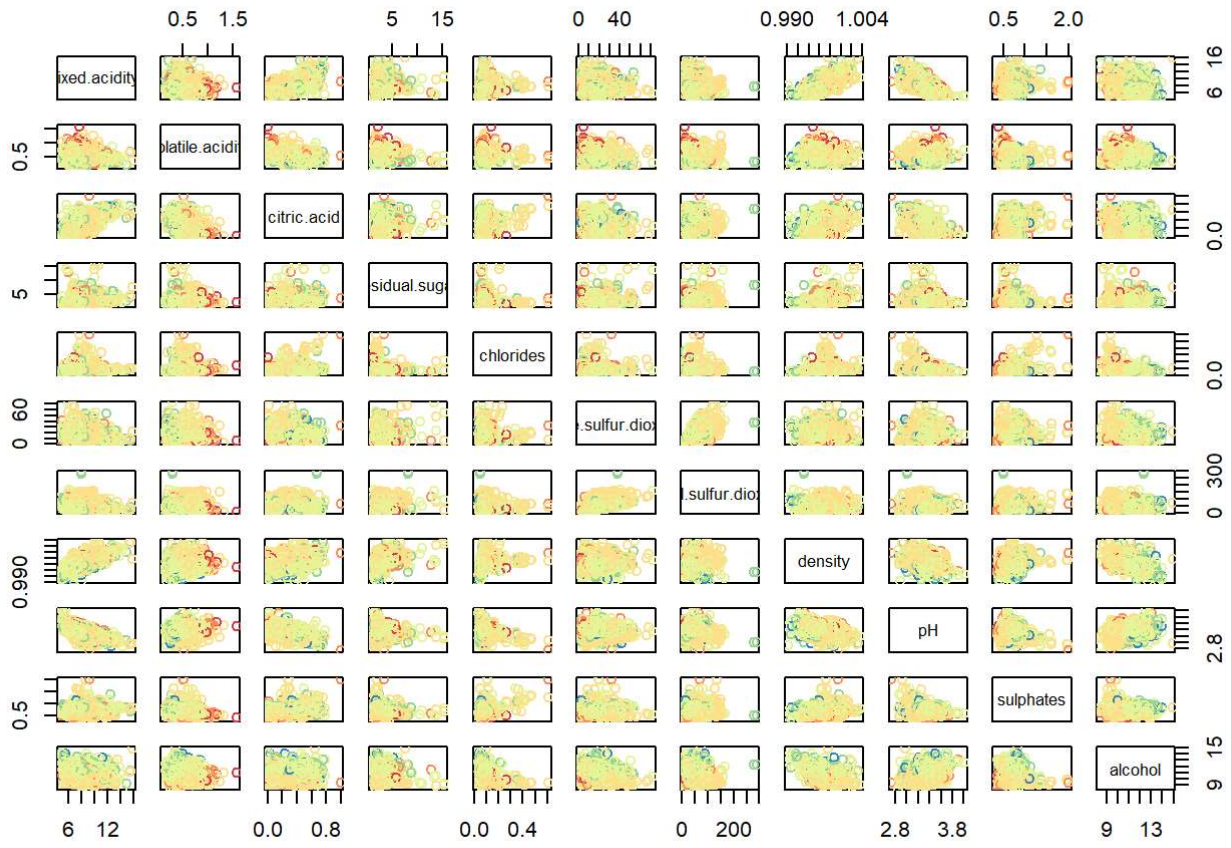
Assignment 7

Section 1

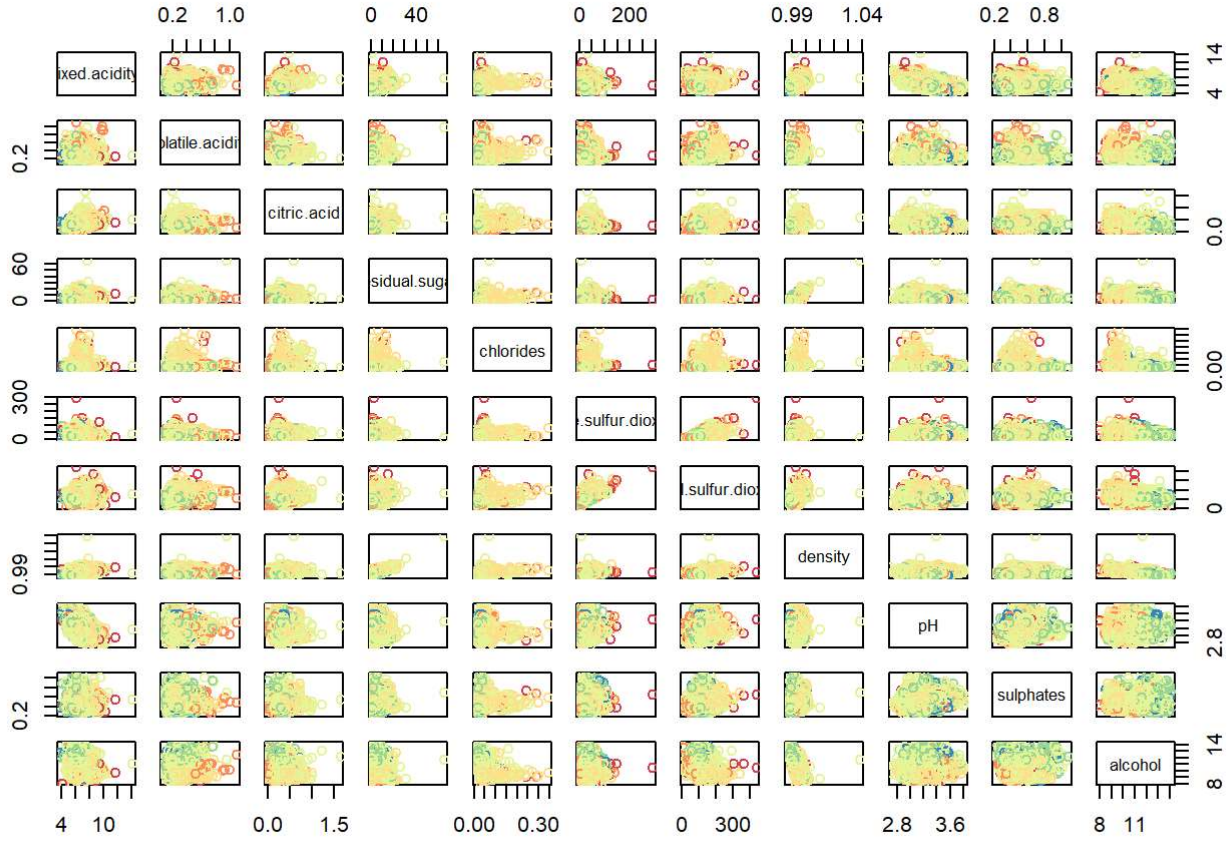
We will be using the wine datasets from the UCI Machine Learning dataset collection.

```
red <- read.csv("winequality-red.csv")
white <- read.csv("winequality-white.csv")
library(RColorBrewer)
colors <- brewer.pal(6, "Spectral")
red$color <- colors[red$quality-2]
white$color <- colors[white$quality-2]
```

```
pairs(red[,1:11], col = red$color)
```

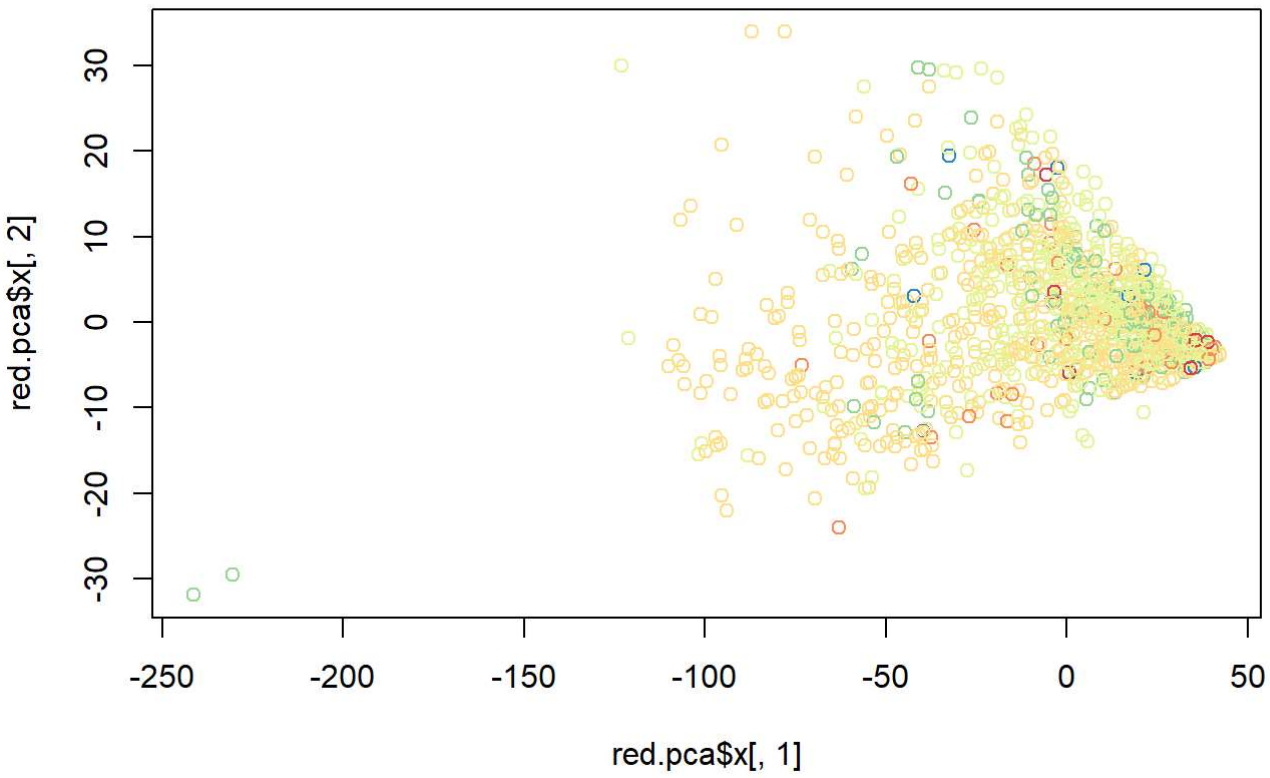



```
pairs(white[,1:11], col = white$color)
```

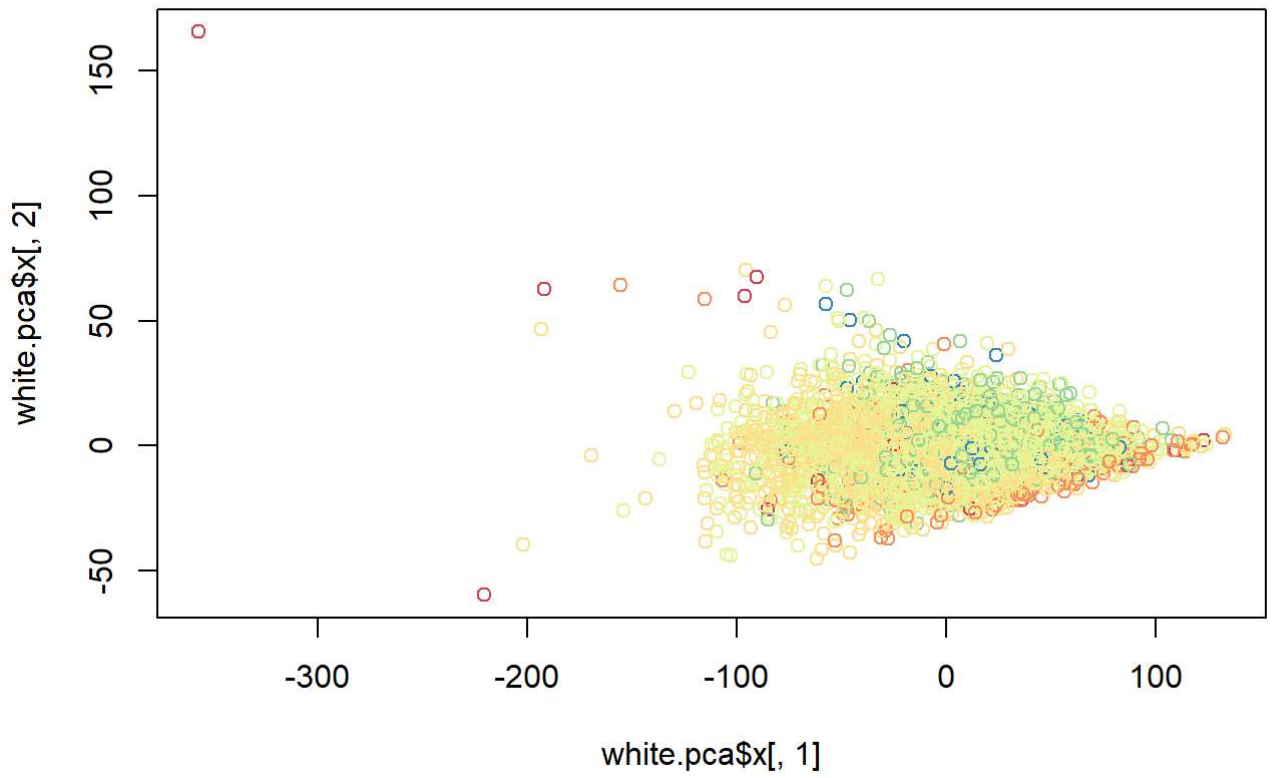


Examining the scatterplots shows that several variables are corelated. Factor analysis or principal components analysis might provide some insights into latent variables behind the correlations. We can also see that none of the variables are too severely skewed. THere are not outliers in the data, so nothing has to be removed.

```
red.pca <- prcomp(red[,1:11])
white.pca <- prcomp(white[,1:11])
plot(red.pca$x[,1], red.pca$x[,2],col=red$color)
```




```
plot(white.pca$x[,1], white.pca$x[,2],col=white$color)
```



Looking a the Principal components plots, we can see that certain quality levels are clustered together. However the boundary between quality levels is fuzzy. Therefore, some sort of nonlinear regression seems like a good idea. Because the distinctions between levels is arbitary, linear regression does not really make sense. Moreover, because the boundaries are not distinct, clustering is not likely to work.

Section 2

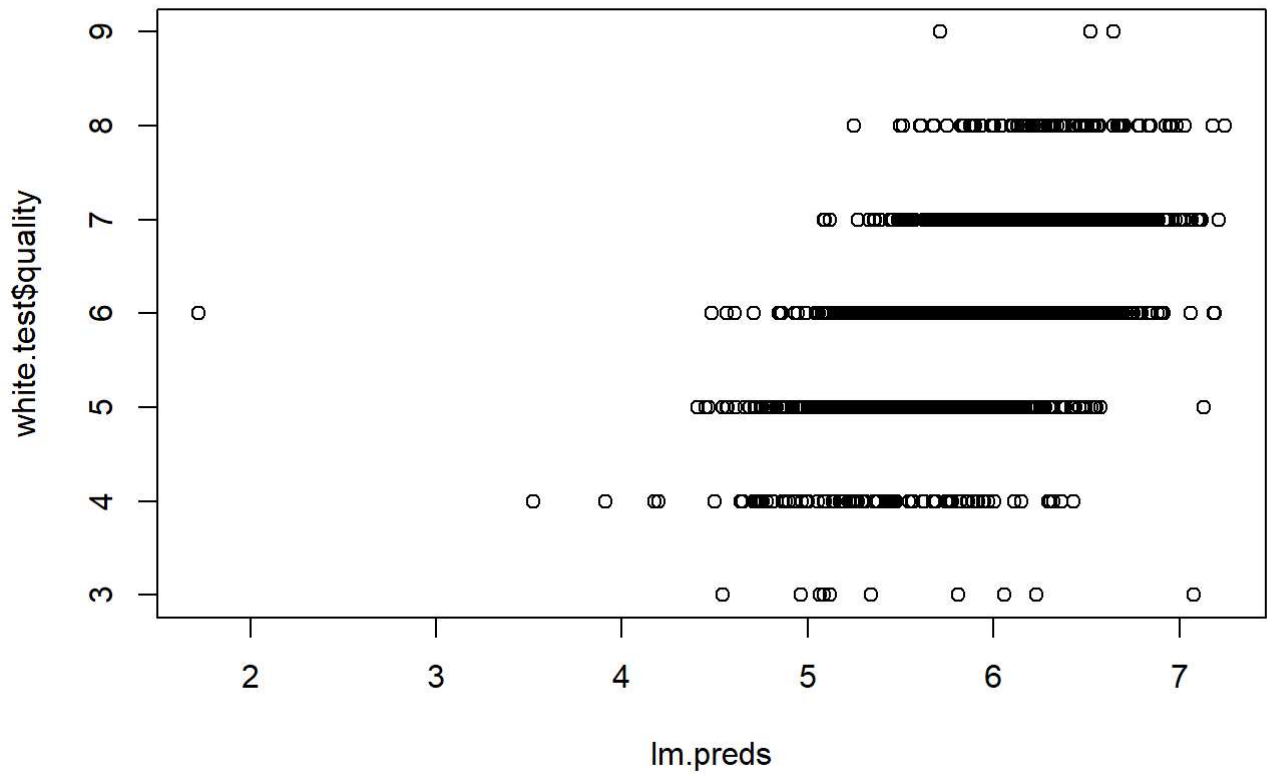
We will be attempting to predict wine quality for white wine. We will do validation using a training set and a training set. We will use training set and test set validation because it is relatively simple to implement.

```
train <- sample(nrow(white), 2000)
white.train <- white[train,]
white.test <- white[-train,]
```

```
lm <- lm(as.numeric(quality)~., white.train[1:12])
summary(lm)
```

```
##
## Call:
## lm(formula = as.numeric(quality) ~ ., data = white.train[1:12])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3829 -0.4753 -0.0392  0.4637  2.4876
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.893e+02  3.574e+01   5.298 1.30e-07 ***
## fixed.acidity         3.397e-02  3.450e-02   0.985 0.324933
## volatile.acidity     -1.951e+00  1.710e-01 -11.414  < 2e-16 ***
## citric.acid           1.418e-01  1.443e-01   0.983 0.325830
## residual.sugar        8.854e-02  1.348e-02   6.567 6.52e-11 ***
## chlorides            -5.137e-01  8.085e-01  -0.635 0.525255
## free.sulfur.dioxide   3.292e-03  1.339e-03   2.459 0.014014 *
## total.sulfur.dioxide  8.618e-04  5.963e-04   1.445 0.148563
## density              -1.890e+02  3.619e+01  -5.221 1.97e-07 ***
## pH                    5.967e-01  1.716e-01   3.477 0.000517 ***
## sulphates             8.042e-01  1.592e-01   5.051 4.80e-07 ***
## alcohol               1.466e-01  4.539e-02   3.231 0.001255 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7377 on 1988 degrees of freedom
## Multiple R-squared:  0.3034, Adjusted R-squared:  0.2996
## F-statistic: 78.73 on 11 and 1988 DF,  p-value: < 2.2e-16
```

```
lm.preds <- predict(lm, white.test)
plot(lm.preds, white.test$quality)
```




```
lm.mse <- sum((lm.preds - white.test$quality)^2)
lm.mse
```

```
## [1] 1699.819
```

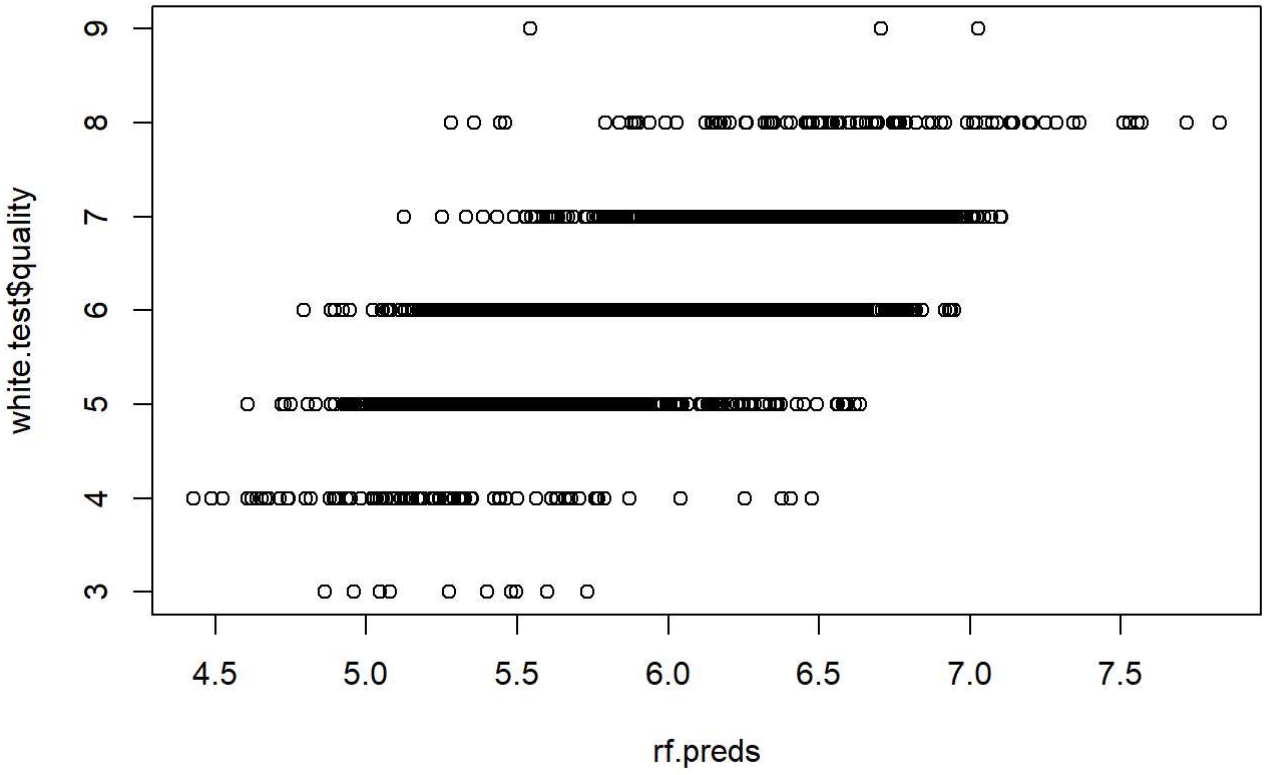
We construct a simple linear model to predict wine quality. We are using a linear model as a baseline so we can judge the accuracy of the random forest model that we will use later. Moreover, the linear model is very easy to interpret. We can see that factors like volatile acidity and alcohol are statistically significant. The small p-value associated with the F-statistic indicates that the linear model is statistically significant. The negative coefficient on density shows that low densities are prefered while the positive coefficient on alcohol shows that high alcohol levels are preferred. The scatterplot of predicted quality vs actual quality shows a positive trend. However, it reveals that our model is not very precise. We get a total mean squared error of 1638 on the test data.

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(forestFloor)
rf = randomForest(
  quality~.,
  white.train[1:12],
  keep.inbag = TRUE,
  importance = TRUE,
  mtry = 3,
  prox = TRUE,
  ntree = 1000,
)
rf.preds <- predict(rf, white.test)
plot(rf.preds, white.test$quality)
```

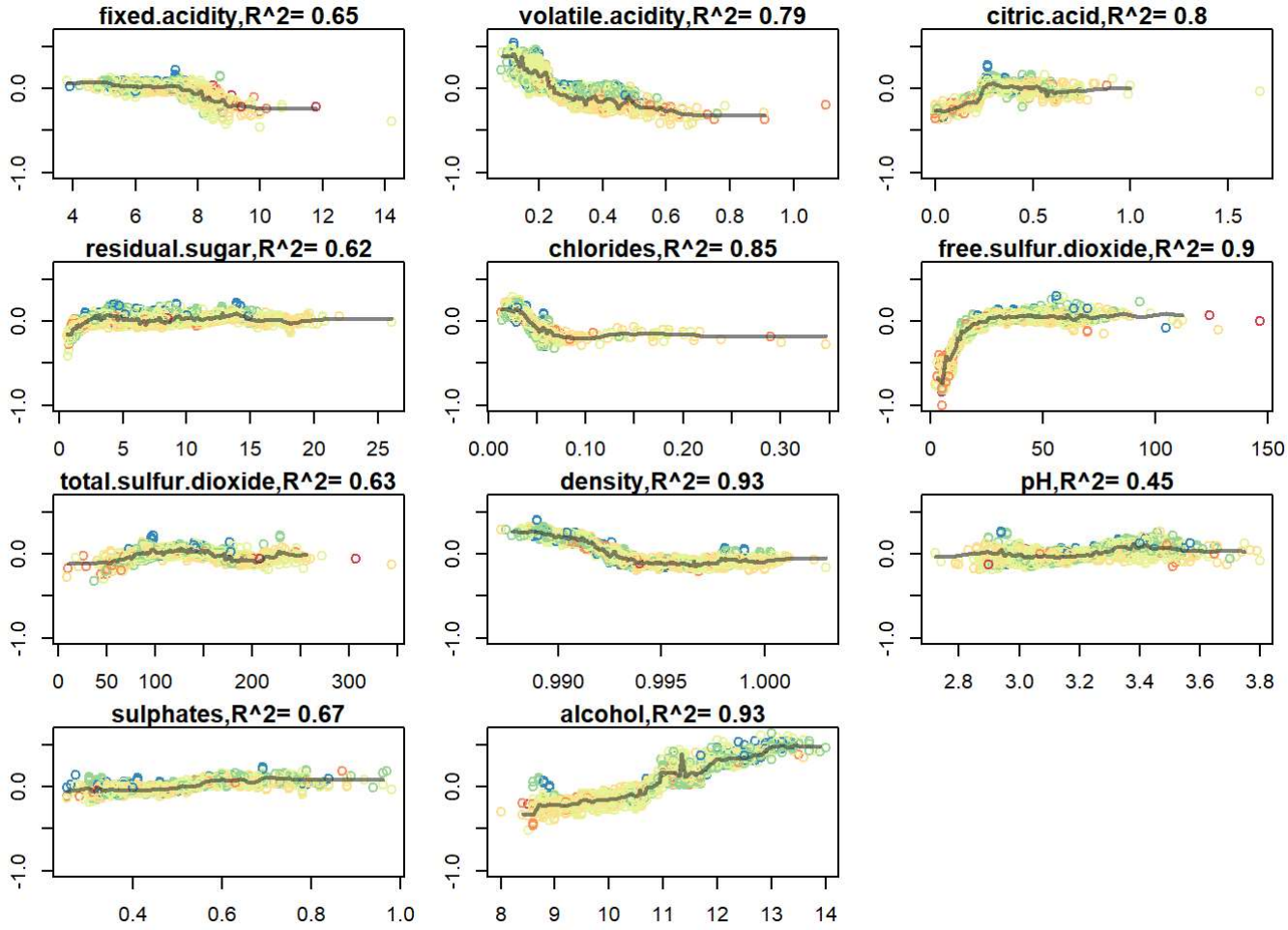



```
rf.mse <- sum((rf.preds - white.test$quality)^2)
rf.mse
```

```
## [1] 1238.043
```

```
ff = forestFloor(
  rf = rf ,       # mandatory
  X = white.train,              # mandatory
  calc_np = FALSE,    # TRUE or FALSE both works, makes no difference
  binary_reg = FALSE  # takes no effect here when rfo$type="regression"
)
plot(ff,  col = white.train$color  ,                 # forestFloor object
     orderByImportance=FALSE    # if TRUE index sequence by importance, else by X column
)
```

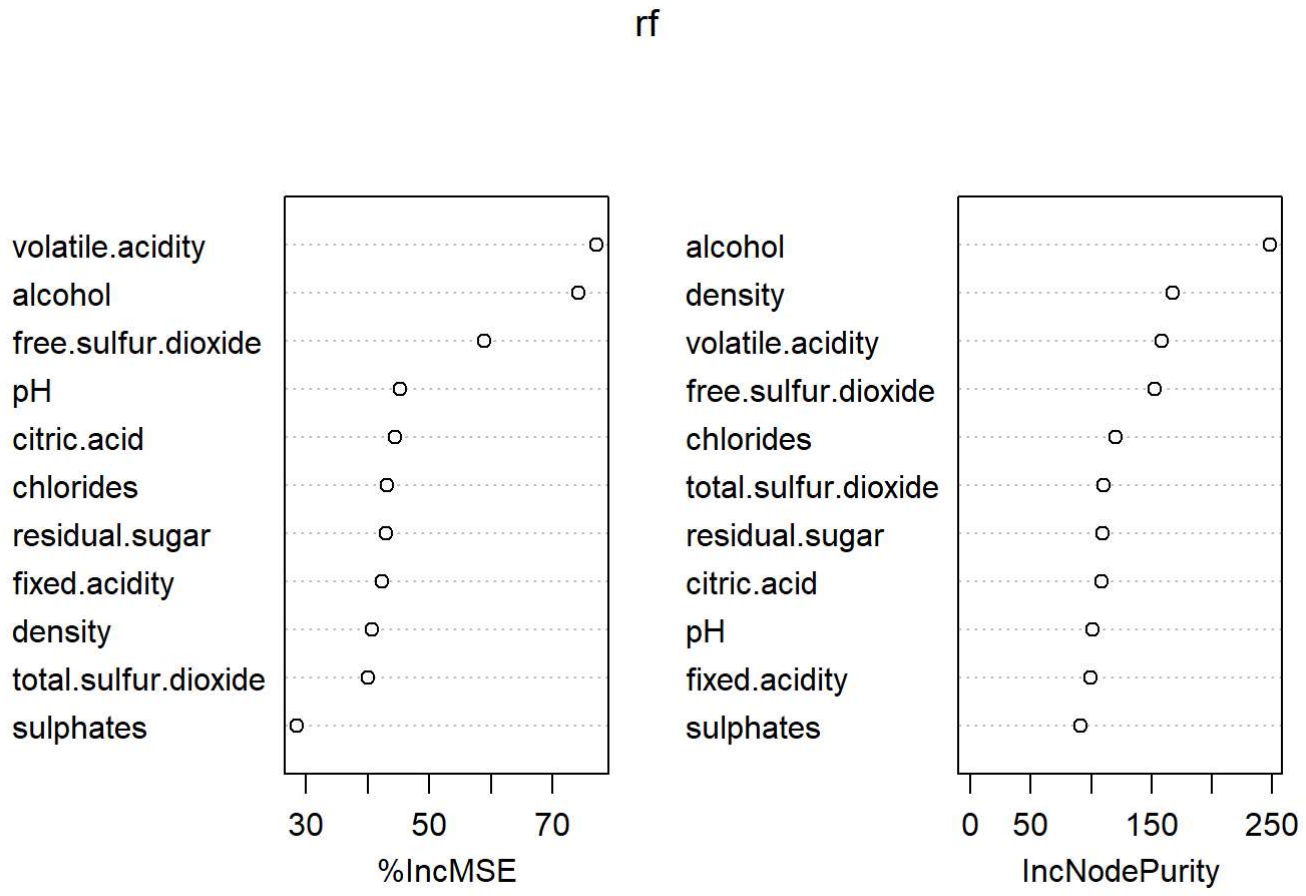
```
## [1] "compute goodness-of-fit with leave-one-out k-nearest neighbor(guassian weighting), kknn
package"
```



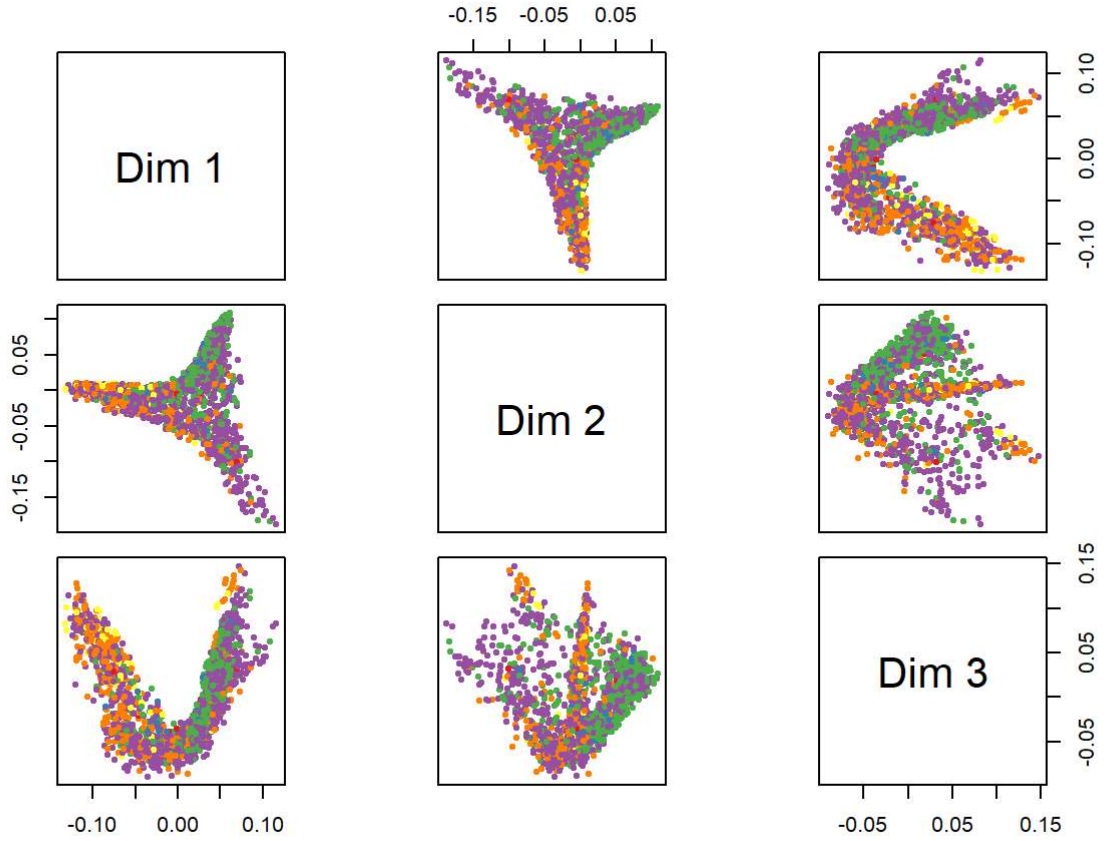
We use a random forest model to predict wine quality. We see that the test mean squared error of 1228 is lower than the linear model's test mean squared error of 1638. This improvement in test accuracy over a linear model we previously determined to be statistically significant shows that the random forest model is also statistically significant. We are using a random forest model because it is very good at making predictions based on quantitative predictors. Unlike the linear model, the random forest model can handle nonlinear trends. For example, in the forest floor plots, we see that the optimal value of total sulfur dioxide is around 100 not at the extremes of the range.On the other hand, we can see that the sulphates variable is not very important. The scatterplot shows that the random forest is still an inexact measure of wine quality.

Section 3

```
varImpPlot(rf)
```




```
randomForest::MDSplot(rf, fac = as.factor(white.train$quality), k=3)
```



The random forest models does not provide us a very tight fit that is super useful for predicting quality. The proximity plots show us that the qualites do not separate perfectly. However, we can see what variables lead to high wine quality. In particular, we can see that lower densitiy and higher alcohol in general lead to higher scores. Moreover, we can see an optimal range for citric acid. Understanding the optimal values for each factor allows us to make decisions in that winemakers could improve wine my finetuning these variables. A winemaker could try to set target citric acid levels among other things because these critical regions are statistically significant. On ther other hand, using this model to predict quality is not reccomended because the variance within qualities is so great.