Red Wines - upper

Katie, Rita, and Chang

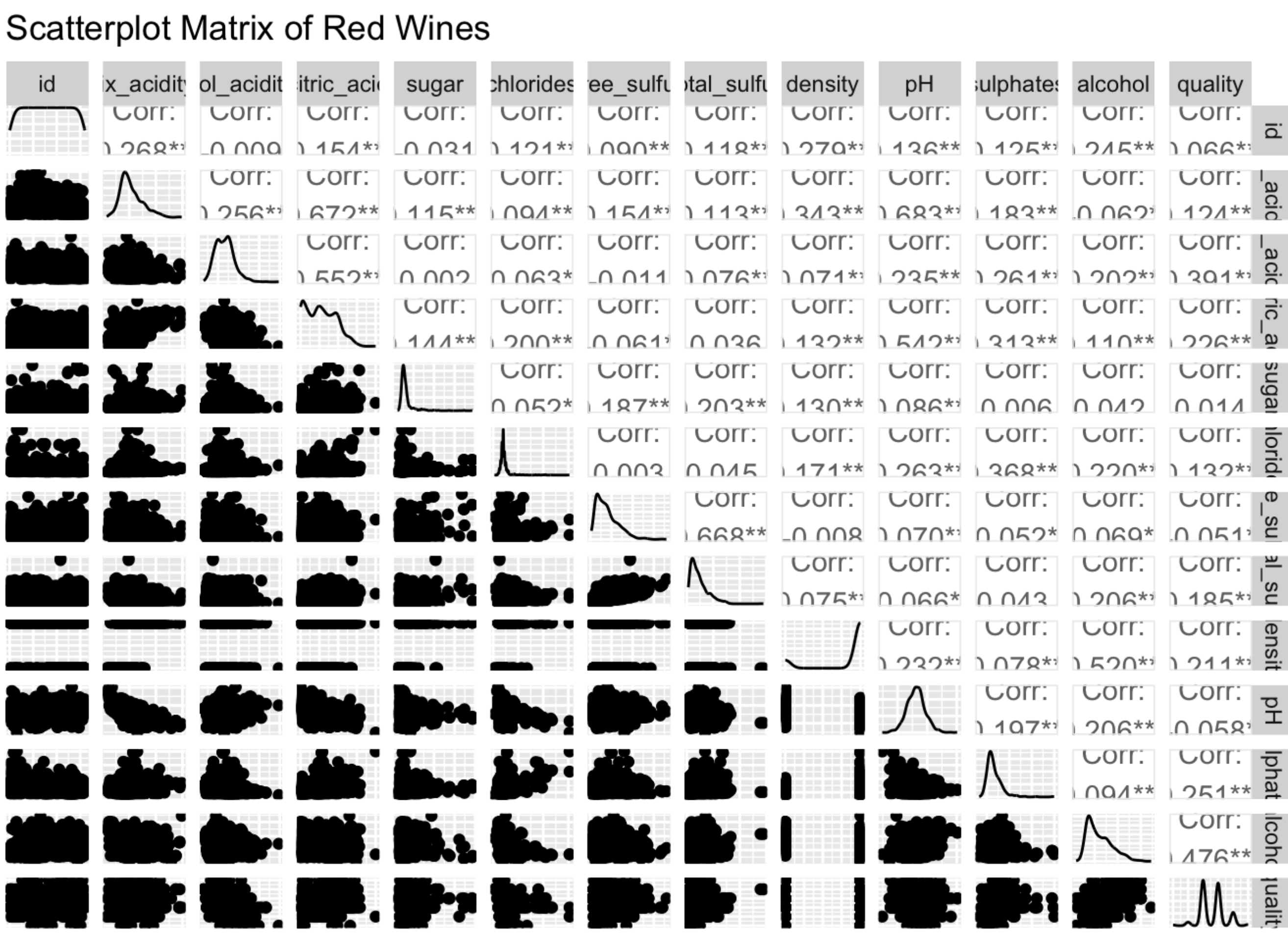
2023-11-01

Scatterplot Matrix

```
library("GGally")
ggpairs(red, axisLabels = "none",
        title = "Scatterplot Matrix of Red Wines")
```

```
# corr codes
```

Scatterplot Matrix



Create Binary Dependent Variable

```
red$highquality = factor((red$quality >= 6))
red$highquality <- as.integer(as.logical(red$highquality))
```

Create Test and Training Data

```
library("caTools")
set.seed = 100
split = sample.split(red$highquality, SplitRatio = 0.6)
train = subset(red, split == TRUE)
test = subset(red, split == FALSE)
print(dim(train)); print(dim(test))
```

```
## [1] 959  14
```

```
## [1] 640  14
```

Descriptive Statistics

```
library("Rmisc")
```

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

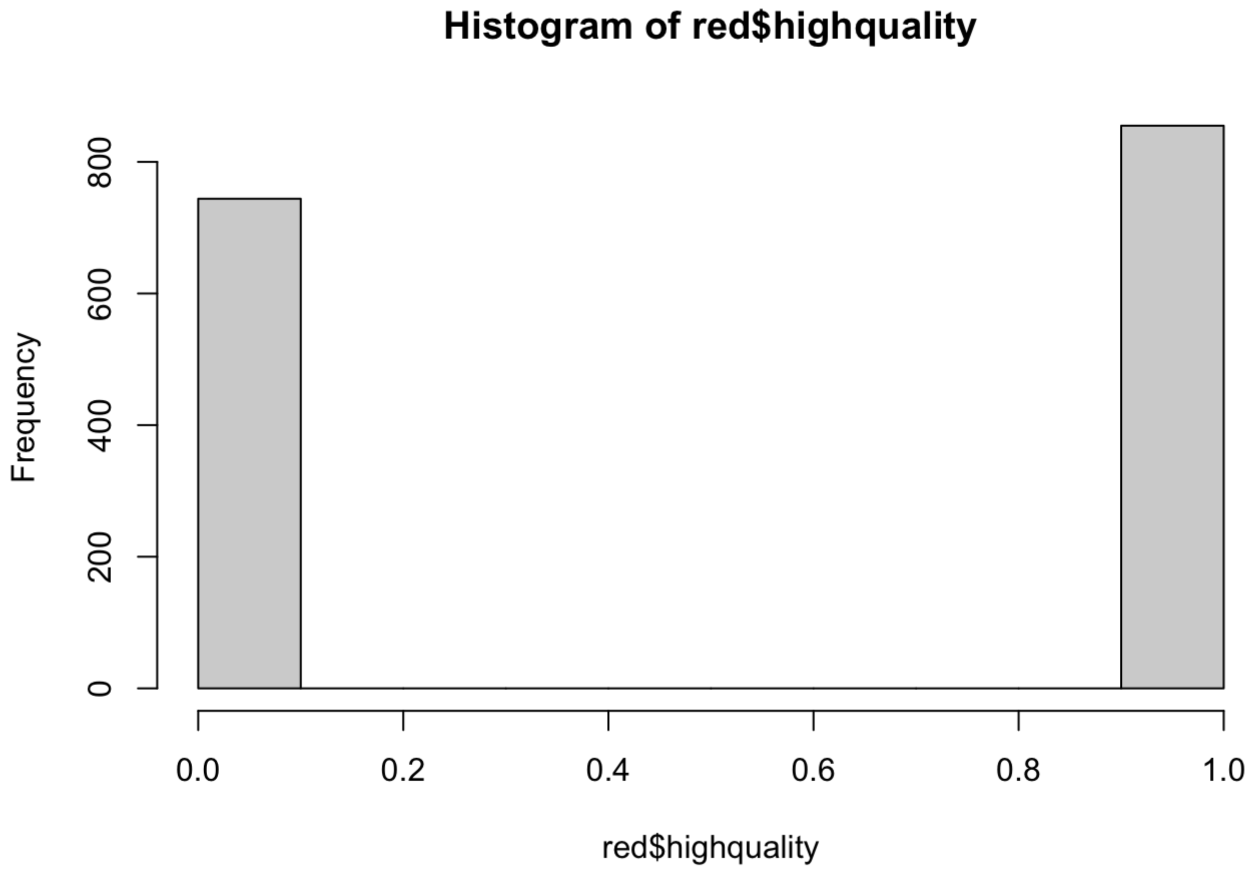
```
## Loading required package: plyr
```

```
sum = summary(red)
sum
```

```
##        id          fix_acidity     vol_acidity      citric_acid
##  Min.   :   1.0   Min.   : 4.60   Min.   :0.1200   Min.   :0.000
##  1st Qu.: 400.5   1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090
##  Median : 800.0   Median : 7.90   Median :0.5200   Median :0.260
##  Mean   : 800.0   Mean   : 8.32   Mean   :0.5284   Mean   :0.271
##  3rd Qu.:1199.5   3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420
##  Max.   :1599.0   Max.   :15.90   Max.   :1.5800   Max.   :1.000
##      sugar           chlorides         free_sulfur     total_sulfur
##  Min.   : 0.900   Min.   :0.01000   Min.   : 1.00   Min.   :  6.00
##  1st Qu.: 1.900   1st Qu.:0.07000   1st Qu.: 7.00   1st Qu.: 22.00
##  Median : 2.200   Median :0.08000   Median :14.00   Median : 38.00
##  Mean   : 2.539   Mean   :0.08787   Mean   :15.87   Mean   : 46.47
##  3rd Qu.: 2.600   3rd Qu.:0.09000   3rd Qu.:21.00   3rd Qu.: 62.00
##  Max.   :15.500   Max.   :0.61000   Max.   :72.00   Max.   :289.00
##     density             pH           sulphates         alcohol
##  Min.   :0.9900   Min.   :2.740   Min.   :0.3300   Min.   : 8.40
##  1st Qu.:1.0000   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50
##  Median :1.0000   Median :3.310   Median :0.6200   Median :10.20
##  Mean   :0.9985   Mean   :3.311   Mean   :0.6581   Mean   :10.42
##  3rd Qu.:1.0000   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
##  Max.   :1.0000   Max.   :4.010   Max.   :2.0000   Max.   :14.90
##     quality       highquality
##  Min.   :3.000   Min.   :0.0000
##  1st Qu.:5.000   1st Qu.:0.0000
##  Median :6.000   Median :1.0000
##  Mean   :5.636   Mean   :0.5347
##  3rd Qu.:6.000   3rd Qu.:1.0000
##  Max.   :8.000   Max.   :1.0000
```

Plot high quality vs low quality distribution

```
hist (red$highquality)
```



Random Forest

```
library("randomForest")
```

```
## randomForest 4.7-1.1
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

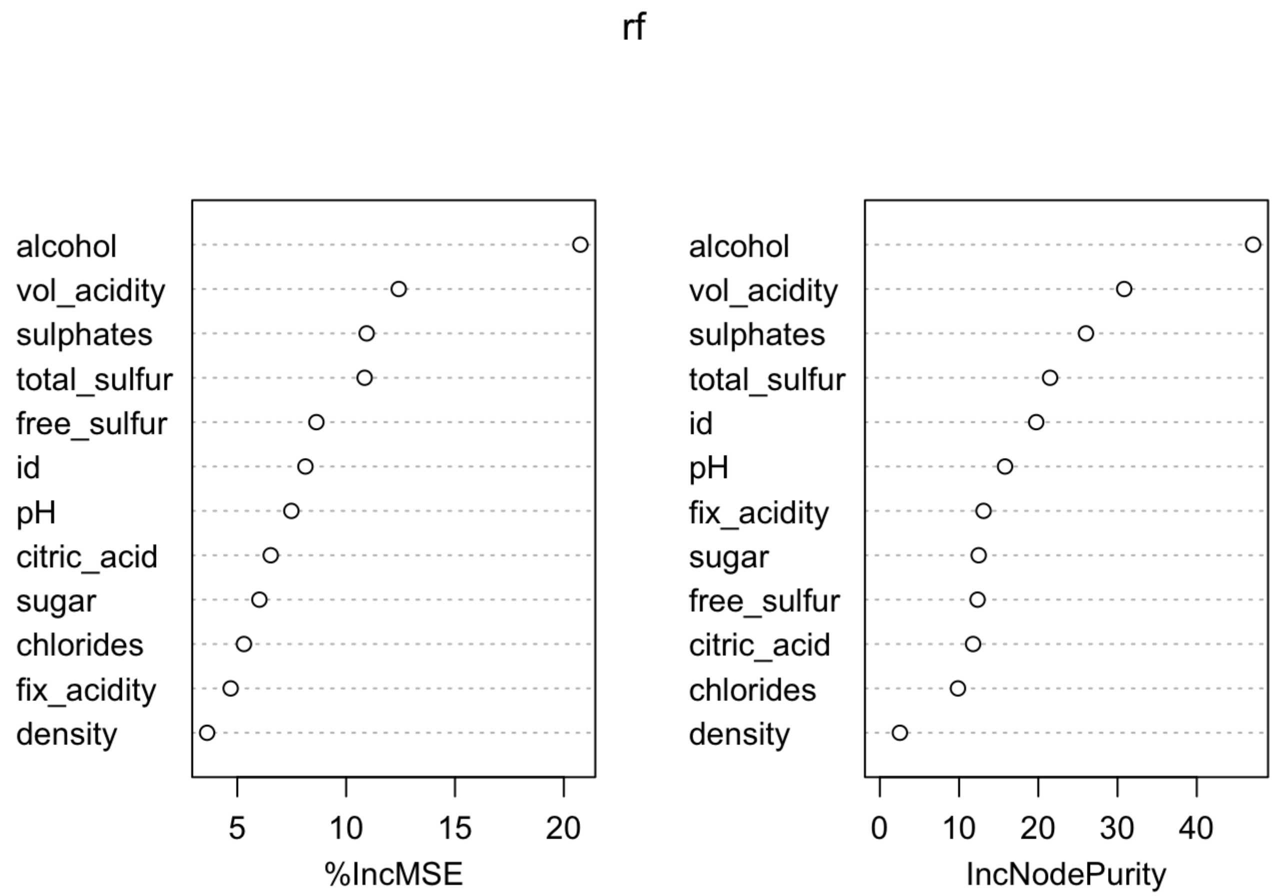
```
library("caret")
library("e1071")
library("rpart")

rf <- randomForest(highquality ~ . - quality, data = train, mtry = 4, importance = TRUE,
ntree = 50, na.action = na.omit)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
print(rf)
```

```
##
## Call:
##  randomForest(formula = highquality ~ . - quality, data = train,      mtry = 4, impor
tance = TRUE, ntree = 50, na.action = na.omit)
##                Type of random forest: regression
##                      Number of trees: 50
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 0.1549242
##                     % Var explained: 37.73
```

```
varImpPlot(rf)
```




```
# predictions on test set
set.seed(100)
predictTest = predict(rf, newdata = test, type = "response")

# confusion matrix on test set
table(test$highquality, predictTest >= 0.5)
```

```
##
##     FALSE TRUE
##   0   236   62
##   1    64  278
```

Random Forest Model

```
# Logit
randomforestmodlogit <- glm(highquality ~ alcohol + sulphates + total_sulfur + vol_acidi
ty, data = red, family = "binomial"(link = "logit"))
summary(randomforestmodlogit)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     vol_acidity, family = binomial(link = "logit"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1638  -0.8675   0.3076   0.8629   2.3262
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.588813   0.795118 -10.802  < 2e-16 ***
## alcohol       0.927362   0.069268  13.388  < 2e-16 ***
## sulphates     2.059047   0.365976   5.626 1.84e-08 ***
## total_sulfur -0.011976   0.001924  -6.225 4.83e-10 ***
## vol_acidity  -3.083277   0.364832  -8.451  < 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: 2209.0  on 1598  degrees of freedom
## Residual deviance: 1684.2  on 1594  degrees of freedom
## AIC: 1694.2
##
## Number of Fisher Scoring iterations: 4
```

```
# Cloglog
randomforestmodcloglog <- glm(highquality ~ alcohol + sulphates + total_sulfur + vol_aci
dity, data = red, family = "binomial"(link = "cloglog"))
summary(randomforestmodcloglog)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     vol_acidity, family = binomial(link = "cloglog"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.5006  -0.9020   0.2185   0.9295   2.0506
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.958517   0.478252 -10.368  < 2e-16 ***
## alcohol       0.505807   0.038543  13.123  < 2e-16 ***
## sulphates     1.324184   0.221318   5.983 2.19e-09 ***
## total_sulfur -0.009109   0.001364  -6.679 2.41e-11 ***
## vol_acidity  -2.022997   0.238813  -8.471  < 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: 2209  on 1598  degrees of freedom
## Residual deviance: 1701  on 1594  degrees of freedom
## AIC: 1711
##
## Number of Fisher Scoring iterations: 7
```

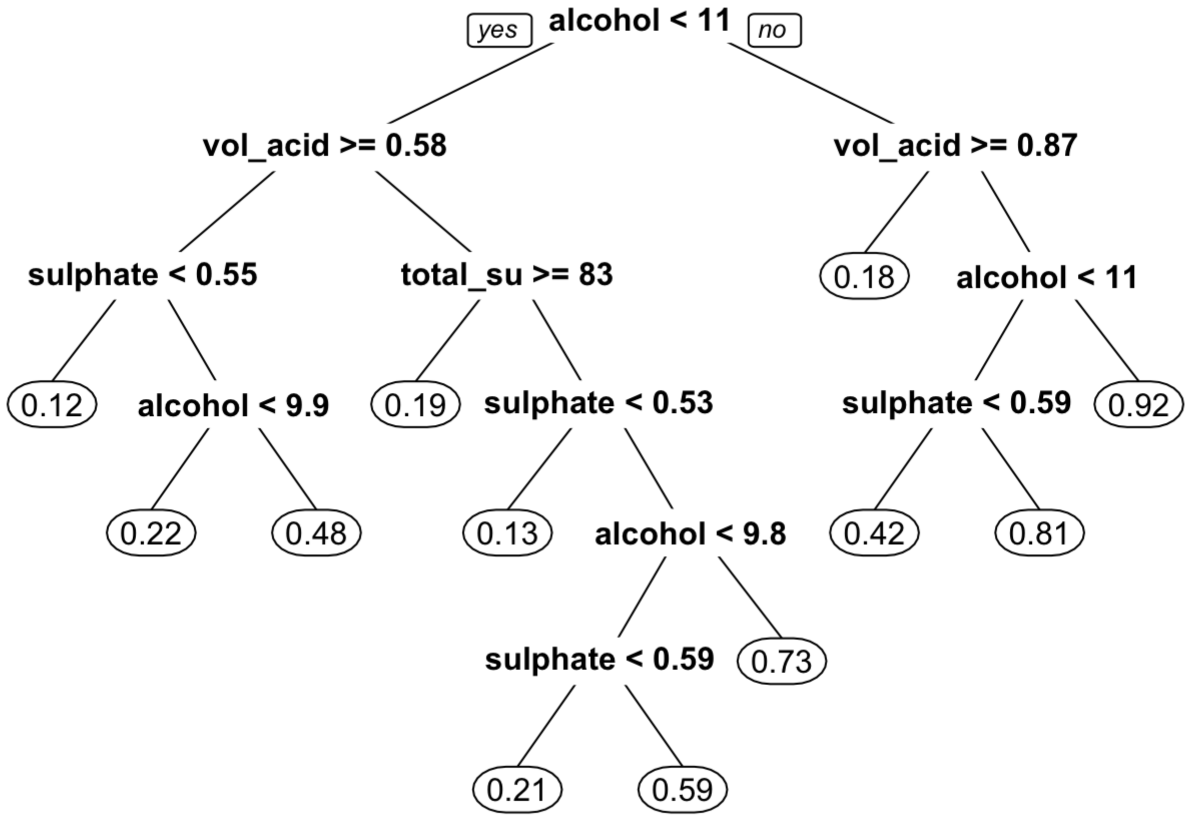
```
# The logit model performed better with a lower AIC value
```

Cart

```
library("caret")
library("e1071")
library("rpart")
library("rpart.plot")

cartmodel = rpart(highquality ~ . - quality, data = train)
print(cartmodel)
```

```
## n= 959
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 959 238.579800 0.5349322
##    2) alcohol< 10.525 589 137.826800 0.3735144
##      4) vol_acidity>=0.575 280  50.442860 0.2357143
##        8) sulphates< 0.545 109  11.449540 0.1192661 *
##        9) sulphates>=0.545 171  36.573100 0.3099415
##         18) alcohol< 9.85 111  18.810810 0.2162162 *
##         19) alcohol>=9.85 60  14.983330 0.4833333 *
##      5) vol_acidity< 0.575 309  77.249190 0.4983819
##       10) total_sulfur>=82.5 53   8.113208 0.1886792 *
##       11) total_sulfur< 82.5 256  63.000000 0.5625000
##         22) sulphates< 0.525 30   3.466667 0.1333333 *
##         23) sulphates>=0.525 226  53.274340 0.6194690
##           46) alcohol< 9.75 111  27.747750 0.5045045
##             92) sulphates< 0.585 24   3.958333 0.2083333 *
##             93) sulphates>=0.585 87  21.103450 0.5862069 *
##           47) alcohol>=9.75 115  22.643480 0.7304348 *
##    3) alcohol>=10.525 370  60.975680 0.7918919
##      6) vol_acidity>=0.87 17   2.470588 0.1764706 *
##      7) vol_acidity< 0.87 353  51.756370 0.8215297
##       14) alcohol< 11.45 182  36.263740 0.7252747
##         28) sulphates< 0.585 40   9.775000 0.4250000 *
##         29) sulphates>=0.585 142  21.866200 0.8098592 *
##       15) alcohol>=11.45 171  12.011700 0.9239766 *
```

```
prp(cartmodel)
```




```
# predictions on test set
set.seed(100)
predictTest = predict(cartmodel, newdata = test, type = "matrix")

# confusion matrix on test set
table(test$highquality, predictTest >= 0.5)
```

```
##
##      FALSE TRUE
##   0   216   82
##   1    86  256
```

Cart Model

```
# Logit
cartmodlogit <- glm(highquality ~ alcohol + sulphates + total_sulfur + vol_acidity, data
= red, family = "binomial"(link = "logit"))
summary(cartmodlogit)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     vol_acidity, family = binomial(link = "logit"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1638  -0.8675   0.3076   0.8629   2.3262
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.588813   0.795118 -10.802  < 2e-16 ***
## alcohol       0.927362   0.069268  13.388  < 2e-16 ***
## sulphates     2.059047   0.365976   5.626 1.84e-08 ***
## total_sulfur -0.011976   0.001924  -6.225 4.83e-10 ***
## vol_acidity  -3.083277   0.364832  -8.451  < 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: 2209.0  on 1598  degrees of freedom
## Residual deviance: 1684.2  on 1594  degrees of freedom
## AIC: 1694.2
##
## Number of Fisher Scoring iterations: 4
```

```
# Cloglog
cartmodcloglog <- glm(highquality ~ alcohol + sulphates + total_sulfur + fix_acidity, da
ta = red, family = "binomial"(link = "cloglog"))
summary(cartmodcloglog)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     fix_acidity, family = binomial(link = "cloglog"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7058  -0.9408   0.3075   0.9490   1.9387
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.835907   0.481268 -14.204  < 2e-16 ***
## alcohol       0.542953   0.037720  14.394  < 2e-16 ***
## sulphates     1.639060   0.217233   7.545 4.52e-14 ***
## total_sulfur -0.009315   0.001384  -6.732 1.67e-11 ***
## fix_acidity   0.027351   0.021284   1.285    0.199
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2209.0  on 1598  degrees of freedom
## Residual deviance: 1777.5  on 1594  degrees of freedom
## AIC: 1787.5
##
## Number of Fisher Scoring iterations: 18
```

```
# The logit model performed better with the lower AIC value
```

Compare best logit model with AIC

```
library("AICcmodavg")
```

```
##
## Attaching package: 'AICcmodavg'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
models <- list(randomforestmodlogit, cartmodlogit)
mod.names <- c('RandomForest', 'Cart')
aictab(cand.set = models, modnames = mod.names)
```

```
## Warning in aictab.AICglm.lm(cand.set = models, modnames = mod.names):
## Check model structure carefully as some models may be redundant
```

```
##
## Model selection based on AICc:
##
##               K    AICc Delta_AICc AICcWt Cum.Wt      LL
## RandomForest 5 1694.21          0    0.5    0.5 -842.09
## Cart         5 1694.21          0    0.5    1.0 -842.09
```

```
# The random forest logit model performed the best
```

Compare best model with BIC

```
library("flexmix")
BIC(randomforestmodlogit)
```

```
## [1] 1721.058
```

```
BIC(randomforestmodcloglog)
```

```
## [1] 1737.845
```

```
BIC(cartmodlogit)
```

```
## [1] 1721.058
```

```
BIC(cartmodcloglog)
```

```
## [1] 1814.418
```

```
# The random forest logit model performed the best
```

Confusion matrix for random forest logit model

```
confusionred = predict(randomforestmodlogit, newdata = red, type = "response")

# confusion matrix on test set
table(red$highquality, confusionred >= 0.5)
```

```
##
##      FALSE TRUE
##    0   548  196
##    1   216  639
```

Predictions for random forest logit model

```
pred_test <- predict(randomforestmodlogit, test, type = "response")

pred_test
```

```
##          1          2          8         10         12         14         15 
## 0.21686414 0.16572704 0.35364079 0.50812152 0.50812152 0.69723806 0.13086405 
##         16         17         18         19         20         25         27 
## 0.13867752 0.64471340 0.56805457 0.20109254 0.62465624 0.49785778 0.51001918 
##         29         30         33         35         37         41         42 
## 0.20629006 0.38187954 0.17799414 0.37517558 0.65305951 0.60541307 0.20653385 
##         45         46         47         52         54         56         60 
## 0.28787990 0.91140600 0.05638181 0.41159910 0.21158666 0.24266029 0.37446288 
##         61         67         69         70         73         74         76 
## 0.35540196 0.36302140 0.76772166 0.66550098 0.11468249 0.17116376 0.67706290 
##         77         78         79         81         82         83         84 
## 0.67706290 0.41857071 0.19429727 0.38434763 0.65378965 0.19907576 0.44994570 
##         86         90         91         95         97         98        102 
## 0.45450681 0.20581734 0.12488112 0.11648871 0.45180389 0.36239110 0.59957642 
##        104        106        107        108        110        113        115 
## 0.17168420 0.17168420 0.65531540 0.25986779 0.05669027 0.21064410 0.46170336 
##        116        119        122        127        129        131        143 
## 0.61534547 0.59589245 0.59589245 0.15237332 0.73731458 0.08091771 0.98548676 
##        144        145        146        147        148        150        152 
## 0.47019352 0.98548676 0.08710726 0.18741376 0.29279616 0.65840863 0.86018228 
##        154        162        164        168        172        173        175 
## 0.24118269 0.45773435 0.10323416 0.18953659 0.42272086 0.42272086 0.40084611 
##        178        180        181        182        183        185        186 
## 0.58567146 0.27931853 0.27931853 0.25686482 0.16145478 0.19399307 0.38321589 
##        187        190        191        193        199        200        208 
## 0.28302491 0.13218057 0.16007768 0.12192200 0.81955290 0.29660130 0.11227661 
##        209        212        215        219        224        229        230 
## 0.27161312 0.23949363 0.30759231 0.37301967 0.39670029 0.56522890 0.64457164 
##        231        232        234        238        243        244        245 
## 0.77766942 0.47608964 0.64457164 0.20849103 0.17002266 0.67646997 0.67646997 
##        248        249        250        252        260        267        268 
## 0.14653731 0.35125228 0.56422747 0.44429231 0.80917975 0.19291723 0.97152743 
##        269        272        273        283        285        287        291 
## 0.50344768 0.87483178 0.61403829 0.22895580 0.32999609 0.63496399 0.69463023 
##        295        301        305        308        315        320        322 
## 0.55129126 0.59572880 0.11220744 0.51647504 0.73885733 0.29176280 0.22451563 
##        326        327        330        333        335        338        343 
## 0.32170068 0.85787373 0.47496687 0.12860589 0.79014250 0.56472573 0.66228068 
##        345        346        352        353        356        359        360 
## 0.68982583 0.34930907 0.28077747 0.31982246 0.86315416 0.76935573 0.55892705 
##        363        368        369        371        372        374        377 
## 0.40286206 0.28955508 0.37745429 0.28282101 0.55754674 0.21115302 0.87515687 
##        379        381        384        389        390        391        392 
## 0.97217666 0.65714868 0.65714868 0.35745073 0.59089106 0.80025092 0.62763464 
##        393        394        396        397        399        400        402 
## 0.45488358 0.09608587 0.95715624 0.11072664 0.63287534 0.16572036 0.75725600 
##        406        409        413        414        419        427        428 
## 0.74785302 0.89010131 0.13582974 0.92335864 0.77020269 0.58695311 0.40277653 
##        431        434        436        437        441        442        443 
## 0.91967155 0.45313982 0.45313982 0.31324574 0.70777919 0.84990517 0.63537526 
##        444        447        448        452        456        458        461 
## 0.86779419 0.64968242 0.83321168 0.64119919 0.95768591 0.22920186 0.82420951 
```

```
##        465        467        468        469        473        477        478 
## 0.55427468 0.80312295 0.98771509 0.46457042 0.68760803 0.69504790 0.94349416 
##        480        482        484        486        489        490        491 
## 0.43048988 0.95378202 0.72818356 0.35050201 0.82229147 0.71109362 0.38820824 
##        492        493        494        495        497        499        501 
## 0.97829349 0.98423867 0.64444930 0.81342325 0.27636155 0.84287574 0.27636155 
##        502        506        507        508        514        515        520 
## 0.88317746 0.94197356 0.92658894 0.36855117 0.87663195 0.87663195 0.69166380 
##        521        523        526        528        542        544        547 
## 0.79768845 0.43596448 0.34605728 0.79882025 0.81658345 0.60467189 0.44918649 
##        551        553        556        563        565        567        568 
## 0.37808419 0.56543070 0.72091694 0.14387840 0.92935383 0.20667837 0.20667837 
##        573        574        575        576        577        579        580 
## 0.77494478 0.28595807 0.53585414 0.75659792 0.51154935 0.26273769 0.52861287 
##        581        583        584        587        588        595        598 
## 0.38002007 0.29689198 0.69766881 0.86089516 0.09780108 0.20381128 0.49453337 
##        599        600        602        605        610        614        620 
## 0.33386682 0.35305750 0.28353983 0.19528536 0.93597592 0.61614952 0.58672865 
##        622        623        627        628        630        632        638 
## 0.17261769 0.30620045 0.23676532 0.23676532 0.13683730 0.60526652 0.03693294 
##        641        644        645        646        647        648        651 
## 0.32321697 0.24690166 0.32321697 0.52329572 0.44260928 0.50449082 0.40958829 
##        653        657        658        661        662        665        668 
## 0.99329501 0.40958829 0.62339714 0.53465925 0.32678756 0.58642668 0.51847411 
##        671        672        674        676        678        679        680 
## 0.52667590 0.24700448 0.24700448 0.63951248 0.22140871 0.15831687 0.50931169 
##        681        684        685        686        687        688        692 
## 0.26144196 0.61274001 0.02353841 0.61274001 0.19820088 0.30853776 0.08792767 
##        695        697        704        705        706        711        713 
## 0.17339004 0.33118031 0.39592542 0.26438374 0.09514858 0.09703634 0.18728926 
##        716        717        720        722        724        727        728 
## 0.27155586 0.30467669 0.22428109 0.15248206 0.74648551 0.59402154 0.43341582 
##        730        732        734        735        736        738        741 
## 0.69434981 0.56601333 0.31147678 0.29287694 0.11882456 0.24236950 0.71932009 
##        742        744        748        752        754        760        761 
## 0.12964712 0.33233049 0.32846257 0.24423781 0.24423781 0.20828698 0.18944864 
##        765        766        769        774        775        782        786 
## 0.21925919 0.21287996 0.19045961 0.58787221 0.57996000 0.49407054 0.36376503 
##        793        797        798        800        802        804        806 
## 0.21742711 0.41405130 0.81010666 0.66715805 0.42345371 0.35326525 0.97678970 
##        809        810        812        813        818        819        822 
## 0.48396705 0.60727563 0.74847855 0.74001422 0.93891093 0.15085766 0.98401461 
##        824        825        828        832        834        835        836 
## 0.43070016 0.59452832 0.61809251 0.75438553 0.57515459 0.21302698 0.16263773 
##        838        839        840        842        843        844        846 
## 0.73172430 0.90163251 0.45005102 0.46356092 0.65608836 0.13891921 0.38077515 
##        848        853        854        860        862        867        868 
## 0.37044591 0.46348365 0.77796635 0.81336010 0.29801703 0.85828874 0.84439382 
##        872        873        877        878        884        886        887 
## 0.53427892 0.53560900 0.74967371 0.68842602 0.17504411 0.44881228 0.39943319 
##        890        893        894        896        898        901        902 
## 0.07413691 0.44299541 0.18472125 0.54837049 0.54837049 0.91308562 0.61749313 
```

```
##        903        907        911        917        920        922        923
## 0.61749313 0.61655717 0.97132513 0.47643991 0.84865159 0.73857291 0.84865159
##        924        927        930        931        933        935        941
## 0.57254719 0.75197156 0.94254625 0.61571145 0.35714165 0.61571145 0.95924086
##        943        944        946        950        951        952        953
## 0.51350024 0.39208642 0.85009985 0.96739607 0.96739607 0.96007673 0.89304382
##        956        958        959        962        963        964        966
## 0.81941210 0.75906185 0.76200136 0.34752279 0.42379344 0.88720837 0.85318054
##        970        972        975        977        978        981        983
## 0.53399341 0.91601469 0.93910091 0.28346202 0.08591010 0.70374096 0.93219598
##        986        987        988        989        992        993        994
## 0.76216515 0.87904782 0.31207909 0.40407434 0.32169795 0.33637960 0.32169795
##        995        997        998        999       1003       1004       1005
## 0.24614218 0.91773179 0.91773179 0.13340402 0.94122194 0.96113831 0.55234587
##       1007       1008       1017       1021       1022       1023       1024
## 0.94122194 0.94084017 0.95824366 0.91151720 0.91151720 0.64109441 0.94561683
##       1025       1028       1032       1039       1040       1043       1044
## 0.57387540 0.53740392 0.71727212 0.94854700 0.80300719 0.80300719 0.82020006
##       1045       1054       1055       1058       1059       1060       1061
## 0.89700536 0.97117164 0.14570490 0.20603080 0.81405117 0.90442354 0.77354568
##       1064       1077       1078       1079       1089       1092       1095
## 0.92780908 0.91711368 0.82775448 0.82775448 0.59911292 0.87464611 0.49531036
##       1100       1111       1114       1116       1120       1121       1123
## 0.34983074 0.55003024 0.64019617 0.67849027 0.91323567 0.95982776 0.87671921
##       1126       1128       1130       1131       1133       1134       1136
## 0.88512330 0.58545658 0.59990536 0.59465043 0.98621359 0.81790944 0.90991385
##       1140       1142       1146       1147       1152       1154       1155
## 0.25663548 0.79255354 0.81699614 0.62188669 0.84747620 0.78814787 0.69856005
##       1156       1162       1163       1164       1165       1166       1168
## 0.32861855 0.68472828 0.93097631 0.34911691 0.34911691 0.64143459 0.95811695
##       1169       1170       1171       1173       1176       1178       1180
## 0.89257541 0.83796361 0.78148743 0.94556825 0.71702966 0.91536349 0.83725702
##       1182       1184       1187       1190       1193       1194       1195
## 0.83146700 0.18325571 0.75844840 0.12264121 0.96428659 0.39866755 0.16205917
##       1196       1198       1201       1203       1204       1205       1209
## 0.36820273 0.43683111 0.43683111 0.90407751 0.12664335 0.84882533 0.84882533
##       1210       1211       1212       1213       1222       1224       1227
## 0.89081590 0.53707859 0.33645556 0.53707859 0.89545422 0.93813689 0.19332528
##       1228       1229       1230       1236       1244       1247       1248
## 0.43944184 0.95997919 0.35954537 0.70226268 0.22116768 0.23445159 0.60960225
##       1249       1250       1251       1256       1258       1259       1262
## 0.82547588 0.61830476 0.61830476 0.50363201 0.60797958 0.69403671 0.37701038
##       1263       1265       1266       1268       1269       1270       1271
## 0.38122840 0.88943306 0.50245646 0.90213067 0.37253912 0.97152715 0.98284751
##       1273       1274       1275       1279       1282       1286       1288
## 0.71983985 0.19873507 0.66138806 0.13504841 0.57652097 0.64986391 0.90845738
##       1291       1292       1293       1294       1301       1306       1307
## 0.59908827 0.59190236 0.93827703 0.25512295 0.87211203 0.20683713 0.23809254
##       1308       1310       1311       1312       1313       1315       1318
## 0.58216493 0.18736687 0.20683713 0.93088901 0.11783860 0.48312749 0.88286689
##       1319       1321       1324       1325       1326       1327       1328
## 0.20771741 0.27840724 0.82228483 0.65726913 0.65726913 0.65726913 0.65726913
```

```
##       1330       1331       1333       1334       1339       1343       1346 
## 0.21628945 0.21628945 0.44771727 0.16577829 0.33063230 0.51740015 0.56699331 
##       1348       1350       1358       1365       1367       1376       1379 
## 0.23966391 0.56190596 0.68815611 0.76732852 0.21556429 0.22071490 0.47985390 
##       1394       1395       1396       1397       1399       1406       1409 
## 0.38905380 0.15440151 0.33954980 0.34921494 0.49039723 0.92222761 0.96956379 
##       1411       1416       1417       1419       1420       1421       1424 
## 0.55013345 0.46359329 0.67996830 0.39766821 0.11430274 0.39766821 0.72836169 
##       1433       1437       1438       1441       1442       1443       1447 
## 0.92574770 0.16556629 0.38062331 0.87184058 0.09511377 0.43582198 0.43582198 
##       1448       1450       1453       1455       1456       1461       1467 
## 0.35375316 0.87394441 0.78758198 0.78198047 0.45960983 0.59735007 0.47414501 
##       1468       1472       1473       1474       1475       1482       1486 
## 0.30916495 0.86017441 0.84882869 0.58301194 0.19663663 0.81973890 0.31103235 
##       1488       1490       1493       1496       1501       1506       1508 
## 0.66158978 0.75070329 0.79984815 0.77693701 0.26400037 0.36554744 0.85817699 
##       1510       1511       1515       1517       1523       1525       1526 
## 0.91307103 0.57620576 0.12840124 0.82595385 0.82595385 0.60658881 0.45939419 
##       1529       1530       1532       1533       1534       1538       1541 
## 0.63560644 0.34422384 0.42672777 0.52363402 0.23235943 0.50712482 0.81445748 
##       1542       1543       1545       1546       1547       1548       1553 
## 0.84898279 0.36046967 0.89062172 0.45450879 0.51435731 0.82069018 0.62386686 
##       1556       1557       1558       1562       1563       1568       1570 
## 0.64128386 0.23011305 0.48014133 0.14727854 0.37557065 0.37557065 0.78071517 
##       1574       1575       1582       1584       1586       1592       1593 
## 0.88033989 0.51978506 0.73417847 0.30660149 0.88371638 0.67450739 0.75118218 
##       1594       1598       1599 
## 0.38765251 0.45044094 0.81940411
```

AUC and ROC

```
library("pROC")
```

```
## Type 'citation("pROC")' for a citation.
```

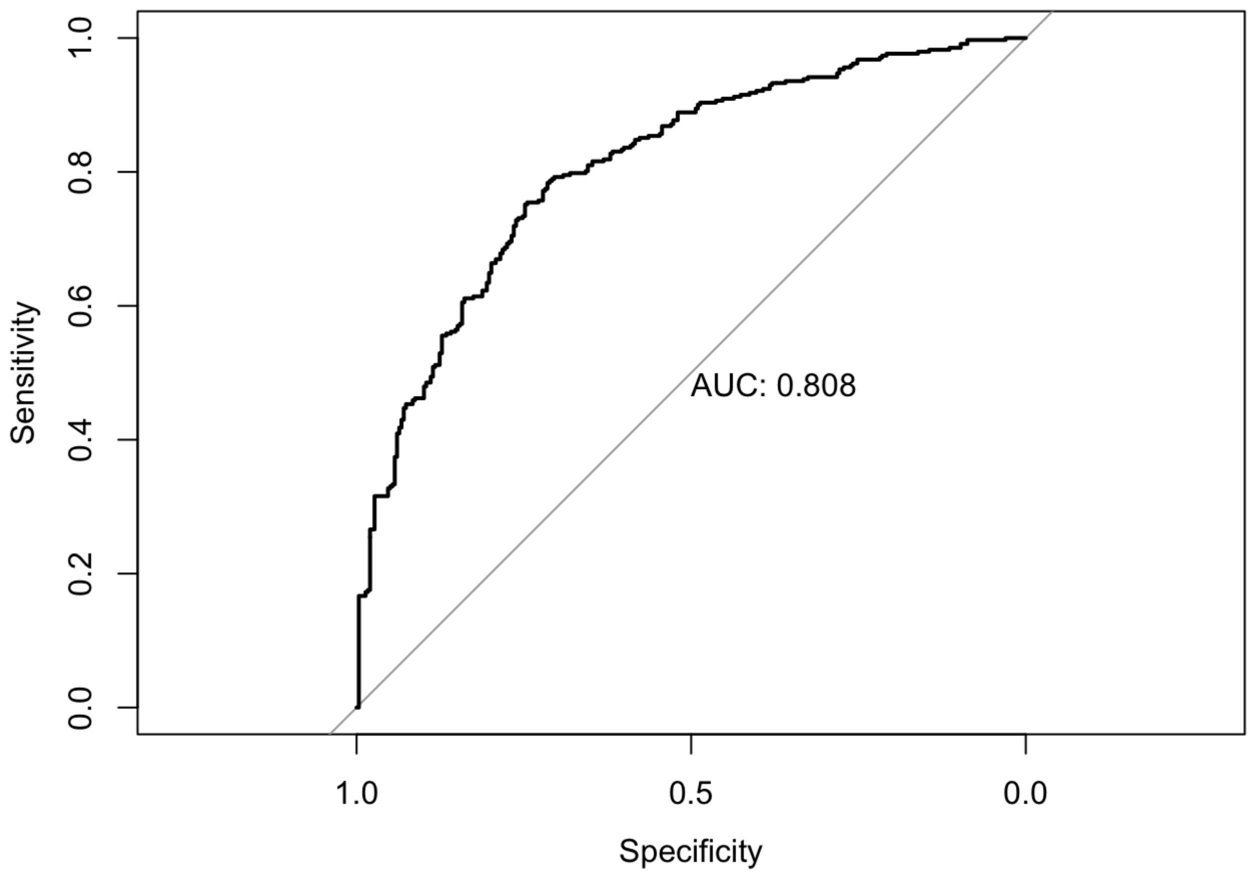
```
## 
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
## 
##     cov, smooth, var
```

```
test_prob = predict(randomforestmodlogit, test, type = "response")

test_roc = roc(test$highquality ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```




```
as.numeric(test_roc$auc)
```

```
## [1] 0.8083324
```

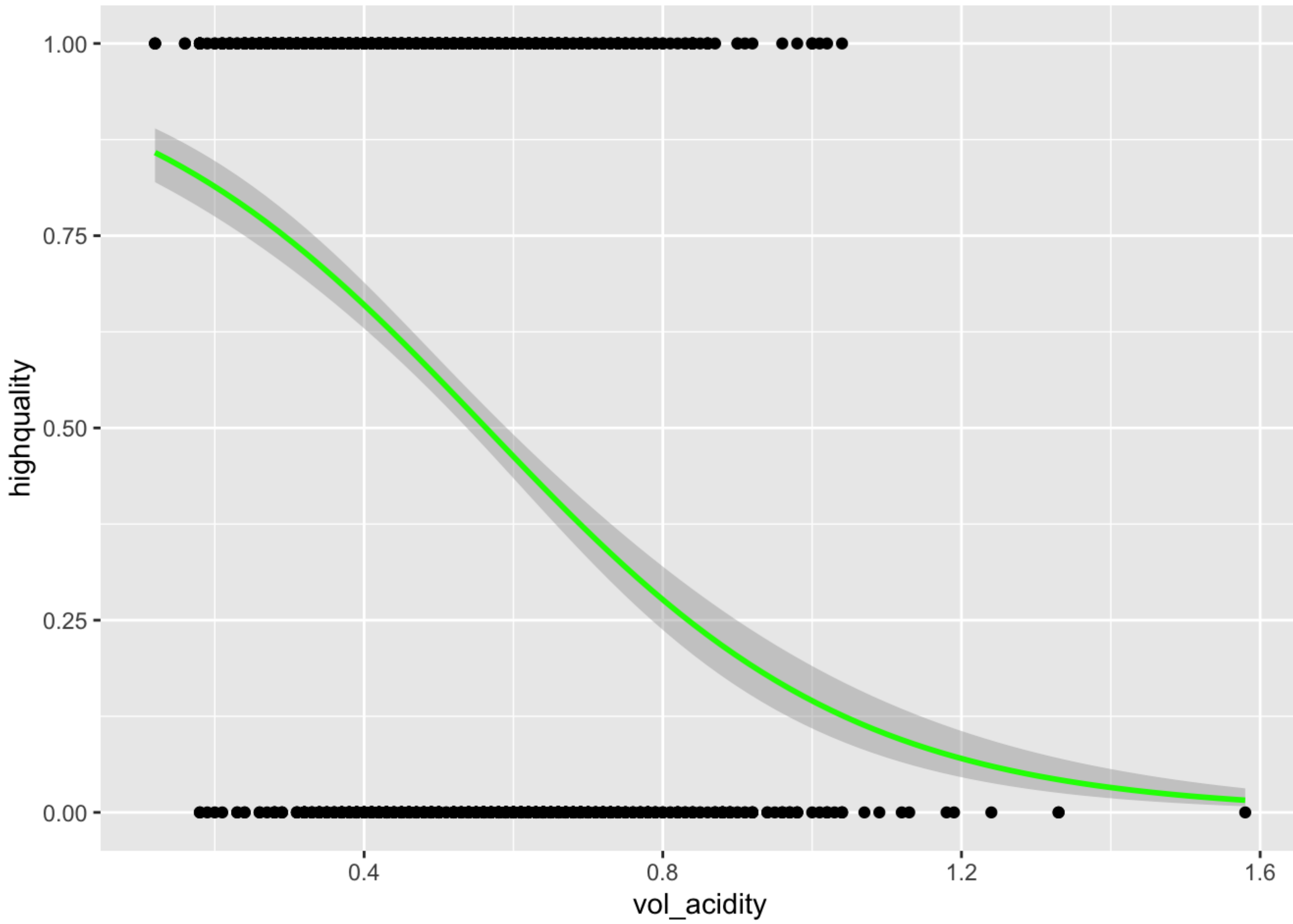
AUC and ROC with just one variable

```
library("ggplot2")

simple <- glm(highquality ~ vol_acidity, data = red, family = "binomial"(link = "logi
t"))
summary(simple)
```

```
##
## Call:
## glm(formula = highquality ~ vol_acidity, family = binomial(link = "logit"),
##     data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8697  -1.1148   0.7156   1.0375   2.0349
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.2874     0.1838   12.45   <2e-16 ***
## vol_acidity  -4.0607     0.3334  -12.18   <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: 2209.0  on 1598  degrees of freedom
## Residual deviance: 2033.4  on 1597  degrees of freedom
## AIC: 2037.4
##
## Number of Fisher Scoring iterations: 4
```

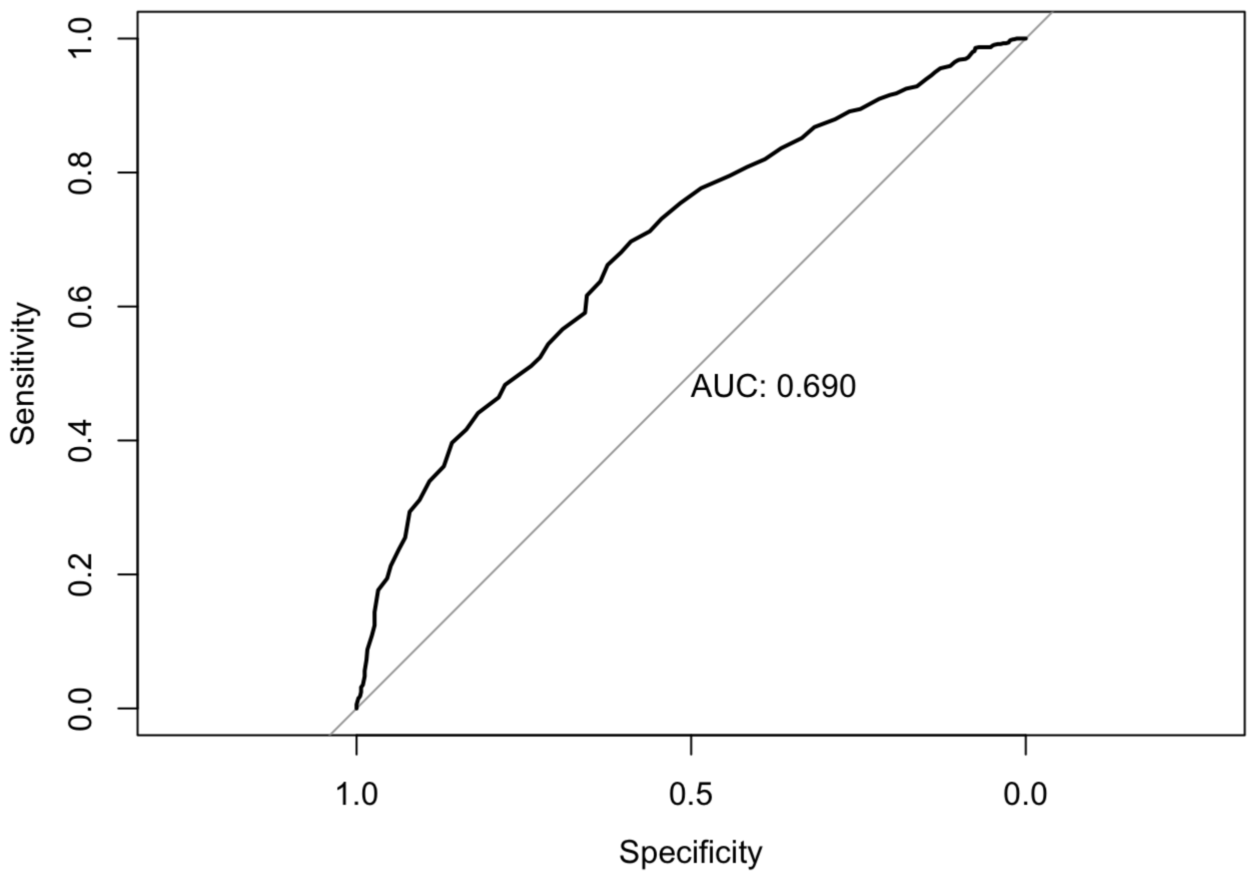
```
ggplot(red, aes(x = vol_acidity, y = highquality)) +geom_point()+stat_smooth(method="gl
m", color="green", se=TRUE, method.args = list(family=binomial))
```

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



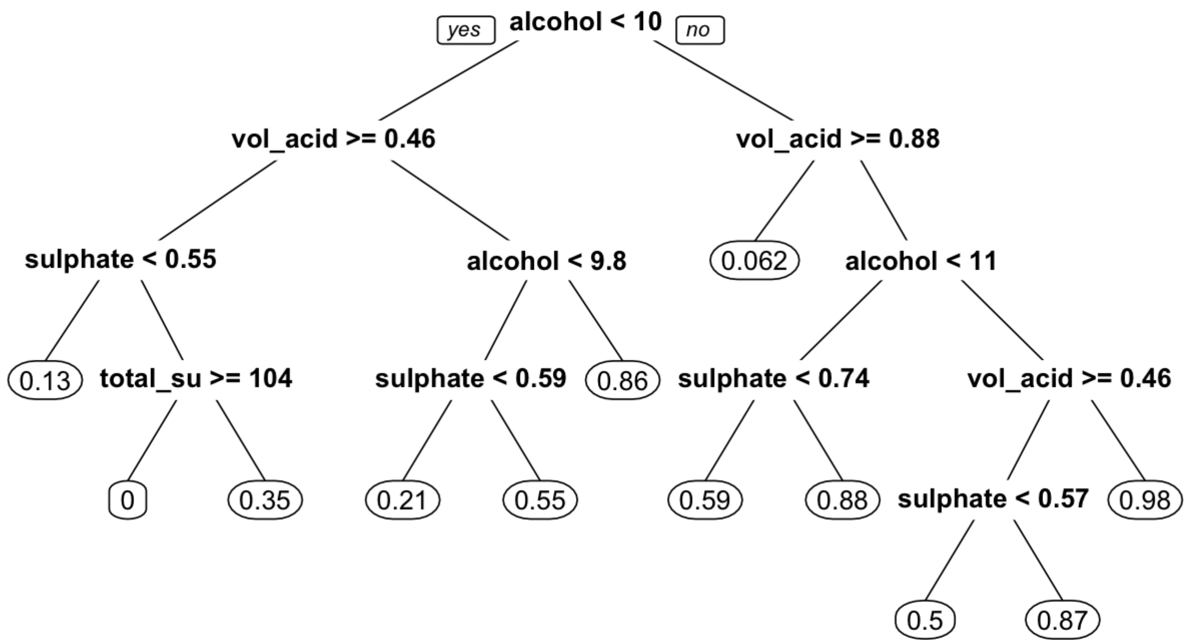

```
test_prop1 = predict(simple, red, type = "response")

test_roc1 = roc(red$highquality ~ test_prop1, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
as.numeric(test_roc1$auc)
```

```
## [1] 0.6900011
```




```
# predictions on test set
set.seed(100)
predictTest = predict(cartmodel, newdata = test, type = "matrix")

# confusion matrix on test set
table(test$highquality, predictTest >= 0.5)
```

```
##
##      FALSE TRUE
##   0    195  103
##   1     82  260
```

Cart Model

```
# Logit
cartmodlogit <- glm(highquality ~ alcohol + sulphates + total_sulfur + fix_acidity, data
= red, family = "binomial"(link = "logit"))
summary(cartmodlogit)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     fix_acidity, family = binomial(link = "logit"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3737  -0.9154   0.3562   0.8762   2.0206
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -12.172146   0.828396 -14.694  < 2e-16 ***
## alcohol        0.989178   0.068276  14.488  < 2e-16 ***
## sulphates      2.587844   0.370028   6.994 2.68e-12 ***
## total_sulfur  -0.011171   0.001895  -5.895 3.75e-09 ***
## fix_acidity    0.109461   0.035511   3.082  0.00205 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2209.0  on 1598  degrees of freedom
## Residual deviance: 1752.5  on 1594  degrees of freedom
## AIC: 1762.5
##
## Number of Fisher Scoring iterations: 4
```

```
# Cloglog
cartmodcloglog <- glm(highquality ~ alcohol + sulphates + total_sulfur + fix_acidity, da
ta = red, family = "binomial"(link = "cloglog"))
summary(cartmodcloglog)
```

```
##
## Call:
## glm(formula = highquality ~ alcohol + sulphates + total_sulfur +
##     fix_acidity, family = binomial(link = "cloglog"), data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7058  -0.9408   0.3075   0.9490   1.9387
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.835907   0.481268 -14.204  < 2e-16 ***
## alcohol       0.542953   0.037720  14.394  < 2e-16 ***
## sulphates     1.639060   0.217233   7.545 4.52e-14 ***
## total_sulfur -0.009315   0.001384  -6.732 1.67e-11 ***
## fix_acidity   0.027351   0.021284   1.285    0.199
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2209.0  on 1598  degrees of freedom
## Residual deviance: 1777.5  on 1594  degrees of freedom
## AIC: 1787.5
##
## Number of Fisher Scoring iterations: 18
```

```
# The logit model performed better with the lower AIC value
```

Compare best logit model with AIC

```
library("AICcmodavg")
```

```
##
## Attaching package: 'AICcmodavg'
```

```
## The following object is masked from 'package:randomForest':
##
##     importance
```

```
models <- list(randomforestmodlogit, cartmodlogit)
mod.names <- c('RandomForest', 'Cart')
aictab(cand.set = models, modnames = mod.names)
```

```
##
## Model selection based on AICc:
##
##              K    AICc Delta_AICc AICcWt Cum.Wt      LL
## RandomForest 5 1694.21       0.00      1      1 -842.09
## Cart         5 1762.56      68.35      0      1 -876.26
```

```
# The random forest logit model performed the best
```

Compare best model with BIC

```
library("flexmix")
BIC(randomforestmodlogit)
```

```
## [1] 1721.058
```

```
BIC(randomforestmodcloglog)
```

```
## [1] 1737.845
```

```
BIC(cartmodlogit)
```

```
## [1] 1789.404
```

```
BIC(cartmodcloglog)
```

```
## [1] 1814.418
```

```
# The random forest logit model performed the best
```

Confusion matrix for random forest logit model

```
confusionred = predict(randomforestmodlogit, newdata = red, type = "response")

# confusion matrix on test set
table(red$highquality, confusionred >= 0.5)
```

```
##
##     FALSE TRUE
##   0   548  196
##   1   216  639
```

Predictions for random forest logit model

```
pred_test <- predict(randomforestmodlogit, test, type = "response")

pred_test
```

```
##          3          4          5         10         16         18         19 
## 0.24006880 0.52789064 0.21686414 0.50812152 0.13867752 0.56805457 0.20109254 
##         22         28         30         38         41         47         48 
## 0.42364540 0.58076699 0.38187954 0.59357547 0.60541307 0.05638181 0.51951414 
##         49         51         53         56         59         60         65 
## 0.43019840 0.30073364 0.43459537 0.24266029 0.36211052 0.37446288 0.48498736 
##         66         70         71         73         74         77         80 
## 0.48498736 0.66550098 0.28235396 0.11468249 0.17116376 0.67706290 0.22946232 
##         81         84         85         87         88         91         92 
## 0.38434763 0.44994570 0.71372134 0.81269343 0.43771981 0.12488112 0.81269343 
##         97        100        102        104        105        106        110 
## 0.45180389 0.24182616 0.59957642 0.17168420 0.30465596 0.17168420 0.05669027 
##        111        112        114        117        119        120        122 
## 0.46170336 0.20639675 0.56940622 0.44937409 0.59589245 0.10540486 0.59589245 
##        125        130        133        136        141        144        145 
## 0.17454268 0.47573192 0.88460563 0.25641373 0.25641373 0.47019352 0.98548676 
##        147        151        153        155        159        161        163 
## 0.18741376 0.71199465 0.24118269 0.44034290 0.21501940 0.15424192 0.45821490 
##        167        168        172        177        185        188        189 
## 0.22409934 0.18953659 0.42272086 0.40084611 0.19399307 0.25256465 0.13003255 
##        190        192        193        201        203        208        211 
## 0.13218057 0.51206935 0.12192200 0.79856559 0.43027971 0.11227661 0.94372425 
##        212        214        215        216        217        218        220 
## 0.23949363 0.29462991 0.30759231 0.43866216 0.56377995 0.14689457 0.12986042 
##        222        224        228        230        231        232        235 
## 0.23247399 0.39670029 0.26752645 0.64457164 0.77766942 0.47608964 0.06682702 
##        237        238        242        243        244        245        247 
## 0.19063712 0.20849103 0.82087382 0.17002266 0.67646997 0.67646997 0.19535300 
##        248        249        251        254        256        264        265 
## 0.14653731 0.35125228 0.70288800 0.11931143 0.12202885 0.42251877 0.77916500 
##        266        269        272        276        279        285        286 
## 0.80711622 0.50344768 0.87483178 0.57130009 0.97446971 0.32999609 0.32999609 
##        287        289        290        295        300        301        302 
## 0.63496399 0.69463023 0.36131597 0.55129126 0.33509483 0.59572880 0.79587338 
##        303        304        307        313        320        321        324 
## 0.41165107 0.27204195 0.20435986 0.25470497 0.29176280 0.68916140 0.26289903 
##        326        327        328        331        333        338        340 
## 0.32170068 0.85787373 0.90657600 0.94054119 0.12860589 0.56472573 0.88983001 
##        344        345        347        354        363        364        366 
## 0.66228068 0.68982583 0.84745884 0.92386355 0.40286206 0.74339678 0.93834097 
##        367        370        373        375        376        377        378 
## 0.44355920 0.97850481 0.87743922 0.78031275 0.90189455 0.87515687 0.97850481 
##        380        381        382        384        385        392        396 
## 0.70150349 0.65714868 0.62763464 0.65714868 0.32025802 0.62763464 0.95715624 
##        397        405        413        419        420        422        423 
## 0.11072664 0.20675079 0.13582974 0.77020269 0.26802335 0.69125350 0.22216089 
##        425        428        430        435        441        445        446 
## 0.22216089 0.40277653 0.32784541 0.62568563 0.70777919 0.93413308 0.32689043 
##        451        460        462        463        465        467        472 
## 0.74289381 0.19464737 0.29995555 0.92477154 0.55427468 0.80312295 0.76084070 
##        474        478        480        481        484        485        488 
## 0.83494087 0.94349416 0.43048988 0.54068853 0.72818356 0.97215159 0.34928339 
```

```
##        489        490        491        494        497        498        501 
## 0.82229147 0.71109362 0.38820824 0.64444930 0.27636155 0.71705477 0.27636155 
##        506        508        510        512        515        519        522 
## 0.94197356 0.36855117 0.87805407 0.40270830 0.87663195 0.93281411 0.35179506 
##        523        525        526        527        539        540        541 
## 0.43596448 0.23574293 0.34605728 0.69166380 0.91684119 0.79067776 0.33434428 
##        545        546        549        550        551        553        555 
## 0.42365538 0.16061952 0.66697078 0.36606459 0.37808419 0.56543070 0.72091694 
##        558        561        563        565        567        569        571 
## 0.72091694 0.74635299 0.14387840 0.92935383 0.20667837 0.76484762 0.87692510 
##        574        575        578        580        582        583        587 
## 0.28595807 0.53585414 0.22627342 0.52861287 0.38002007 0.29689198 0.86089516 
##        590        593        595        597        598        600        602 
## 0.83511782 0.33060442 0.20381128 0.43790682 0.49453337 0.35305750 0.28353983 
##        607        611        613        616        621        624        626 
## 0.94521424 0.40906300 0.40122182 0.24600419 0.16493964 0.92396940 0.38478177 
##        632        633        638        639        640        641        644 
## 0.60526652 0.63251372 0.03693294 0.28835011 0.89903179 0.32321697 0.24690166 
##        649        652        653        655        657        658        660 
## 0.79606704 0.19901970 0.99329501 0.34792620 0.40958829 0.62339714 0.51004428 
##        664        665        671        672        673        677        683 
## 0.81425791 0.58642668 0.52667590 0.24700448 0.01313453 0.49457701 0.37713648 
##        685        688        691        692        693        695        696 
## 0.02353841 0.30853776 0.19926076 0.08792767 0.47830984 0.17339004 0.92968944 
##        697        699        703        704        706        708        711 
## 0.33118031 0.21822469 0.32282879 0.39592542 0.09514858 0.56086414 0.09703634 
##        713        715        716        719        720        723        731 
## 0.18728926 0.16052021 0.27155586 0.31642036 0.22428109 0.50744957 0.39042992 
##        733        734        737        740        743        744        746 
## 0.17713379 0.31147678 0.11882456 0.21223885 0.30029207 0.33233049 0.45423361 
##        749        750        751        753        754        756        759 
## 0.47090034 0.45423361 0.24423781 0.25521181 0.24423781 0.24061283 0.17874689 
##        762        764        766        770        771        772        773 
## 0.21832154 0.21832154 0.21287996 0.23550990 0.19045961 0.07048717 0.09786165 
##        775        776        779        781        783        789        790 
## 0.57996000 0.16206695 0.66183909 0.28239984 0.12989081 0.37973849 0.09437875 
##        791        801        805        810        811        814        815 
## 0.42161900 0.11964960 0.48671568 0.60727563 0.60772687 0.86887538 0.84554487 
##        819        820        824        825        827        836        837 
## 0.15085766 0.21151380 0.43070016 0.59452832 0.88097191 0.16263773 0.73172430 
##        838        839        840        841        842        847        850 
## 0.73172430 0.90163251 0.45005102 0.92522550 0.46356092 0.38077515 0.38523107 
##        851        852        853        854        855        856        860 
## 0.44134957 0.44134957 0.46348365 0.77796635 0.77796635 0.78141884 0.81336010 
##        862        863        866        868        870        873        877 
## 0.29801703 0.59081792 0.18470531 0.84439382 0.59381839 0.53560900 0.74967371 
##        878        882        883        885        886        887        888 
## 0.68842602 0.66426561 0.94676764 0.27035098 0.44881228 0.39943319 0.87920584 
##        890        891        892        895        898        899        900 
## 0.07413691 0.57461423 0.18472125 0.21369246 0.54837049 0.95762419 0.34173363 
##        903        904        908        909        912        917        922 
## 0.61749313 0.64721685 0.79467120 0.57631602 0.85616038 0.47643991 0.73857291 
```

```
##        923        924        925        927        928        930        935 
## 0.84865159 0.57254719 0.84029182 0.75197156 0.14108907 0.94254625 0.61571145 
##        939        940        943        945        948        950        952 
## 0.96015192 0.82418379 0.51350024 0.93921165 0.96007673 0.96739607 0.96007673 
##        953        955        958        964        965        966        976 
## 0.89304382 0.91523844 0.75906185 0.88720837 0.82503678 0.85318054 0.28346202 
##        977        981        982        983        984        988        991 
## 0.28346202 0.70374096 0.26394962 0.93219598 0.70374096 0.31207909 0.40407434 
##        994        995       1000       1002       1004       1007       1008 
## 0.32169795 0.24614218 0.89973442 0.78374144 0.96113831 0.94122194 0.94084017 
##       1012       1019       1021       1023       1024       1027       1029 
## 0.72592191 0.90328264 0.91151720 0.64109441 0.94561683 0.94593820 0.50667880 
##       1031       1032       1036       1042       1048       1051       1052 
## 0.77233191 0.71727212 0.77472663 0.49873341 0.60752858 0.60752858 0.67804325 
##       1054       1059       1064       1067       1068       1072       1078 
## 0.97117164 0.81405117 0.92780908 0.92897417 0.89593617 0.08796075 0.82775448 
##       1079       1080       1082       1086       1087       1089       1091 
## 0.82775448 0.40453879 0.37324283 0.28421079 0.91462483 0.59911292 0.87689507 
##       1092       1094       1096       1098       1099       1102       1103 
## 0.87464611 0.95181067 0.66933405 0.34983074 0.97135198 0.88355318 0.74464695 
##       1108       1110       1111       1115       1116       1118       1120 
## 0.94262703 0.67961164 0.55003024 0.96048872 0.67849027 0.67849027 0.91323567 
##       1121       1124       1126       1129       1131       1133       1134 
## 0.95982776 0.82569184 0.88512330 0.42390723 0.59465043 0.98621359 0.81790944 
##       1137       1139       1141       1142       1145       1147       1150 
## 0.85680402 0.20660676 0.34979431 0.79255354 0.55703889 0.62188669 0.91677782 
##       1153       1157       1159       1160       1164       1169       1182 
## 0.32861855 0.90970330 0.84798042 0.75007048 0.34911691 0.89257541 0.83146700 
##       1184       1185       1186       1187       1191       1195       1199 
## 0.18325571 0.39228204 0.83216767 0.75844840 0.93726418 0.16205917 0.80568967 
##       1200       1201       1203       1205       1208       1209       1211 
## 0.18261742 0.43683111 0.90407751 0.84882533 0.44462253 0.84882533 0.53707859 
##       1212       1214       1215       1217       1218       1221       1222 
## 0.33645556 0.79664197 0.71480329 0.25697732 0.90659821 0.89545422 0.89545422 
##       1225       1228       1230       1231       1233       1234       1235 
## 0.77019805 0.43944184 0.35954537 0.93764299 0.35954537 0.47608892 0.82878055 
##       1236       1242       1246       1247       1248       1252       1257 
## 0.70226268 0.44890191 0.60960225 0.23445159 0.60960225 0.31158618 0.15058729 
##       1258       1260       1263       1264       1265       1267       1270 
## 0.60797958 0.69403671 0.38122840 0.18918454 0.88943306 0.50245646 0.97152715 
##       1272       1277       1278       1281       1284       1285       1288 
## 0.78782193 0.91209807 0.35315581 0.57652097 0.35403270 0.72038518 0.90845738 
##       1289       1292       1293       1296       1298       1299       1300 
## 0.52796774 0.59190236 0.93827703 0.15149263 0.87066317 0.85966749 0.06683777 
##       1301       1303       1304       1309       1310       1312       1315 
## 0.87211203 0.89325440 0.81440601 0.23809254 0.18736687 0.93088901 0.48312749 
##       1317       1320       1322       1323       1325       1326       1328 
## 0.79987457 0.37260225 0.79987457 0.86432689 0.65726913 0.65726913 0.65726913 
##       1330       1331       1339       1341       1343       1350       1351 
## 0.21628945 0.21628945 0.33063230 0.57850387 0.51740015 0.56190596 0.29613263 
##       1354       1355       1356       1357       1366       1369       1371 
## 0.31505119 0.40899359 0.57777135 0.57441114 0.36658102 0.15039302 0.30081941 
```

```
##       1372       1374       1376       1377       1378       1379       1380
## 0.91160867 0.08195041 0.22071490 0.18491208 0.83265348 0.47985390 0.62870483
##       1384       1386       1387       1392       1396       1399       1400
## 0.18145955 0.06251845 0.32286340 0.60775822 0.33954980 0.49039723 0.69724047
##       1401       1403       1404       1407       1408       1409       1411
## 0.10211360 0.94896201 0.85527482 0.92066898 0.73685309 0.96956379 0.55013345
##       1422       1423       1426       1428       1430       1433       1435
## 0.35313054 0.80043906 0.66331164 0.81035058 0.92040632 0.92574770 0.18849461
##       1437       1438       1439       1443       1444       1445       1449
## 0.16556629 0.38062331 0.50395641 0.43582198 0.77179526 0.48436224 0.37160792
##       1450       1451       1453       1455       1456       1458       1466
## 0.87394441 0.87184058 0.78758198 0.78198047 0.45960983 0.17120492 0.33900665
##       1469       1470       1474       1476       1477       1479       1483
## 0.47414501 0.11210856 0.58301194 0.96416743 0.19663663 0.28090696 0.55985828
##       1484       1485       1486       1487       1488       1491       1492
## 0.80671170 0.47494678 0.31103235 0.45669868 0.66158978 0.95967599 0.79522301
##       1494       1497       1499       1500       1501       1505       1516
## 0.14135590 0.14135590 0.42318892 0.71090938 0.26400037 0.85817699 0.13049129
##       1517       1527       1528       1529       1530       1531       1532
## 0.82595385 0.43733113 0.70351630 0.63560644 0.34422384 0.85054796 0.42672777
##       1533       1534       1535       1536       1543       1552       1553
## 0.52363402 0.23235943 0.81214100 0.37906001 0.36046967 0.24069329 0.62386686
##       1558       1559       1560       1561       1567       1569       1571
## 0.48014133 0.12503776 0.14727854 0.14727854 0.89505671 0.25504396 0.96429949
##       1572       1576       1579       1581       1582       1585       1587
## 0.82810158 0.86164481 0.72404537 0.90818659 0.73417847 0.91848799 0.88684408
##       1594       1597       1598
## 0.38765251 0.75118218 0.45044094
```

Model Diagnostics

```
accuracy = (548+639)/(548+196+216+639)
accuracy
```

```
## [1] 0.742339
```

```
sensitivity = 639/(639+196)
sensitivity
```

```
## [1] 0.7652695
```

```
specificity = 548/(548+216)
specificity
```

```
## [1] 0.7172775
```

AUC and ROC

```
library("pROC")
```

```
## Type 'citation("pROC")' for a citation.
```

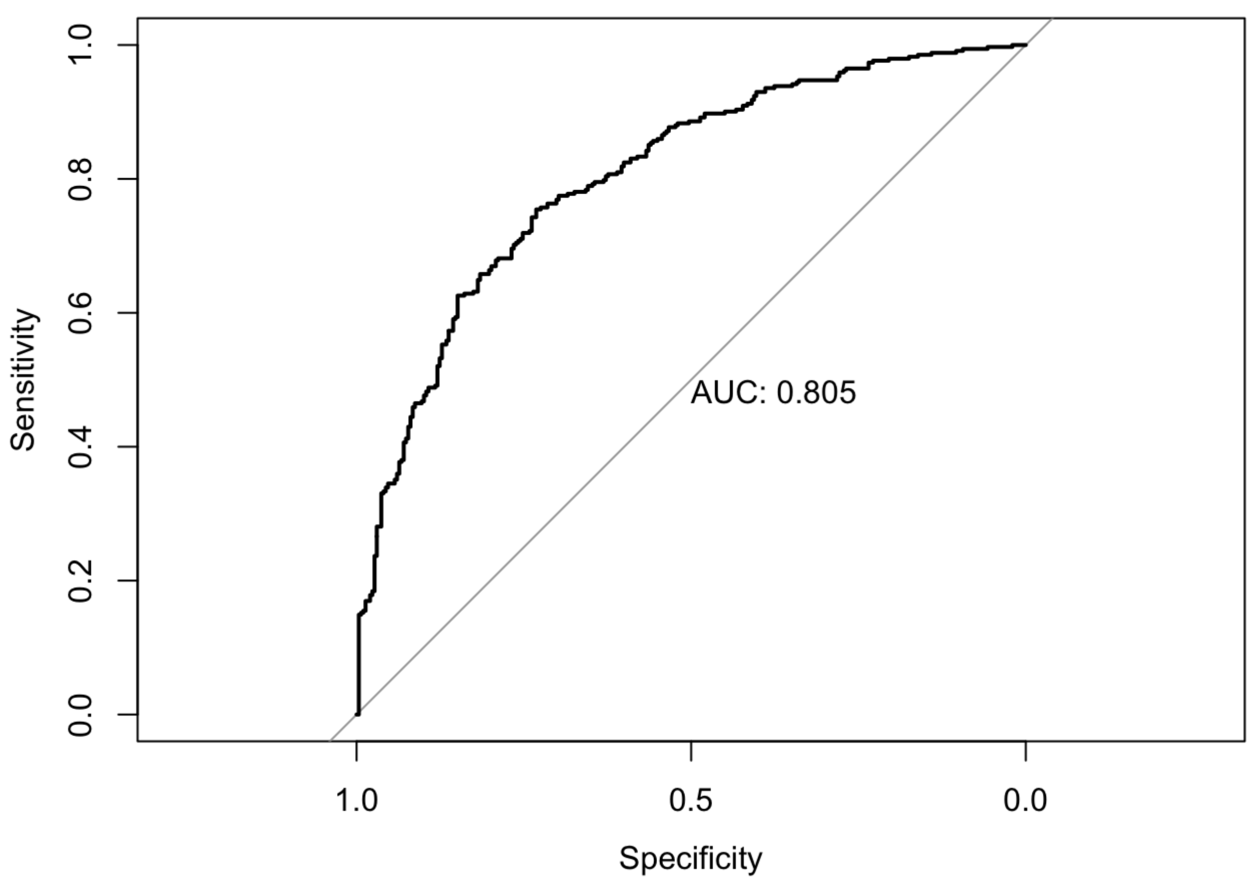
```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
test_prob = predict(randomforestmodlogit, test, type = "response")

test_roc = roc(test$highquality ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```



```
as.numeric(test_roc$auc)
```

```
## [1] 0.8052023
```

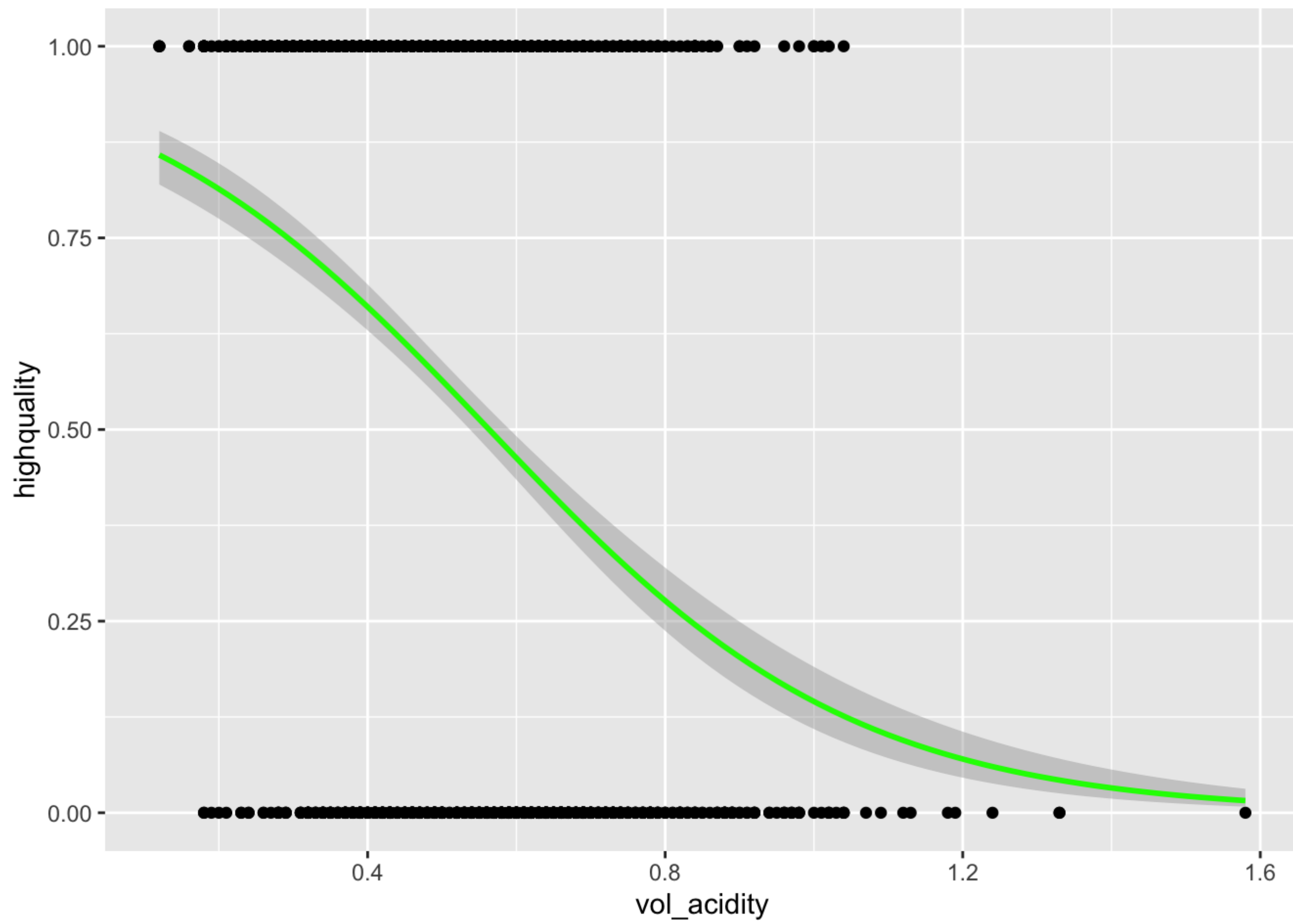
AUC and ROC with just one variable

```
library("ggplot2")

simple <- glm(highquality ~ vol_acidity, data = red, family = "binomial"(link = "logi
t"))
summary(simple)
```

```
##
## Call:
## glm(formula = highquality ~ vol_acidity, family = binomial(link = "logit"),
##     data = red)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8697  -1.1148   0.7156   1.0375   2.0349
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.2874     0.1838   12.45   <2e-16 ***
## vol_acidity  -4.0607     0.3334  -12.18   <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: 2209.0  on 1598  degrees of freedom
## Residual deviance: 2033.4  on 1597  degrees of freedom
## AIC: 2037.4
##
## Number of Fisher Scoring iterations: 4
```

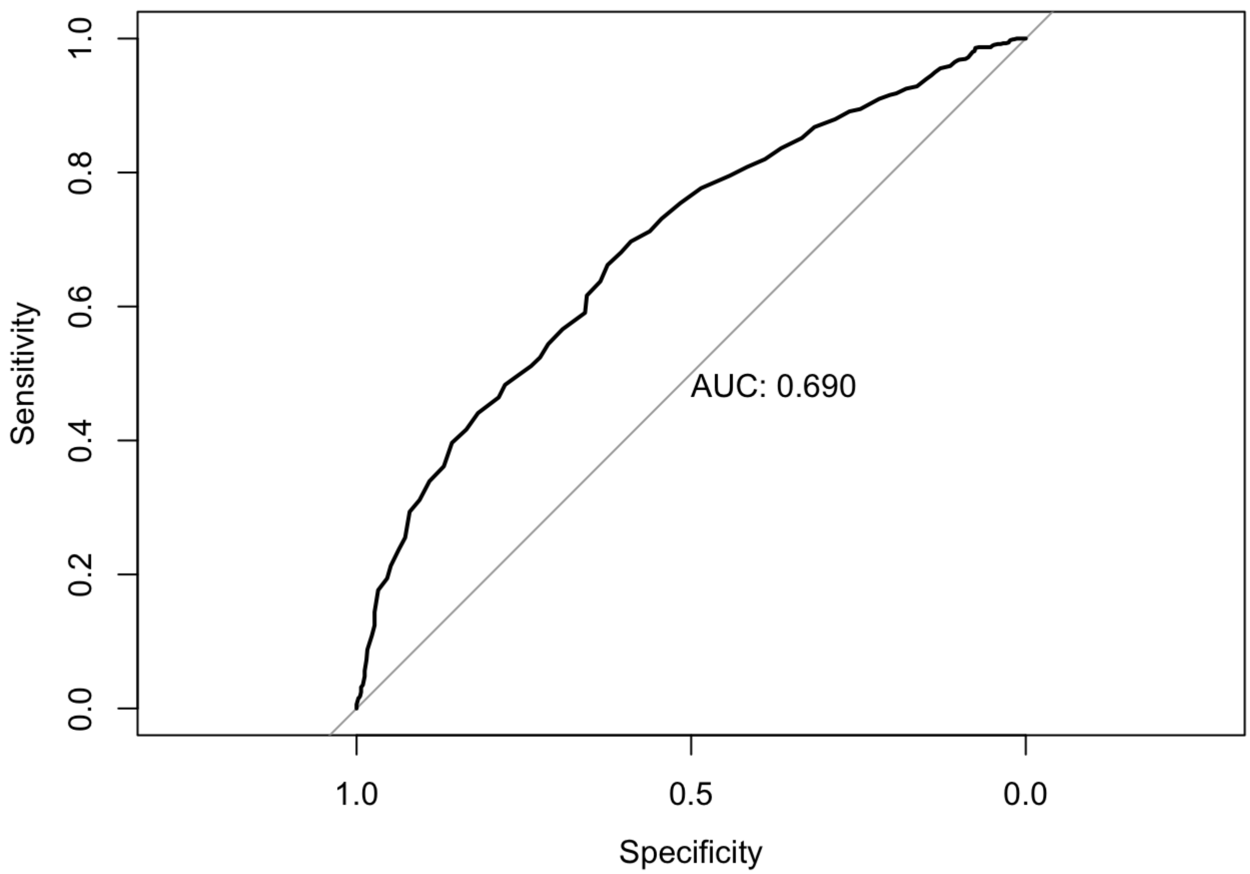
```
ggplot(red, aes(x = vol_acidity, y = highquality)) +geom_point()+stat_smooth(method="gl
m", color="green", se=TRUE, method.args = list(family=binomial))
```

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




```
test_prop1 = predict(simple, red, type = "response")

test_roc1 = roc(red$highquality ~ test_prop1, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```




```
as.numeric(test_roc1$auc)
```

```
## [1] 0.6900011
```