Predicting the quality of red wine

Synopsis

Mini-research project for Statistic course. Includes analyzing and predicting the quality of red wine. Based on Portuguese "Vinho Verde" red wine dataset. Includes tips for begginers in data science from Kaggle: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009/kernels (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009/kernels).

Some useful packages:

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

Reading and cleaning data

```
wine <- read.csv("wine_quality.csv")
head(wine)
```

	fixed.acidity <dbl>	volatile.acidity <dbl>	citric.acid <dbl>	residual.sugar <dbl>	chlorides <dbl>	free.sulfur.dioxide <dbl>
1	7.4	0.70	0.00	1.9	0.076	11
2	7.8	0.88	0.00	2.6	0.098	25
3	7.8	0.76	0.04	2.3	0.092	15

	fixed.acidity <dbl>	volatile.acidity <dbl>	citric.acid <dbl>	residual.sugar <dbl>	chlorides <dbl>	free.sulfur.dioxide <dbl>
4	11.2	0.28	0.56	1.9	0.075	17
5	7.4	0.70	0.00	1.9	0.076	11
6	7.4	0.66	0.00	1.8	0.075	13

6 rows | 1-7 of 13 columns

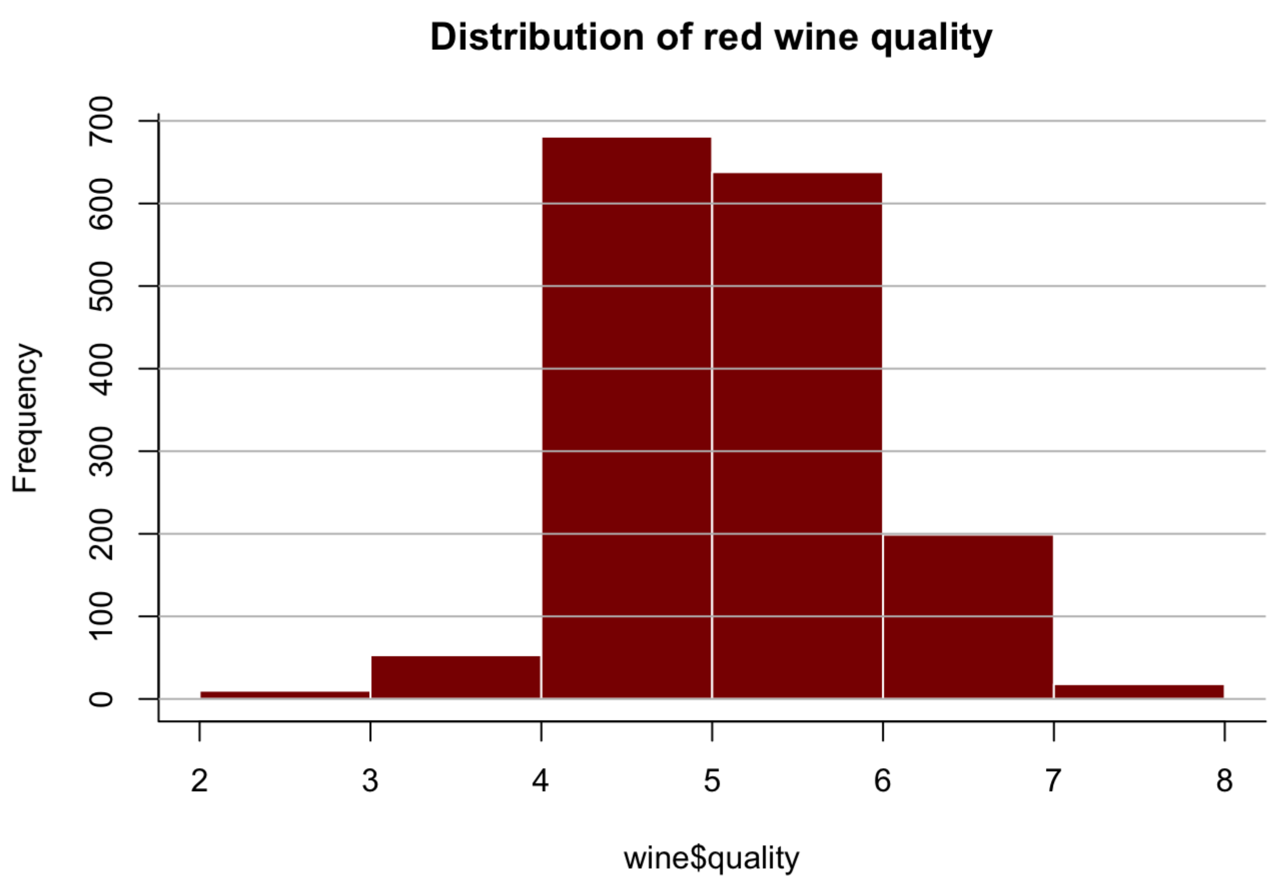
Disrtibution of wine quality

As one can see, the largest number of wines have a mediocre quality,

```
table(wine$quality)
```

```
##
##   3   4   5   6   7   8
##  10  53 681 638 199  18
```

```
hist(wine$quality,col="darkred",border="white", main="Distribution of red wine quality", breaks=c(2, 3, 4, 5, 6, 7, 8))
box(bty="l")
grid(nx=NA,ny=NULL,lty=1,lwd=1,col="gray")
```



Clasification

Splitting the wines into two category for visual analysis and performaing later regressions, so we have two possible classes:

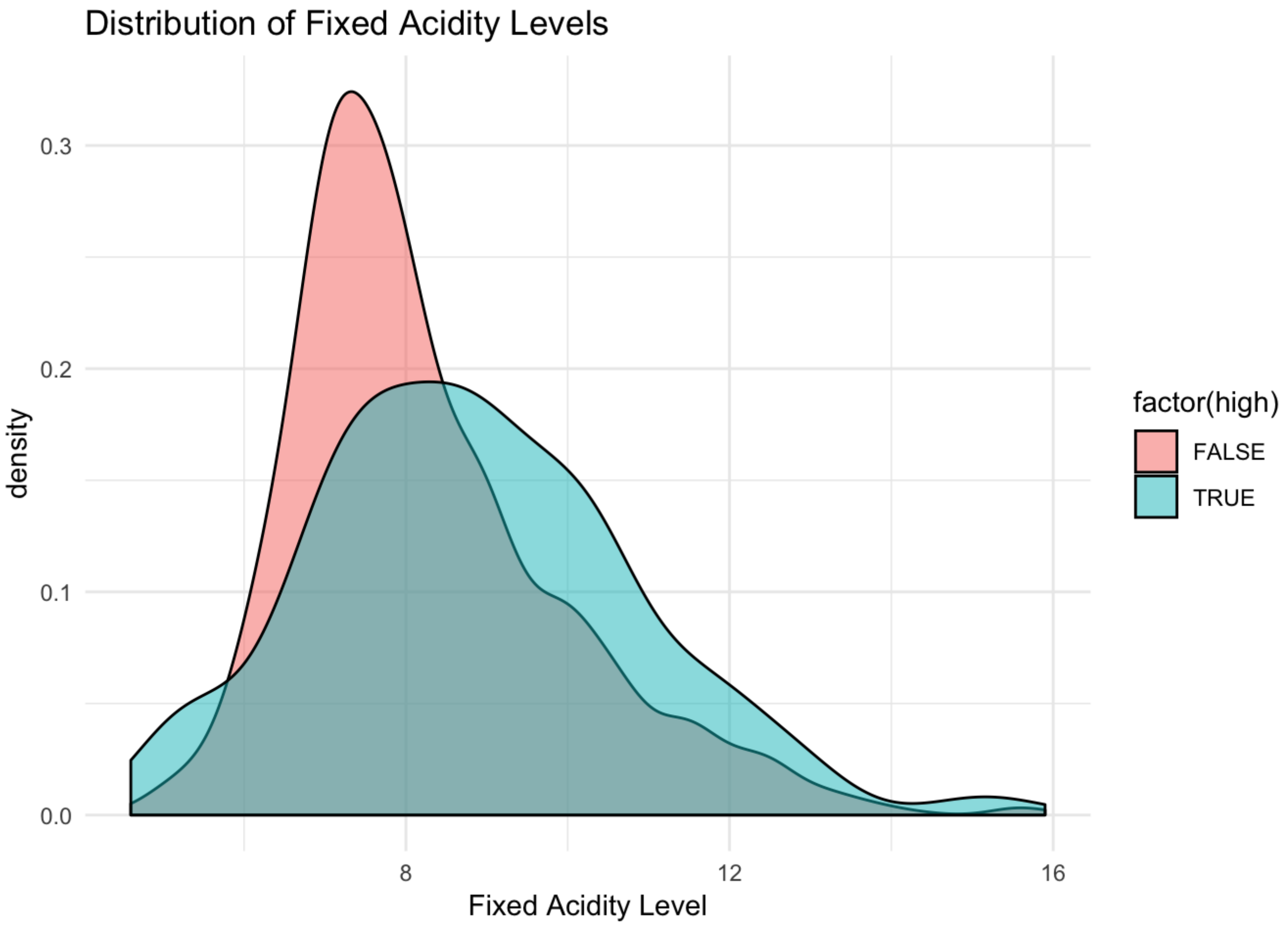
- *low quality wine*
- *high quality wine*

```
wine$high <- wine$quality >= 7
summary(wine)
```

```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00
##  Median :0.07900   Median :14.00       Median : 38.00
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00
##     density             pH          sulphates         alcohol
##  Min.   :0.9901   Min.   :2.740   Min.   :0.3300   Min.   : 8.40
##  1st Qu.:0.9956   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50
##  Median :0.9968   Median :3.310   Median :0.6200   Median :10.20
##  Mean   :0.9967   Mean   :3.311   Mean   :0.6581   Mean   :10.42
##  3rd Qu.:0.9978   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
##  Max.   :1.0037   Max.   :4.010   Max.   :2.0000   Max.   :14.90
##     quality         high
##  Min.   :3.000   Mode :logical
##  1st Qu.:5.000   FALSE:1382
##  Median :6.000   TRUE :217
##  Mean   :5.636
##  3rd Qu.:6.000
##  Max.   :8.000
```

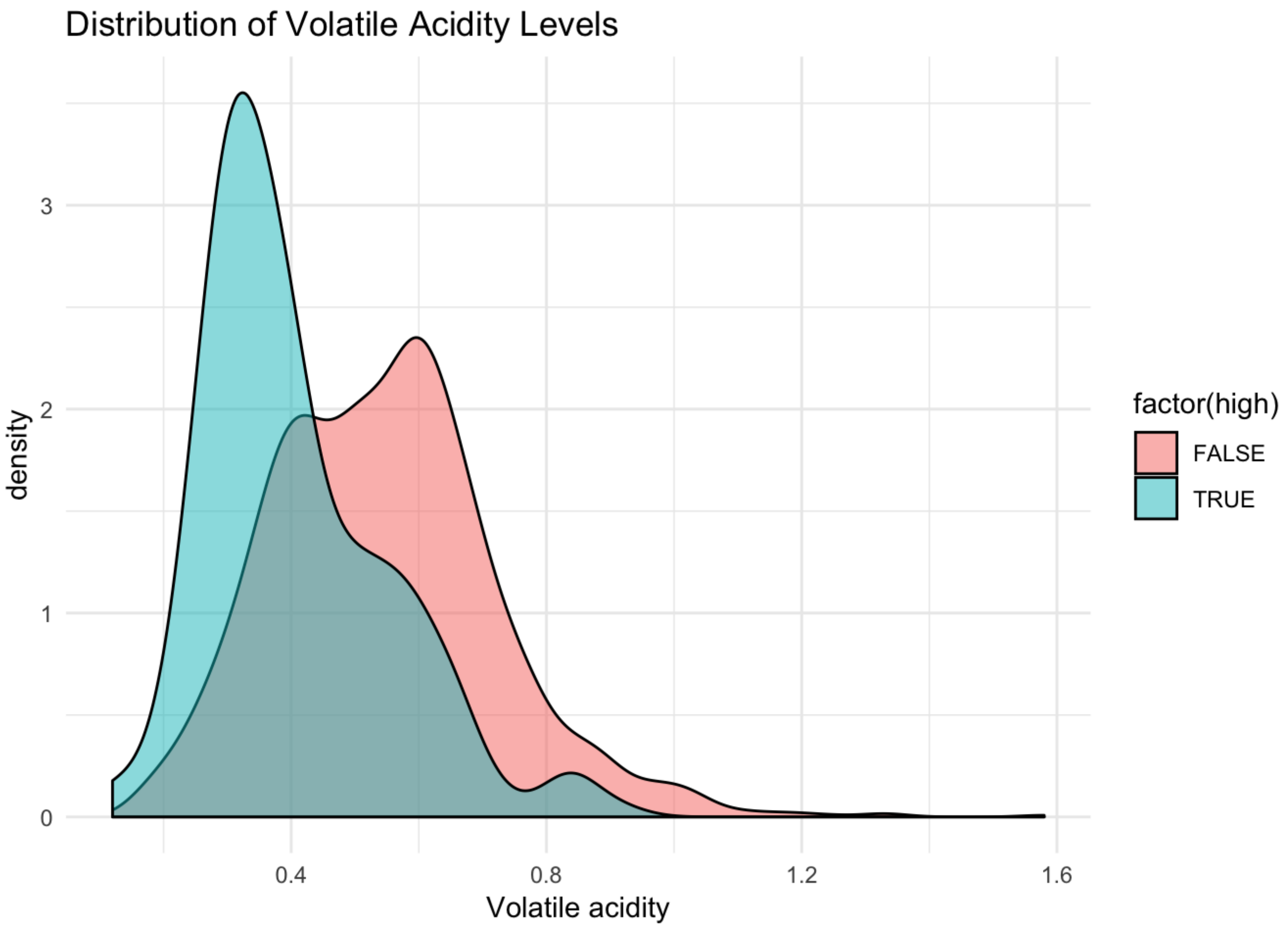
Analyzing influence of different parameters on wine quality

```
ggplot(wine,aes(x=fixed.acidity,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Fixed Acidity Level")+
  ggtitle("Distribution of Fixed Acidity Levels")+
  theme_minimal()
```

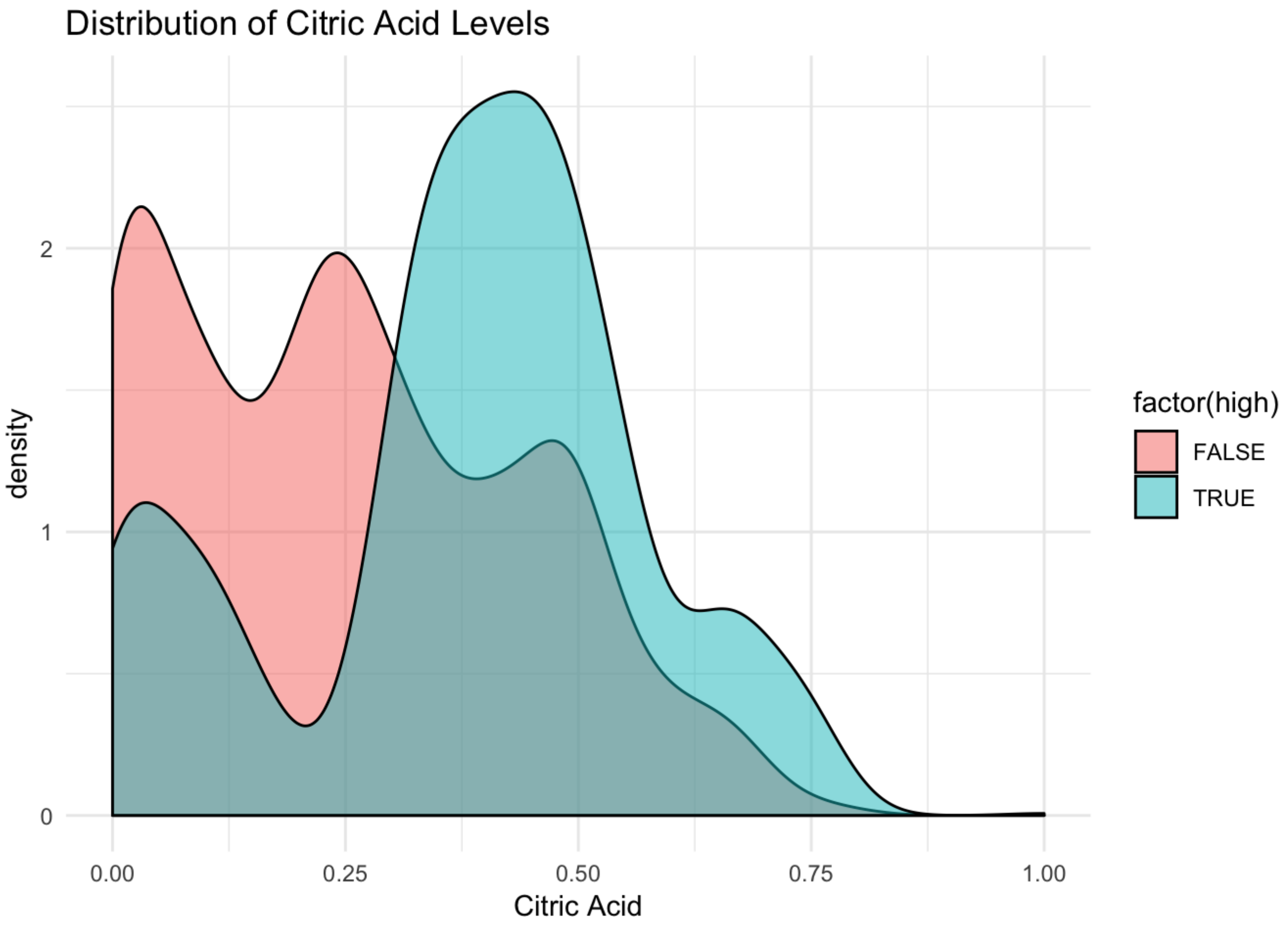



```
ggplot(wine,aes(x=alcohol,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Alcohol level")+
  ggtitle("Distribution of Alcohol Levels")+
  theme_minimal()
```

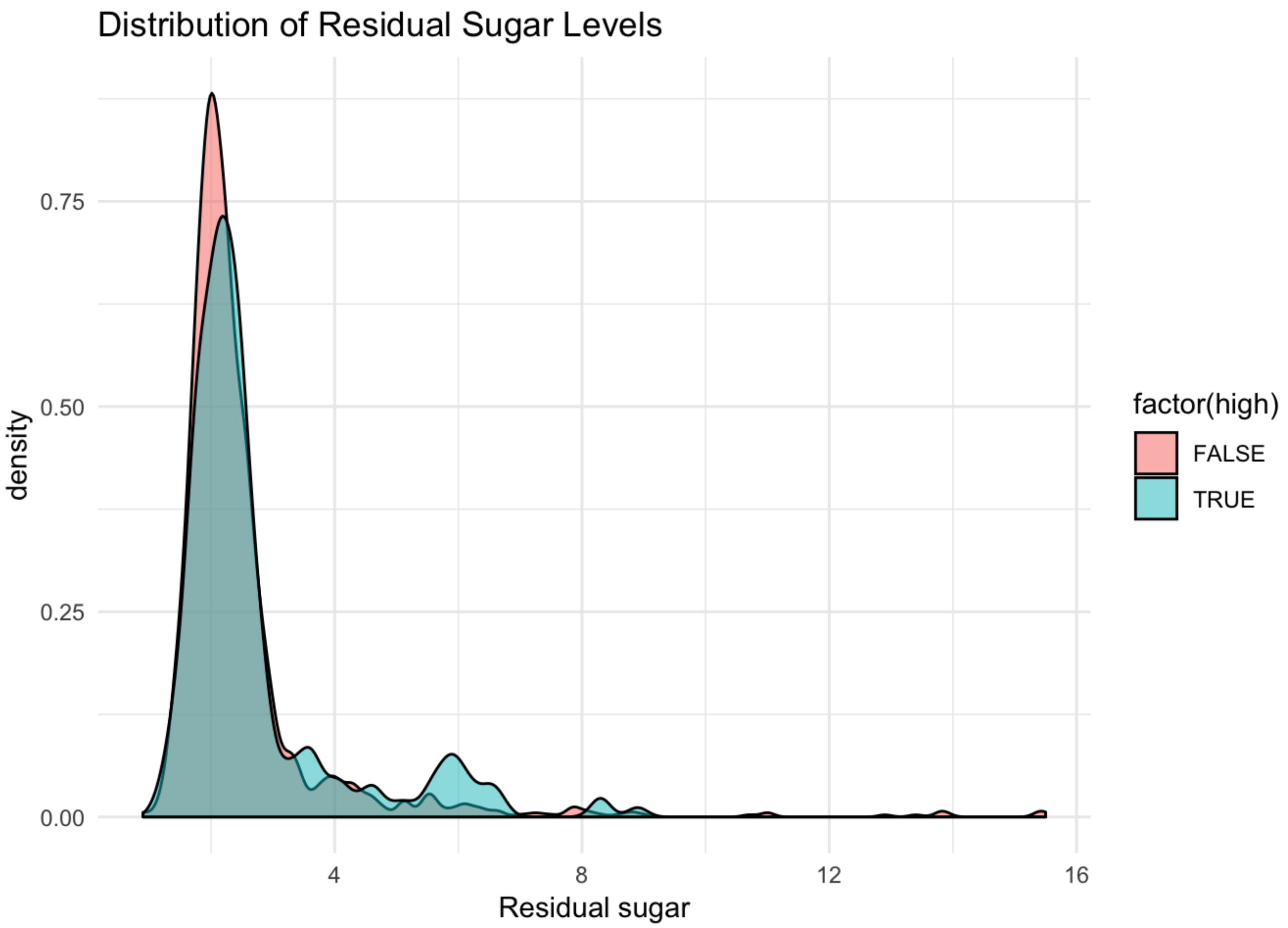
```
ggplot(wine,aes(x=volatile.acidity,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Volatile acidity")+
  ggtitle("Distribution of Volatile Acidity Levels")+
  theme_minimal()
```



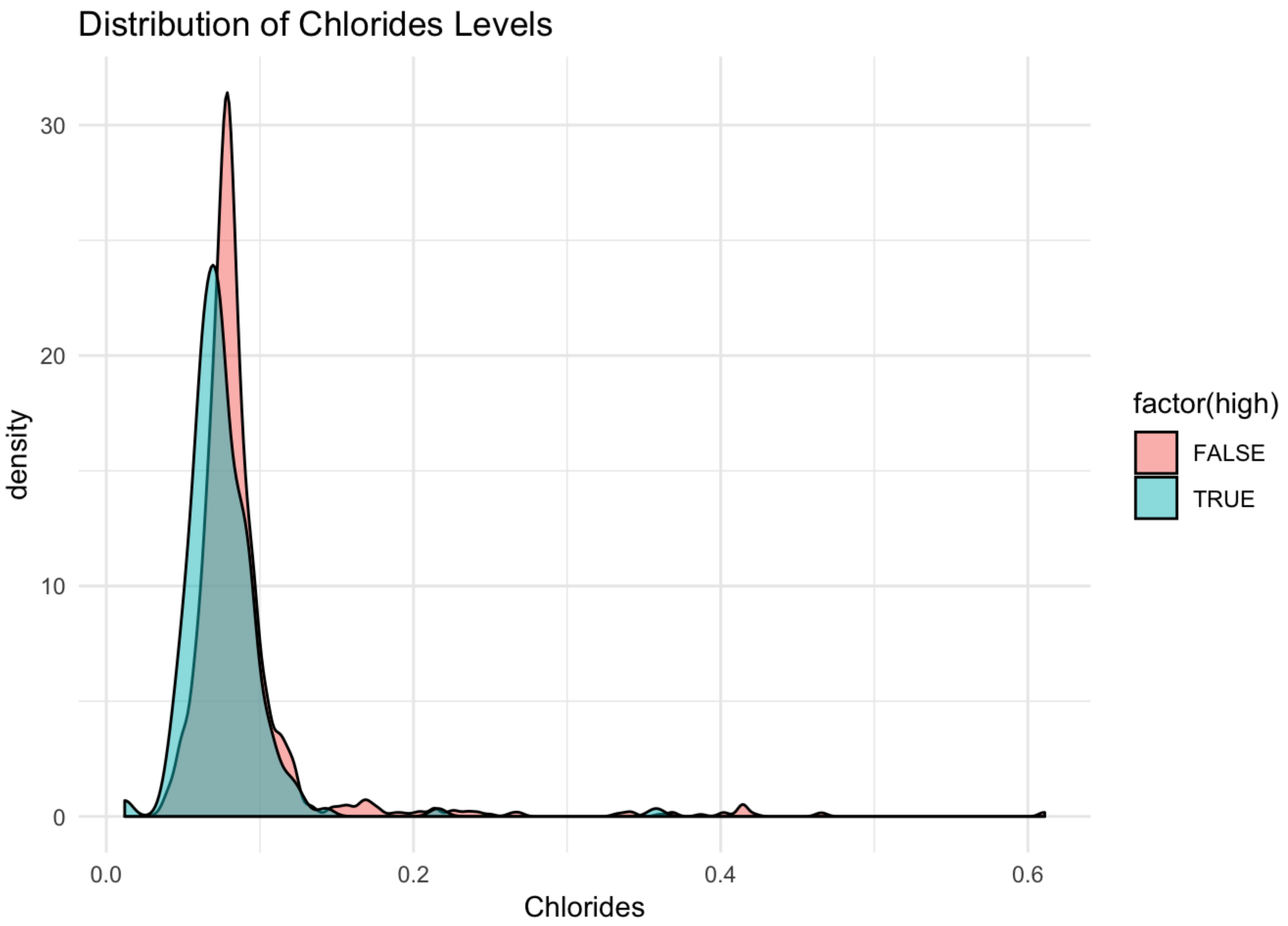

```
ggplot(wine,aes(x=citric.acid,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label =  "Citric Acid")+
  ggtitle("Distribution of Citric Acid Levels")+
  theme_minimal()
```



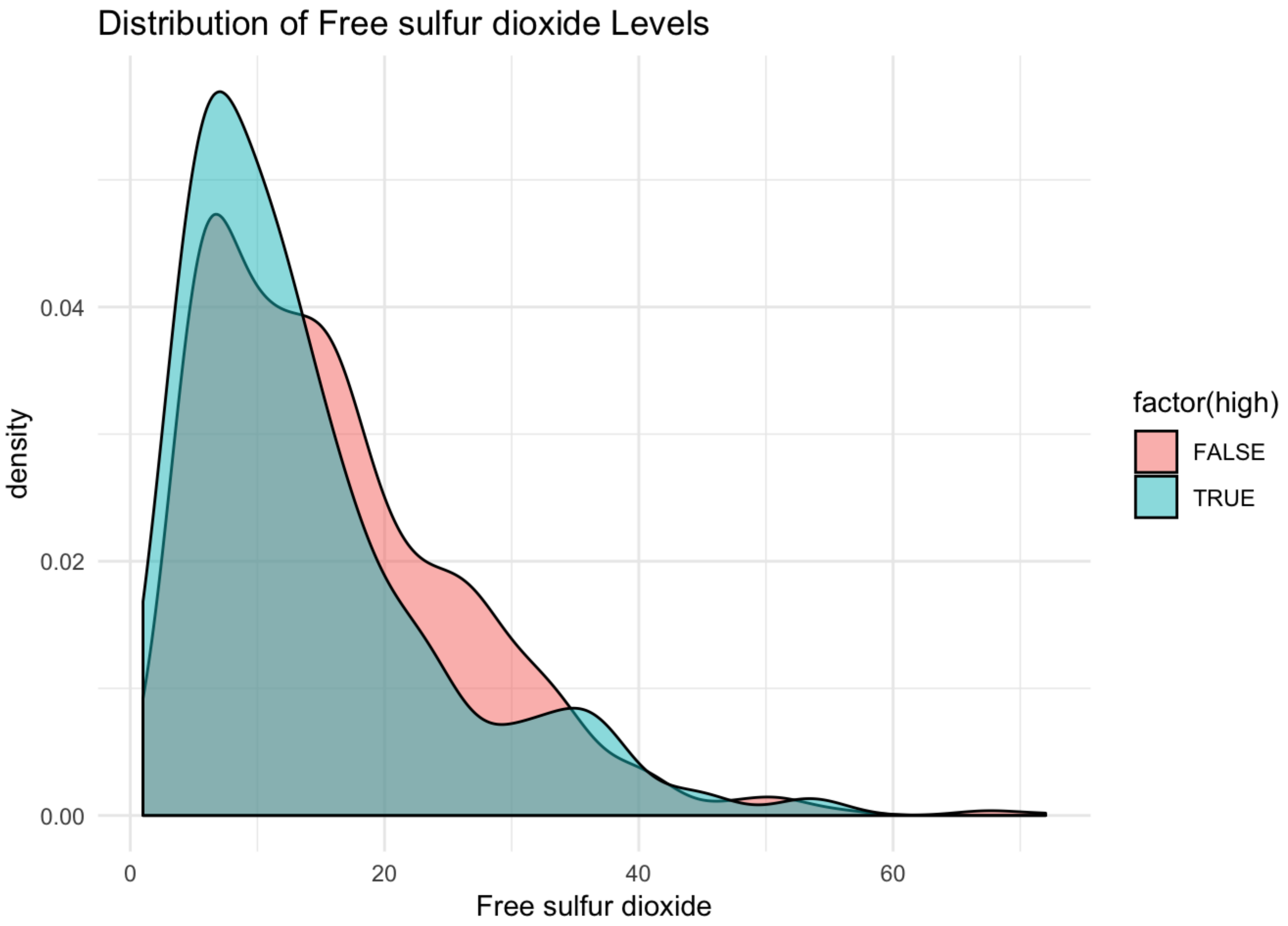

```
ggplot(wine,aes(x=residual.sugar,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Residual sugar")+
  ggtitle("Distribution of Residual Sugar Levels")+
  theme_minimal()
```



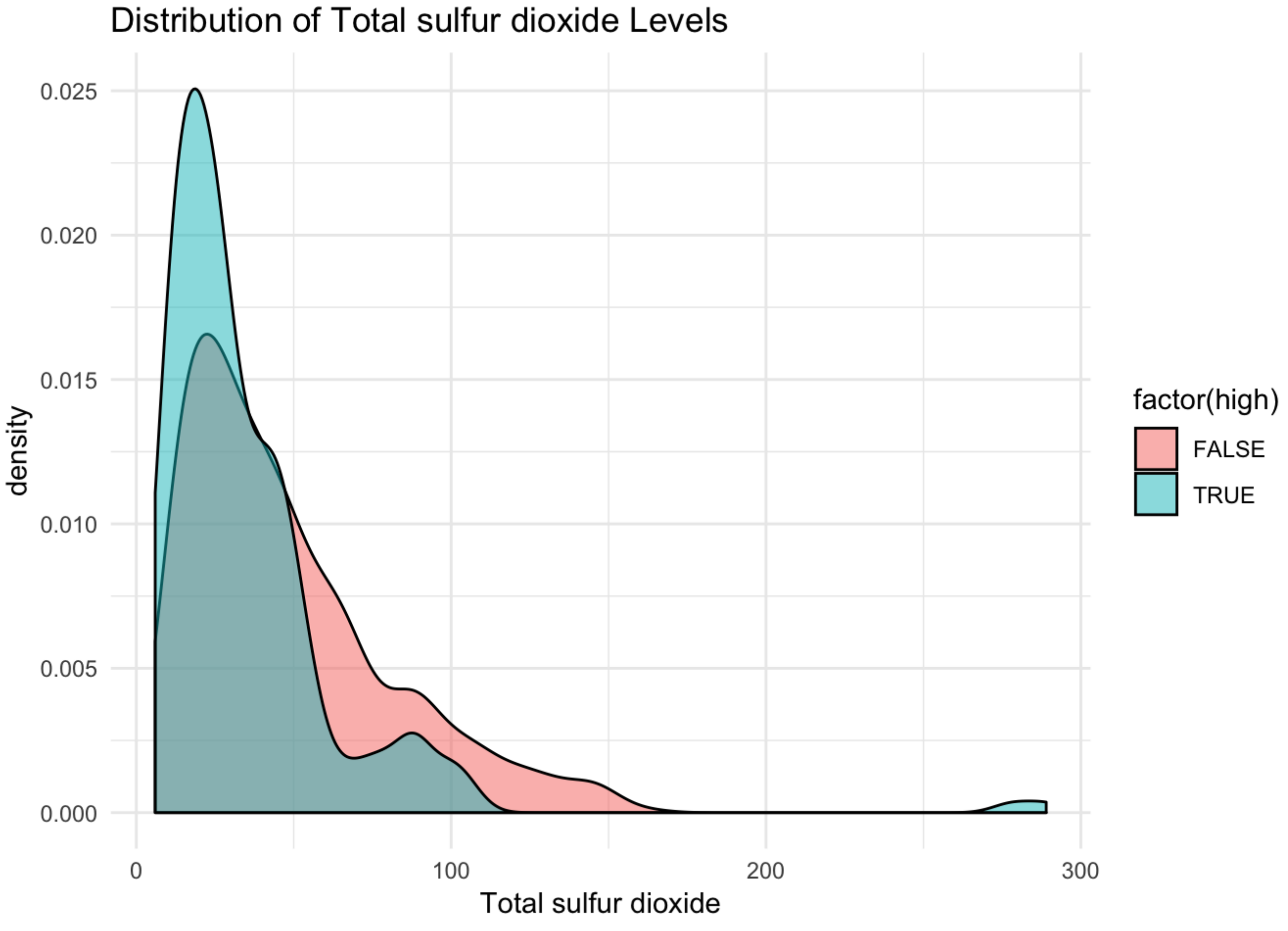
```
ggplot(wine,aes(x=chlorides,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Chlorides")+
  ggtitle("Distribution of Chlorides Levels")+
  theme_minimal()
```



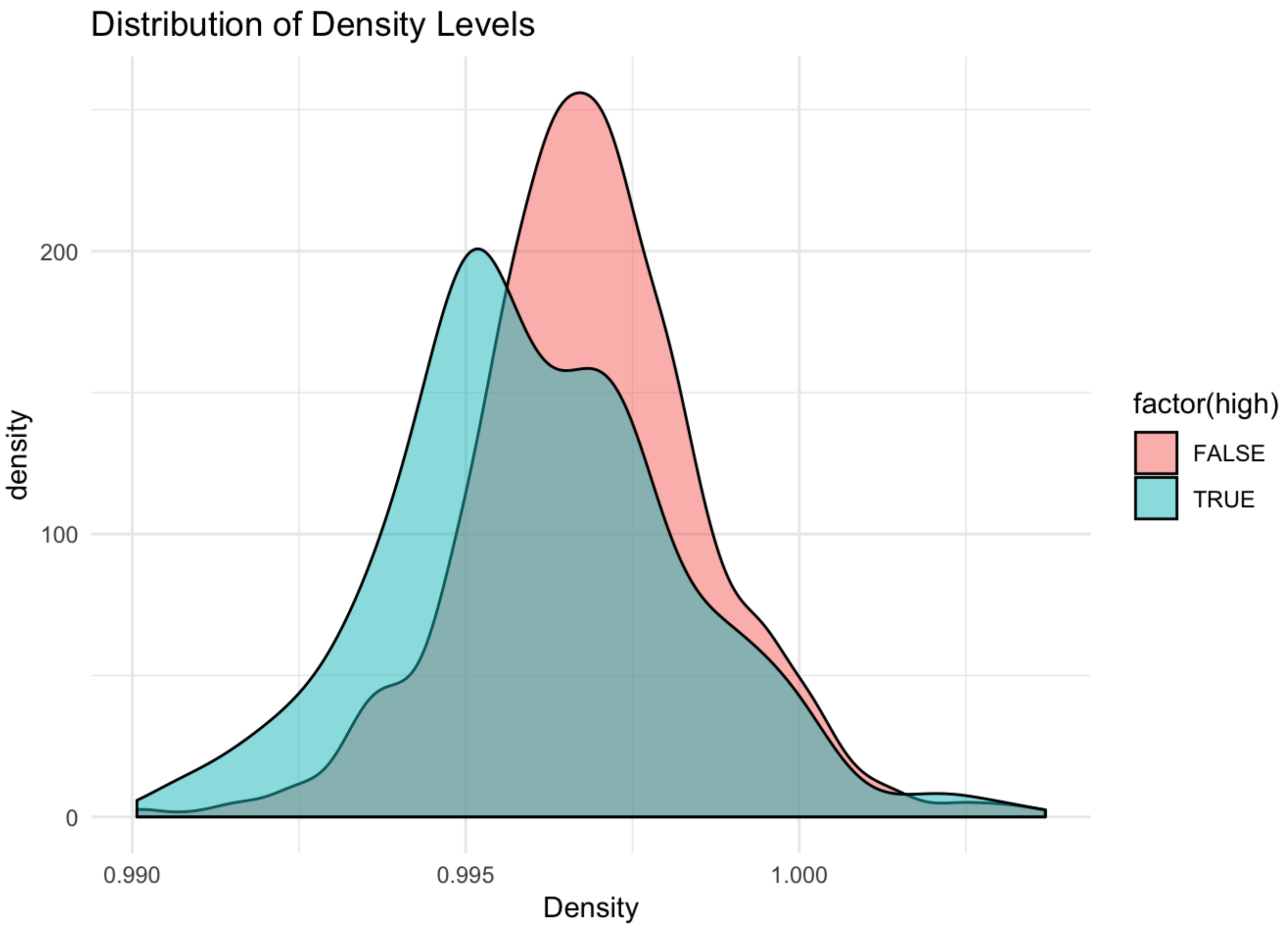

```
ggplot(wine,aes(x=free.sulfur.dioxide,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Free sulfur dioxide")+
  ggtitle("Distribution of Free sulfur dioxide Levels")+
  theme_minimal()
```



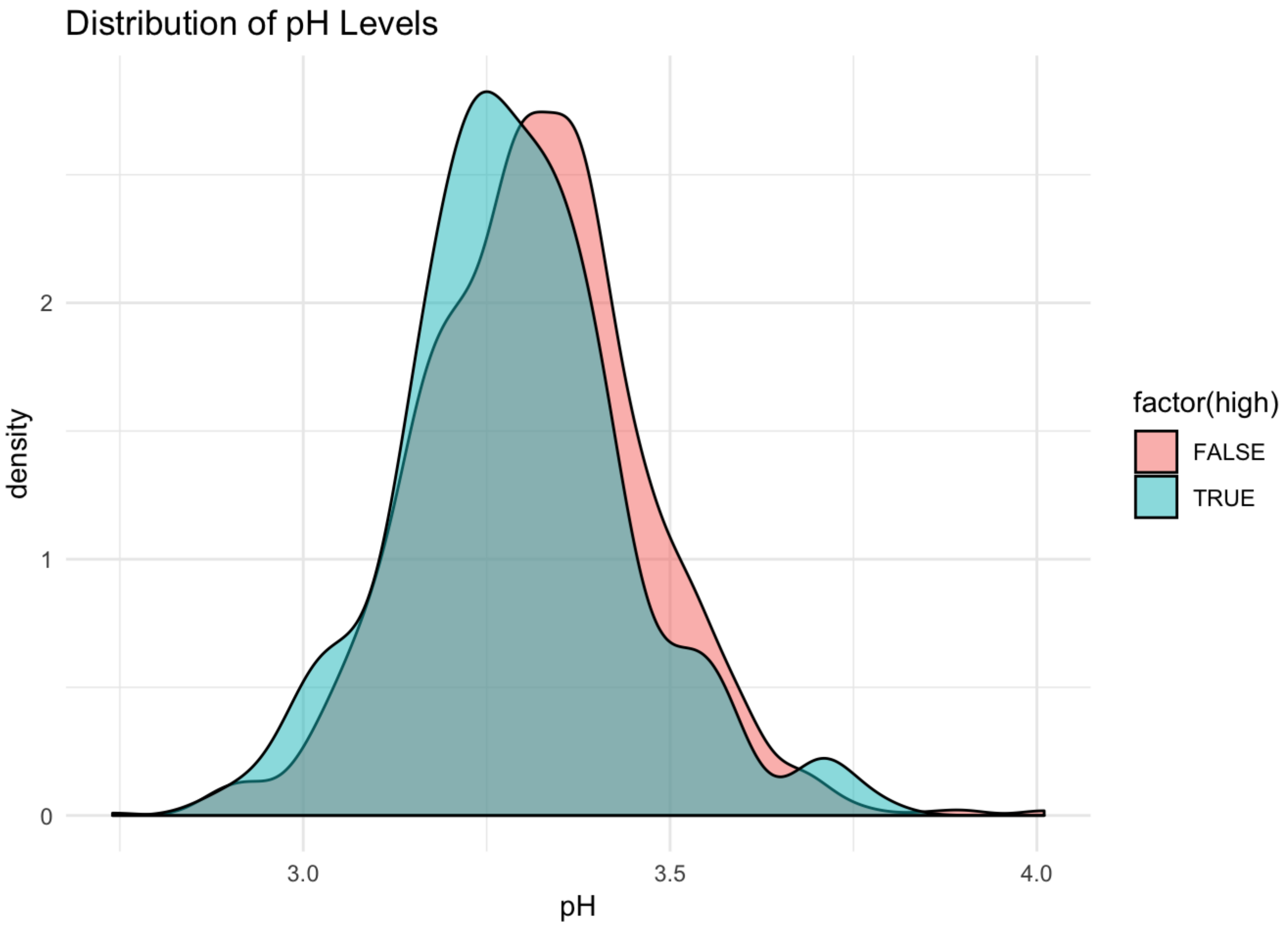

```
ggplot(wine,aes(x=total.sulfur.dioxide,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Total sulfur dioxide")+
  ggtitle("Distribution of Total sulfur dioxide Levels")+
  theme_minimal()
```



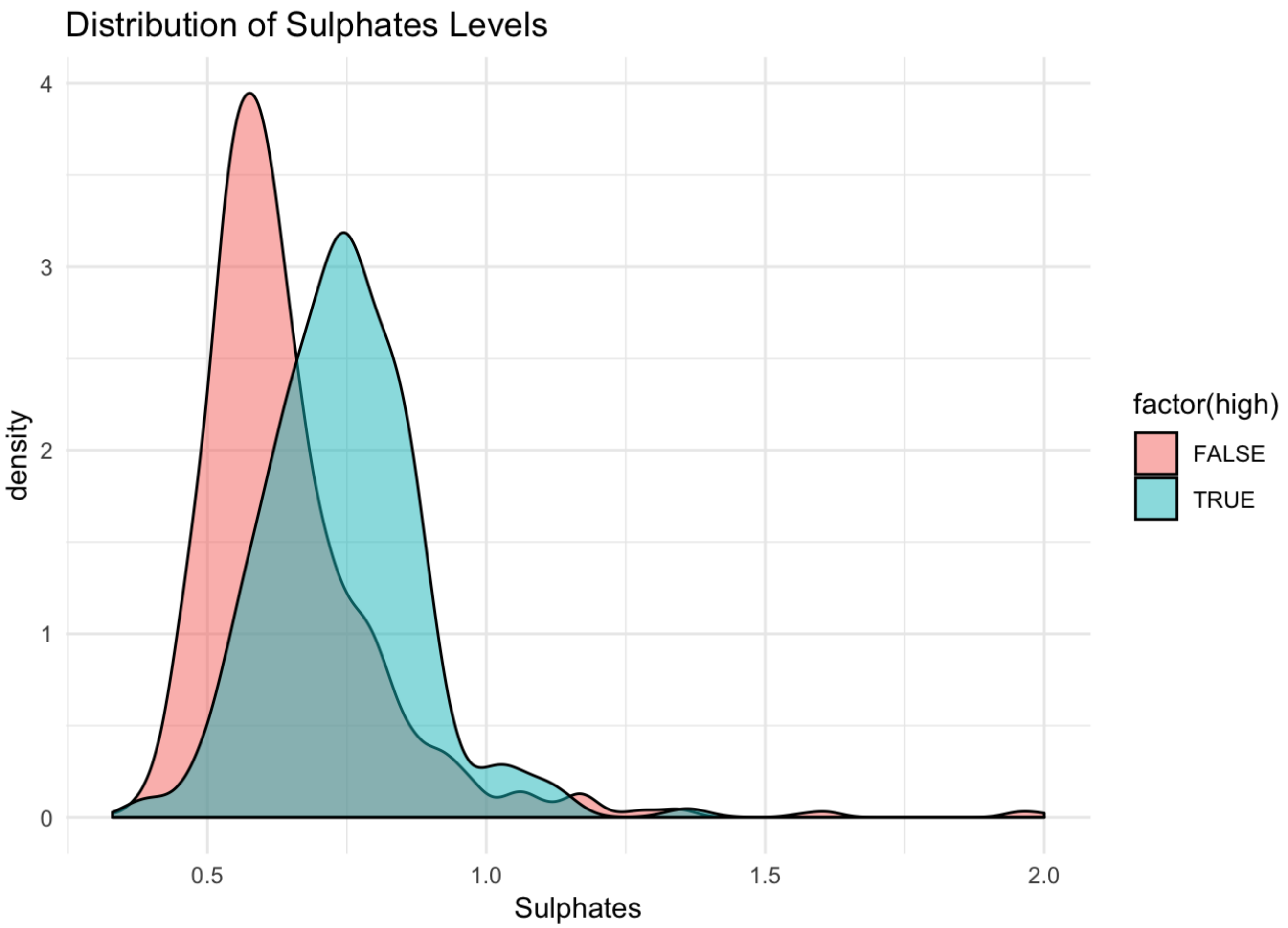
```
ggplot(wine,aes(x=density,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Density")+
  ggtitle("Distribution of Density Levels")+
  theme_minimal()
```




```
ggplot(wine,aes(x=pH,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "pH")+
  ggtitle("Distribution of pH Levels")+
  theme_minimal()
```




```
ggplot(wine,aes(x=sulphates,fill=factor(high)))+geom_density(alpha=0.5)+
  xlab(label = "Sulphates")+
  ggtitle("Distribution of Sulphates Levels")+
  theme_minimal()
```



Hypothesis testing

H_0 - Alcohol level and quality are assigned independently
H_1 - There is a dependence between the alcohol level and quality of the wine

Organazing data into dataframe

```
alcohols <- unique(wine$alcohol)
q.3 <- c()
q.4 <- c()
q.5 <- c()
q.6 <- c()
q.7 <- c()
q.8 <- c()
n.i <- c()
n.j <- c()



for (i in 1:length(unique(wine$alcohol))){
  q.3[i] <- 0
  q.4[i] <- 0
  q.5[i] <- 0
  q.6[i] <- 0
  q.7[i] <- 0
  q.8[i] <- 0
}

for (row in 1:nrow(wine)){
  index <- which(alcohols == wine[row, "alcohol"])
  q <- wine[row, "quality"]
  if (q == 3){
    q.3[index] <- q.3[index] + 1
  }
  else if (q == 4){
    q.4[index] <- q.4[index] + 1
  }
  else if (q == 5){
    q.5[index] <- q.5[index] + 1
  }
  else if (q == 6){
    q.6[index] <- q.6[index] + 1
  }
  else if (q == 7){
    q.7[index] <- q.7[index] + 1
  }
  else if (q == 8){
    q.8[index] <- q.8[index] + 1
  }

}

df1 <- data.frame(q.3, q.4, q.5, q.6, q.7, q.8, row.names=alcohols)
```

I used chi-squared statistics for testing wine quality and wine alcohol level fot statistical independence. The value of test statistics:

$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{O_{ij} - E_{ij}}{E_{ij}}$

Then, we compare it with $\chi^2_{(c-1)(r-1)}$ distribution and calculate p-value.

```
chisq.test(df1)
```

```
## Warning in chisq.test(df1): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  df1
## X-squared = 1124.5, df = 320, p-value < 2.2e-16
```

Conclusion

p-value is almost zero, so we reject H_0. The quality of red wine and its alcohol level are dependent.

Performing loglinear regression

Firstly I desided to find the correlation between different factors and wine quality:

```
correlation <- cor(wine)
print(correlation[,"quality"])
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##           0.12405165          -0.39055778           0.22637251
##       residual.sugar            chlorides  free.sulfur.dioxide
##           0.01373164          -0.12890656          -0.05065606
## total.sulfur.dioxide              density                   pH
##          -0.18510029          -0.17491923          -0.05773139
##            sulphates              alcohol              quality
##           0.25139708           0.47616632           1.00000000
##                 high
##           0.71019625
```

Quality he most correlates with volatile.acidity, citric.acid, total.sulfur.dioxide, density, sulphates, alcohol

Model training

```
wine$high <- as.factor(wine$high)
train <- wine[1:1280, ]
test <- wine[1281:1599,]
```

There aren't any correlation coefficient equals +-1, so we can't build a strong linear model. Let's try performing binomial logistic regression to predict whether the wine has a high quality. It works as follows:
1. Building a model for finding probability that wine is good
$P(y = 1|x) = f(z)$,
$z = \theta^T x = \theta_0 + \theta_1 x_1 + \ldots \theta_n x_n$
$f(z) = \frac{1}{1+e^{-z}}$,
$\theta_0, \ldots \theta_n$ are regression coefficients
Actually, building a model mean specifying θ. 2. For every wine in testing dataset, we estimate the probability that it has quality via model. If P(y = 1|x) > 0.5, classify the wine as good.
3. Comparing predicted quality with actual data and calculating the accuracy of the model as relative frequencies:
Accuracy = $\frac{\text{True positive} + \text{True Negative}}{\text{True positive} + \text{False positive} + \text{True negative} + \text{False negative}}$

```
model <- glm(high ~ alcohol + total.sulfur.dioxide +  citric.acid + volatile.acidity + density + sulpha
tes ,
                      data = train, family=binomial(link='logit'))
```

Model testing

```
prediction <- predict.glm(model, newdata = test, type = 'response')
prediction <- ifelse(prediction > 0.5,TRUE,FALSE)
result <- data.frame(prediction)
result$prediction <- as.factor(result$prediction)
confusionMatrix(result$prediction, test$high)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction FALSE TRUE
##      FALSE   288   19
##      TRUE      7    5
## 
##                Accuracy : 0.9185
##                  95% CI : (0.8829, 0.9461)
##     No Information Rate : 0.9248
##     P-Value [Acc > NIR] : 0.70958
## 
##                   Kappa : 0.2396
##  Mcnemar's Test P-Value : 0.03098
## 
##             Sensitivity : 0.9763
##             Specificity : 0.2083
##          Pos Pred Value : 0.9381
##          Neg Pred Value : 0.4167
##              Prevalence : 0.9248
##          Detection Rate : 0.9028
##    Detection Prevalence : 0.9624
##       Balanced Accuracy : 0.5923
## 
##        'Positive' Class : FALSE
## 
```

Conclusion

The accuracy is high, so model fits the data and can be used to predict wine quality!