

# CMTH642 Assignment 3

*Geoffrey Clark*

*April 11, 2018*

## Data Prep

Preparation: The dataset is related to white Portuguese “Vinh o Verde” wine. For more info: <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>

Import to R the following file: <http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv>

```
# Download remote content
# wine.df <- read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv")
wine.df <- read.csv("winequality-white.csv", header=T, sep=";")
```

## QUESTIONS

1. Check data characteristics. Is there missing data?

```
sum(is.na(wine.df)) #0
```

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

```
str(wine.df)
```

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

```
sapply(wine.df, function(x) sum(is.na(x))) # 0 NAs
```

```
##      fixed.acidity    volatile.acidity    citric.acid
##           0              0              0
##      residual.sugar    chlorides    free.sulfur.dioxide
##           0              0              0
##      total.sulfur.dioxide    density    pH
##           0              0              0
##      sulphates    alcohol    quality
##           0              0              0
```

```
sapply(wine.df, function(x) sum(is.null(x))) # 0 NULLs
```

```
##      fixed.acidity    volatile.acidity    citric.acid
##              0              0              0
##      residual.sugar    chlorides    free.sulfur.dioxide
##              0              0              0
## total.sulfur.dioxide    density    pH
##              0              0              0
##      sulphates    alcohol    quality
##              0              0              0
```

## 2. What is the correlation between the attributes other than wine quality?

```
# For Visualizing Correlation:
# install.packages("corrplot")
library(corrplot)
```

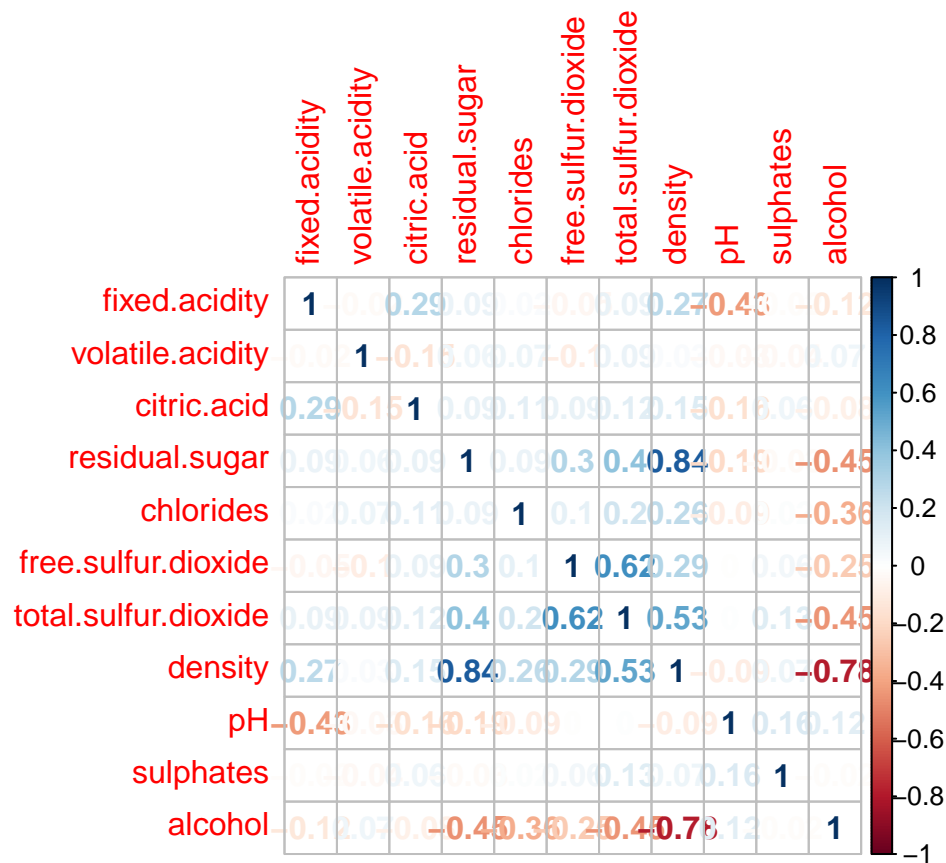
```
## corrplot 0.84 loaded
```

```
(wine.cor <- cor(wine.df[, -12]))
```

```
##      fixed.acidity    volatile.acidity    citric.acid
## fixed.acidity      1.00000000    -0.02269729    0.28918070
## volatile.acidity    -0.02269729      1.00000000   -0.14947181
## citric.acid         0.28918070    -0.14947181    1.00000000
## residual.sugar      0.08902070     0.06428606    0.09421162
## chlorides           0.02308564     0.07051157    0.11436445
## free.sulfur.dioxide -0.04939586    -0.09701194    0.09407722
## total.sulfur.dioxide 0.09106976     0.08926050    0.12113080
## density             0.26533101     0.02711385    0.14950257
## pH                 -0.42585829    -0.03191537   -0.16374821
## sulphates          -0.01714299    -0.03572815    0.06233094
## alcohol            -0.12088112     0.06771794   -0.07572873
##      residual.sugar    chlorides    free.sulfur.dioxide
## fixed.acidity      0.08902070    0.02308564    -0.0493958591
## volatile.acidity    0.06428606    0.07051157    -0.0970119393
## citric.acid         0.09421162    0.11436445     0.0940772210
## residual.sugar      1.00000000    0.08868454     0.2990983537
## chlorides           0.08868454    1.00000000     0.1013923521
## free.sulfur.dioxide 0.29909835    0.10139235     1.0000000000
## total.sulfur.dioxide 0.40143931    0.19891030     0.6155009650
## density             0.83896645    0.25721132     0.2942104109
## pH                 -0.19413345   -0.09043946    -0.0006177961
## sulphates          -0.02666437    0.01676288     0.0592172458
## alcohol            -0.45063122   -0.36018871    -0.2501039415
##      total.sulfur.dioxide    density    pH
## fixed.acidity      0.091069756    0.26533101   -0.4258582910
## volatile.acidity    0.089260504    0.02711385   -0.0319153683
## citric.acid         0.121130798    0.14950257   -0.1637482114
## residual.sugar      0.401439311    0.83896645   -0.1941334540
## chlorides           0.198910300    0.25721132   -0.0904394560
## free.sulfur.dioxide 0.615500965    0.29421041   -0.0006177961
## total.sulfur.dioxide 1.000000000    0.52988132    0.0023209718
## density             0.529881324    1.00000000   -0.0935914935
## pH                 0.002320972   -0.09359149    1.0000000000
## sulphates           0.134562367    0.07449315    0.1559514973
```

```
## alcohol                -0.448892102 -0.78013762  0.1214320987
##                sulphates      alcohol
## fixed.acidity      -0.01714299 -0.12088112
## volatile.acidity   -0.03572815  0.06771794
## citric.acid         0.06233094 -0.07572873
## residual.sugar     -0.02666437 -0.45063122
## chlorides           0.01676288 -0.36018871
## free.sulfur.dioxide 0.05921725 -0.25010394
## total.sulfur.dioxide 0.13456237 -0.44889210
## density             0.07449315 -0.78013762
## pH                  0.15595150  0.12143210
## sulphates           1.00000000 -0.01743277
## alcohol             -0.01743277  1.00000000
```

```
corrplot(wine.cor, method="number") # I did this because I appreciate the visual
```



```
# highest correlation:
```

```
# density & residual.sugar: 0.84
```

```
# density & alcohol: -0.78
```

```
# I found a cool SO thread on how to do this programmatically:
```

```
# https://stackoverflow.com/questions/7074246/show-correlations-as-an-ordered-list-not-as-a-large-matrix
```

```
# as.data.frame(as.table(wine.cor))
```

```
# install.packages("reshape")
```

```
library(reshape) # includes melt()
```

```
wine.cor.list <- wine.cor
wine.cor.list[wine.cor.list == 1] <- NA
wine.cor.list <- na.omit(melt(wine.cor.list))
wine.cor.list[order(-abs(wine.cor.list$value)),]
```

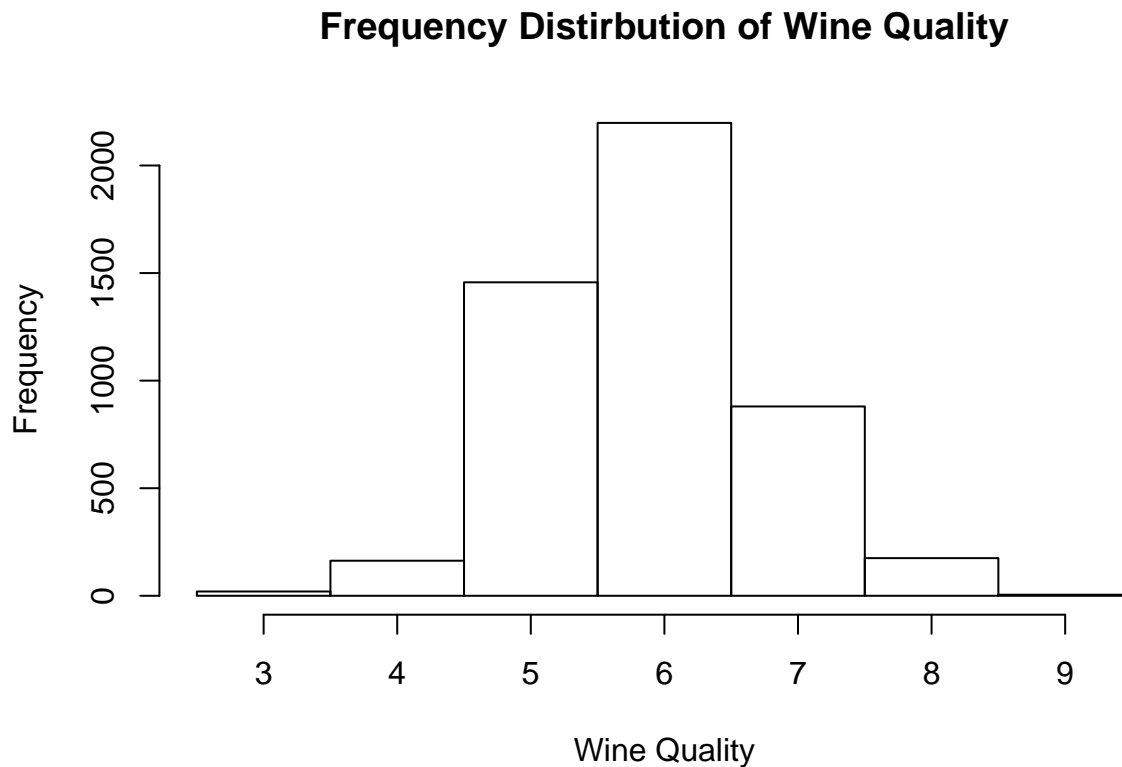
| ##     | X1                   | X2                   | value         |
|--------|----------------------|----------------------|---------------|
| ## 41  | density              | residual.sugar       | 0.8389664549  |
| ## 81  | residual.sugar       | density              | 0.8389664549  |
| ## 88  | alcohol              | density              | -0.7801376214 |
| ## 118 | density              | alcohol              | -0.7801376214 |
| ## 62  | total.sulfur.dioxide | free.sulfur.dioxide  | 0.6155009650  |
| ## 72  | free.sulfur.dioxide  | total.sulfur.dioxide | 0.6155009650  |
| ## 74  | density              | total.sulfur.dioxide | 0.5298813239  |
| ## 84  | total.sulfur.dioxide | density              | 0.5298813239  |
| ## 44  | alcohol              | residual.sugar       | -0.4506312220 |
| ## 114 | residual.sugar       | alcohol              | -0.4506312220 |
| ## 77  | alcohol              | total.sulfur.dioxide | -0.4488921021 |
| ## 117 | total.sulfur.dioxide | alcohol              | -0.4488921021 |
| ## 9   | pH                   | fixed.acidity        | -0.4258582910 |
| ## 89  | fixed.acidity        | pH                   | -0.4258582910 |
| ## 40  | total.sulfur.dioxide | residual.sugar       | 0.4014393112  |
| ## 70  | residual.sugar       | total.sulfur.dioxide | 0.4014393112  |
| ## 55  | alcohol              | chlorides            | -0.3601887121 |
| ## 115 | chlorides            | alcohol              | -0.3601887121 |
| ## 39  | free.sulfur.dioxide  | residual.sugar       | 0.2990983537  |
| ## 59  | residual.sugar       | free.sulfur.dioxide  | 0.2990983537  |
| ## 63  | density              | free.sulfur.dioxide  | 0.2942104109  |
| ## 83  | free.sulfur.dioxide  | density              | 0.2942104109  |
| ## 3   | citric.acid          | fixed.acidity        | 0.2891806977  |
| ## 23  | fixed.acidity        | citric.acid          | 0.2891806977  |
| ## 8   | density              | fixed.acidity        | 0.2653310138  |
| ## 78  | fixed.acidity        | density              | 0.2653310138  |
| ## 52  | density              | chlorides            | 0.2572113204  |
| ## 82  | chlorides            | density              | 0.2572113204  |
| ## 66  | alcohol              | free.sulfur.dioxide  | -0.2501039415 |
| ## 116 | free.sulfur.dioxide  | alcohol              | -0.2501039415 |
| ## 51  | total.sulfur.dioxide | chlorides            | 0.1989102996  |
| ## 71  | chlorides            | total.sulfur.dioxide | 0.1989102996  |
| ## 42  | pH                   | residual.sugar       | -0.1941334540 |
| ## 92  | residual.sugar       | pH                   | -0.1941334540 |
| ## 31  | pH                   | citric.acid          | -0.1637482114 |
| ## 91  | citric.acid          | pH                   | -0.1637482114 |
| ## 98  | sulphates            | pH                   | 0.1559514973  |
| ## 108 | pH                   | sulphates            | 0.1559514973  |
| ## 30  | density              | citric.acid          | 0.1495025706  |
| ## 80  | citric.acid          | density              | 0.1495025706  |
| ## 14  | citric.acid          | volatile.acidity     | -0.1494718106 |
| ## 24  | volatile.acidity     | citric.acid          | -0.1494718106 |
| ## 76  | sulphates            | total.sulfur.dioxide | 0.1345623669  |
| ## 106 | total.sulfur.dioxide | sulphates            | 0.1345623669  |
| ## 99  | alcohol              | pH                   | 0.1214320987  |
| ## 119 | pH                   | alcohol              | 0.1214320987  |
| ## 29  | total.sulfur.dioxide | citric.acid          | 0.1211307977  |

|        |                      |                      |               |
|--------|----------------------|----------------------|---------------|
| ## 69  | citric.acid          | total.sulfur.dioxide | 0.1211307977  |
| ## 11  | alcohol              | fixed.acidity        | -0.1208811232 |
| ## 111 | fixed.acidity        | alcohol              | -0.1208811232 |
| ## 27  | chlorides            | citric.acid          | 0.1143644484  |
| ## 47  | citric.acid          | chlorides            | 0.1143644484  |
| ## 50  | free.sulfur.dioxide  | chlorides            | 0.1013923521  |
| ## 60  | chlorides            | free.sulfur.dioxide  | 0.1013923521  |
| ## 17  | free.sulfur.dioxide  | volatile.acidity     | -0.0970119393 |
| ## 57  | volatile.acidity     | free.sulfur.dioxide  | -0.0970119393 |
| ## 26  | residual.sugar       | citric.acid          | 0.0942116243  |
| ## 36  | citric.acid          | residual.sugar       | 0.0942116243  |
| ## 28  | free.sulfur.dioxide  | citric.acid          | 0.0940772210  |
| ## 58  | citric.acid          | free.sulfur.dioxide  | 0.0940772210  |
| ## 86  | pH                   | density              | -0.0935914935 |
| ## 96  | density              | pH                   | -0.0935914935 |
| ## 7   | total.sulfur.dioxide | fixed.acidity        | 0.0910697562  |
| ## 67  | fixed.acidity        | total.sulfur.dioxide | 0.0910697562  |
| ## 53  | pH                   | chlorides            | -0.0904394560 |
| ## 93  | chlorides            | pH                   | -0.0904394560 |
| ## 18  | total.sulfur.dioxide | volatile.acidity     | 0.0892605036  |
| ## 68  | volatile.acidity     | total.sulfur.dioxide | 0.0892605036  |
| ## 4   | residual.sugar       | fixed.acidity        | 0.0890207014  |
| ## 34  | fixed.acidity        | residual.sugar       | 0.0890207014  |
| ## 38  | chlorides            | residual.sugar       | 0.0886845359  |
| ## 48  | residual.sugar       | chlorides            | 0.0886845359  |
| ## 33  | alcohol              | citric.acid          | -0.0757287301 |
| ## 113 | citric.acid          | alcohol              | -0.0757287301 |
| ## 87  | sulphates            | density              | 0.0744931485  |
| ## 107 | density              | sulphates            | 0.0744931485  |
| ## 16  | chlorides            | volatile.acidity     | 0.0705115715  |
| ## 46  | volatile.acidity     | chlorides            | 0.0705115715  |
| ## 22  | alcohol              | volatile.acidity     | 0.0677179428  |
| ## 112 | volatile.acidity     | alcohol              | 0.0677179428  |
| ## 15  | residual.sugar       | volatile.acidity     | 0.0642860601  |
| ## 35  | volatile.acidity     | residual.sugar       | 0.0642860601  |
| ## 32  | sulphates            | citric.acid          | 0.0623309403  |
| ## 102 | citric.acid          | sulphates            | 0.0623309403  |
| ## 65  | sulphates            | free.sulfur.dioxide  | 0.0592172458  |
| ## 105 | free.sulfur.dioxide  | sulphates            | 0.0592172458  |
| ## 6   | free.sulfur.dioxide  | fixed.acidity        | -0.0493958591 |
| ## 56  | fixed.acidity        | free.sulfur.dioxide  | -0.0493958591 |
| ## 21  | sulphates            | volatile.acidity     | -0.0357281469 |
| ## 101 | volatile.acidity     | sulphates            | -0.0357281469 |
| ## 20  | pH                   | volatile.acidity     | -0.0319153683 |
| ## 90  | volatile.acidity     | pH                   | -0.0319153683 |
| ## 19  | density              | volatile.acidity     | 0.0271138455  |
| ## 79  | volatile.acidity     | density              | 0.0271138455  |
| ## 43  | sulphates            | residual.sugar       | -0.0266643659 |
| ## 103 | residual.sugar       | sulphates            | -0.0266643659 |
| ## 5   | chlorides            | fixed.acidity        | 0.0230856437  |
| ## 45  | fixed.acidity        | chlorides            | 0.0230856437  |
| ## 2   | volatile.acidity     | fixed.acidity        | -0.0226972901 |
| ## 12  | fixed.acidity        | volatile.acidity     | -0.0226972901 |
| ## 110 | alcohol              | sulphates            | -0.0174327719 |

```
## 120      sulphates      alcohol -0.0174327719
## 10      sulphates      fixed.acidity -0.0171429850
## 100     fixed.acidity      sulphates -0.0171429850
## 54      sulphates      chlorides 0.0167628837
## 104     chlorides      sulphates 0.0167628837
## 75      pH total.sulfur.dioxide 0.0023209718
## 95     total.sulfur.dioxide      pH 0.0023209718
## 64      pH free.sulfur.dioxide -0.0006177961
## 94     free.sulfur.dioxide      pH -0.0006177961
```

3. Graph the frequency distribution of wine quality .

```
hist(wine.df$quality, freq=T, breaks=seq(2.5,9.5,1), main="Frequency Distirbution of Wine Quality", xlab="Wine Quality", ylab="Frequency")
```



4. Reduce the levels of rating for quality to three levels as high, medium and low.

```
wine.df$quality <- cut(wine.df$quality, 3, labels=c("Low","Medium","High"))
table(wine.df$quality)
```

```
##
##   Low Medium   High
## 1640  3078   180
```

```
plot(wine.df$quality, main="Distribution of Wine Quality Levels", xlab="Wine Quality Level", ylab="Frequency")
```



5. Normalize the data set.

```
# scale() returns a matrix. Also requires numerical values
# (exclude class attribute (factor): column 12)
wine.df.norm <- as.data.frame(scale(wine.df[, -12], center=T, scale=T))
wine.df.norm <- cbind(wine.df.norm, wine.df[, 12]) # Re-add class attribute
names(wine.df.norm)[12] <- names(wine.df)[12] # Re-name the class attribute
```

6. Divide the data to training and testing groups.

```
# I wanted replicable results for discussion. Remove below call to set.seed() to
# re-introduce pseudo random numbers in division of test & training set.
set.seed(42)

wine.train_index <- sample(1:nrow(wine.df.norm), 0.7*nrow(wine.df.norm))
wine.train <- wine.df.norm[wine.train_index,]
wine.test <- wine.df.norm[-wine.train_index,]
(nrow(wine.df) == (nrow(wine.train) + nrow(wine.test))) # Did we include all observations?

## [1] TRUE
```

7. Use the KNN algorithm to predict the quality of wine using its attributes.

```
library(class) # needed for knn()

##
## Attaching package: 'class'
## The following object is masked from 'package:reshape':
##
##      condense

# I chose to reference the test group directly in the following calls to knn().
# wine.train_labels <- wine.train$quality
# wine.test_labels <- wine.test$quality
#
# Note that subsetting & removing the attribute quality (wine.test[, -12]):
# done to separate class attribute from training & test sets.
# Explicitly defined as cl parameter.

wine.knn <- list(k3=factor(), k5=factor(), k7=factor())
wine.knn$k3 <- knn(train=wine.train[, -12], test=wine.test[, -12], cl=wine.train$quality, k=3, prob=T)
wine.knn$k5 <- knn(train=wine.train[, -12], test=wine.test[, -12], cl=wine.train$quality, k=5)
wine.knn$k15 <- knn(train=wine.train[, -12], test=wine.test[, -12], cl=wine.train$quality, k=15)
```

8. Evaluate the model performance .

```
library(gmodels) # Necessary for CrossTable()

# I prefer actual results along the top of the table: I find it easier to read
# This seems pretty standard, and is likewise on Wikipedia: https://en.wikipedia.org/wiki/Confusion\_matrix
CrossTable(x=wine.knn$k3, y=wine.test$quality, prop.chisq = F)
```

```
##
##
##      Cell Contents
## |-----|
## |                      N |
## |          N / Row Total |
## |          N / Col Total |
## |          N / Table Total |
## |-----|
##
##
## Total Observations in Table:  1470
##
##
##      | wine.test$quality
## wine.knn$k3 |      Low |      Medium |      High | Row Total |
## -----|-----|-----|-----|
##      Low |      305 |      161 |         4 |      470 |
##      |      0.649 |      0.343 |      0.009 |      0.320 |
##      |      0.642 |      0.171 |      0.074 |      |
##      |      0.207 |      0.110 |      0.003 |      |
```



```
## -----|-----|-----|-----|-----|
##      Medium |      167 |      766 |      40 |      973 |
##      |      0.172 |      0.787 |      0.041 |      0.662 |
##      |      0.352 |      0.814 |      0.741 |      |
##      |      0.114 |      0.521 |      0.027 |      |
## -----|-----|-----|-----|
##      High |      3 |      14 |      10 |      27 |
##      |      0.111 |      0.519 |      0.370 |      0.018 |
##      |      0.006 |      0.015 |      0.185 |      |
##      |      0.002 |      0.010 |      0.007 |      |
## -----|-----|-----|-----|
## Column Total |      475 |      941 |      54 |      1470 |
##      |      0.323 |      0.640 |      0.037 |      |
## -----|-----|-----|-----|
##
##
```

```
CrossTable(x=wine.knn$k5, y=wine.test$quality, prop.chisq = F)
```

```
##
##
##      Cell Contents
## -----|
##      |      N |
##      |      N / Row Total |
##      |      N / Col Total |
##      |      N / Table Total |
## -----|
##
##
## Total Observations in Table:  1470
##
##
##      | wine.test$quality
## wine.knn$k5 |      Low |      Medium |      High | Row Total |
## -----|-----|-----|-----|-----|
##      Low |      301 |      153 |      2 |      456 |
##      |      0.660 |      0.336 |      0.004 |      0.310 |
##      |      0.634 |      0.163 |      0.037 |      |
##      |      0.205 |      0.104 |      0.001 |      |
## -----|-----|-----|-----|
##      Medium |      170 |      779 |      48 |      997 |
##      |      0.171 |      0.781 |      0.048 |      0.678 |
##      |      0.358 |      0.828 |      0.889 |      |
##      |      0.116 |      0.530 |      0.033 |      |
## -----|-----|-----|-----|
##      High |      4 |      9 |      4 |      17 |
##      |      0.235 |      0.529 |      0.235 |      0.012 |
##      |      0.008 |      0.010 |      0.074 |      |
##      |      0.003 |      0.006 |      0.003 |      |
## -----|-----|-----|-----|
## Column Total |      475 |      941 |      54 |      1470 |
##      |      0.323 |      0.640 |      0.037 |      |
## -----|-----|-----|-----|
##
```

```
##
```

```
CrossTable(x=wine.knn$k15, y=wine.test$quality, prop.chisq = F)
```

```
##
```

```
##
```

```
##      Cell Contents
```

```
## |-----|
## |                N |
## |      N / Row Total |
## |      N / Col Total |
## |      N / Table Total |
## |-----|
```

```
##
```

```
##
```

```
## Total Observations in Table:  1470
```

```
##
```

```
##
```

```
##      | wine.test$quality
## wine.knn$k15 |      Low |      Medium |      High | Row Total |
## -----|-----|-----|-----|-----|
##      Low |      293 |      143 |        2 |      438 |
##      |      0.669 |      0.326 |      0.005 |      0.298 |
##      |      0.617 |      0.152 |      0.037 |      |
##      |      0.199 |      0.097 |      0.001 |      |
## -----|-----|-----|-----|-----|
##      Medium |      182 |      798 |      52 |      1032 |
##      |      0.176 |      0.773 |      0.050 |      0.702 |
##      |      0.383 |      0.848 |      0.963 |      |
##      |      0.124 |      0.543 |      0.035 |      |
## -----|-----|-----|-----|-----|
## Column Total |      475 |      941 |      54 |      1470 |
##      |      0.323 |      0.640 |      0.037 |      |
## -----|-----|-----|-----|-----|
```

```
##
```

```
##
```

```
# Different values of k produce very similar results with the best being a toss up between k=3 & k=5
# I think that overall this model performed acceptably. It would be interesting to compare to other
# models (Naive Bayes, Decesion Tree, etc) and see how it compares in terms of accuracy.
```

```
#
```

```
# I think another important concept that wasn't included in this assignment is the
# concept of "research question" and "metrics of success".
# For example, below I listed the True Positive rates for the three levels: low, medium and high.
# However, perhaps True Positives aren't the most important metrics for your research question.
# Maybe False Positives are a bigger issue, or False Negatives. This is context specific but extremely
# relevant with these types of models. For example, look at the classification of High Quality wines:
# The TP rates are 37%, 23% and 0% with k set to 3, and 15 respectively.
# That might not be considered very high depending on your research question.
```

```
#
```

```
# Also note the low success TP rate for a high k: 0%. This would seem like a result of the
# "majority vote" combined with a class imbalance problem. As there are so few Wines in
# the "high" category it is unlikely that with a higher K, who collects votes from more
# "Nearest Neighbors" there being enough votes in favour of the wine quality being "high".
# This is definitely an issue to be aware of when using these models.
```

```
#
# TP rates (k=3): 64.9%, 78.7%, 37.0% for Low, Medium & High respectively
#
# I also did a Prime Component Analysis (PCA) and ran the kNN model to compare with the above results.
# It produced a very marginal improvement in TP rates for k=3: 65.2%, 78.8%, and 40.0% for Low, Medium
# & High, respectively.
```

## Extra

### Visualization

I tried to plot the KNN output of this assignment visually and ran into some trouble I still believe this to be a great exercise but perhaps not ideal for this particular assignment.

### Some resources:

- <https://cran.r-project.org/web/packages/ElemStatLearn/ElemStatLearn.pdf> # pg 8: mixture.example examples
- <https://stats.stackexchange.com/questions/21572/how-to-plot-decision-boundary-of-a-k-nearest-neighbor-classifier-from-21602#21602>
- <https://stackoverflow.com/questions/31234621/variation-on-how-to-plot-decision-boundary-of-a-k-nearest-neighbor-classifier>

I adapted my code as much as possible from the examples and discussion on Stack Overflow but in the end had to settle for a relatively straightforward cluster plot based on kNN algorithm results. The results of this attempt made me wonder if performing Dimensionality Reduction would help: it didn't. My results are below. In the end I chose to plot pH vs. free.sulfur.dioxide because these are the least correlated factors in the dataset.

```
require(dplyr)
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:reshape':
##
##   rename

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

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

classif <- wine.knn$k3
prob <- attr(classif, "prob")
wine.gf.df <- bind_rows(mutate(wine.test[, -12],
                              prob=prob,
                              cls="Low",
                              prob_cls=ifelse(classif==cls, 1, 0)),
                       mutate(wine.test[, -12],
                              prob=prob,
```

```

        cls="Medium",
        probb_cls=ifelse(classif==cls, 1, 0)),
mutate(wine.test[, -12],
        probb=probb,
        cls="High",
        probb_cls=ifelse(classif==cls, 1, 0)))

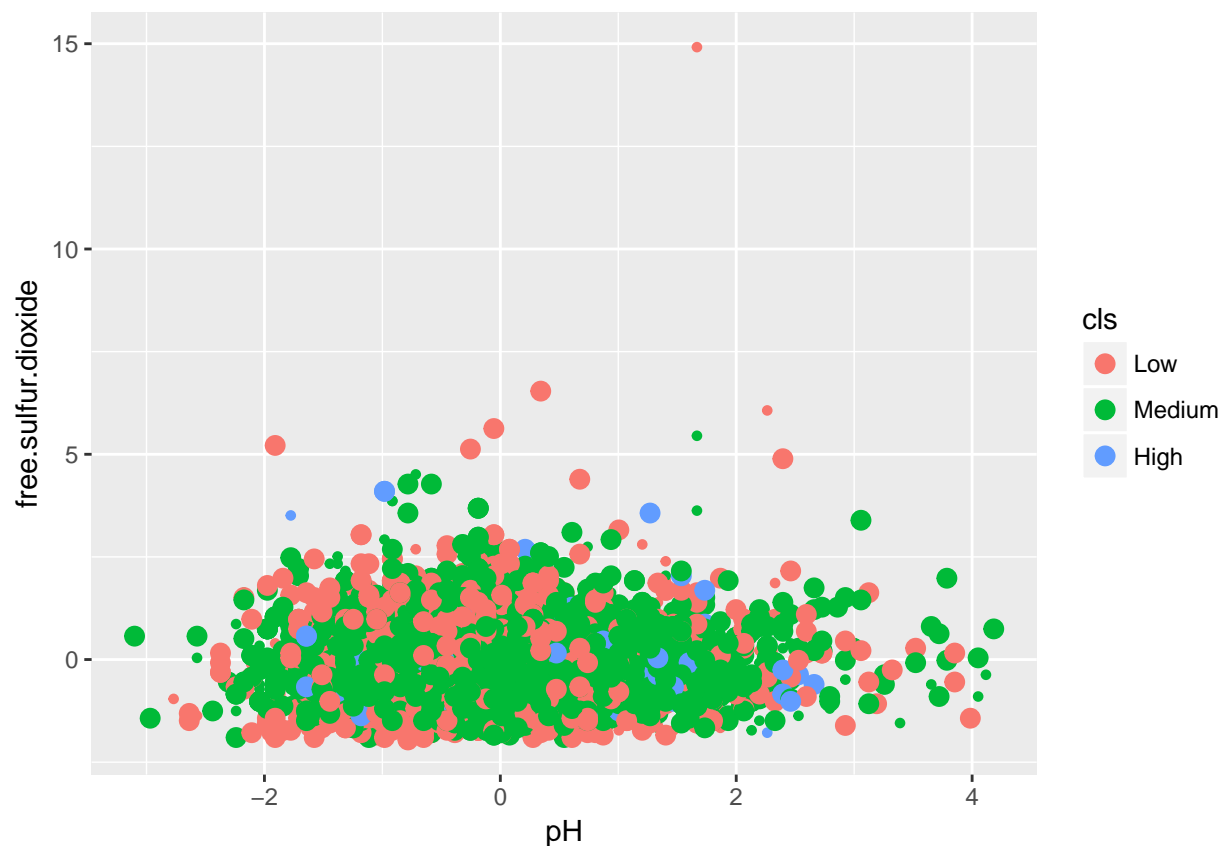
# wine.gf.df.unq <- wine.gf.df[!duplicated(wine.gf.df[, c('pH', 'free.sulfur.dioxide')]), ]

require(ggplot2)

## Loading required package: ggplot2

ggplot(wine.gf.df) +
  geom_point(aes(x=pH, y=free.sulfur.dioxide, col=cls),
             data=mutate(wine.test[, -12], cls=classif),
             size=1.2) +
  # geom_contour(aes(x=pH, y=free.sulfur.dioxide, z=probb_cls, group=cls, color=cols),
  #             bins=2,
  #             data=wine.gf.df.unq) +
  geom_point(aes(x=x, y=y, col=cls),
             size=3,
             data=data.frame(x=wine.train$pH, y=wine.train$free.sulfur.dioxide, cls=wine.train$quality))

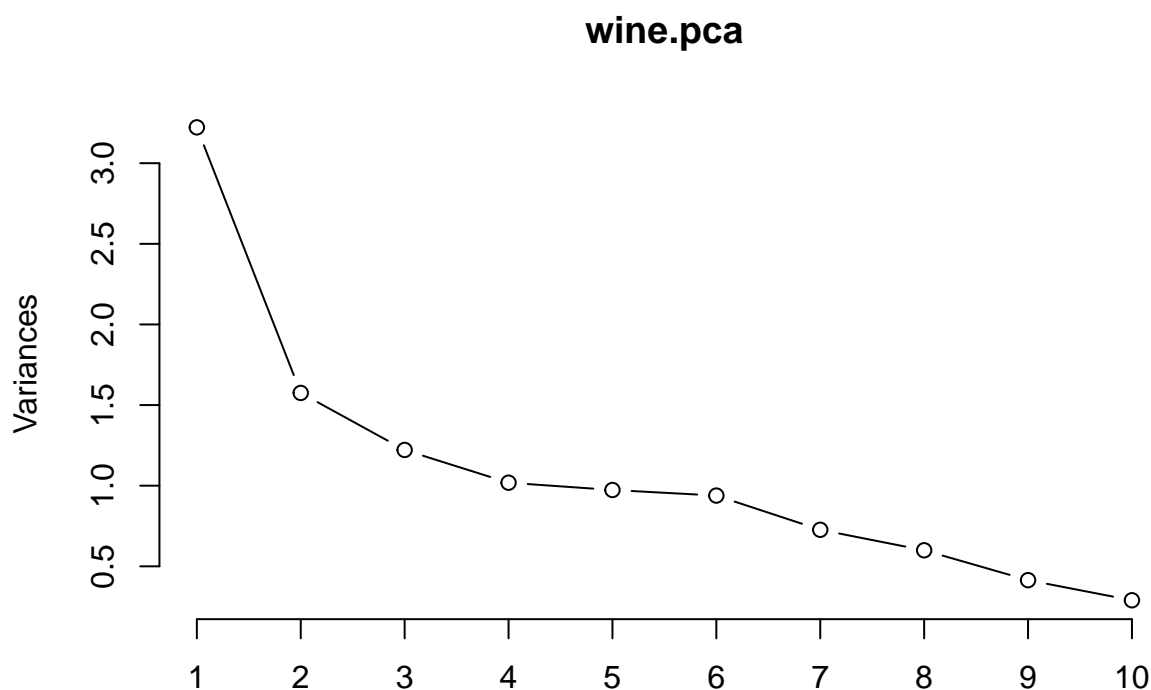
```



## PCA

I decided to spend some more time on this dataset and applied PCA. My goal was the same as above; to organize the data in such a way to be able to create an effective visual. I was unsuccessful in that regard but did notice that the model performs better, with regards to the TP Rate, than without doing PCA.

```
wine.pca <- prcomp(wine.df[,-12], scale. = T, center = T)
screeplot(wine.pca, type="lines")
```



```
wine.pca.knn <- knn(train=wine.pca$x[wine.train_index,], test=wine.pca$x[-wine.train_index,], cl=wine.d
CrossTable(x=wine.pca.knn, y=wine.df[-wine.train_index,'quality'], prop.chisq = F)
```

```
##
##
##   Cell Contents
## |-----|
## |               N |
## |   N / Row Total |
## |   N / Col Total |
## |   N / Table Total |
## |-----|
##
##
## Total Observations in Table:  1470
##
##
```

```
## | wine.df[-wine.train_index, "quality"]
## wine.pca.knn |      Low |      Medium |      High | Row Total |
## -----|-----|-----|-----|-----|
##      Low |      306 |      160 |       3 |      469 |
##      |      0.652 |      0.341 |      0.006 |      0.319 |
##      |      0.644 |      0.170 |      0.056 |      |
##      |      0.208 |      0.109 |      0.002 |      |
## -----|-----|-----|-----|-----|
##      Medium |      166 |      769 |      41 |      976 |
##      |      0.170 |      0.788 |      0.042 |      0.664 |
##      |      0.349 |      0.817 |      0.759 |      |
##      |      0.113 |      0.523 |      0.028 |      |
## -----|-----|-----|-----|-----|
##      High |       3 |      12 |      10 |      25 |
##      |      0.120 |      0.480 |      0.400 |      0.017 |
##      |      0.006 |      0.013 |      0.185 |      |
##      |      0.002 |      0.008 |      0.007 |      |
## -----|-----|-----|-----|-----|
## Column Total |      475 |      941 |      54 |      1470 |
##      |      0.323 |      0.640 |      0.037 |      |
## -----|-----|-----|-----|-----|
##
##
```

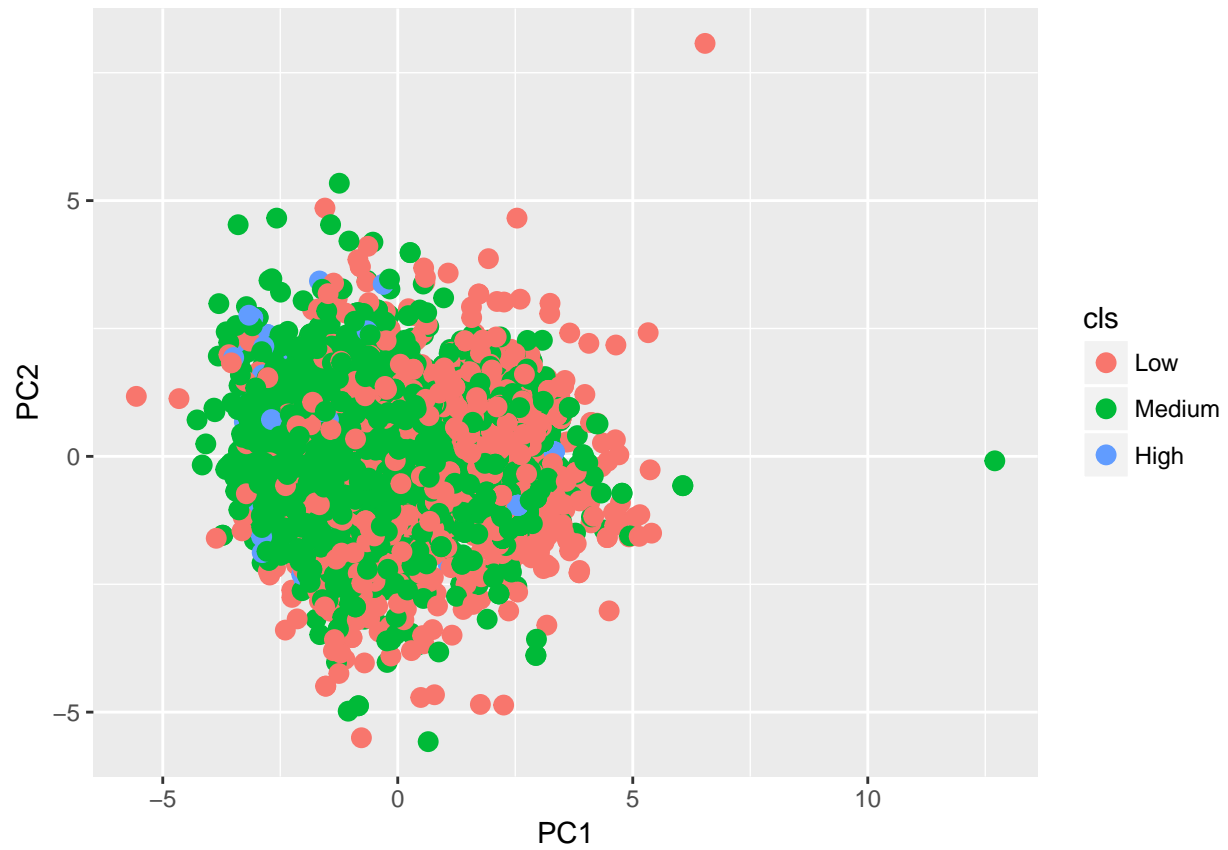
```
prob <- attr(wine.pca.knn, "prob")
test <- as.data.frame(wine.pca$x)

wine.pca.gf.df <- bind_rows(mutate(test[-wine.train_index,],
                                   prob=prob,
                                   cls="Low",
                                   prob_cls=ifelse(wine.pca.knn==cls, 1, 0)),
                             mutate(test[-wine.train_index,],
                                   prob=prob,
                                   cls="Medium",
                                   prob_cls=ifelse(wine.pca.knn==cls, 1, 0)),
                             mutate(test[-wine.train_index,],
                                   prob=prob,
                                   cls="High",
                                   prob_cls=ifelse(wine.pca.knn==cls, 1, 0)))

wine.pca.gf.df$cols <- vector(length=nrow(wine.pca.gf.df))
wine.pca.gf.df$cols[wine.pca.gf.df$cls == 'Low'] <- 'c'
wine.pca.gf.df$cols[wine.pca.gf.df$cls == 'Medium'] <- 's'
wine.pca.gf.df$cols[wine.pca.gf.df$cls == 'High'] <- 'v'

ggplot(wine.gf.df) +
  geom_point(aes(x=PC1, y=PC2, col=cls),
             data=mutate(test[-wine.train_index,], cls=wine.pca.knn),
             size=1.2) +
  # geom_contour(aes(x=PC1, y=PC2, z=prob_cls, group=cls, color=cols),
  #             bins=2,
  #             data=wine.pca.gf.df) +
  geom_point(aes(x=x, y=y, col=cls),
```

```
size=3,  
data=data.frame(x=wine.pca$x[,1], y=wine.pca$x[,2], cls=wine.df$quality))
```



Save your R codes in an RMD file. Send your RMD and PDF files to the course shell.