Project: 354

Meeting 3

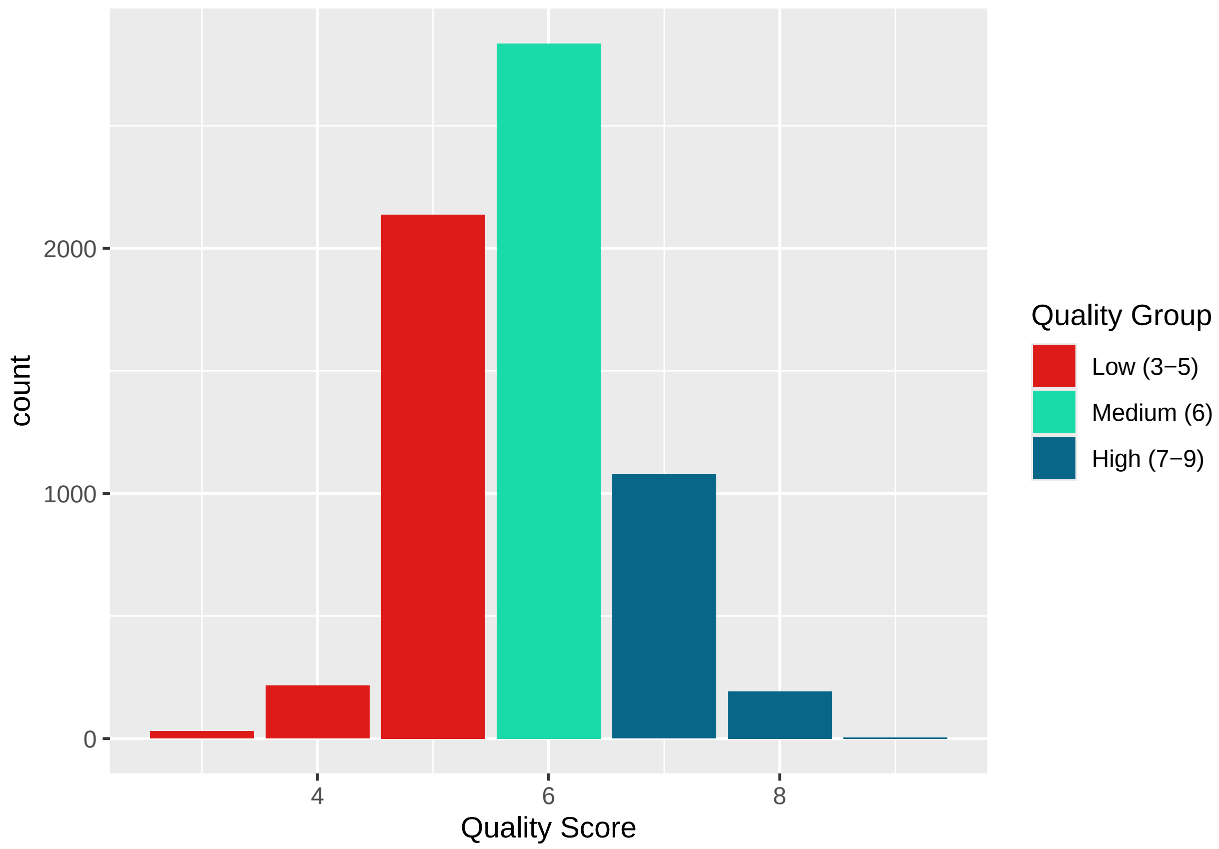
Adam 'One (1)' Nordquist

Setup

Set arbitrary quality groups

```r
wine_correct <- wine_correct %>%
  mutate(qgroup = case_when(quality %in% c(3, 4, 5) ~ "Low",
                            quality %in% c(6) ~ "Medium",
                            quality %in% c(7, 8, 9) ~ "High"),
         qgroup = factor(qgroup, levels = c("Low", "Medium", "High")))

wine_correct %>%
  ggplot(aes(x=quality, fill = qgroup))+
  geom_bar()+
  scale_fill_manual(values = c("Low" = "#DD1C1A", "Medium"="#1BDAAA", "High"="#086788"),
                    labels = c("Low" = "Low (3-5)", "Medium"="Medium (6)", "High"="High (7-9)"))+
  labs(x="Quality Score", fill="Quality Group")
```



```
wine_correct %>%
  count(qgroup)
```

```
##   qgroup    n
## 1    Low 2384
## 2 Medium 2836
## 3   High 1277
```

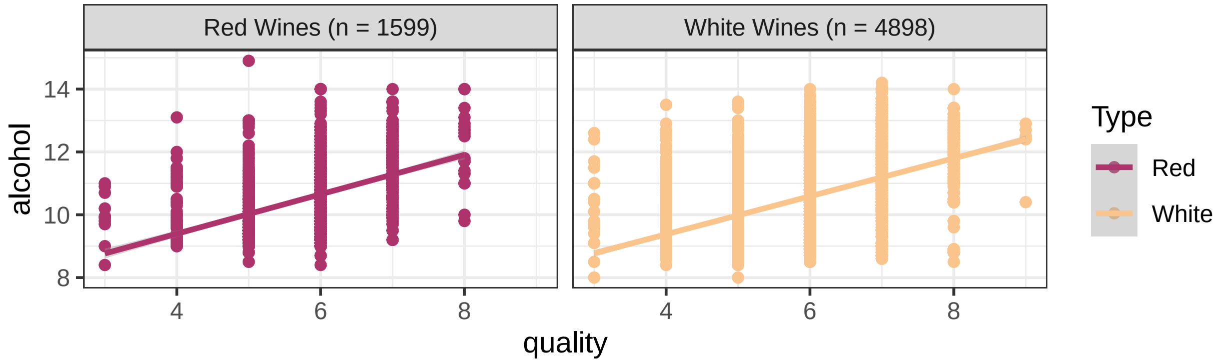
Establish relationship between quality and type

EDA

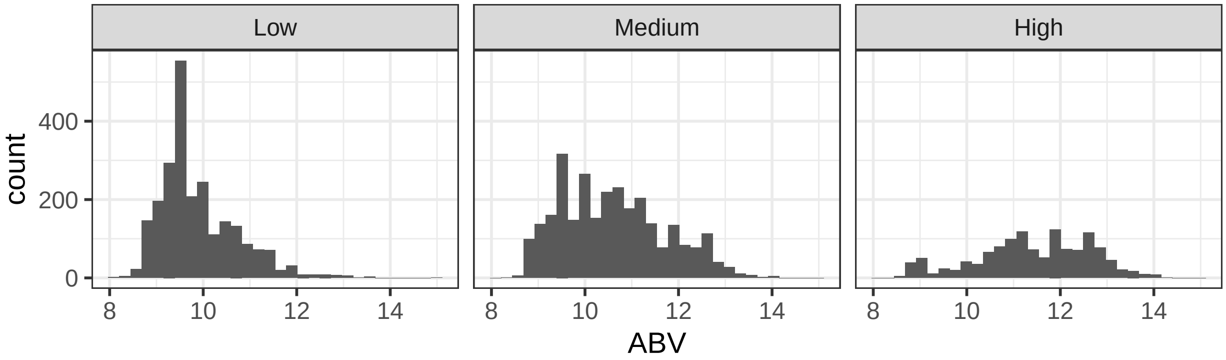
Long story short:

- Some linear relationship b/w ABV and quality
- ABV and volatile acidity look distinguishable enough

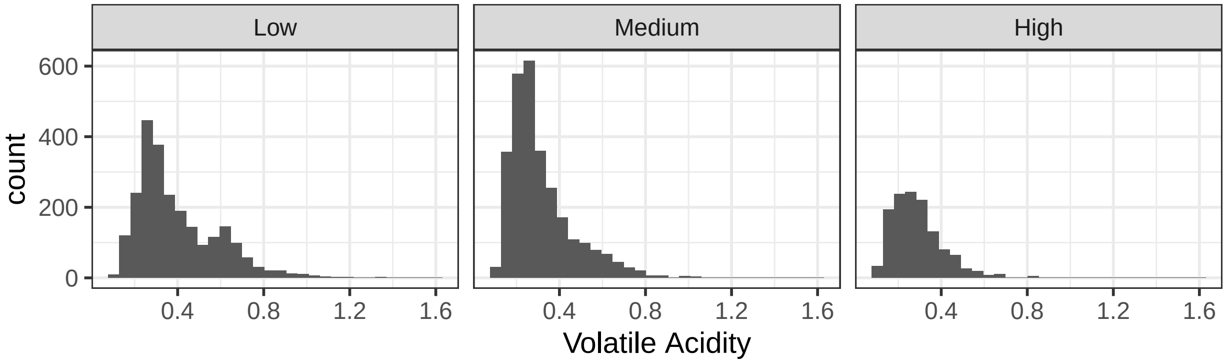
```
## 'geom_smooth()' using formula = 'y ~ x'
```



```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



Experimentation

The gameplan:

- Set the seed to make sure we grab the same observation every time
- Grab the top observation from a dataframe of 10 randomly drawn wines
- Observe its characteristics

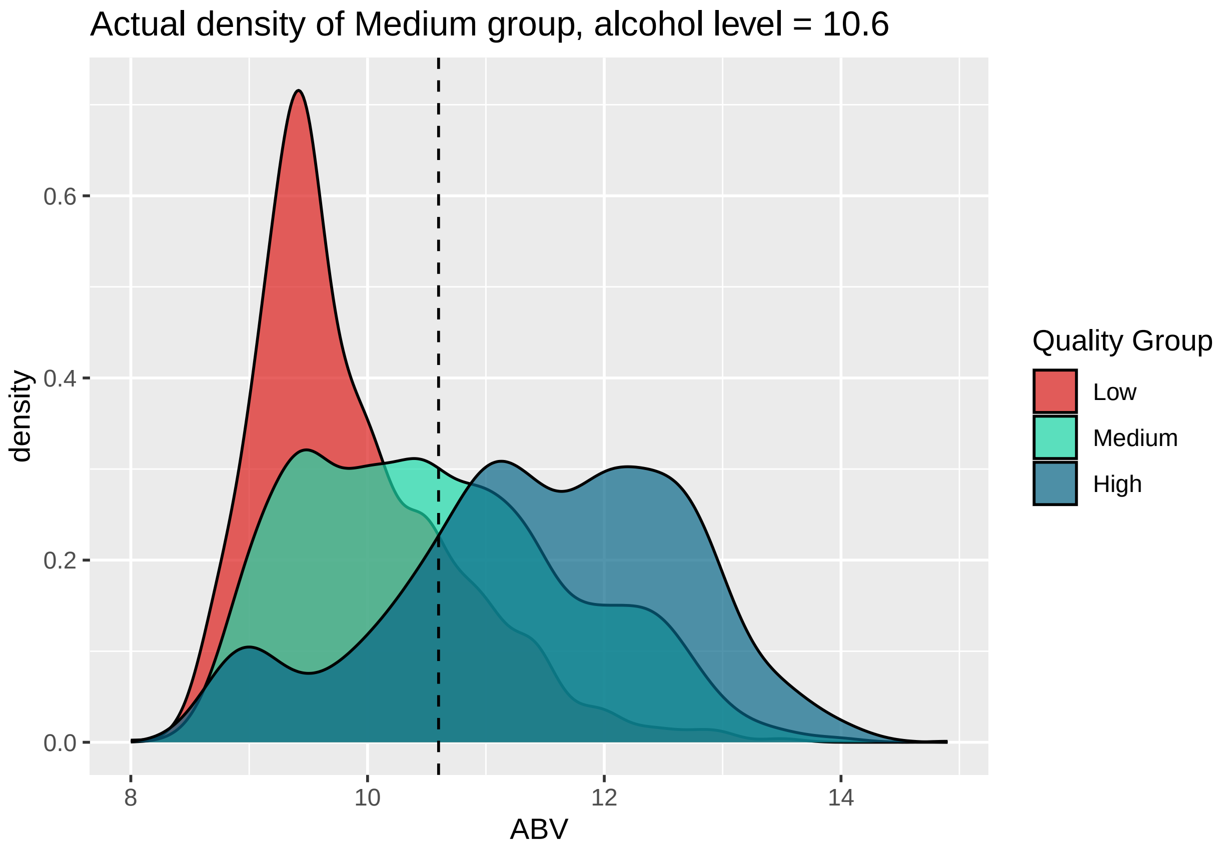
- For our purposes, explain how the model works

This first part is the slower, manual version - just to prove that it works; the "quick", "automatic" model will show that this is accurate math.

Following the Book (Manual Naive Bayes Classification)

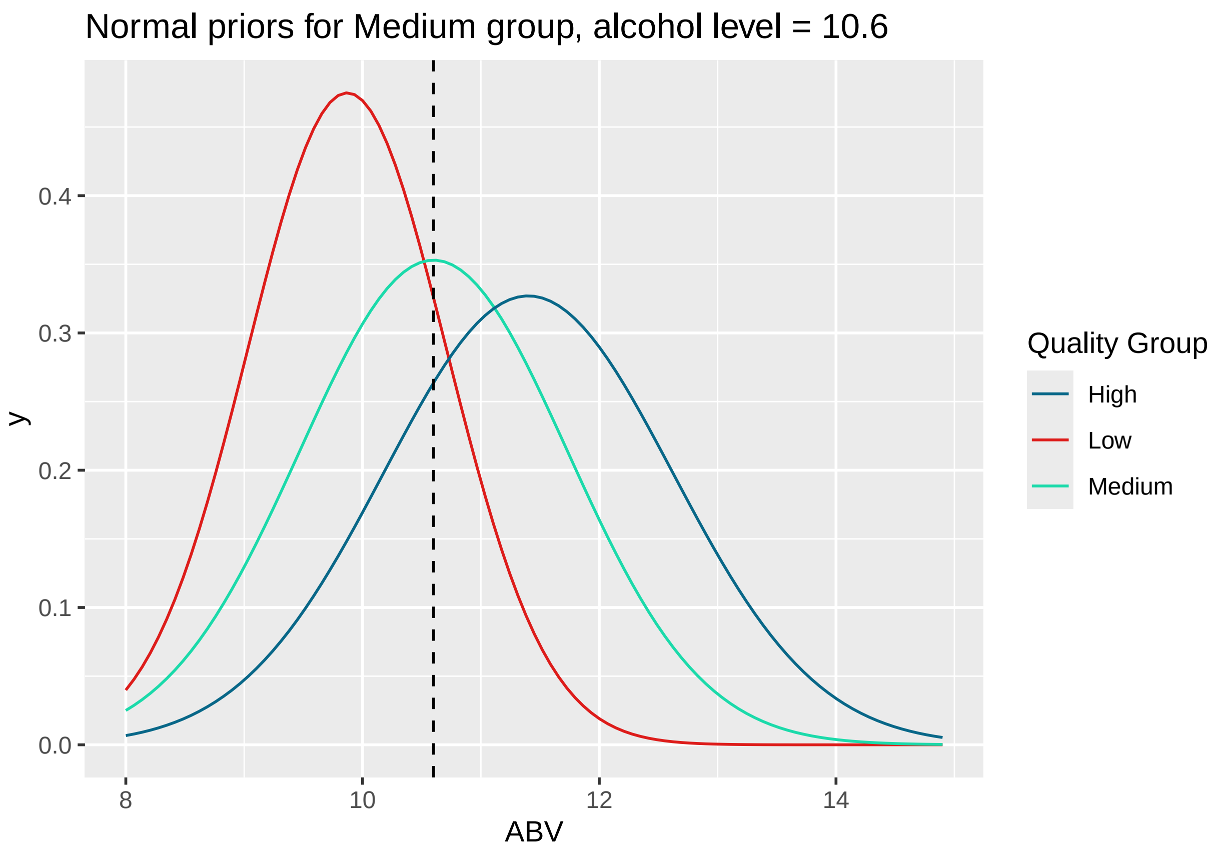
The Chosen Wine

ABV	Volatile Acidity	Quality Score	Quality Group	Wine Type
10.6	0.685	6	Medium	R



The below table informs our normal, naive classification priors.

Quality Group	Mean	Median	Std. Dev.
Low	9.87	9.60	0.84
Medium	10.59	10.50	1.13
High	11.43	11.50	1.22



Calculations

The below calculations use Bayes' Rule; the probabilities, normalizing constant, and likelihoods I already calculated and were kind of cluttering up the document, so they are hidden away. Rest assured they are calculated correctly.

```
p_low #Probability this wine is in the Low quality group
```

```
## [1] 0.3669747
```

```
p_med #Probability this wine is in the Medium quality group
```

```
## [1] 0.4736862
```

```
p_hi #Probability this wine is in the High quality group
```

```
## [1] 0.1593391
```

```
#..and do they add up to 1?
p_low + p_med + p_hi
```

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

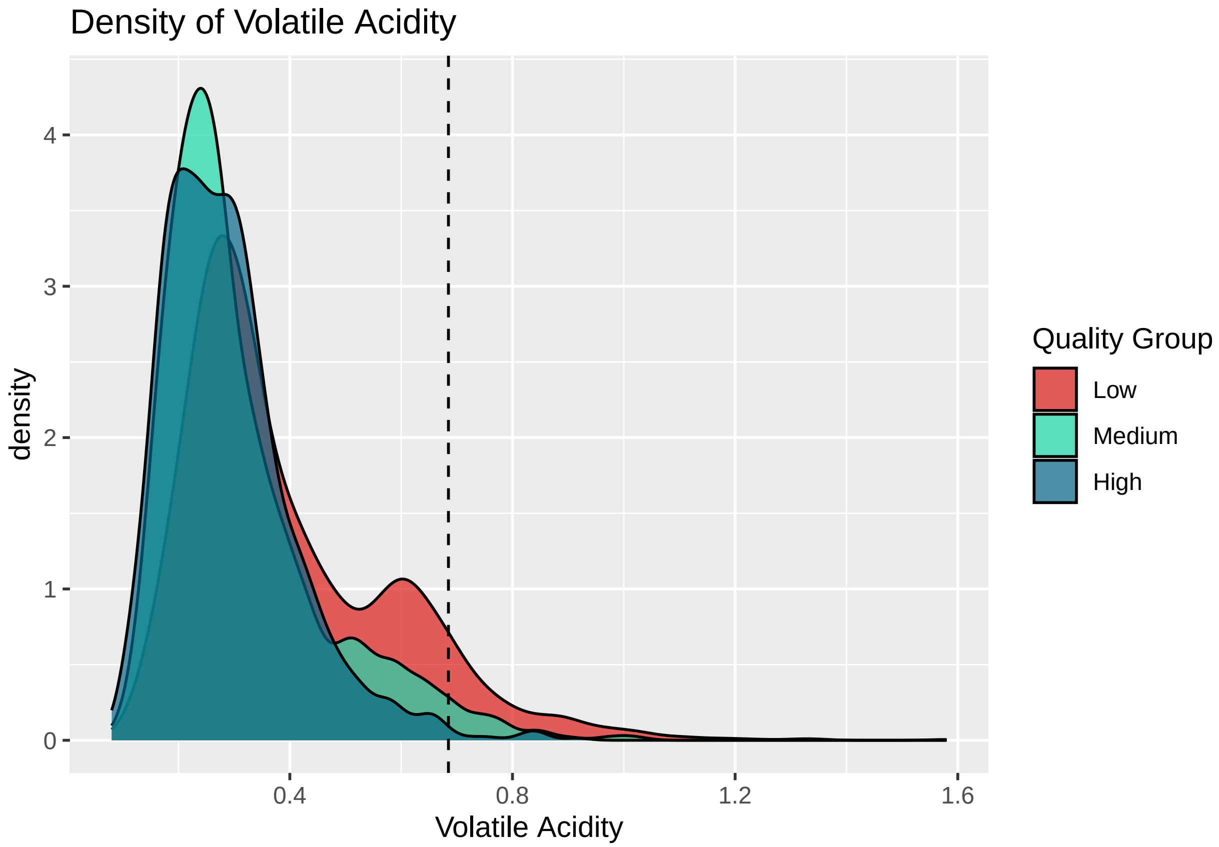
```
#Yes, they do. Compare with the predictions below...
```

Following the Book ("Automatic" Naive Bayes Classification)

```
##            Low    Medium      High
## [1,] 0.3682596 0.4750267 0.1567136
```

This quick, one-predictor model believes there is roughly a 47% chance this seeded observation is from the Medium quality group (which it is). The question, then: Is there more to this? Can we make the model better by adding in more predictors?

Model 2: Volatile Acidity



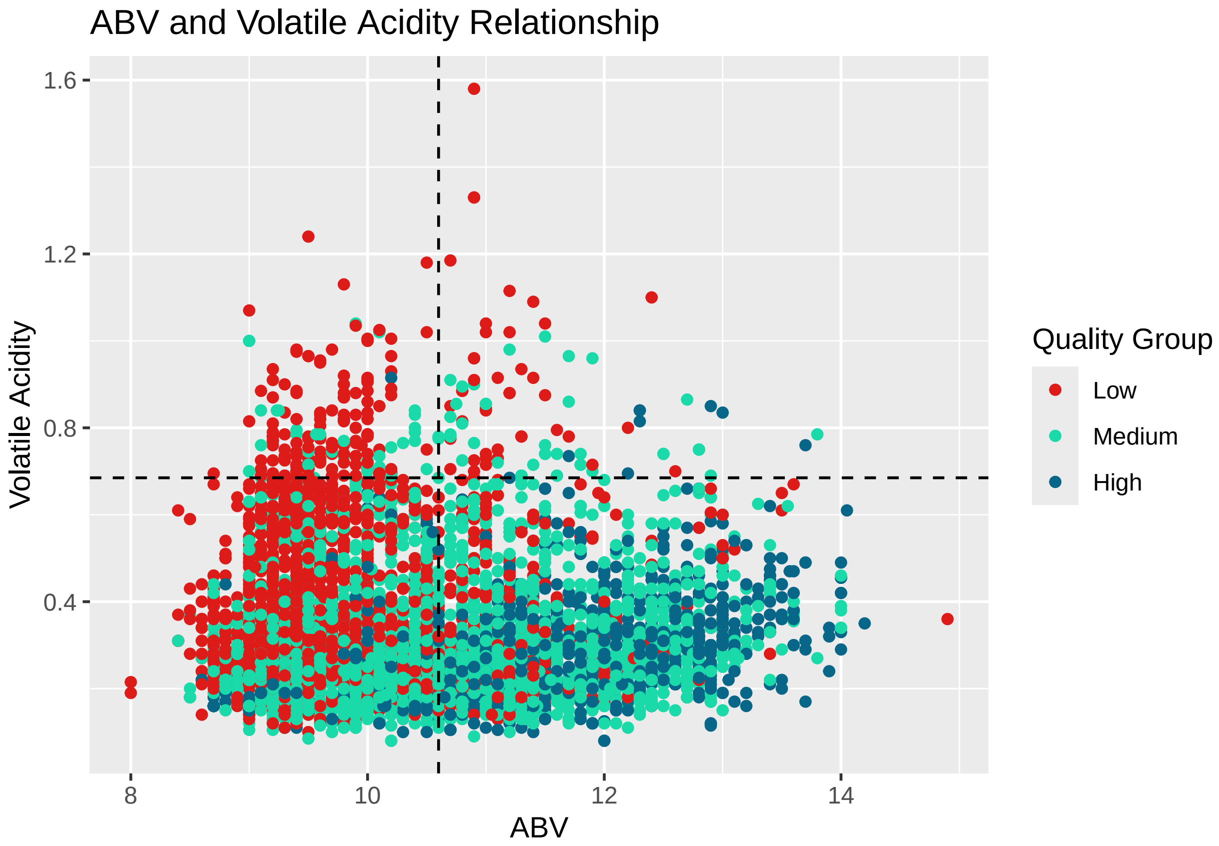
It is worth noting (and perhaps obvious) that this would be.. *difficult* to discern quality from. You can see below that this model has kind of no idea with this single predictor - in fact, it is very convinced this is a Low-quality wine based solely on the volatile acidity, which makes sense when you look at this density graph.

```
##            Low    Medium        High
## [1,] 0.8250329 0.1675097 0.007457404
```

Let's improve this by using two predictors, the ones we've spoken about already (ABV and volatile acidity).

Model 3: ABV / Volatile Acidity

The observation we picked from earlier is plotted below. Note that it is in the Medium group but surrounded by a sea of Low observations, mostly due to its volatile acidity content.



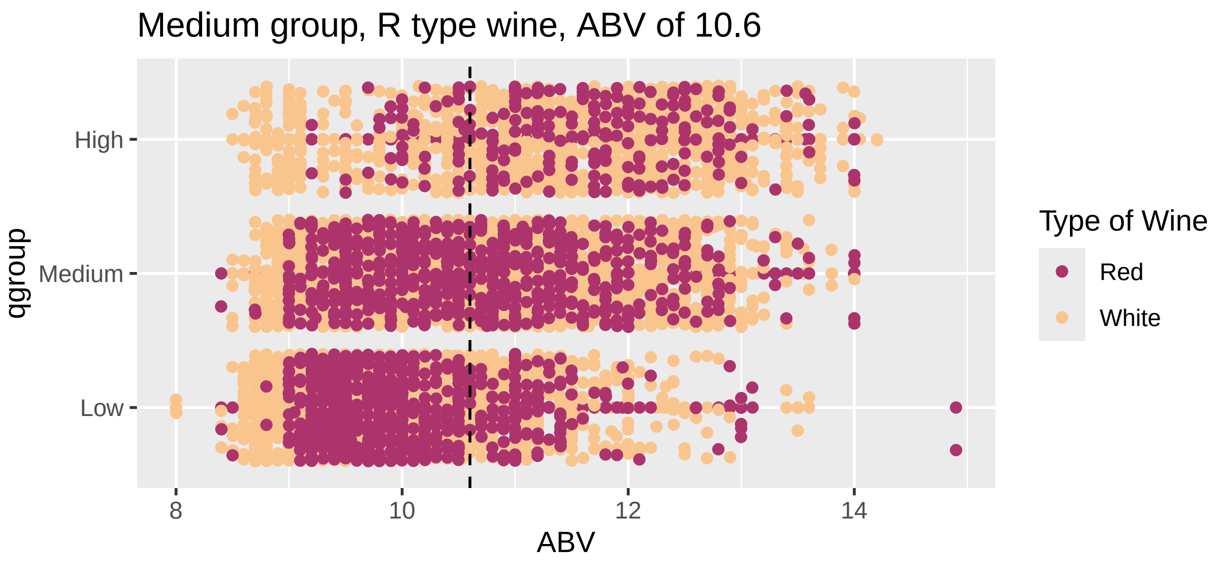
So let's predict, given there is some separation here. Is this new model ok at predicting quality based on both characteristics?

```
##            Low    Medium        High
## [1,] 0.8147714 0.1793777 0.005850867
```

The answer is "for this one, it doesn't seem to be *great*." That said, our simulated Chosen Wine is kind of an outlier in regards to these two combined characteristics, so this makes some sense.

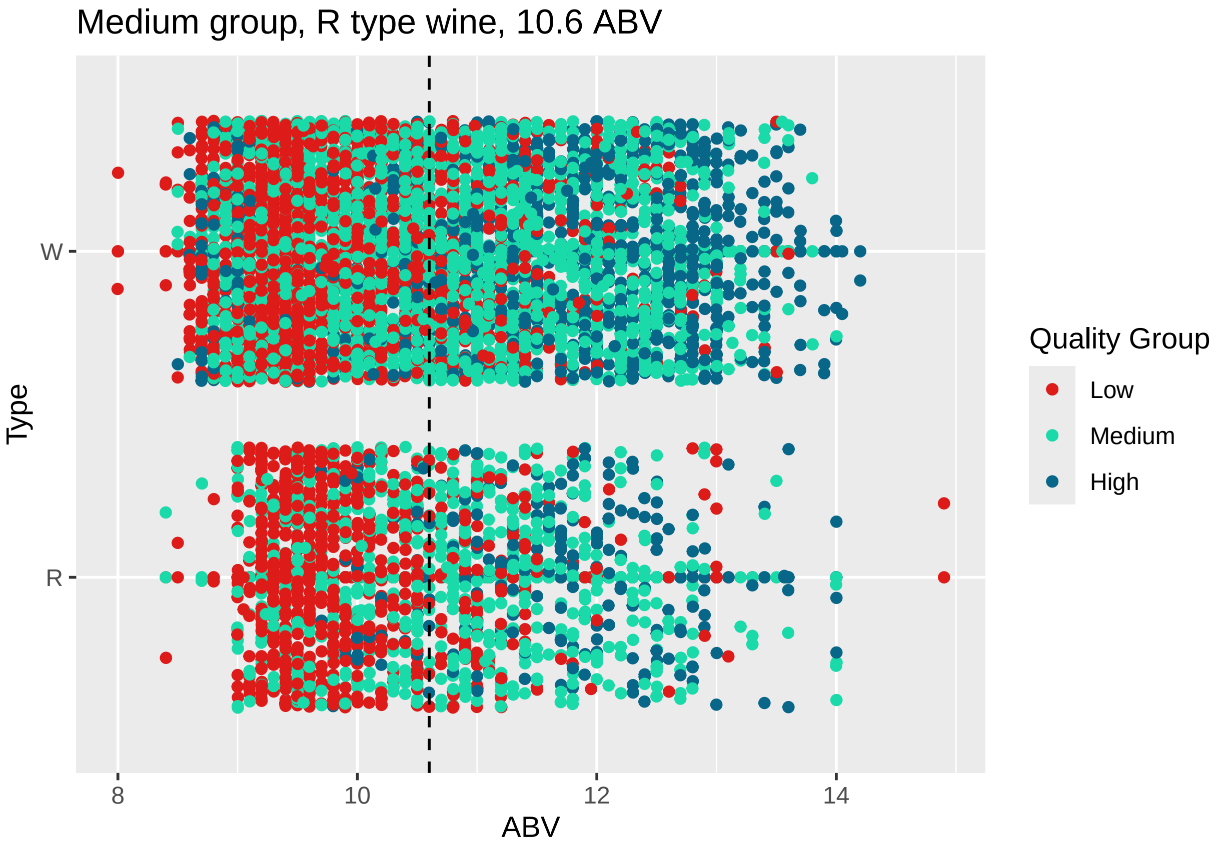
While we're at it, let's figure out if we can use the wine type, as well as the ABV, to determine wine quality group.

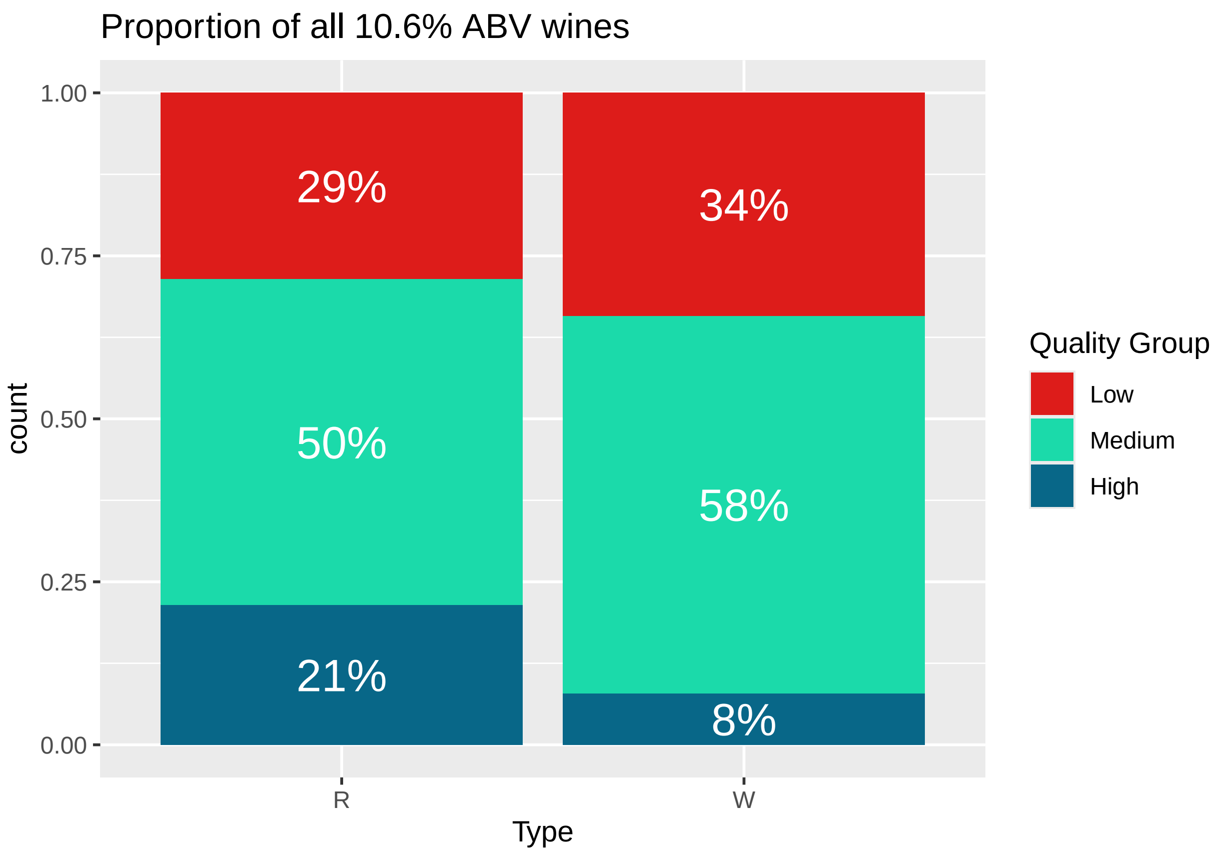
Model 4: ABV / Type



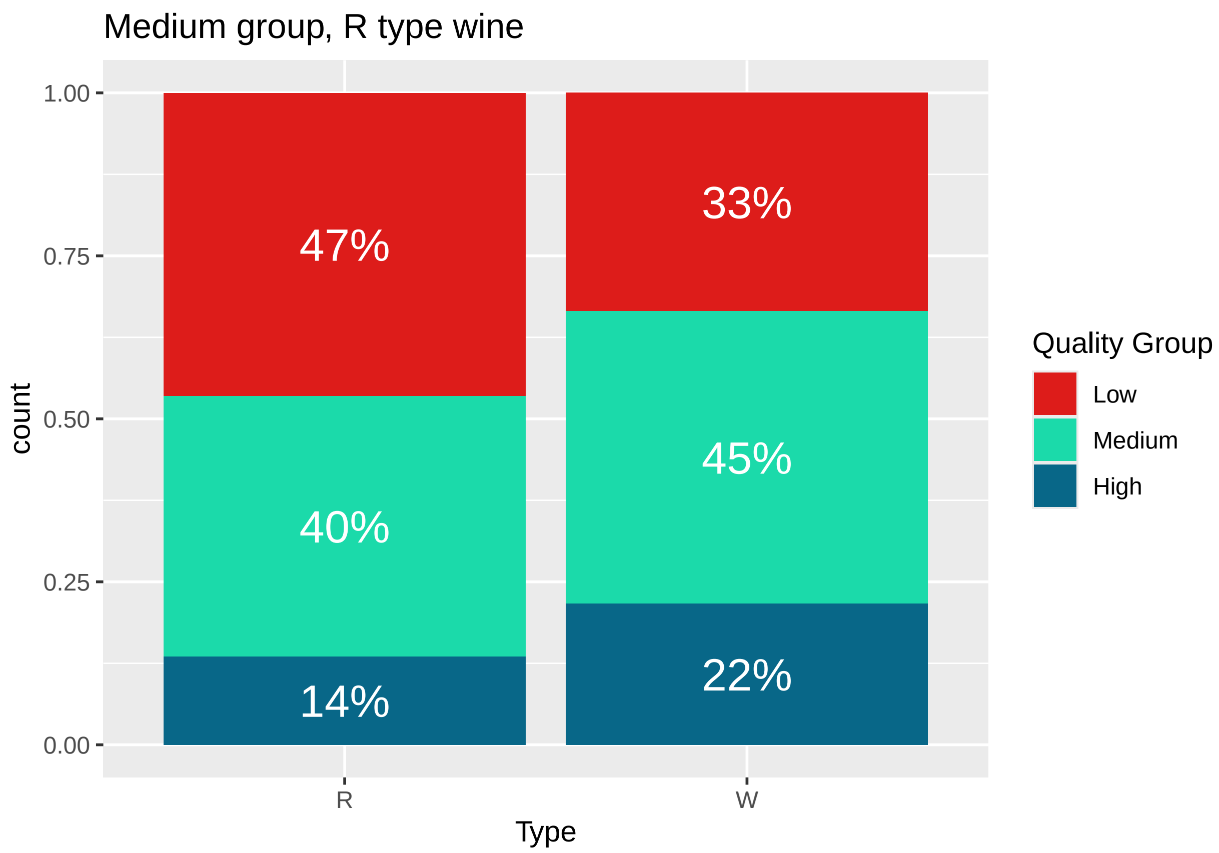
The above graph is kind of unnecessary, but it does give us an okay glimpse of what we can expect. it is largely a reversed graph of the first experiment we ran, and it gives us the same idea: as quality goes up, so does ABV. For our observation, it could be medium or low, and it's only moderately likely to be of the High group.

Let's zoom in a little on the wine type.





Our wine could be either low, medium, or high at first glance, but the quality group fill gives some clarity - it is smack in the middle of where most Medium wines are, especially as a Red wine. We will discover soon that this will end up being useful. It is also important to note that, among all wines that are 10.6% ABV, the most likely outcome by far is a medium wine, regardless of type. When it's factored in that this is a red wine, it is a neat coin toss for "medium or not".



This set of graphs shows it is not impossible to figure out quality group based on wine type if there are other characteristics involved, such as the already observed relationship b/w ABV and quality score. Let's give the single Type predictor a shot.

Model 4.1: Single-Var Type

```
##             Low    Medium      High
## [1,] 0.4652908 0.3989994 0.1357098
```

The single "Type" variable does pretty okay! However, it is.. basically just replicating the data, since all it knows about is what proportions of quality come with what types of wine. What about both ABV and wine type?

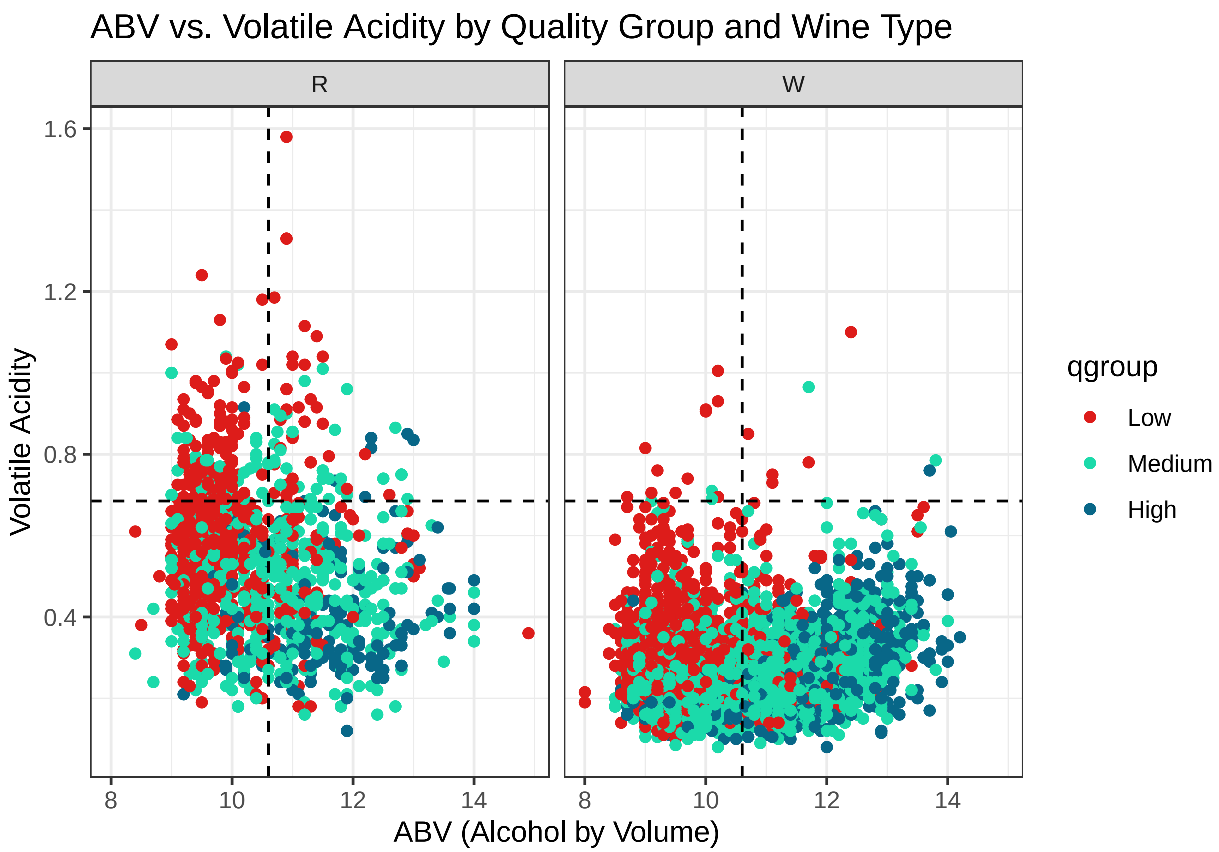
Model 4.2: Both ABV and Type

```
##             Low    Medium      High
## [1,] 0.4626282 0.4301737 0.1071981
```

Slightly better.. what about all three? We will see in Model 5.

Model 5 (for fun): ABV / Volatile Acidity / Type

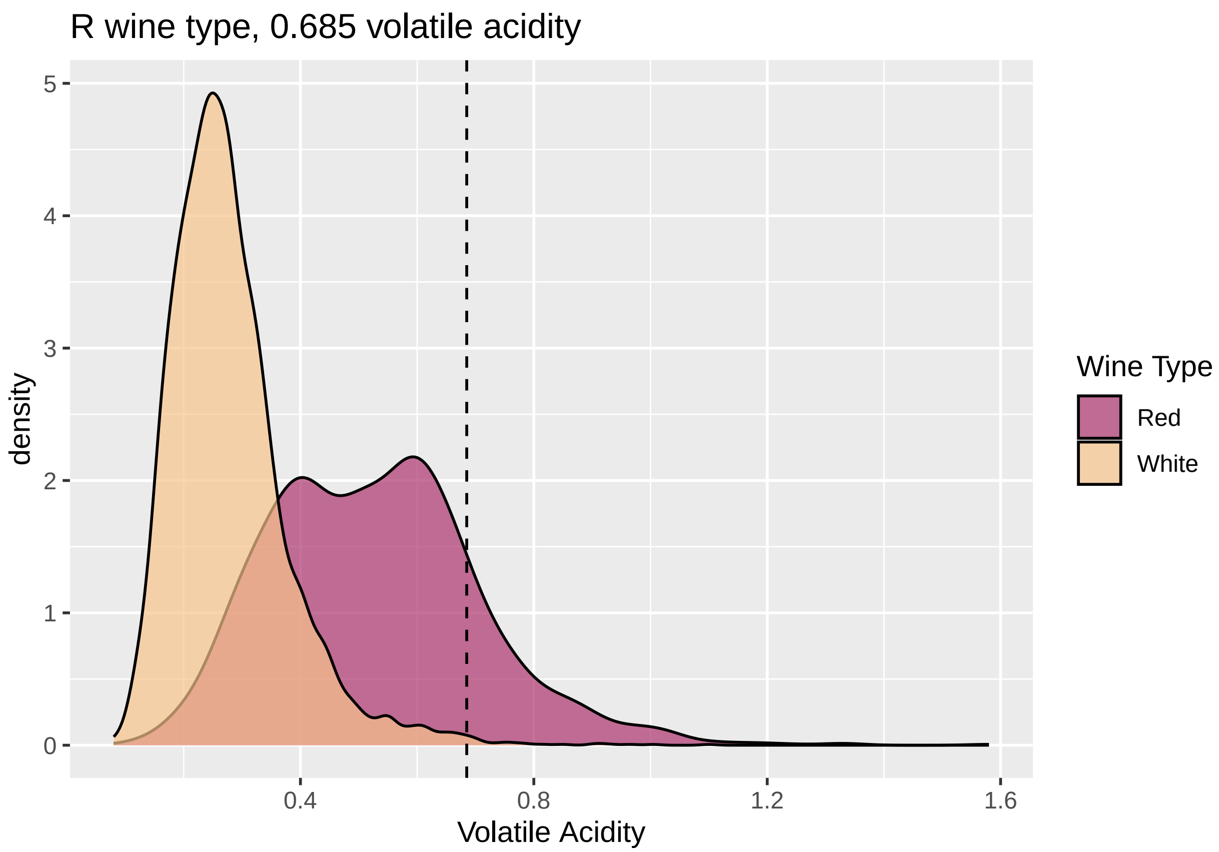
My last attempt hits quality by everything we've talked about: ABV, volatile acidity, and type.



```
##              Low     Medium        High
## [1,] 0.8601326 0.1365042 0.003363195
```

Model Deviations: Predicting wine type

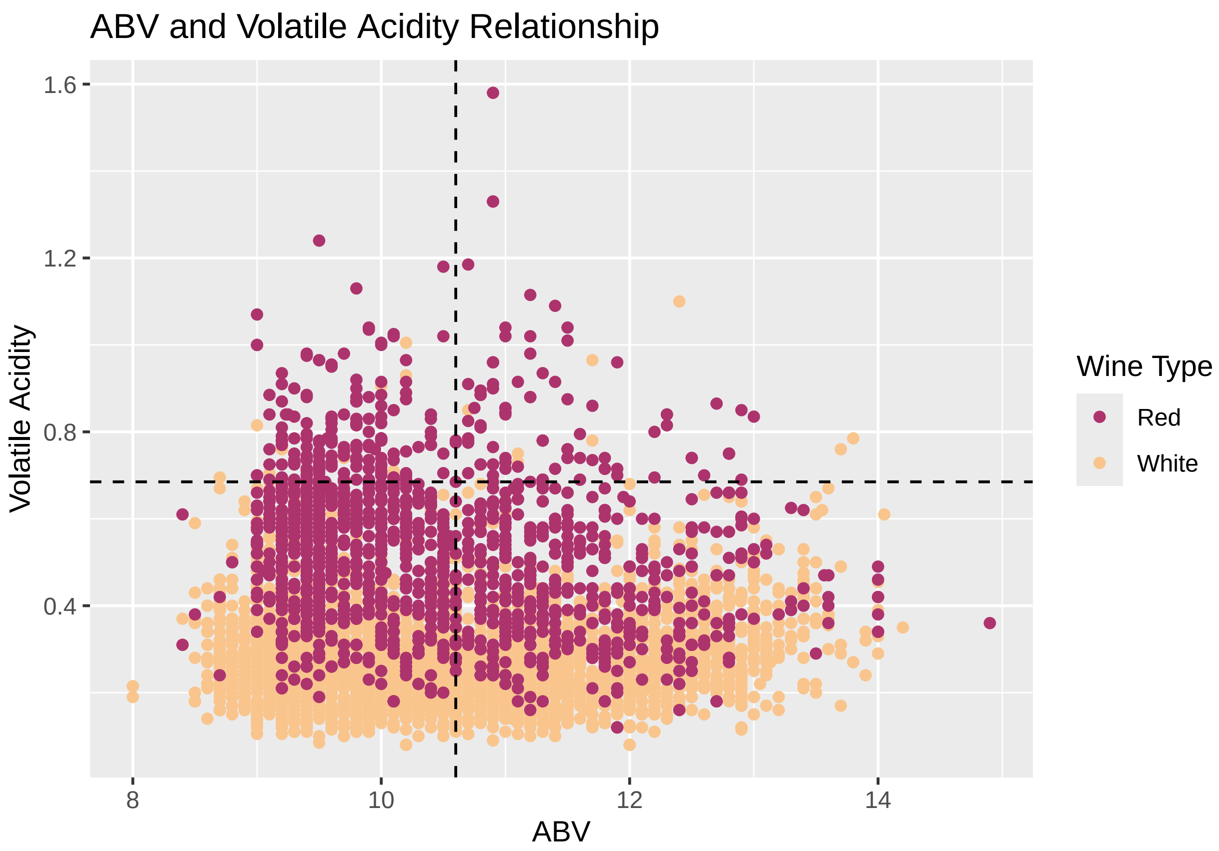
For this, I briefly deviated and found that, for this specific observation, there is a really good model to predict **wine type** once you throw in the volatile acidity - most likely due to the fact that **volatile acidity is typically high in Reds** and a wine with a volatile acidity this high would not possibly be a white wine. This knowledge came to me far too late in the final project process to change anything, so I'm not. However, I think if I were to do this again, I'd maybe try predicting wine type.



This first attempt at prediction adds ABV, quality group, and volatile acidity as predictors.

```
##              R           W
## [1,] 0.9977134 0.002286585
```

```
ggplot(wine_correct, aes(x = alcohol, y= volatile.acidity, color = Type)) +
  geom_point()+
  geom_vline(xintercept = linevalue, linetype = "dashed")+
  geom_hline(yintercept = dioxvalue, linetype = "dashed")+
  scale_color_manual(values = c("R" = "#ad336d", "W" = "#f9c58d"),
                     labels = c("R" = "Red", "W" = "White"))+
  labs(title = "ABV and Volatile Acidity Relationship",
       x="ABV", y="Volatile Acidity", color="Wine Type")
```



The second goes to ABV and VA, and the third does alcohol and quality group.

```
##              R           W
## [1,] 0.9979664 0.002033578
```

```
##              R         W
## [1,] 0.2489132 0.7510868
```

How Good Are the Models?

To recap, we have:

- first, which is qgroup ~ alcohol,
- volmodel, qgroup ~ volatile.acidity,
- twopredmodel, qgroup ~ alcohol + volatile.acidity,
- typesinglemodel, qgroup ~ Type,
- second_attempt_model, qgroup ~ alcohol + Type,
- threepredmodel, qgroup ~ alcohol + volatile.acidity + Type,
- reverse_many_model, Type ~ alcohol + qgroup + volatile.acidity,
- reverse_model, Type ~ alcohol + volatile.acidity, and
- reverse_model_2, Type ~ alcohol + qgroup.

Cross-Validation

Model	Max Accuracy	Average Accuracy
ABV	0.5639445	0.5257909
Volatile Acidity	0.5261538	0.4809885
ABV + Volatile Acidity	0.5753846	0.5436309
Wine Type	0.4769231	0.4528266
ABV + Wine Type	0.5738462	0.5277739
ABV + Volatile Acidity + Wine Type	0.5830769	0.5410113
Type ~ ABV + Quality Group + Volatile Acidity	0.7019868	0.6135132
Type ~ ABV + Volatile Acidity	0.6802721	0.6069966
Type ~ ABV + Quality Group*	0	0

*What I can discern from this is that the CV isn't training well enough on simply ABV and qgroup, and there's probably something skewing the training data to say "Just predict a white wine, you'll be right 75.3% (pct of whites in the total) of the time".

Conclusion

Confusion Matrices for Models 3 and 5

Model 3 (two-predictor model, ABV + Volatile Acidity)

```
##  qgroup            Low         Medium         High
##     Low 66.57% (1,587) 32.09%   (765)  1.34%  (32)
##  Medium 33.50%   (950) 54.55% (1,547) 11.95% (339)
##    High 11.98%   (153) 56.85%   (726) 31.17% (398)
```

Model 5 (three-predictor model, ABV + Volatile Acidity + Wine Type)

```
##  qgroup            Low         Medium         High
##     Low 53.90% (1,285) 44.55% (1,062)  1.55%  (37)
##  Medium 23.66%   (671) 63.82% (1,810) 12.52% (355)
##    High  5.09%    (65) 62.02%   (792) 32.89% (420)
```

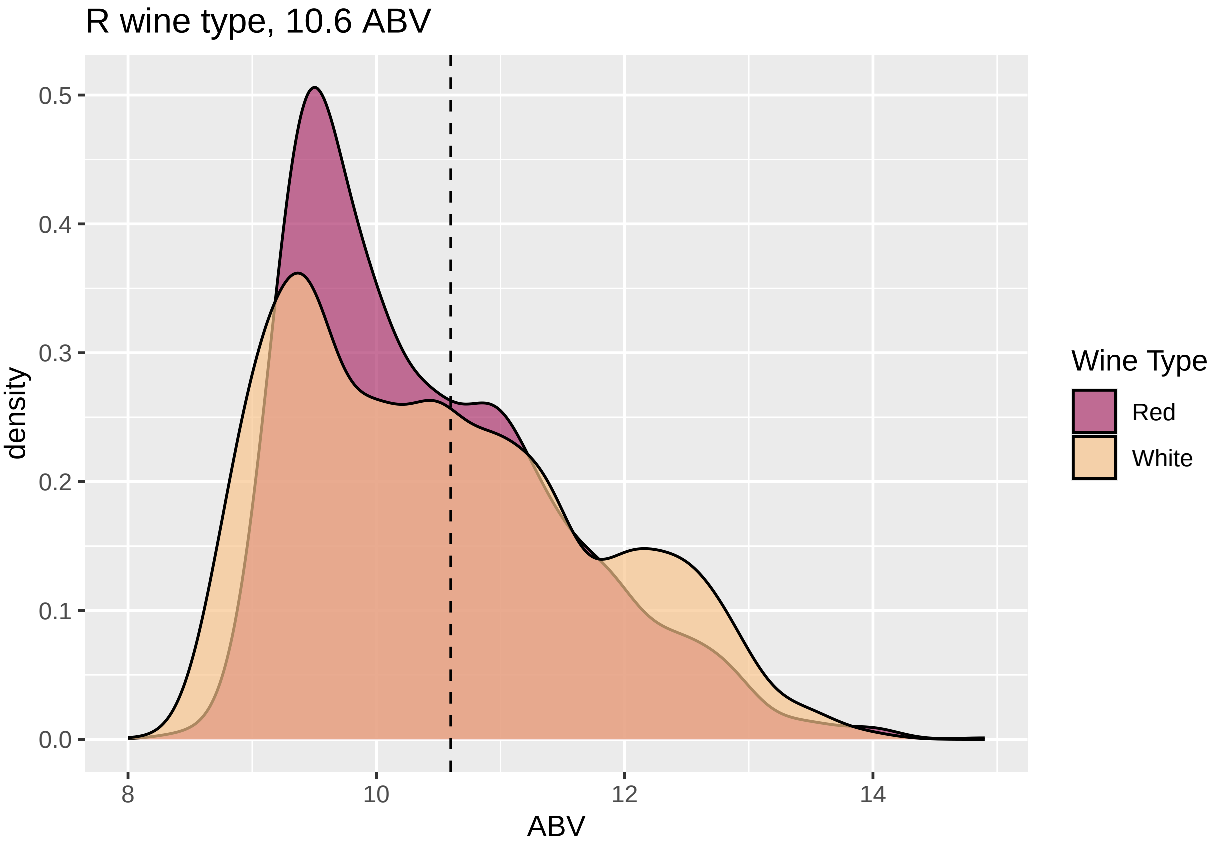
These two models are largely identical in accuracy, so I'm noting both their confusion matrices. What's interesting is that what makes them similar - yet different - is that the two-predictor model is relatively good at predicting Low quality wines as Low (66.57% accuracy), while the three-predictor model isn't (53.61%); similarly, the three-predictor model is relatively good at predicting Medium-quality wines as Medium (64.03%) while the two-predictor model is not (54.41%). Both models are kind of equally garbage at predicting High-quality wines, which is (likely) mostly due to two things:

- There are not many High wines (n=1277)
- Because of that, the characteristics are less separated than they are at the L and M levels

Pt 2

In general, I've found that the quality group is not very easy to predict well. The highest accuracy score of any model we tested was the three-predictor model (ABV + Volatile Acidity + Wine Type), with an accuracy score of .583, which is okay for our purposes, but wouldn't be great if this were something that had any worldly dependencies. The three-predictor model had the highest single accuracy score, but the highest average accuracy goes to the two-predictor model (ABV + Volatile Acidity) by ≈ .002, or 0.2%. Both can effectively share the title, as it's so close some seed changes might swap that order.

The type of wine is a little easier to process, it seems, as all the models (except for the q-group model) had higher average accuracy scores than any of the quality group prediction models. The ABV is not as great at discerning type, but volatile acidity is.


R wine type, 10.6 ABV
density
ABV
Wine Type
Red
White