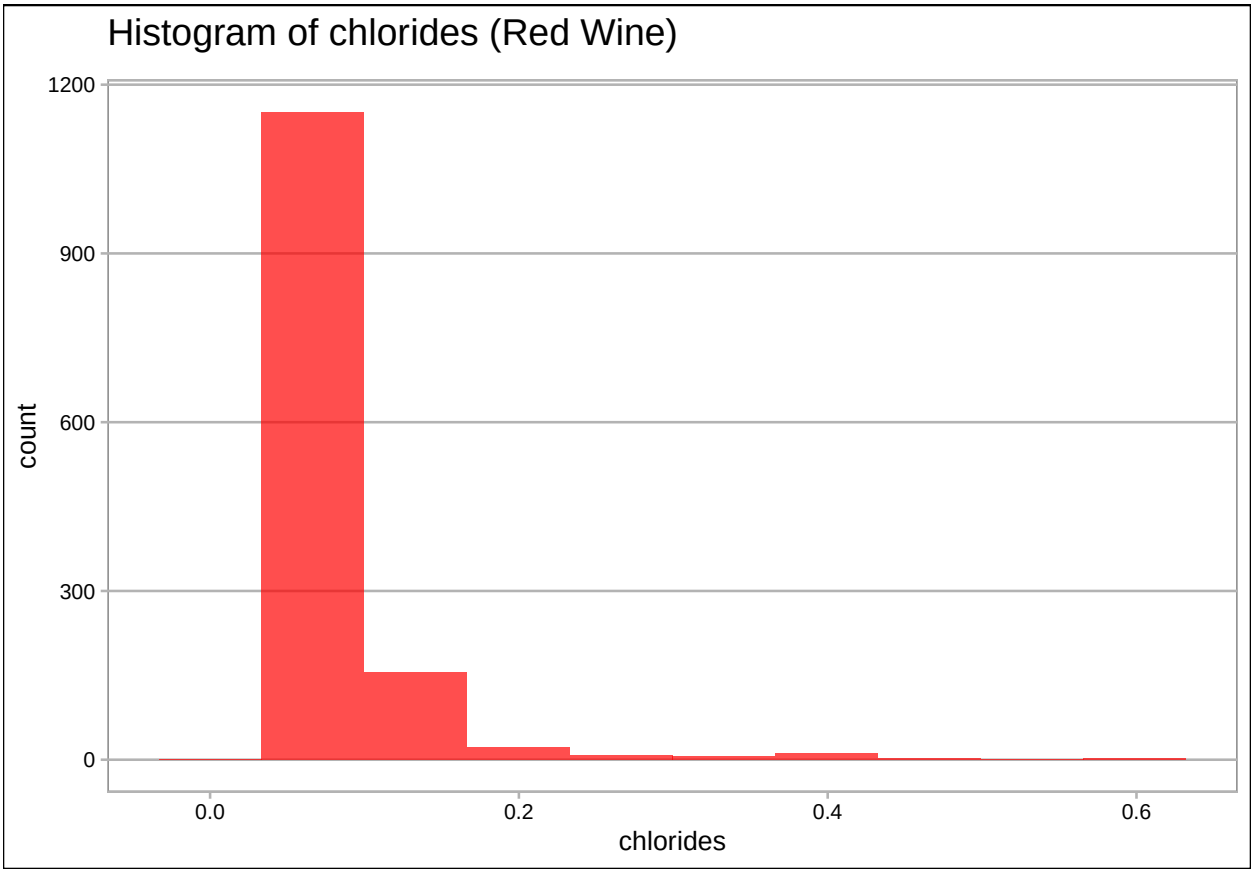
```r
# White wine
shapiro.test(white_wine$chlorides)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  white_wine$chlorides
## W = 0.57789, p-value < 2.2e-16
```

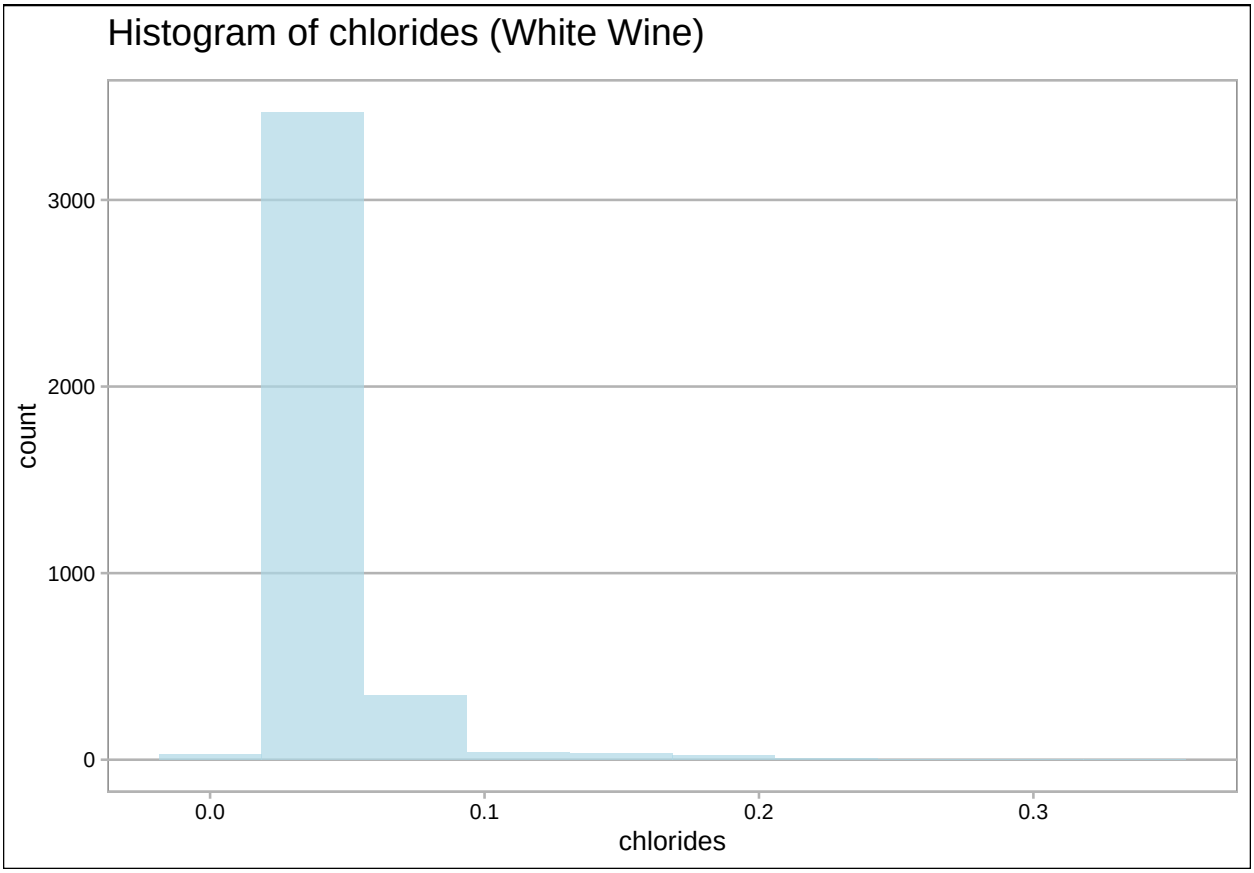
p<0.05: Not normally distributed → Mann-Whitney U Test.

Assumptions for Mann-Whitney U Test Checking for same shape

```r
#plotting the distribution of chlorides for red  wine
  ggplot(red_wine, aes(x = chlorides)) +
    geom_histogram(alpha = 0.7, bins =10,fill = "red") +
    labs(title ="Histogram of chlorides (Red Wine)") +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```




```r
#plotting the distribution of chlorides for white wine
  ggplot(white_wine, aes(x = chlorides)) +
    geom_histogram(alpha = 0.7, bins =10, fill = "lightblue") +
    labs(title ="Histogram of chlorides (White Wine)") +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```




```
# Mann-Whitney U Test / Wilcox rank sum test
wilcox.test(chlorides ~ wine_type,data = wine_data)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  chlorides by wine_type
## W = 5071770, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

p<0.05: we reject Null Hypothesis (H):The chlorides distributions differ significantly between the two wine types

Hypothesis Test 4 Hypotheses: Null Hypothesis (H): Median Alcohol level is equal across all quality type Alternative Hypothesis (H): Median Alcohol is not equal across all quality type

```
# Normality check for alcohol for all quality types
byf.shapiro(alcohol ~ quality_type, data = wine_data_copy)
```

```
##
##  Shapiro-Wilk normality tests
##
## data:  alcohol by quality_type
##
##             W   p-value
```

```
## Low    0.9742 0.0002668 ***
## Medium 0.9452 < 2.2e-16 ***
## High   0.9870 8.491e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

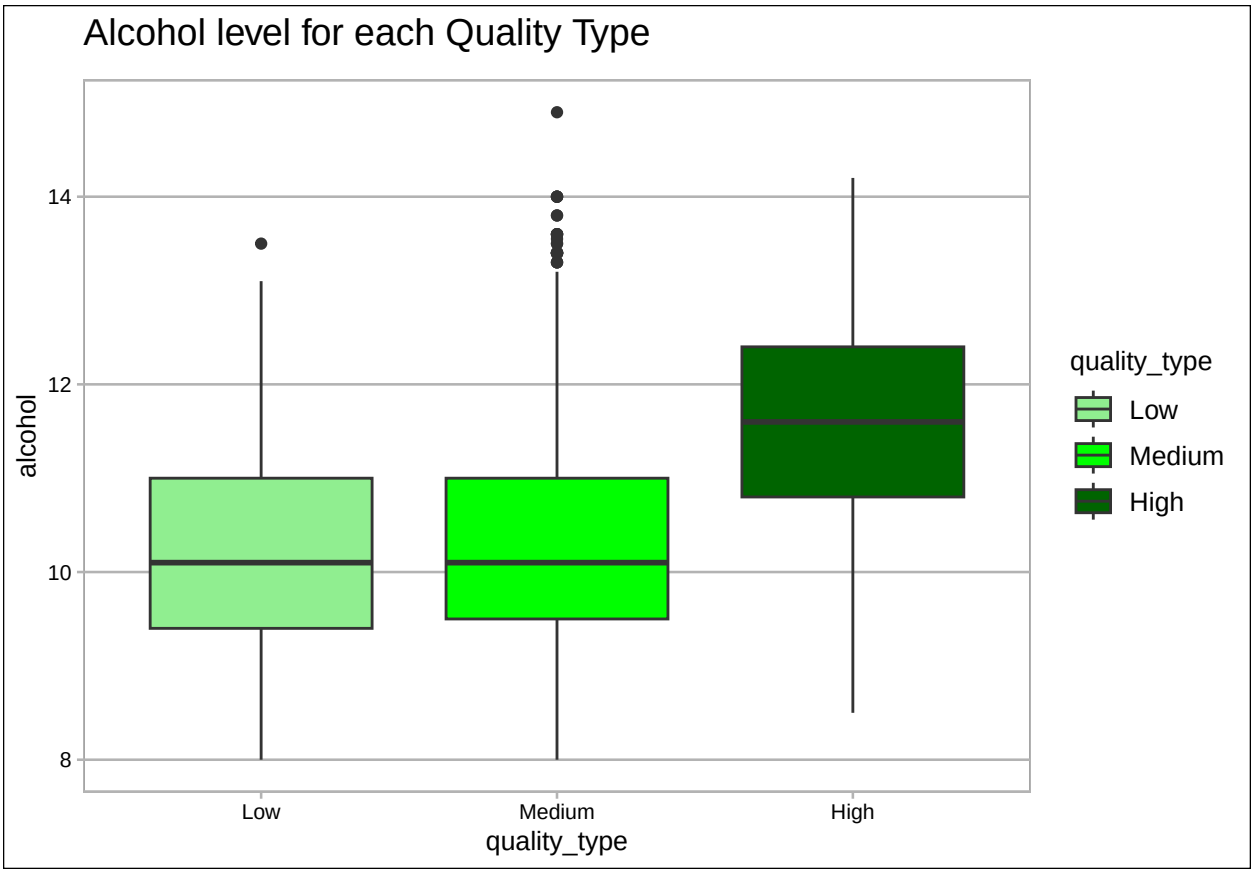
P<0.05, Reject null hypothesis. Not normally distributed.

```r
# Test for homogeneity of variance
bartlett.test(alcohol ~ quality_type, data = wine_data_copy)
```

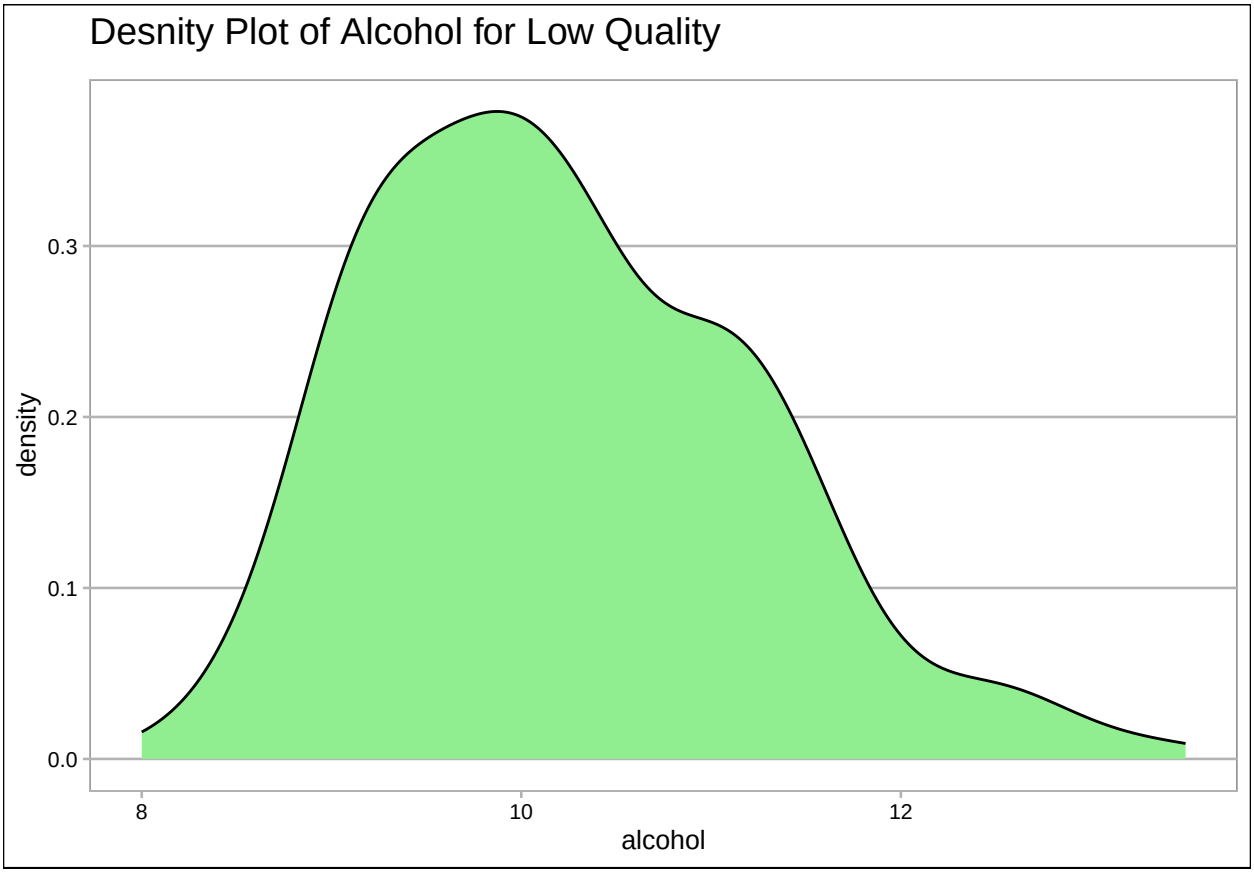
```
##
##  Bartlett test of homogeneity of variances
##
## data:  alcohol by quality_type
## Bartlett's K-squared = 5.4887, df = 2, p-value = 0.06429
```

P > 0.05, Accept Null hypothesis.

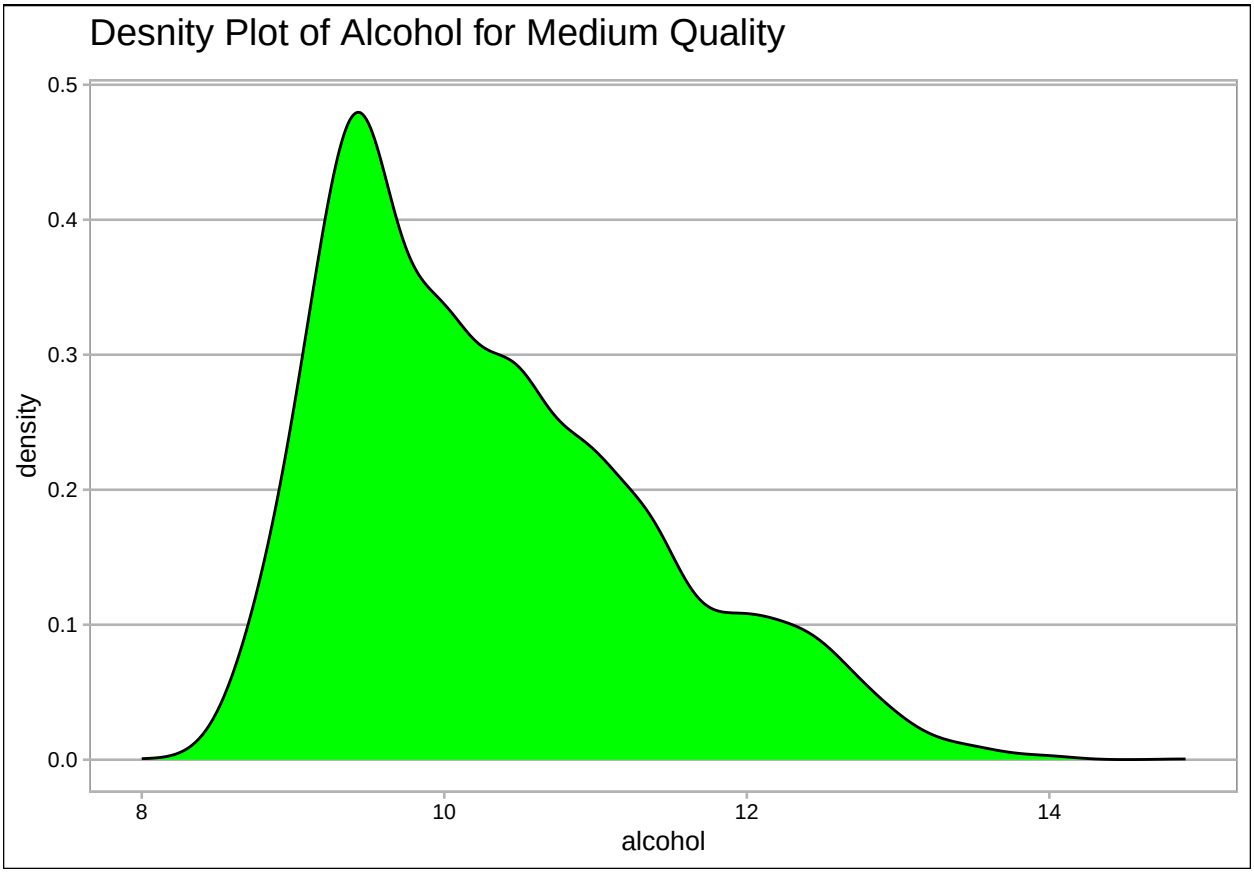
```r
# Distribution of alcohol in across quality type
ggplot(data=wine_data_copy, aes(x = quality_type, y = alcohol, fill=quality_type)) +
  geom_boxplot() +
  labs(title = paste("Alcohol level for each Quality Type")) +
  scale_fill_manual(values = c("Low" = "lightgreen", "Medium" = "green", "High" = "darkgreen")) +
  theme_calc() +
  theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```



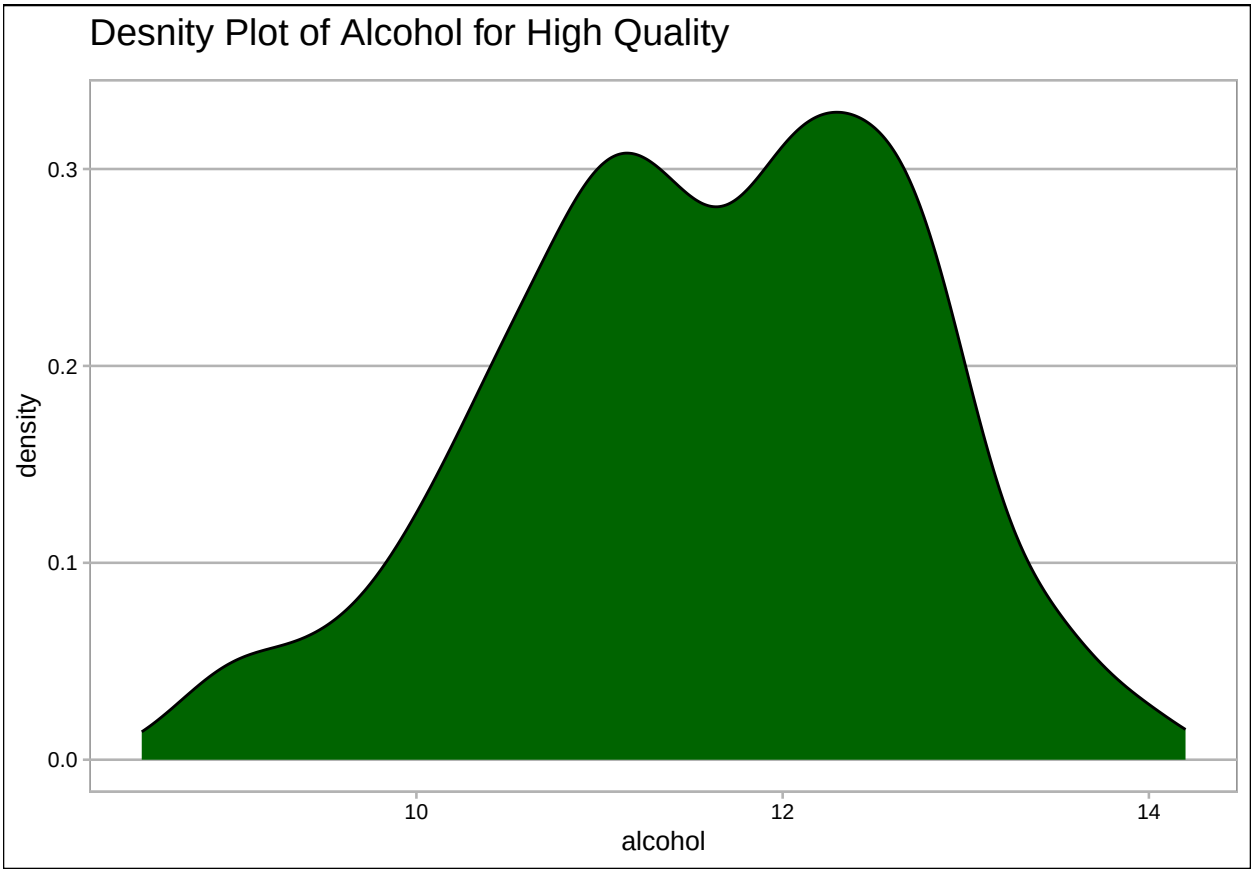
```r
#plotting the distribution of Alcohol for low quality
wine_data_copy %>%
  filter(quality_type == 'Low') %>%
  ggplot(aes(x = alcohol)) +
    geom_density(fill = 'lightgreen') +
    labs(title ="Desnity Plot of Alcohol for Low Quality") +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```




```r
#plotting the distribution of Alcohol for medium quality
wine_data_copy %>%
  filter(quality_type == 'Medium') %>%
  ggplot(aes(x = alcohol)) +
    geom_density(fill = 'green') +
    labs(title ="Desnity Plot of Alcohol for Medium Quality") +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```




```
#plotting the distribution of Alcohol for high quality
wine_data_copy %>%
  filter(quality_type == 'High') %>%
  ggplot(aes(x = alcohol)) +
    geom_density(fill = 'darkgreen') +
    labs(title ="Desnity Plot of Alcohol for High Quality") +
    theme_calc() +
    theme(plot.title = element_text(size = 14, margin = margin(b = 10)))
```




```
# Kruskal-Wallace test for non parametric models.
kruskal.test(alcohol ~ quality_type, data = wine_data_copy)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  alcohol by quality_type
## Kruskal-Wallis chi-squared = 838.84, df = 2, p-value < 2.2e-16
```

$p < 0.05$, We reject Null hypothesis and carry out Dunes test, to check which is actually different.

POST-HOC TEST

```
# Dunns test
dunnTest(wine_data_copy$alcohol, wine_data_copy$quality_type, method = 'bonferroni')
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Bonferroni method.

##      Comparison         Z       P.unadj         P.adj
## 1    High - Low 14.819268  1.099756e-49  3.299269e-49
## 2 High - Medium 28.692487 4.734457e-181 1.420337e-180
## 3  Low - Medium -0.935171  3.497002e-01  1.000000e+00
```

Regression

Model 1 Question: Can wine type and physiochemical properties be used to predict wine quality. Logistic Regression 1

```
# preparing data for logistic regression
# changing the quality to binary for logistic regression
wine_data_logistic <- wine_data %>%
  mutate(quality_binary = case_when(
    quality %in% c(3, 4,5,6) ~ 0,
    quality %in% c(7,8, 9) ~ 1
  ))
head(wine_data_logistic)
```

```
## # A tibble: 6 x 14
##   `fixed acidity` `volatile acidity` `citric acid` `residual sugar` chlorides
##             <dbl>              <dbl>         <dbl>             <dbl>     <dbl>
## 1             7.4               0.7           0                 1.9     0.076
## 2             7.8               0.88          0                 2.6     0.098
## 3             7.8               0.76          0.04              2.3     0.092
## 4            11.2               0.28          0.56              1.9     0.075
## 5             7.4               0.66          0                 1.8     0.075
## 6             7.9               0.6           0.06              1.6     0.069
## # i 9 more variables: `free sulfur dioxide` <dbl>,
## #   `total sulfur dioxide` <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
## #   alcohol <dbl>, quality <ord>, wine_type <fct>, quality_binary <dbl>
```

Backward Step wise

```
# Quailty prediction model 1
quality_prediction_1 <- glm(
  quality_binary ~ wine_type + `fixed acidity` + `volatile acidity` +
  `citric acid`+ `residual sugar` + chlorides+ `free sulfur dioxide` +
  `total sulfur dioxide`+ sulphates + density + pH + alcohol,
  data = wine_data_logistic,family = "binomial"
)

summary(quality_prediction_1)
```

```
##
## Call:
## glm(formula = quality_binary ~ wine_type + `fixed acidity` +
##     `volatile acidity` + `citric acid` + `residual sugar` + chlorides +
##     `free sulfur dioxide` + `total sulfur dioxide` + sulphates +
##     density + pH + alcohol, family = "binomial", data = wine_data_logistic)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             4.207e+02  7.391e+01   5.692 1.26e-08 ***
## wine_typeWhite         -5.508e-01  2.718e-01  -2.026  0.04274 *
## `fixed acidity`         5.326e-01  7.580e-02   7.027 2.12e-12 ***
## `volatile acidity`     -3.044e+00  4.294e-01  -7.088 1.36e-12 ***
## `citric acid`           3.541e-01  3.922e-01   0.903  0.36662
```

```
## `residual sugar`        2.033e-01  2.972e-02   6.839 7.96e-12 ***
## chlorides              -8.481e+00  2.797e+00  -3.033  0.00243 **
## `free sulfur dioxide`   1.695e-02  3.399e-03   4.986 6.18e-07 ***
## `total sulfur dioxide` -6.252e-03  1.553e-03  -4.026 5.67e-05 ***
## sulphates               2.537e+00  3.266e-01   7.767 8.02e-15 ***
## density                -4.451e+02  7.492e+01  -5.940 2.85e-09 ***
## pH                      3.305e+00  4.100e-01   8.062 7.52e-16 ***
## alcohol                 4.764e-01  9.045e-02   5.267 1.39e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5168.2  on 5319  degrees of freedom
## Residual deviance: 3937.0  on 5307  degrees of freedom
## AIC: 3963
##
## Number of Fisher Scoring iterations: 6
```

```r
# Quality prediction model 2
quality_prediction_2 <- glm(
  quality_binary ~ wine_type + `fixed acidity` + `volatile acidity` +
  `residual sugar` + chlorides+ `free sulfur dioxide` +
  `total sulfur dioxide`+ sulphates + density + pH + alcohol,
  data = wine_data_logistic,family = "binomial"
)

summary(quality_prediction_2)
```

```
##
## Call:
## glm(formula = quality_binary ~ wine_type + `fixed acidity` +
##     `volatile acidity` + `residual sugar` + chlorides + `free sulfur dioxide` +
##     `total sulfur dioxide` + sulphates + density + pH + alcohol,
##     family = "binomial", data = wine_data_logistic)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             4.168e+02  7.384e+01   5.645 1.66e-08 ***
## wine_typeWhite         -5.368e-01  2.718e-01  -1.975  0.04832 *
## `fixed acidity`         5.474e-01  7.409e-02   7.388 1.49e-13 ***
## `volatile acidity`     -3.177e+00  4.042e-01  -7.860 3.85e-15 ***
## `residual sugar`        2.028e-01  2.973e-02   6.821 9.03e-12 ***
## chlorides              -8.272e+00  2.793e+00  -2.962  0.00306 **
## `free sulfur dioxide`   1.677e-02  3.392e-03   4.944 7.65e-07 ***
## `total sulfur dioxide` -6.102e-03  1.544e-03  -3.953 7.73e-05 ***
## sulphates               2.546e+00  3.264e-01   7.800 6.18e-15 ***
## density                -4.412e+02  7.485e+01  -5.894 3.77e-09 ***
## pH                      3.291e+00  4.099e-01   8.027 1.00e-15 ***
## alcohol                 4.879e-01  8.964e-02   5.443 5.24e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 5168.2  on 5319  degrees of freedom
## Residual deviance: 3937.9  on 5308  degrees of freedom
## AIC: 3961.9
##
## Number of Fisher Scoring iterations: 6
```

```
# Important features for predicting quality
quality_feature_importance <- varImp(quality_prediction_2, scale=False)
quality_feature_importance <- quality_feature_importance %>% arrange(desc(Overall))
quality_feature_importance
```

```
##                         Overall
## pH                     8.026870
## `volatile acidity`     7.859567
## sulphates              7.800256
## `fixed acidity`        7.388163
## `residual sugar`       6.821217
## density                5.894072
## alcohol                5.442944
## `free sulfur dioxide`  4.944129
## `total sulfur dioxide` 3.952710
## chlorides              2.961765
## wine_typeWhite         1.974587
```

Most important features when predicting quality are pH, volatile acidity,sulphates,residual sugar,fixed acidity density,alcohol.

```
# Using only the most important features
quality_prediction_3 <- glm(
  quality_binary ~ `fixed acidity` + `volatile acidity` +`residual sugar`+
  sulphates + density + pH + alcohol,
  data = wine_data_logistic,family = "binomial"
)

summary(quality_prediction_3)
```

```
##
## Call:
## glm(formula = quality_binary ~ `fixed acidity` + `volatile acidity` +
##     `residual sugar` + sulphates + density + pH + alcohol, family = "binomial",
##     data = wine_data_logistic)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         346.70846   52.85178   6.560 5.38e-11 ***
## `fixed acidity`       0.51703    0.06996   7.390 1.46e-13 ***
## `volatile acidity`   -3.12069    0.38370  -8.133 4.18e-16 ***
## `residual sugar`      0.17001    0.02193   7.751 9.15e-15 ***
## sulphates             2.54946    0.31641   8.057 7.80e-16 ***
## density            -372.59169   54.04946  -6.894 5.44e-12 ***
## pH                    3.23696    0.39187   8.260  < 2e-16 ***
## alcohol               0.61888    0.07199   8.597  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5168.2  on 5319  degrees of freedom
## Residual deviance: 3989.0  on 5312  degrees of freedom
## AIC: 4005
##
## Number of Fisher Scoring iterations: 6
```

```
# Multicolinearity check
vif(quality_prediction_2)
```

```
##              wine_type        `fixed acidity`     `volatile acidity`
##               7.084829               6.289306                1.576465
##       `residual sugar`              chlorides  `free sulfur dioxide`
##               8.692678               2.175781                2.154992
## `total sulfur dioxide`              sulphates                 density
##               3.695894               1.629069               27.810807
##                     pH                alcohol
##               2.739525               6.122816
```

```
# Quality prediction model 4
quality_prediction_4 <- glm(
  quality_binary ~ `volatile acidity` + wine_type +
  `residual sugar` + chlorides+ `free sulfur dioxide`+ sulphates + pH + alcohol,
  data = wine_data_logistic,family = "binomial"
)

summary(quality_prediction_4)
```

```
##
## Call:
## glm(formula = quality_binary ~ `volatile acidity` + wine_type +
##     `residual sugar` + chlorides + `free sulfur dioxide` + sulphates +
##     pH + alcohol, family = "binomial", data = wine_data_logistic)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -14.263451   1.066681 -13.372  < 2e-16 ***
## `volatile acidity`     -3.905121   0.399006  -9.787  < 2e-16 ***
## wine_typeWhite         -0.508452   0.177197  -2.869 0.004112 **
## `residual sugar`        0.028225   0.011058   2.552 0.010699 *
## chlorides             -11.213856   2.855071  -3.928 8.58e-05 ***
## `free sulfur dioxide`   0.007837   0.002615   2.997 0.002725 **
## sulphates               1.903407   0.303562   6.270 3.60e-10 ***
## pH                      0.883657   0.258802   3.414 0.000639 ***
## alcohol                 0.983468   0.041382  23.766  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 5168.2  on 5319  degrees of freedom
## Residual deviance: 4019.1  on 5311  degrees of freedom
## AIC: 4037.1
##
## Number of Fisher Scoring iterations: 6
```

```
# Multicollinearity check
vif(quality_prediction_4)
```

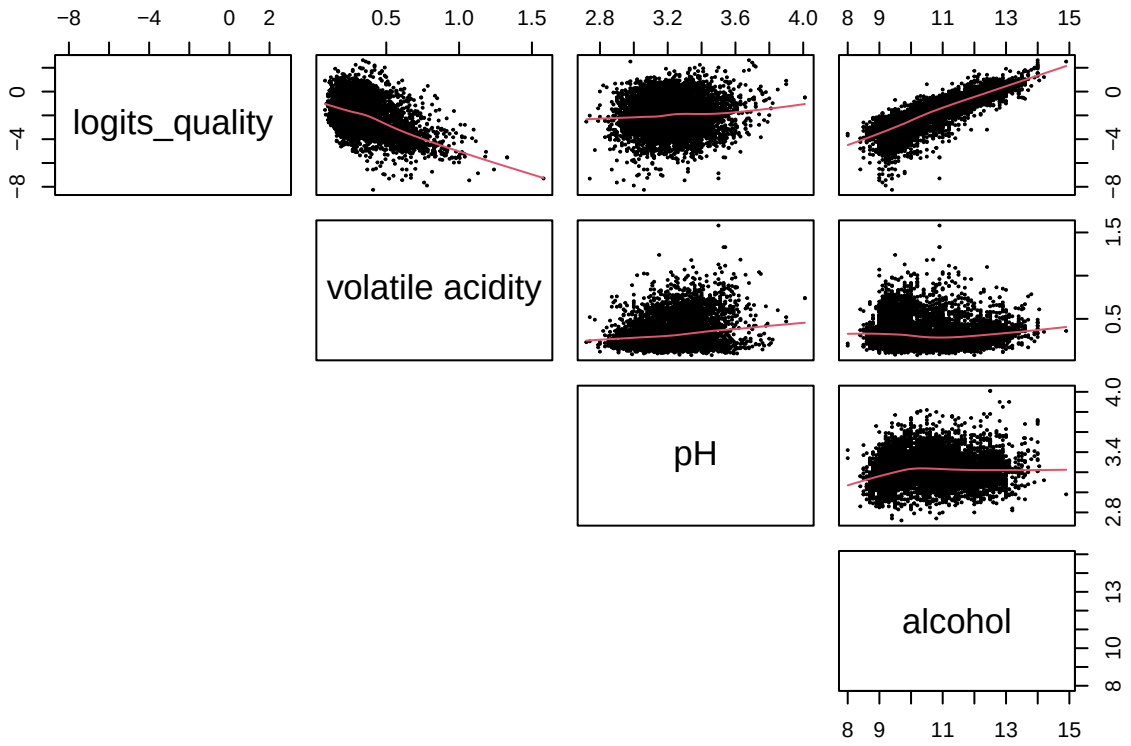
```
##    `volatile acidity`             wine_type      `residual sugar`
##              1.533879              2.977820              1.274633
##             chlorides `free sulfur dioxide`             sulphates
##              2.100503              1.311159              1.399575
##                    pH               alcohol
##              1.109331              1.335072
```

Linearity Check

```
# Calculating pi
probs_quality <- predict(quality_prediction_4, data=wine_data_logistic,type="response")
wine_data_logistic$probs_quality <- probs_quality
```

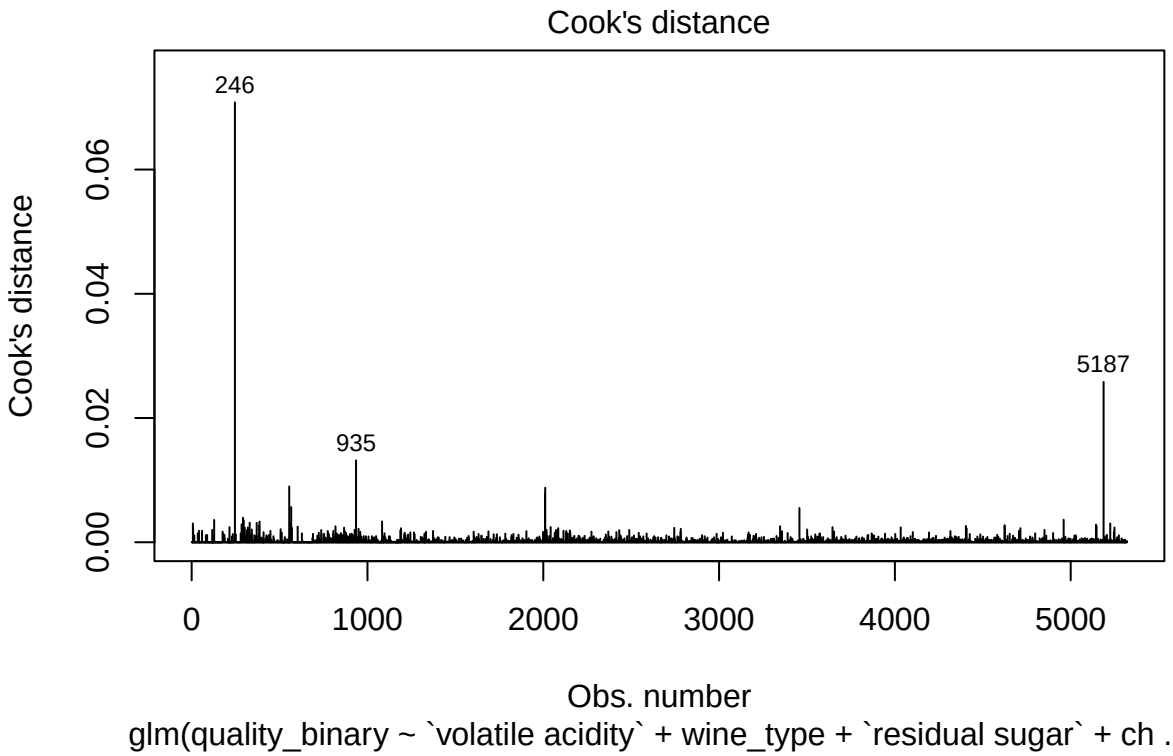
```
# Calculating logit(pi)
wine_data_logistic$logits_quality <- log(probs_quality/(1-probs_quality))
```

```
# scatter plot for linearity check
pairs(wine_data_logistic[,c(16,2,9,11)], lower.panel = NULL,
      upper.panel = panel.smooth, pch = 19,cex = 0.2)
```



Logits appears to have a linear relationship with volatile acidity, pH and alcohol.

```
# Cooks disatnce test for influential values
plot(quality_prediction_4, which = 4, id.n = 3)
```



It is worth noting that observation 246,935, and 5187, might have outliers.

```
# Model Evaluation
predicted <- ifelse(probs_quality > 0.5,1,0)
actual <- wine_data_logistic$quality_binary
confusionMatrix(factor(predicted),factor(actual))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4116  732
##          1  195  277
##
##                Accuracy : 0.8258
##                  95% CI : (0.8153, 0.8359)
##     No Information Rate : 0.8103
##     P-Value [Acc > NIR] : 0.002007
##
##                   Kappa : 0.288
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9548
##             Specificity : 0.2745
##          Pos Pred Value : 0.8490
```

```
##          Neg Pred Value : 0.5869
##              Prevalence : 0.8103
##          Detection Rate : 0.7737
##    Detection Prevalence : 0.9113
##       Balanced Accuracy : 0.6146
##
##        'Positive' Class : 0
##
```

Model 2 Question: Can wine quality and physiochemical properties be used to predict wine type?

Logistic Regression Model 2

```
# data for prediction (removing columns observed with substantial outliers)
wine_data_logistic_2 <- wine_data %>%
  mutate(wine_type_numeric = ifelse(wine_type == "Red", 0, 1)) %>%
  slice(-c(3654,919,929))
```

Backward Step wise

```
# First Prediction model for wine type
type_prediction_1 <- glm(
  wine_type_numeric ~ quality + `fixed acidity` + `volatile acidity` +
  `citric acid`+ `residual sugar` + chlorides+ `free sulfur dioxide` +
  `total sulfur dioxide`+ sulphates + density + pH + alcohol,
  data = wine_data_logistic_2,family = "binomial"
)

summary(type_prediction_1)
```

```
##
## Call:
## glm(formula = wine_type_numeric ~ quality + `fixed acidity` +
##     `volatile acidity` + `citric acid` + `residual sugar` + chlorides +
##     `free sulfur dioxide` + `total sulfur dioxide` + sulphates +
##     density + pH + alcohol, family = "binomial", data = wine_data_logistic_2)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             2.456e+03  2.807e+02   8.749  < 2e-16 ***
## quality.L               2.275e+00  5.887e+02   0.004  0.99692
## quality.Q               7.197e+00  5.664e+02   0.013  0.98986
## quality.C               2.977e+00  4.239e+02   0.007  0.99440
## quality^4               1.142e+00  2.510e+02   0.005  0.99637
## quality^5               1.300e+00  1.133e+02   0.011  0.99084
## quality^6              -2.932e-01  3.416e+01  -0.009  0.99315
## `fixed acidity`         8.806e-01  2.786e-01   3.161  0.00157 **
## `volatile acidity`     -7.236e+00  1.341e+00  -5.394 6.87e-08 ***
## `citric acid`           1.979e+00  1.506e+00   1.314  0.18881
## `residual sugar`        9.768e-01  1.163e-01   8.401  < 2e-16 ***
## chlorides              -1.945e+01  4.546e+00  -4.277 1.89e-05 ***
## `free sulfur dioxide`  -7.436e-02  1.671e-02  -4.451 8.56e-06 ***
## `total sulfur dioxide`  6.238e-02  6.512e-03   9.580  < 2e-16 ***
```

```
## sulphates              -2.135e+00  1.561e+00  -1.368  0.17146
## density                -2.460e+03  2.424e+02 -10.152  < 2e-16 ***
## pH                      3.607e+00  1.719e+00   2.098  0.03594 *
## alcohol                -2.551e+00  3.710e-01  -6.876 6.15e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6040.1  on 5316  degrees of freedom
## Residual deviance:  270.3  on 5299  degrees of freedom
## AIC: 306.3
##
## Number of Fisher Scoring iterations: 15
```

```
# Second prediction model for wine type.
type_prediction_2 <- glm(
  wine_type_numeric ~ `volatile acidity` +
  `residual sugar`+ `total sulfur dioxide`+ density + alcohol,
  data = wine_data_logistic_2,family = "binomial"
)

summary(type_prediction_2)
```

```
##
## Call:
## glm(formula = wine_type_numeric ~ `volatile acidity` + `residual sugar` +
##     `total sulfur dioxide` + density + alcohol, family = "binomial",
##     data = wine_data_logistic_2)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             2.094e+03  1.287e+02  16.265   <2e-16 ***
## `volatile acidity`     -7.555e+00  8.880e-01  -8.508   <2e-16 ***
## `residual sugar`        8.131e-01  6.521e-02  12.468   <2e-16 ***
## `total sulfur dioxide`  4.911e-02  4.288e-03  11.454   <2e-16 ***
## density                -2.086e+03  1.279e+02 -16.311   <2e-16 ***
## alcohol                -2.129e+00  2.117e-01 -10.057   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6040.1  on 5316  degrees of freedom
## Residual deviance:  372.4  on 5311  degrees of freedom
## AIC: 384.4
##
## Number of Fisher Scoring iterations: 9
```

```
# Most important variables when predicting wine type
type_feature_importance <- varImp(type_prediction_2, scale=False)
type_feature_importance <- type_feature_importance %>% arrange(desc(Overall))
type_feature_importance
```

```
##                          Overall
## density                16.310806
## `residual sugar`       12.467764
## `total sulfur dioxide` 11.454406
## alcohol                10.057075
## `volatile acidity`      8.508301
```

Most important features when predicting wine type are density, total sulfur dioxide, residual sugar, volatile acidity and alcohol

```
# Multicolinearity check
vif(type_prediction_2)
```

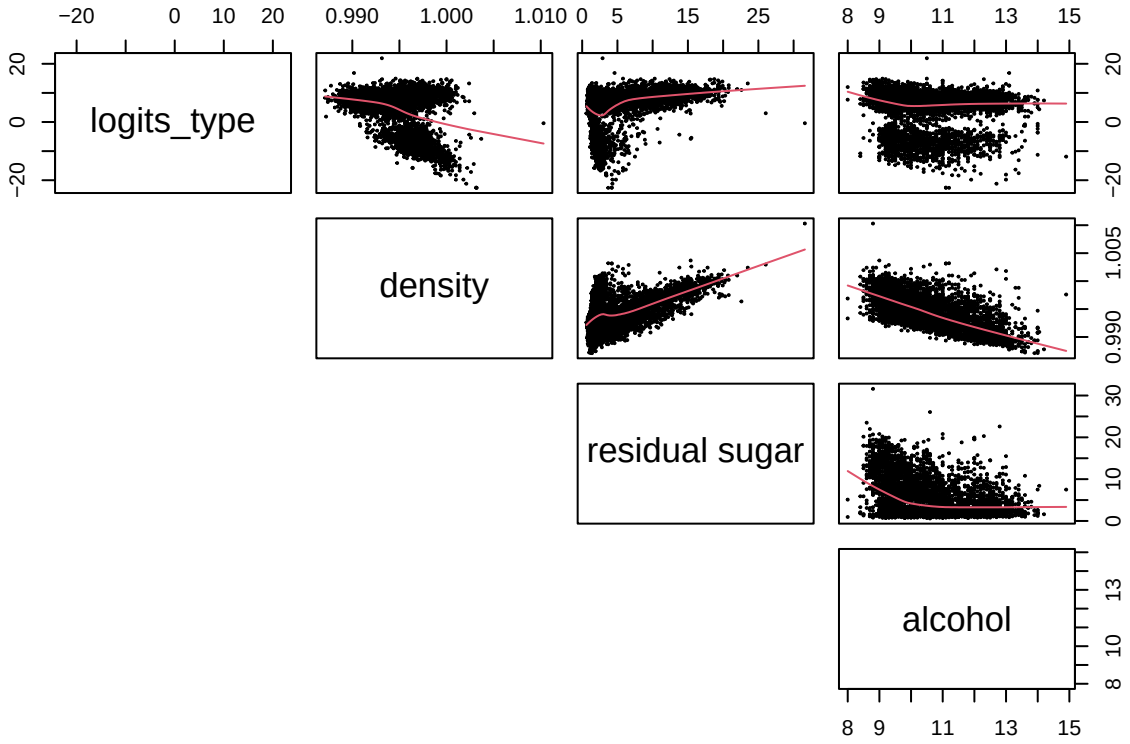
```
##     `volatile acidity`       `residual sugar` `total sulfur dioxide`
##               1.013125               2.041409               1.220932
##                density                alcohol
##               4.334865               2.750990
```

Linearity check

```
# Calculating pi values
probs_type <- predict(type_prediction_2, data=wine_data_logistic_2,type="response")
wine_data_logistic_2$probs_type <- probs_type
```

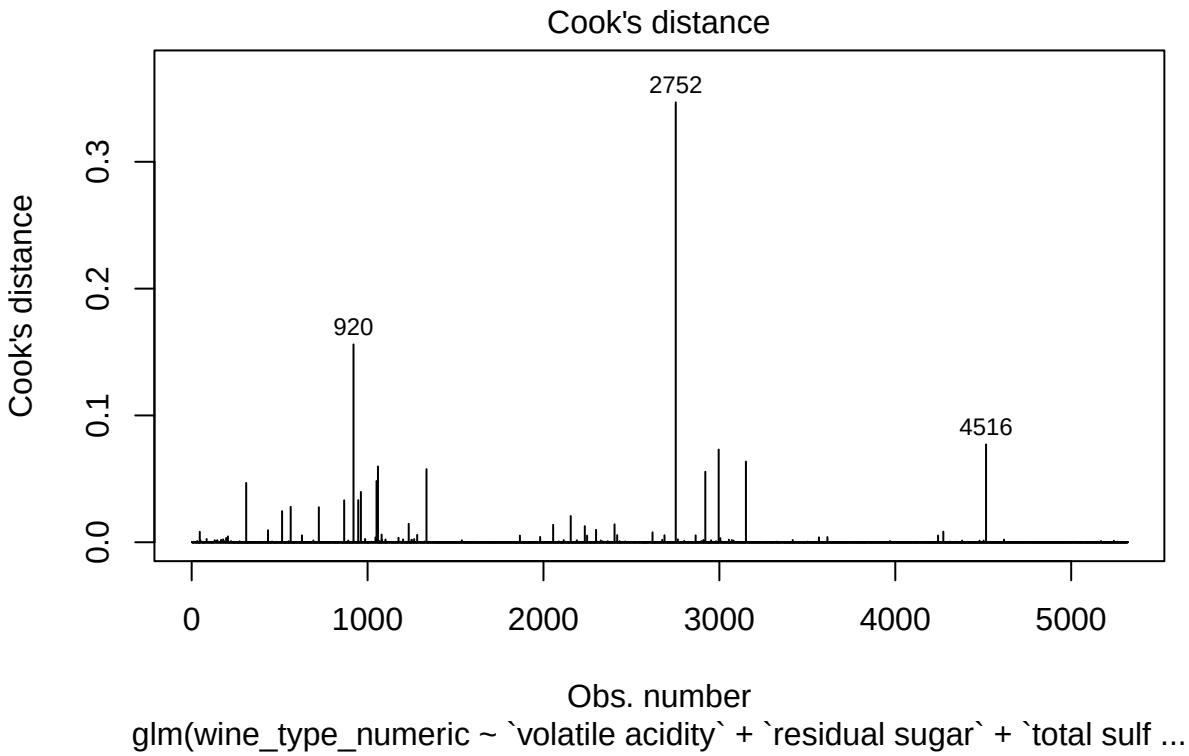
```
# calculating (logit(pi))
wine_data_logistic_2$logits_type <- log(probs_type/(1-probs_type))
```

```
# Scatter plot to confirm linearity
pairs(wine_data_logistic_2[,c(16,8,4,11)], lower.panel = NULL,
      upper.panel = panel.smooth, pch = 19,cex = 0.2)
```



Logits has an approximately linear relationship with density,residual sugar and alcohol.

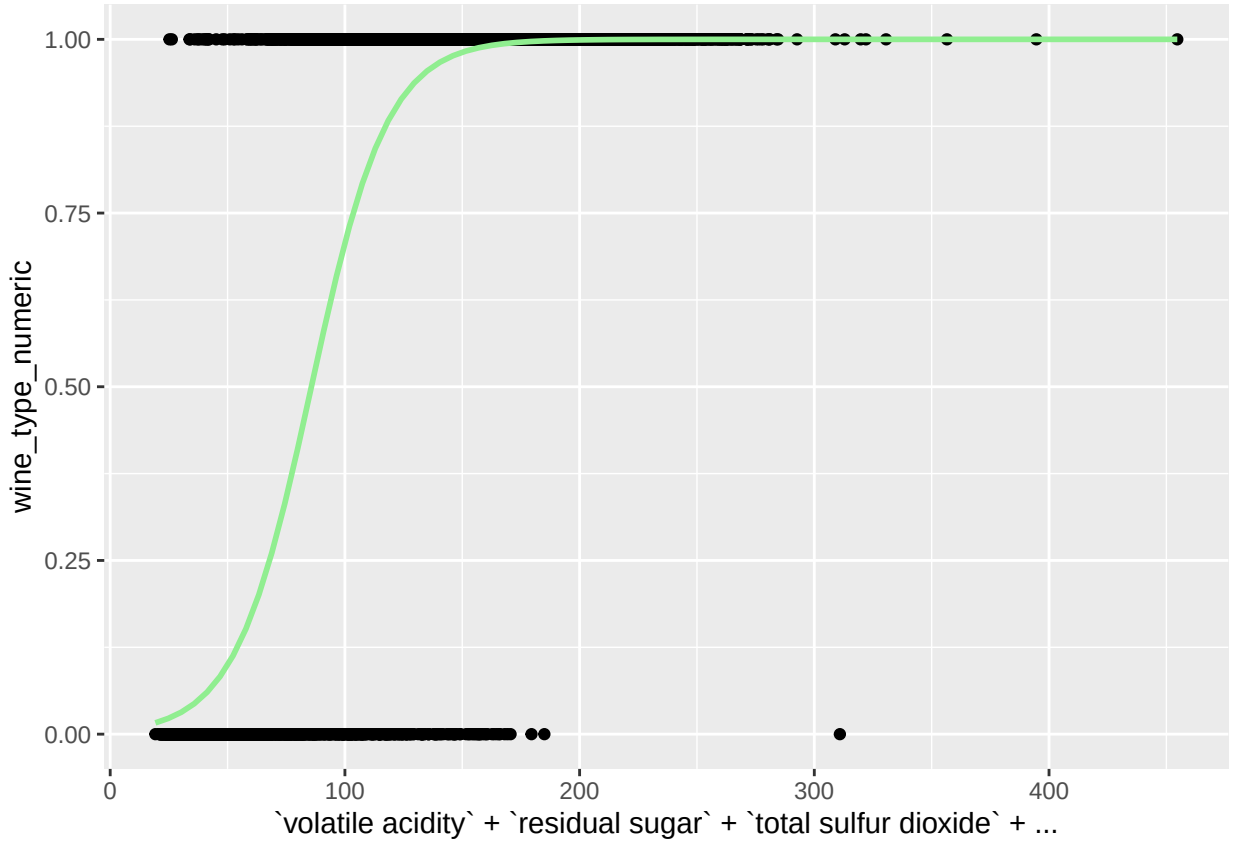
```
# checking for influential values.
plot(type_prediction_2, which = 4, id.n = 3)
```



Noticed that at observation 920,2752 and 4516, have some influential values worth looking out for.

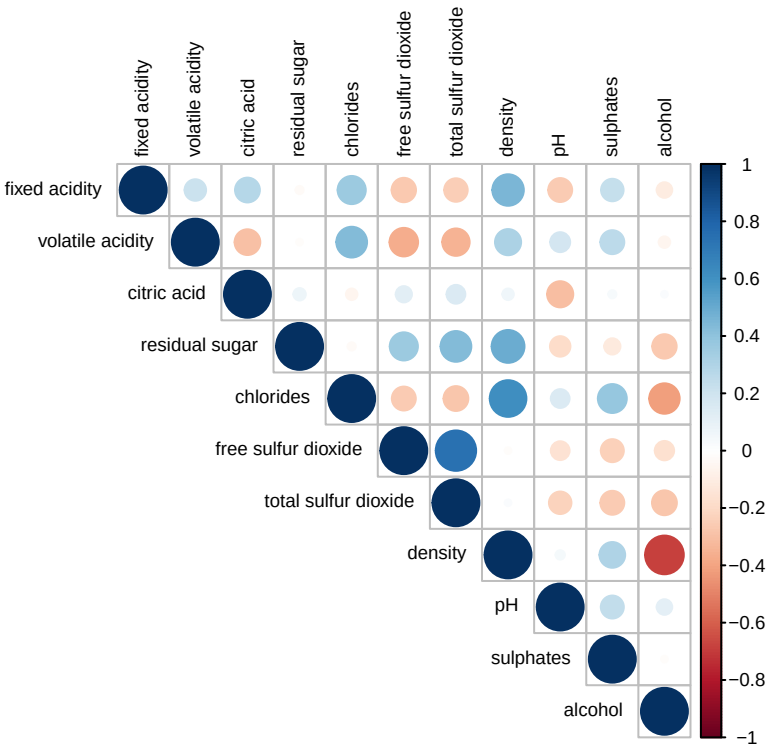
```
# Regression curve to predict wine type.
ggplot(wine_data_logistic_2, aes(x=`volatile acidity` +
  `residual sugar`+ `total sulfur dioxide`+ density + alcohol,
  y=wine_type_numeric)) +
geom_point() +
stat_smooth(method="glm", color="lightgreen", se=FALSE,
method.args = list(family=binomial))
```

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



MODEL 3 Question: can the other physiochemical properties be used to predict the density of wines? Multiple Linear Regression model.

```
# Visualizing the relationship between density and other properties
corrplot(wine_data_continuous_cor_matrix,
         method = "circle",
         type = "upper",
         title = "Relationship Between the Physiochemical Properties",
         tl.col = "black",
         number.cex = 0.6,
         tl.cex = 0.6,
         cl.cex = 0.6,
         mar = c(1, 0, 2, 0))
```



It seems like Density has a good correlation with alcohol, fixed acidity, volatile acidity and residual sugar.

```
# removing outliers using Z score
compute_z_scores <- function(data) {
  as.data.frame(scale(data))}
```

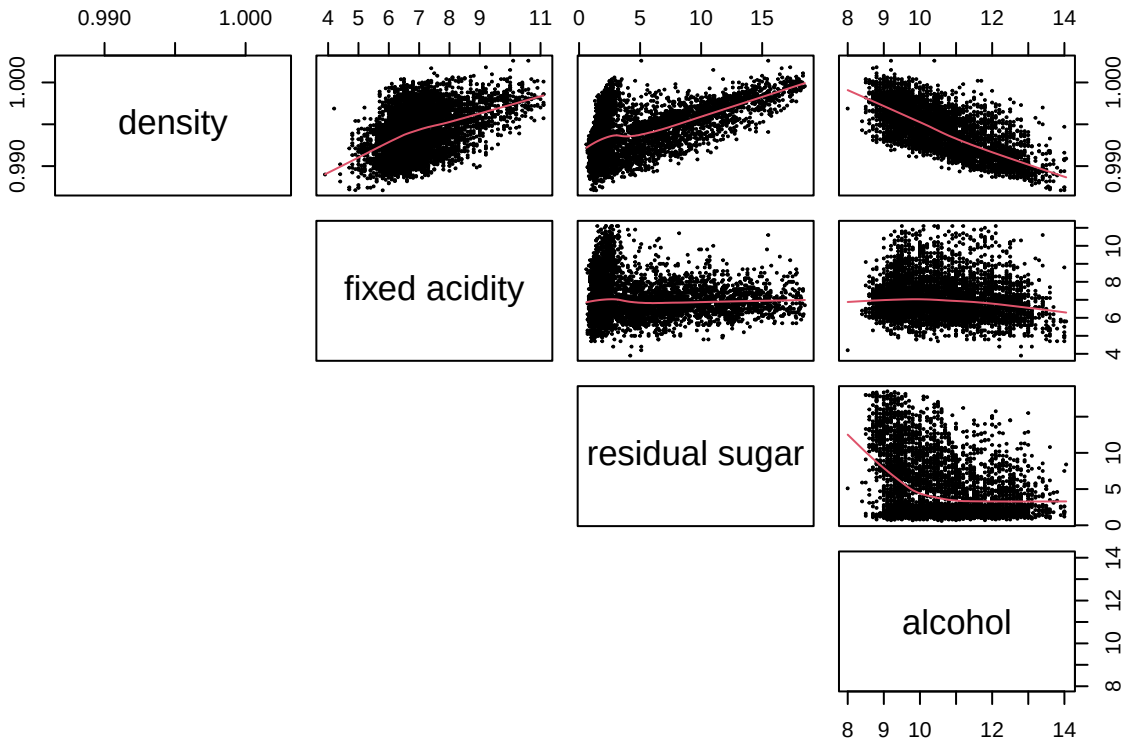
```
# Removing outliers
z_scores <- compute_z_scores(wine_data_continuous)
outlier_rows <- apply(z_scores, 1, function(row) any(abs(row) > 3))
wine_no_outliers <- wine_data_continuous[!outlier_rows, ]
```

```
#Multiple Linear regression model
density_prediction <- lm(density ~ alcohol +`residual sugar` +
                         `fixed acidity`+ `volatile acidity`,
                          wine_no_outliers)
summary(density_prediction)
```

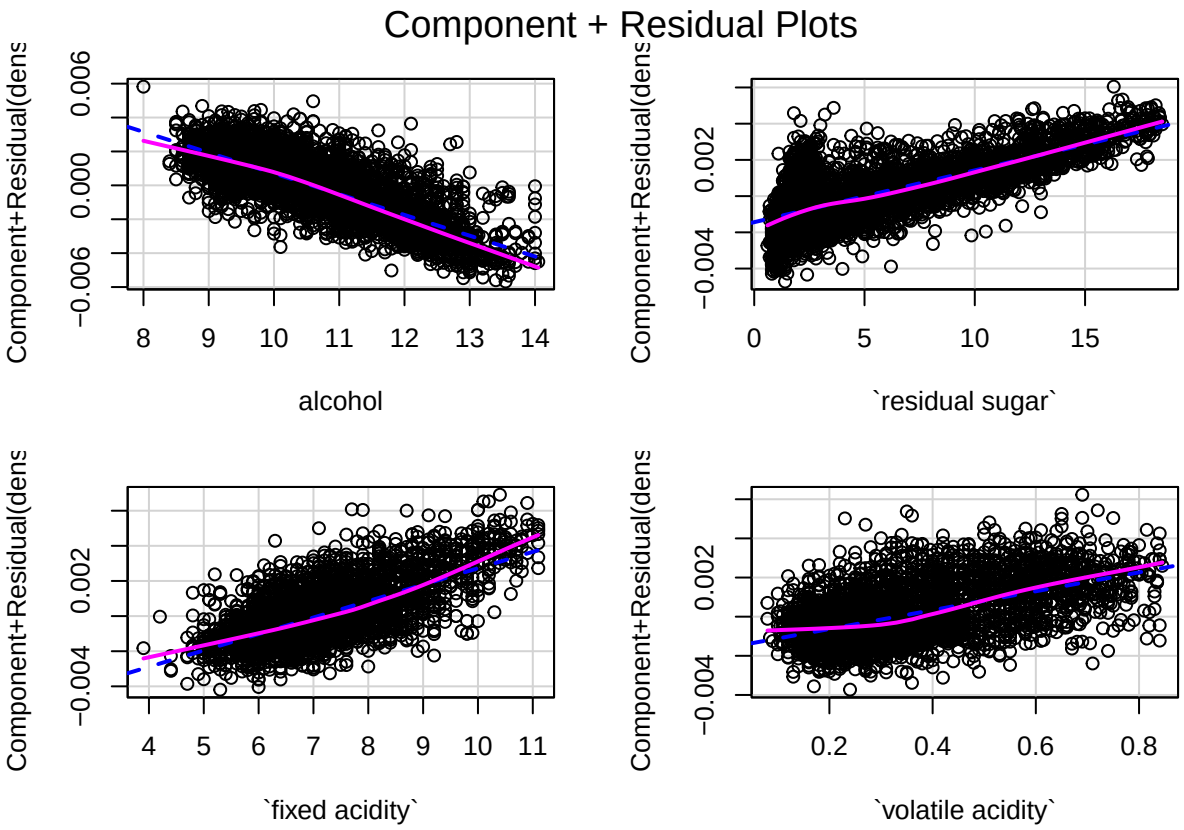
```
##
## Call:
## lm(formula = density ~ alcohol + `residual sugar` + `fixed acidity` +
##     `volatile acidity`, data = wine_no_outliers)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0042204 -0.0007157 -0.0000925  0.0006164  0.0055076
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          9.976e-01  2.089e-04 4776.57   <2e-16 ***
## alcohol             -1.225e-03  1.438e-05  -85.17   <2e-16 ***
## `residual sugar`     2.860e-04  4.013e-06   71.26   <2e-16 ***
## `fixed acidity`      9.355e-04  1.523e-05   61.44   <2e-16 ***
## `volatile acidity`   4.826e-03  1.099e-04   43.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001107 on 4884 degrees of freedom
## Multiple R-squared:  0.8428, Adjusted R-squared:  0.8427
## F-statistic:  6548 on 4 and 4884 DF,  p-value: < 2.2e-16
```

```
# Linearity between the IVs and DV
pairs(wine_no_outliers[,c(8,1,4,11)], lower.panel = NULL,
      upper.panel = panel.smooth,pch = 19,cex = 0.2)
```

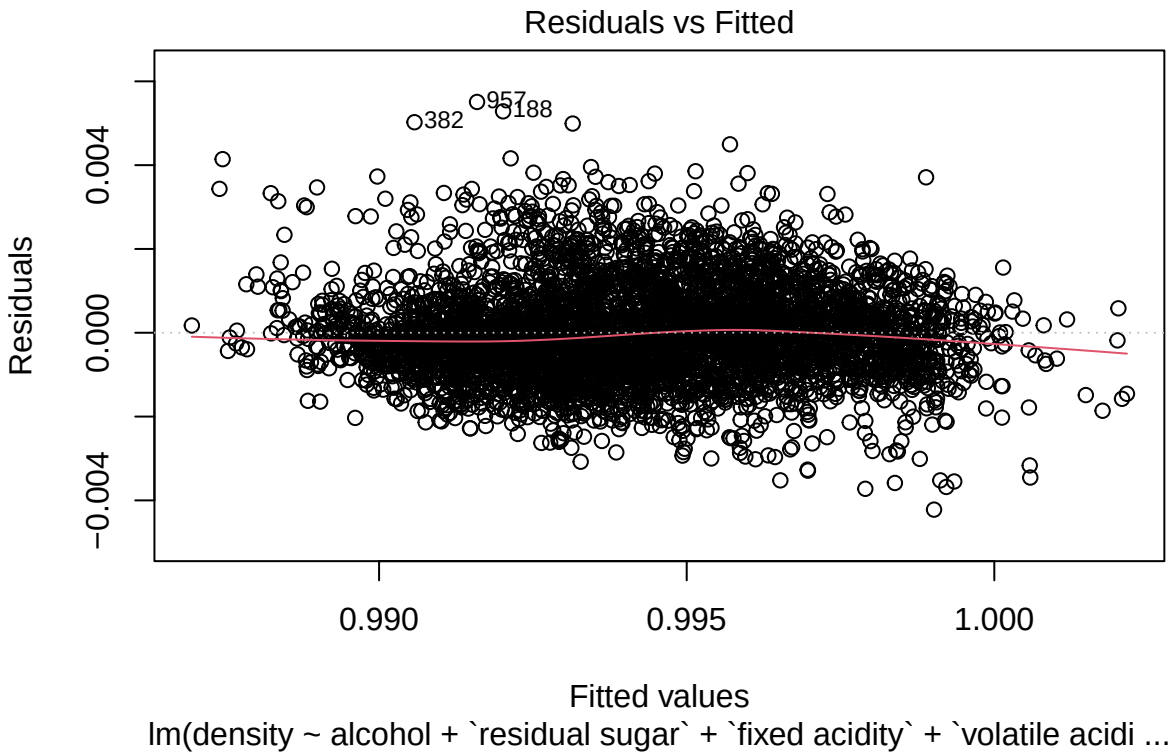



```
# Linearity between the IVs and DV
crPlots(density_prediction)
```



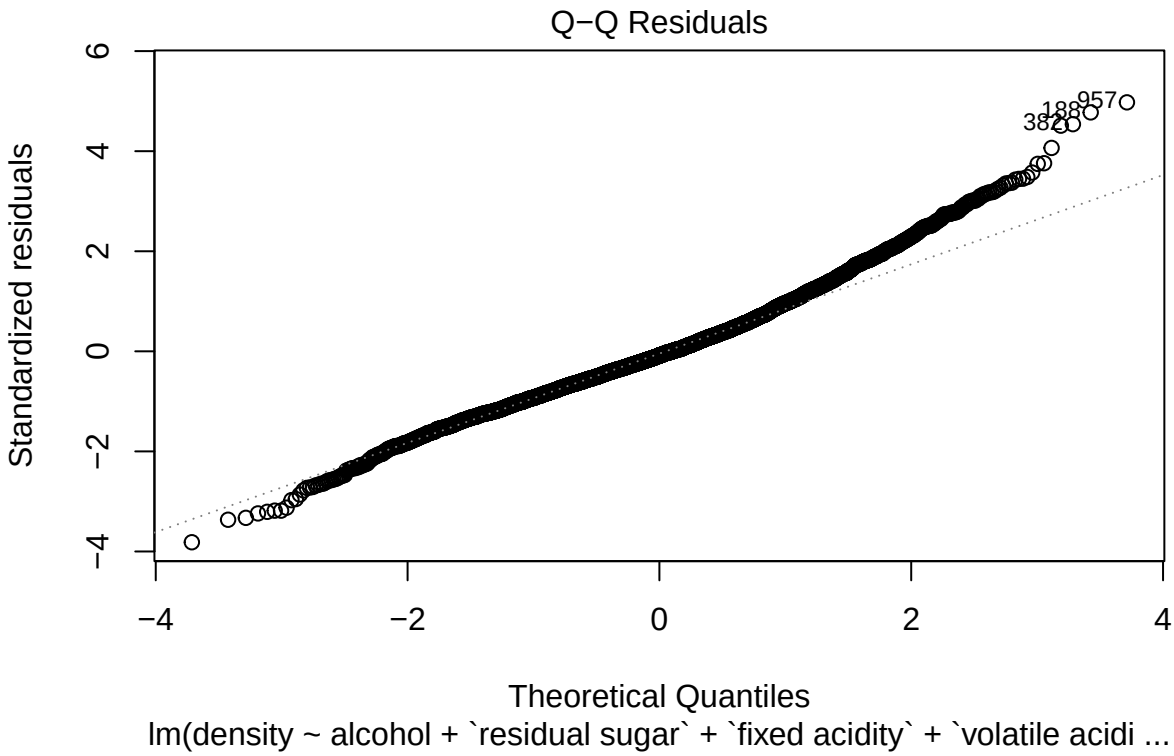
All the independent variables have a linear relationship with density (Dependent variable)

```
# Residual Independence
plot(density_prediction,1)
```



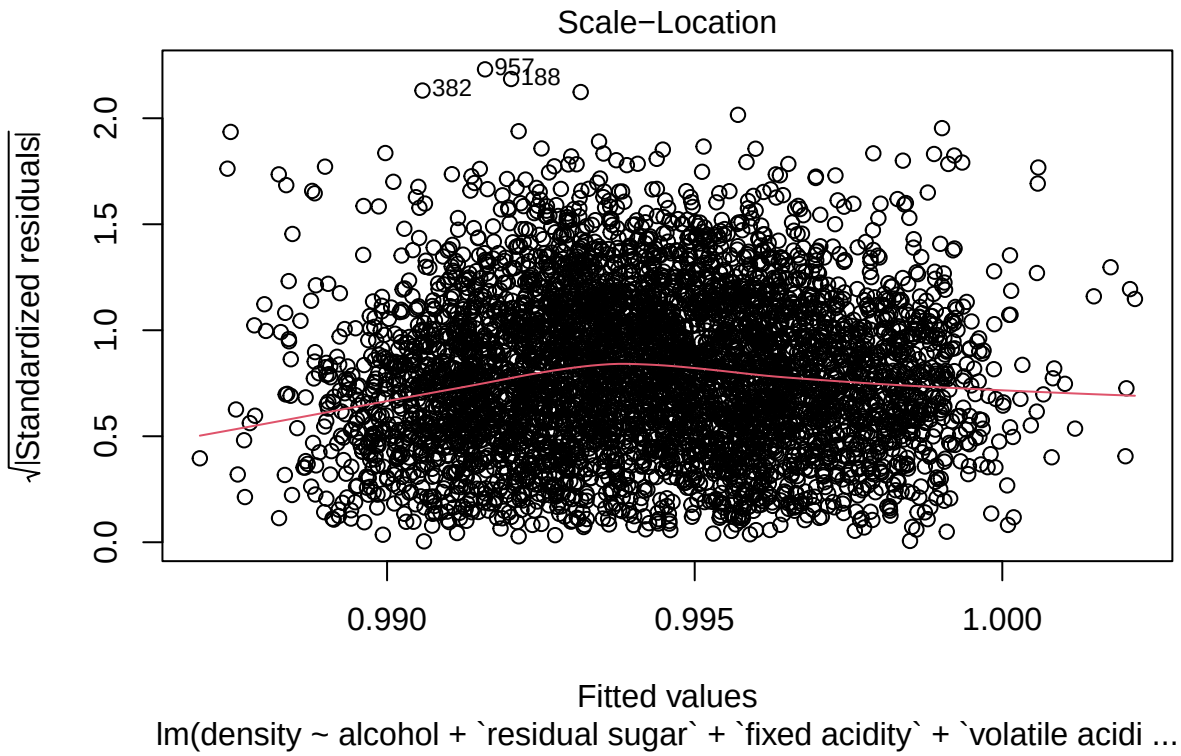
The correlation between the residuals is approximately 0, therefore theresiduals are independent.

```
# Normality of residuals
plot(density_prediction,2)
```



The residuals appear approximately normally distributed, with some deviations at the tail

```
# Homoscedacity test
plot(density_prediction,3)
```



There is no clear pattern among residuals, they appear randomly scattered with equal variablilty.

```
# Multicollinearity
vif(density_prediction)
```

```
##            alcohol   `residual sugar`    `fixed acidity` `volatile acidity`
##           1.161331           1.180526           1.062238           1.079728
```

The VIF of all the IVs are all less than 2, which indicates little to no correlation between the IVs and DV.