project-revised

STA 210 - Summer 2022

Team 2 - Alicia Gong, Ashley Chen, Abdel Shehata, Claire Tam

6-16-2022

Setup

```
library(tidyverse)
library(tidymodels)
library(dplyr)
library(ggplot2)
library(cowplot)
library(knitr)
library(recipes)
library(caret)
library(InformationValue)
library(ISLR)
library(MASS)
library(nnet)
library(Stat2Data)
```

```
redwine <- read.csv("data/winequality-red.csv", sep = ";")
whitewine <- read.csv("data/winequality-white.csv", sep = ";")
redwine <- redwine %>% mutate(color="red")
whitewine <- whitewine %>% mutate(color="white")
wine <- redwine %>% full_join(whitewine)
```

Load packages and data:

```
Joining, by = c("fixed.acidity", "volatile.acidity", "citric.acid",
"residual.sugar", "chlorides", "free.sulfur.dioxide", "total.sulfur.dioxide",
"density", "pH", "sulphates", "alcohol", "quality", "color")
```

```r
wine <- slice(wine, sample(1:n()))
```

Introduction and Data

Introduction About 234 million hectoliters of wine were consumed in 2020, worldwide, with the US making up approximetly 14% of that consumption (Karlsson 2020). Since Wine composition and wine quality varies widely, it raises the question: what makes a good wine?

To answer that question, we will analyze the wine quality dataset from Vinho Verde vinyard in Portogal, and more importantly try to narrow down our question to make it possible for it to be supported by evidence. Below is the introduction to our research:

Project Goal: To identify variables that are important in explaining variation in the response. "Vinho Verde" is the kind of wine exclusively produced in the demarcated region of Vinho Verde in northwestern Portugal. The Vinho Verde wine has its own unique system of production and is only produced from the indigenous grape varieties of the region. Vinho Verde region is one of the largest and oldest wine regions in the world, and is home to thousands of producers, generating a wealth of economic activity and jobs, and strongly contributes to the development of Minho province and the country. The Vinho Verde wine also enjoys high reputation worldwide. It is recurrently awarded in national and international competitions. The goal of this dataset is to model wine quality based on physicochemical tests. We believe that this dataset can also be used to analyze the relationship between different chemical compositions and the ratings of wine quality. We believe that this dataset can also be used to analyze what chemical factors are attributable to the final rating of Vinho Verde wine. Our research may shed light on future research and development directions for improving the quality of Vinho Verde wine, which may also contribute to the competitiveness of Portuguese wine industry.

Our goal is to produce a classification model that best explains how different chemical compositions of the Portuguese "Vinho Verde" wine affects the variation of the wine quality.

Data Introduction The Wine Quality dataset was collected from Vinho Verde wine Samples, from the North of Portugal. The data was originally donated in 2009 by Professor Cortez. The specific mechanism of the collection of the data was lab work done on different wines to measure their chemical attributites (like acidity etc.). The quality of the wine however was obtained through the average rating of three wine experts. The dataset is divided into two: Red wine and White wine. Red Wine has 1599 observations, and white wine has 4898 observations (each observation being a specific wine). Information about the wine include but are not limited to:PH,Density,Acidity, and alcohol content.

Each observation is a specific wine from the Vinho Verde region. Thus, there might be a little uncertainity in collecting the exact numbers for each numbers. However, since the Vinho Verde region is a vast region spreading 15500 hectareas of vineyards in far-north Portugal, this

uncertainity shouldn't be significant in our analysis or project. Thus, we will assume that the datas are independent and random

```
glimpse(wine)
```

```
Rows: 6,497
Columns: 13
$ fixed.acidity        <dbl> 6.6, 7.5, 5.7, 3.8, 6.3, 6.5, 5.4, 6.0, 7.6, 5.7,~
$ volatile.acidity     <dbl> 0.290, 0.250, 0.220, 0.310, 0.250, 0.230, 0.255, ~
$ citric.acid          <dbl> 0.31, 0.32, 0.25, 0.02, 0.53, 0.38, 0.33, 0.26, 0~
$ residual.sugar       <dbl> 3.90, 8.20, 1.10, 11.10, 1.80, 1.30, 1.20, 6.80, ~
$ chlorides            <dbl> 0.027, 0.024, 0.050, 0.036, 0.021, 0.032, 0.051, ~
$ free.sulfur.dioxide  <dbl> 39, 53, 97, 20, 41, 29, 29, 22, 32, 28, 22, 36, 3~
$ total.sulfur.dioxide <dbl> 96, 209, 175, 114, 101, 112, 122, 93, 155, 173, 1~
$ density              <dbl> 0.990350, 0.995630, 0.990990, 0.992480, 0.989315,~
$ pH                   <dbl> 3.24, 3.12, 3.44, 3.75, 3.19, 3.29, 3.37, 3.15, 3~
$ sulphates            <dbl> 0.60, 0.46, 0.62, 0.44, 0.31, 0.54, 0.66, 0.42, 0~
$ alcohol              <dbl> 12.6, 10.8, 11.1, 12.4, 13.0, 9.7, 11.3, 11.0, 11~
$ quality              <int> 8, 6, 6, 6, 6, 5, 6, 6, 6, 6, 6, 6, 5, 6, 5, 6~
$ color                <chr> "white", "white", "white", "white", "white", "whi~
```

There are 6497 observations and 13 variables (14 if you include the new response variable added later).

```
any(is.na(wine))
```

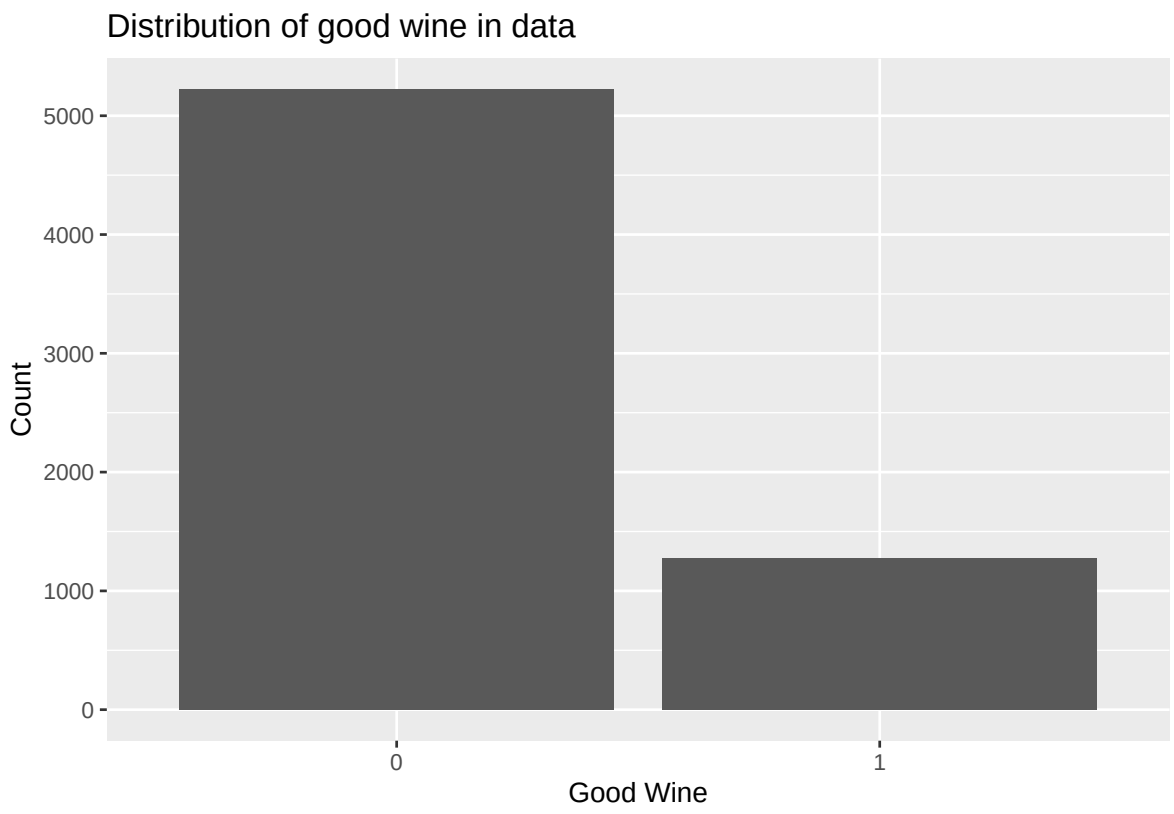
```
[1] FALSE
```

There are no NAs in our data, so we shouldn't be concerned about missing data.

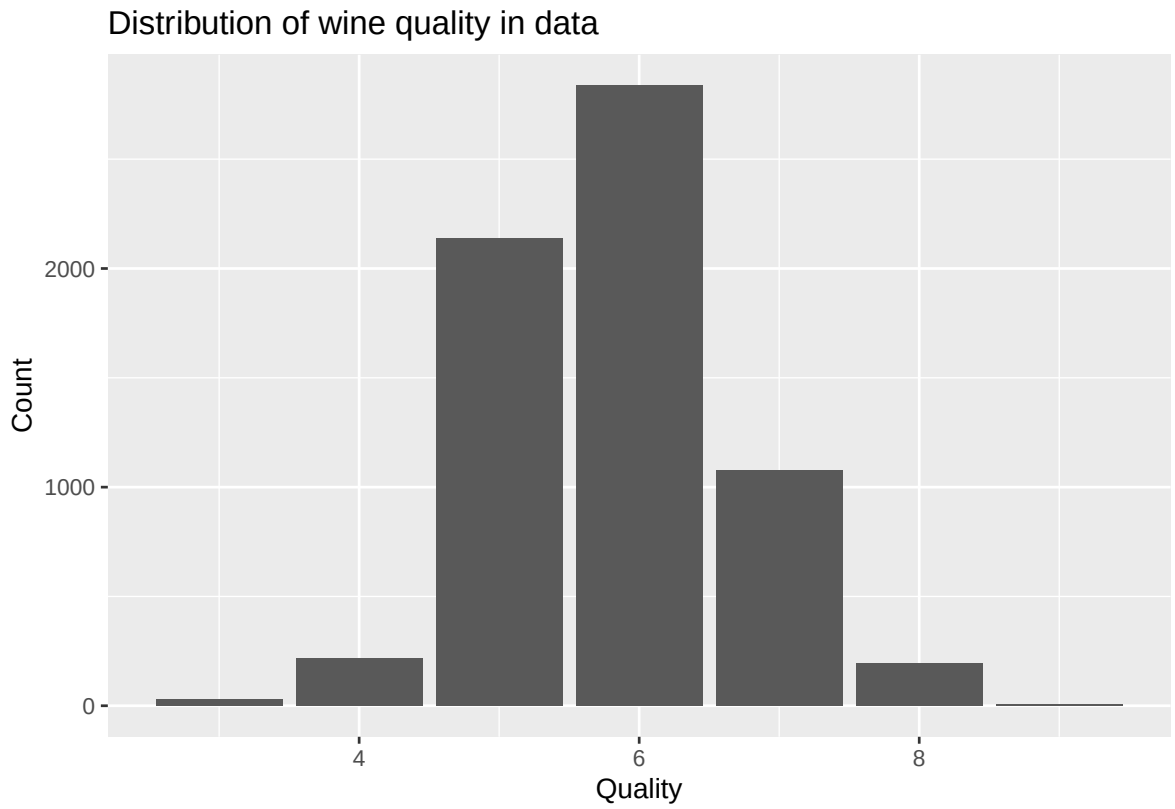
```
wine <- wine %>%
  mutate(good_wine = if_else(quality >= 7,"1","0"))
wine <- wine %>%
  mutate(good_wine = as.factor(good_wine))
wine <- wine %>%
  mutate(good_wine_names = if_else(good_wine=="1","Good wine","Bad or subpar wine"))

no1 <- colnames(wine)[1:11]
colnames(wine)[1:11] = paste("c_", no1, sep = "")
```

Data Wrangling

```
ggplot(wine, aes(x = good_wine)) +
  geom_bar() +
  labs(title = "Distribution of good wine in data",
       y = "Count",
       x = "Good Wine"
  )
```




```
ggplot(wine, aes(x = quality)) +
  geom_bar() +
  labs(title = "Distribution of wine quality in data",
       y = "Count",
       x = "Quality"
  )
```



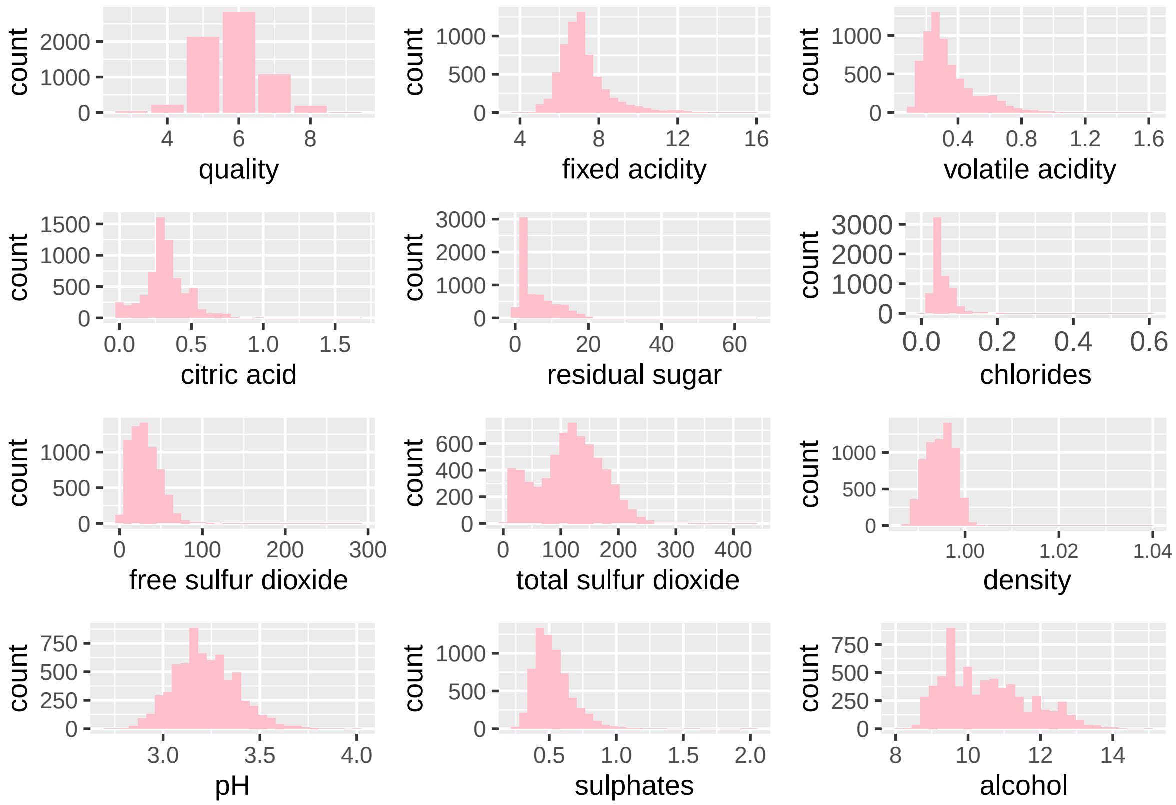

```
p1  <- ggplot(data = wine, aes(x = quality) ) +
  geom_bar(fill = "pink") +
  labs(x = "quality")

p2  <- ggplot(data = wine, aes(x = c_fixed.acidity) ) +
  geom_histogram(fill = "pink") +
  labs(x = "fixed acidity")

p3  <- ggplot(data = wine, aes(x = c_volatile.acidity) ) +
  theme(axis.text=element_text(size=9)) +
  geom_histogram(fill = "pink") +
  labs(x = "volatile acidity")

p4  <- ggplot(data = wine, aes(x = c_citric.acid) ) +
  theme(axis.text = element_text(size=9)) +
  geom_histogram(fill = "pink") +
  labs(x = "citric acid")
```

```
p5  <- ggplot(data = wine, aes(x = c_residual.sugar) ) +
  geom_histogram(fill = "pink") +
  labs(x = "residual sugar")

p6  <- ggplot(data = wine, aes(x = c_chlorides) ) +
  theme(axis.text = element_text(size = 11)) +
  geom_histogram(fill = "pink") +
  labs(x = "chlorides")

p7  <- ggplot(data = wine, aes(x = c_free.sulfur.dioxide) ) +
  theme(axis.text = element_text(size=9)) +
  geom_histogram(fill = "pink") +
  labs(x = "free sulfur dioxide")

p8  <- ggplot(data = wine, aes(x = c_total.sulfur.dioxide) ) +
  theme(axis.text = element_text(size=9)) +
  geom_histogram(fill = "pink") +
  labs(x = "total sulfur dioxide")

p9  <- ggplot(data = wine, aes(x = c_density) ) +
  theme(axis.text = element_text(size = 7.5)) +
  geom_histogram(fill= "pink") +
  labs(x = "density")

p10 <- ggplot(data = wine, aes(x = c_pH) ) +
  geom_histogram(fill = "pink") +
  labs(x = "pH")

p11 <- ggplot(data = wine, aes(x = c_sulphates) ) +
  theme(axis.text = element_text(size=9)) +
  geom_histogram(fill= "pink") +
  labs(x = "sulphates")

p12 <- ggplot(data = wine, aes(x = c_alcohol) ) +
  geom_histogram(fill= "pink") +
  labs(x = "alcohol")

plot_grid(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, ncol = 3, nrow = 4)
```

Exploratory Data Analysis

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

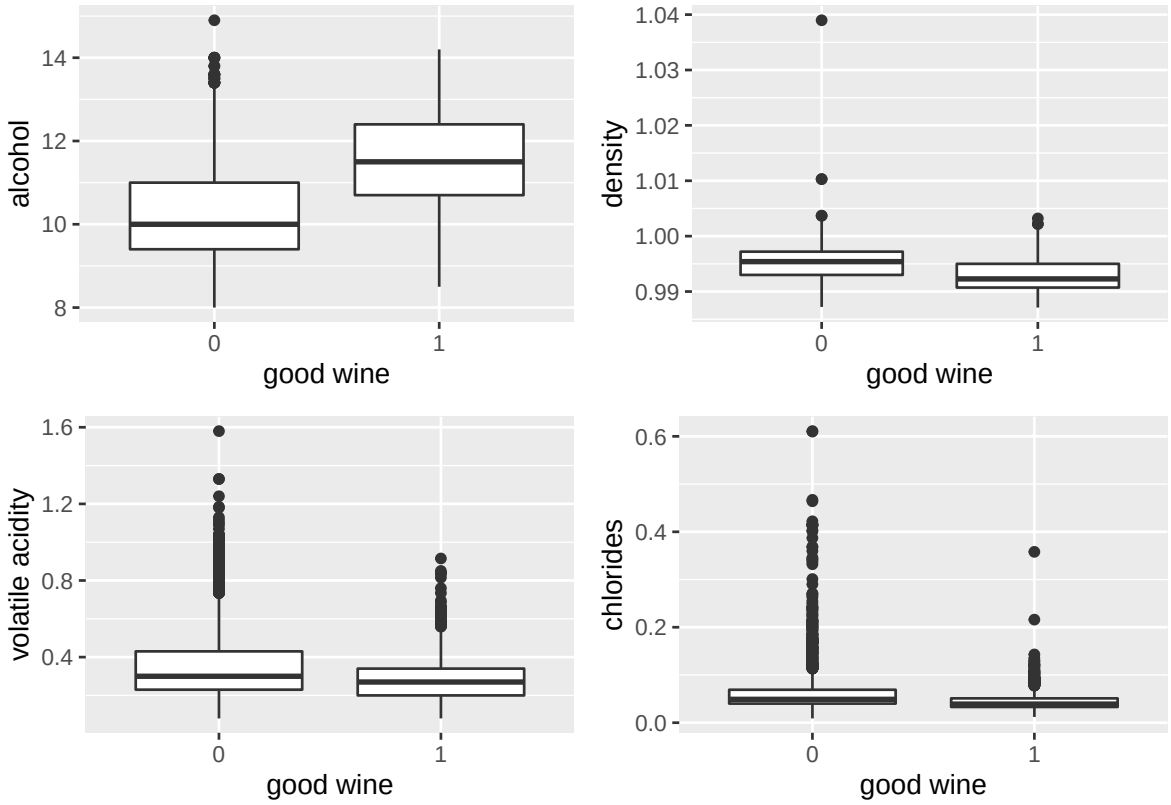
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`.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



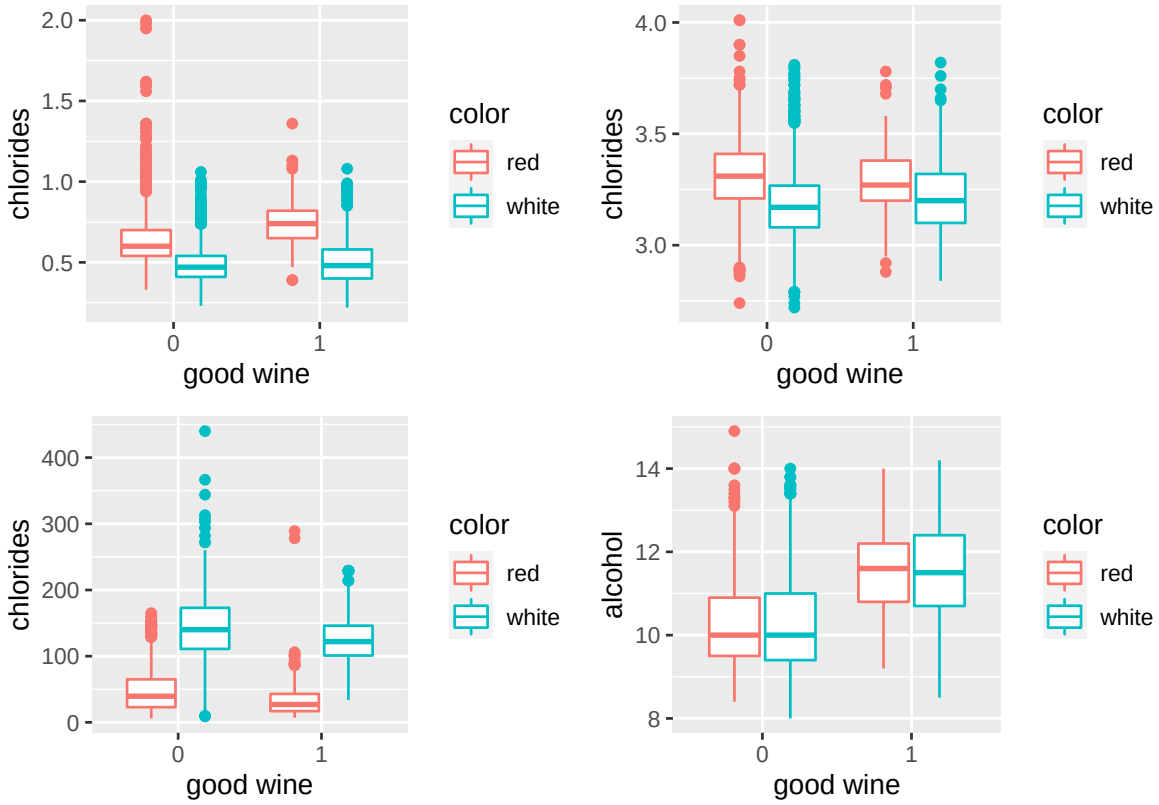

```
#cor(wine$c_alcohol, wine$quality)
#cor(wine$c_density, wine$quality)
#cor(wine$c_volatile.acidity, wine$quality)
#cor(wine$c_chlorides, wine$quality)
#cor(wine$c_residual.sugar, wine$quality)
#cor(wine$c_fixed.acidity, wine$quality)
#cor(wine$c_free.sulfur.dioxide, wine$quality)
```

```
#cor(wine$c_total.sulfur.dioxide, wine$quality)
#cor(wine$c_pH, wine$quality)
#cor(wine$c_sulphates, wine$quality)
#cor(wine$c_citric.acid, wine$quality)
```

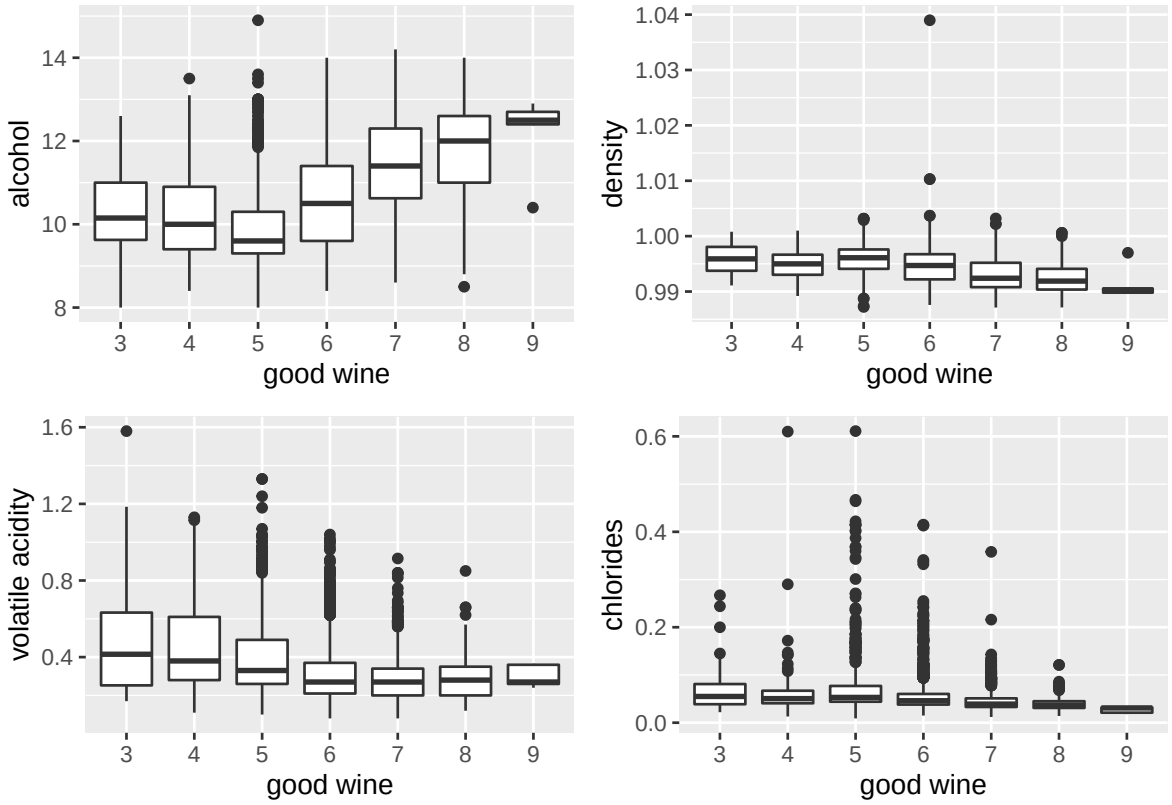
```
a1 <- ggplot(wine, aes(y = c_alcohol, x = as.factor(good_wine))) +
  geom_boxplot() +
  labs(x = "good wine", y = "alcohol")

a2 <- ggplot(wine, aes(y = c_density, x = as.factor(good_wine))) +
  geom_boxplot() +
  labs(x = "good wine", y = "density")

a3 <- ggplot(wine, aes(y = c_volatile.acidity, x = as.factor(good_wine))) +
  geom_boxplot() +
  labs(x = "good wine", y = "volatile acidity")

a4 <- ggplot(wine, aes(y = c_chlorides, x = as.factor(good_wine))) +
  geom_boxplot() +
  labs(x = "good wine", y = "chlorides")

plot_grid(a1, a2, a3, a4, ncol = 2, nrow = 2)
```



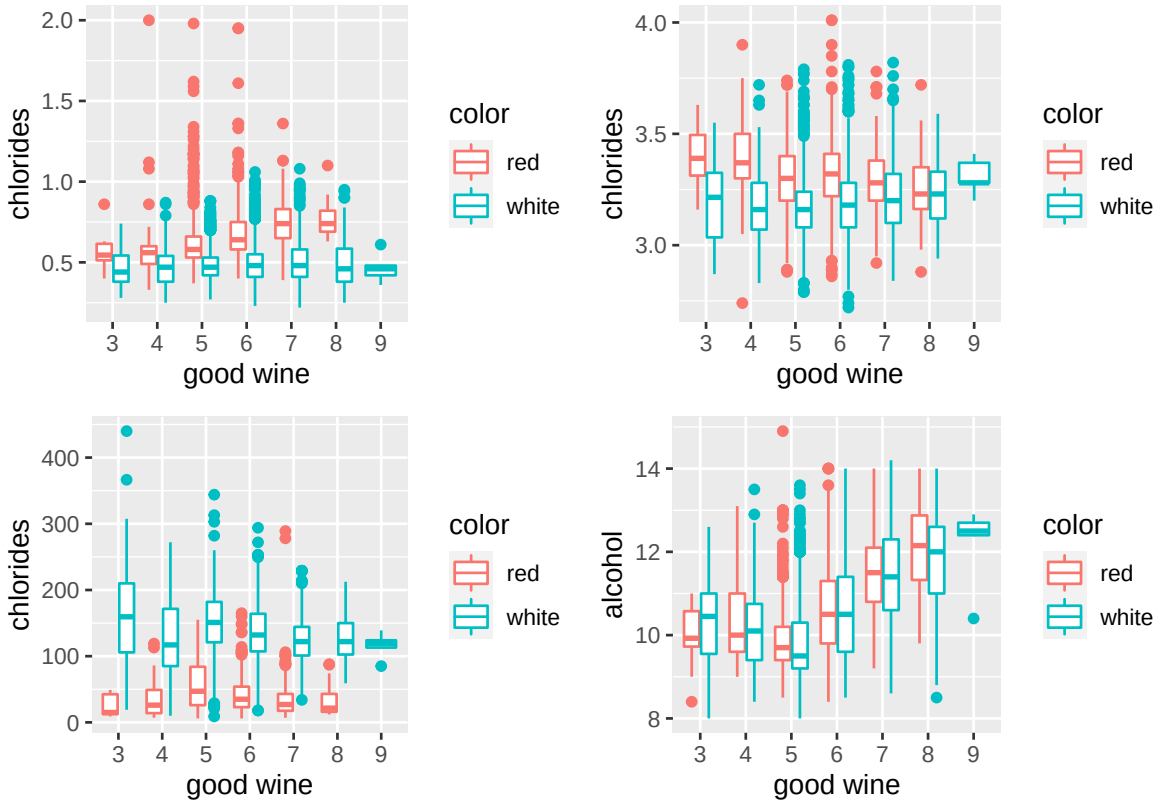

```
# exploring interaction effects
a5 <- ggplot(wine, aes(y = c_sulphates, x = as.factor(good_wine), color = color)) +
  geom_boxplot() +
  labs(x = "good wine", y = "chlorides")

a6 <- ggplot(wine, aes(y = c_pH, x = as.factor(good_wine), color = color)) +
  geom_boxplot() +
  labs(x = "good wine", y = "chlorides")

a7 <- ggplot(wine, aes(y = c_total.sulfur.dioxide, x = as.factor(good_wine), color = color))
  geom_boxplot() +
  labs(x = "good wine", y = "chlorides")

a8 <- ggplot(wine, aes(y = c_alcohol, x = as.factor(good_wine), color = color)) +
  geom_boxplot() +
  labs(x = "good wine", y = "alcohol")

plot_grid(a5, a6, a7, a8, ncol = 2, nrow = 2)
```




```
a1 <- ggplot(wine, aes(y = c_alcohol, x = as.factor(quality))) +
  geom_boxplot() +
  labs(x = "good wine", y = "alcohol")

a2 <- ggplot(wine, aes(y = c_density, x = as.factor(quality))) +
  geom_boxplot() +
  labs(x = "good wine", y = "density")

a3 <- ggplot(wine, aes(y = c_volatile.acidity, x = as.factor(quality))) +
  geom_boxplot() +
  labs(x = "good wine", y = "volatile acidity")

a4 <- ggplot(wine, aes(y = c_chlorides, x = as.factor(quality))) +
  geom_boxplot() +
  labs(x = "good wine", y = "chlorides")

plot_grid(a1, a2, a3, a4, ncol = 2, nrow = 2)
```




```
# exploring interaction effects
a5 <- ggplot(wine, aes(y = c_sulphates, x = as.factor(quality), color = color)) +
  geom_boxplot() +
  labs(x = "good wine", y = "chlorides")

a6 <- ggplot(wine, aes(y = c_pH, x = as.factor(quality), color = color)) +
  geom_boxplot() +
  labs(x = "good wine", y = "chlorides")

a7 <- ggplot(wine, aes(y = c_total.sulfur.dioxide, x = as.factor(quality), color = color)) +
  geom_boxplot() +
  labs(x = "good wine", y = "chlorides")

a8 <- ggplot(wine, aes(y = c_alcohol, x = as.factor(quality), color = color)) +
  geom_boxplot() +
  labs(x = "good wine", y = "alcohol")

plot_grid(a5, a6, a7, a8, ncol = 2, nrow = 2)
```



Methodology

Initial Logistic Model Data split: we split the data into 25% testing set and 75% training set.

```
set.seed(222)

wine_split <- initial_split(wine, prop = 3/4)
wine_train <- training(wine_split)
wine_test <- training(wine_split)
```

```
wine_spec <- logistic_reg() %>% set_engine("glm")
```

We plan to compare a full model and a reduced model. For the full model, we decide to include interactive terms. For the reduced model, we adopt the stepwise AIC test to get a best AIC model.

Initial speculation of the model. We'll include all the predictor variables and include all the interactive terms with colors at first. ::: {.cell}

```
wine_rec_initial <- recipe(
  good_wine ~., data = wine_train) %>%
  step_rm(good_wine_names) %>%
  step_rm(quality) %>%
  step_interact(terms = ~starts_with("c_"):color) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors())
```

:::

```
wine_wflow_initial <- workflow() %>%
  add_model(wine_spec) %>%
  add_recipe(wine_rec_initial)
```

```
wine_fit_initial <- wine_wflow_initial %>%
  fit(data = wine_train)
```

```
Warning: Categorical variables used in `step_interact` should probably be
avoided; This can lead to differences in dummy variable values that are produced
by `step_dummy`. Please convert all involved variables to dummy variables first.
```

```
kable(tidy(wine_fit_initial), digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	214.287	122.689	1.747	0.081
c_fixed.acidity	0.289	0.143	2.025	0.043
c_volatile.acidity	-2.314	0.912	-2.537	0.011
c_citric.acid	0.713	0.980	0.728	0.467
c_residual.sugar	0.219	0.086	2.558	0.011
c_chlorides	-8.354	3.948	-2.116	0.034
c_free.sulfur.dioxide	0.016	0.014	1.150	0.250
c_total.sulfur.dioxide	-0.019	0.006	-3.197	0.001
c_density	-231.611	125.241	-1.849	0.064
c_pH	0.556	1.128	0.493	0.622
c_sulphates	3.280	0.653	5.026	0.000
c_alcohol	0.877	0.153	5.731	0.000
c_fixed.acidity_x_colorwhite	0.276	0.177	1.558	0.119

term	estimate	std.error	statistic	p.value
c_volatile.acidity_x_colorwhite	-1.050	1.069	-0.982	0.326
c_citric.acid_x_colorwhite	-1.038	1.080	-0.961	0.337
c_residual.sugar_x_colorwhite	0.073	0.095	0.774	0.439
c_chlorides_x_colorwhite	-4.376	5.907	-0.741	0.459
c_free.sulfur.dioxide_x_colorwhite	-0.007	0.015	-0.469	0.639
c_total.sulfur.dioxide_x_colorwhite	0.018	0.006	2.852	0.004
c_density_x_colorwhite	-438.935	166.379	-2.638	0.008
c_pH_x_colorwhite	2.919	1.231	2.370	0.018
c_sulphates_x_colorwhite	-1.359	0.765	-1.777	0.076
c_alcohol_x_colorwhite	-0.747	0.201	-3.706	0.000
color_white	432.926	163.484	2.648	0.008

From the table we can see that there are many variables with p-values greater than 0.05 under 95% CI, so we can drop some of those valuables if we were to conduct to a hypothesis test.

Full Logistic Model $log\ odds(good\ wine) = 290.153 + 0.371 * fixed\ acidity - 3.217 * volatile + 0.256 * residual\ sugar - 11.862 * chlorides - 0.011 * total.sulfur.dioxide - 304.879 * density + 3.63 * sulphates + 0.766 * alcohol + 0.015 * total.sulfur.dioxidexcolorwhite - 265.692 * densityxcolorwhite - 1.534 * sulphatesxcolorwhite - 0.511 * alcoholxcolorwhite + 259.744 * colorwhite$

```
wine_rec_full <- recipe(
  good_wine ~., data = wine_train) %>%
  step_rm(good_wine_names) %>%
  step_rm(quality) %>%
  step_rm(c_pH) %>%
  step_rm(c_free.sulfur.dioxide) %>%
  step_rm(c_citric.acid) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms = ~starts_with("c_"):starts_with("color")) %>%
  step_rm(c_chlorides_x_color_white) %>%
  step_rm(c_residual.sugar_x_color_white) %>%
  step_rm(c_volatile.acidity_x_color_white) %>%
  step_rm(c_fixed.acidity_x_color_white) %>%
  step_zv(all_predictors())
prep(wine_rec_full)%>%bake(wine_train)
```

```
# A tibble: 4,872 x 14
   c_fixed.acidity c_volatile.acidity c_residual.sugar c_chlorides
```

```
             <dbl>              <dbl>             <dbl>       <dbl>
 1             6                 0.34               15.9       0.046
 2             6.8               0.52               13.2       0.044
 3             6.3               0.23               10.4       0.043
 4             7.2               0.54                2.6       0.084
 5             9                 0.245               5.9       0.045
 6             5.8               0.2                 1.4       0.042
 7             7.4               0.785               5.2       0.094
 8             5.8               0.415               1.4       0.04
 9             9.5               0.42                2.3       0.034
10             7.1               0.46                2.8       0.076
# ... with 4,862 more rows, and 10 more variables:
#   c_total.sulfur.dioxide <dbl>, c_density <dbl>, c_sulphates <dbl>,
#   c_alcohol <dbl>, good_wine <fct>, color_white <dbl>,
#   c_total.sulfur.dioxide_x_color_white <dbl>, c_density_x_color_white <dbl>,
#   c_sulphates_x_color_white <dbl>, c_alcohol_x_color_white <dbl>
```

```
wine_wflow_full <- workflow() %>%
  add_model(wine_spec) %>%
  add_recipe(wine_rec_full)
```

```
wine_fit_full <- wine_wflow_full %>%
  fit(data = wine_train)
kable(tidy(wine_fit_full), digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-3.788	74.245	-0.051	0.959
c_fixed.acidity	0.086	0.052	1.647	0.100
c_volatile.acidity	-3.589	0.428	-8.384	0.000
c_residual.sugar	0.132	0.025	5.250	0.000
c_chlorides	-12.161	3.031	-4.012	0.000
c_total.sulfur.dioxide	-0.016	0.004	-3.819	0.000
c_density	-9.272	74.159	-0.125	0.901
c_sulphates	3.089	0.616	5.014	0.000
c_alcohol	1.033	0.124	8.349	0.000
color_white	246.587	77.963	3.163	0.002
c_total.sulfur.dioxide_x_color_white	0.017	0.004	3.817	0.000
c_density_x_color_white	-243.759	77.484	-3.146	0.002
c_sulphates_x_color_white	-1.439	0.720	-1.998	0.046
c_alcohol_x_color_white	-0.432	0.145	-2.982	0.003

All the p-values are very small. We'll now apply this recipe to test set to make prediction.

```
wine_full_test <- wine_wflow_full %>%
  fit(data = wine_test)

tidy(wine_full_test) %>%
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-3.788	74.245	-0.051	0.959
c_fixed.acidity	0.086	0.052	1.647	0.100
c_volatile.acidity	-3.589	0.428	-8.384	0.000
c_residual.sugar	0.132	0.025	5.250	0.000
c_chlorides	-12.161	3.031	-4.012	0.000
c_total.sulfur.dioxide	-0.016	0.004	-3.819	0.000
c_density	-9.272	74.159	-0.125	0.901
c_sulphates	3.089	0.616	5.014	0.000
c_alcohol	1.033	0.124	8.349	0.000
color_white	246.587	77.963	3.163	0.002
c_total.sulfur.dioxide_x_color_white	0.017	0.004	3.817	0.000
c_density_x_color_white	-243.759	77.484	-3.146	0.002
c_sulphates_x_color_white	-1.439	0.720	-1.998	0.046
c_alcohol_x_color_white	-0.432	0.145	-2.982	0.003

```
wine_full_fit <- predict(wine_fit_full, wine_train, type = "prob") %>% bind_cols(wine_train)
wine_full_fit
```

```
# A tibble: 4,872 x 17
   .pred_0 .pred_1 c_fixed.acidity c_volatile.acidity c_citric.acid
     <dbl>   <dbl>           <dbl>              <dbl>         <dbl>
 1   0.933 0.0673              6                0.34          0.66
 2   0.945 0.0554              6.8              0.52          0.32
 3   0.898 0.102               6.3              0.23          0.5
 4   0.938 0.0623              7.2              0.54          0.27
 5   0.882 0.118               9                0.245         0.38
 6   0.502 0.498               5.8              0.2           0.16
 7   0.997 0.00319             7.4              0.785         0.19
 8   0.912 0.0884              5.8              0.415         0.13
 9   0.901 0.0986              9.5              0.42          0.41
10   0.934 0.0665              7.1              0.46          0.14
```

```
# ... with 4,862 more rows, and 12 more variables: c_residual.sugar <dbl>,
#   c_chlorides <dbl>, c_free.sulfur.dioxide <dbl>,
#   c_total.sulfur.dioxide <dbl>, c_density <dbl>, c_pH <dbl>,
#   c_sulphates <dbl>, c_alcohol <dbl>, quality <int>, color <chr>,
#   good_wine <fct>, good_wine_names <chr>
```

Stepwise AIC Model

- We need to include reasons here. Why choose stepwise AIC model as the reduced model?

```
AIC_fit <- logistic_reg() %>%
  set_engine("glm") %>%
  fit(good_wine~.-c_citric.acid-quality-good_wine_names, data = wine_train)

AIC_fit <- repair_call(AIC_fit, data = wine_train)
AIC_fit_engine <- AIC_fit %>% extract_fit_engine()
```

```
best_AIC_model <- stepAIC(AIC_fit_engine, direction="forward", trace=FALSE)
tidy(best_AIC_model)
```

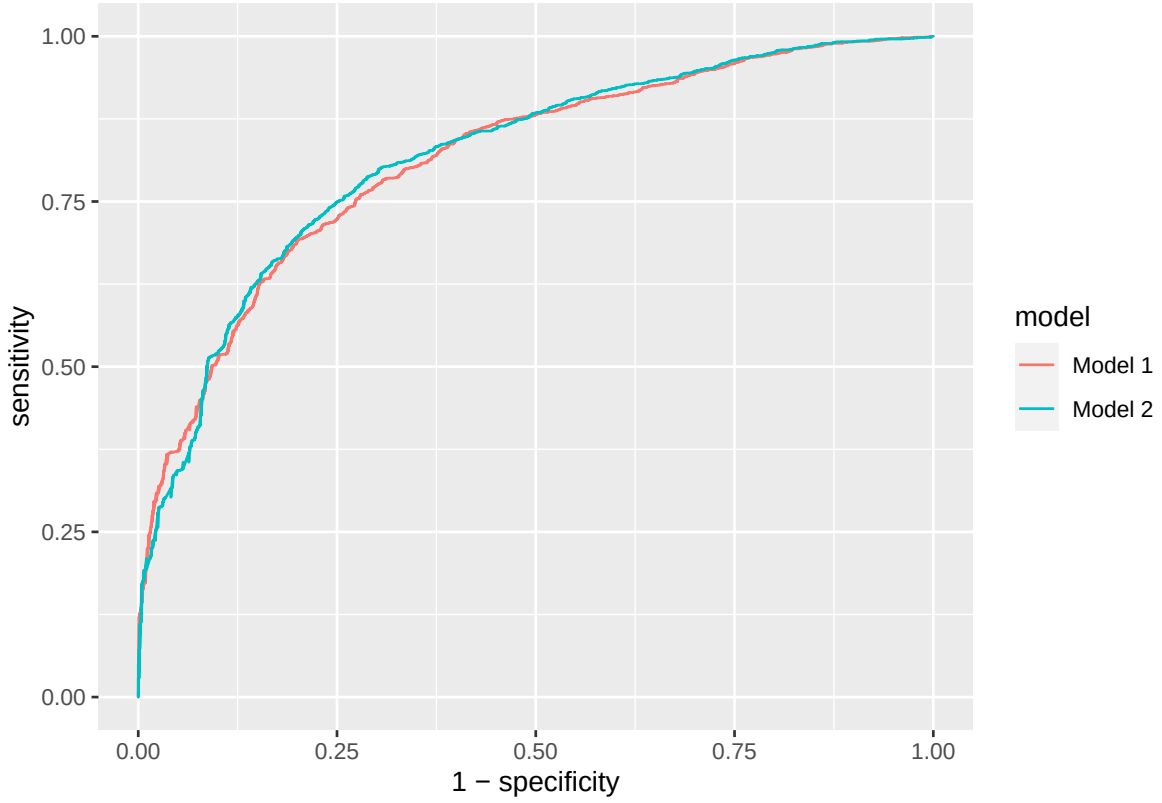
```
# A tibble: 12 x 5
   term                   estimate std.error statistic  p.value
   <chr>                     <dbl>     <dbl>     <dbl>    <dbl>
 1 (Intercept)             384.       74.9          5.13 2.92e- 7
 2 c_fixed.acidity           0.487     0.0756       6.44 1.20e-10
 3 c_volatile.acidity       -3.37      0.419       -8.04 9.09e-16
 4 c_residual.sugar          0.208     0.0300       6.95 3.57e-12
 5 c_chlorides              -7.42      2.85        -2.61 9.16e- 3
 6 c_free.sulfur.dioxide     0.0119    0.00340      3.51 4.53e- 4
 7 c_total.sulfur.dioxide   -0.00474   0.00154     -3.09 2.01e- 3
 8 c_density              -405.       75.9         -5.34 9.23e- 8
 9 c_pH                      2.66      0.415        6.40 1.56e-10
10 c_sulphates               2.10      0.330        6.35 2.19e-10
11 c_alcohol                 0.496     0.0916       5.41 6.28e- 8
12 colorwhite               -0.611     0.280       -2.18 2.89e- 2
```

```
best_AIC_fit <- predict(best_AIC_model, wine_train) %>% bind_cols(wine_train)
```

```
New names:
* `` -> ...1
```

```
best_AIC_fit <- best_AIC_fit %>% mutate(.pred_1 = exp(...1) / (1 + exp(...1))) %>% mutate(.pr
```

```
wine_full_fit %>% roc_curve(truth = as.factor(good_wine), .pred_0) %>% mutate(model = "Model
  bind_rows(best_AIC_fit %>% roc_curve(truth = as.factor(good_wine), .pred_0) %>% mutate(mode
  ggplot(aes(x = 1 - specificity, y = sensitivity, color = model)) +
  geom_line()
```




```
wine_full_fit %>%
  roc_auc(truth = as.factor(good_wine), .pred_0)
```

Model Selection

```
# A tibble: 1 x 3
```

```
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.813
```

```
best_AIC_fit %>%
  roc_auc(truth = as.factor(good_wine), .pred_0)
```

```
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 roc_auc binary         0.817
```

```
best_AIC_rec <- recipe(
  good_wine ~ c_fixed.acidity + c_volatile.acidity +
    c_citric.acid + c_residual.sugar + c_chlorides + c_free.sulfur.dioxide +
    c_total.sulfur.dioxide + c_density + c_pH + c_sulphates +
    c_alcohol + quality + color + good_wine_names, data = wine_train) %>%
  step_rm(c_citric.acid) %>%
  step_rm(quality) %>%
  step_rm(good_wine_names) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_zv(all_predictors())

wine_wflow_reduce <- workflow() %>%
  add_model(wine_spec) %>%
  add_recipe(best_AIC_rec)
```

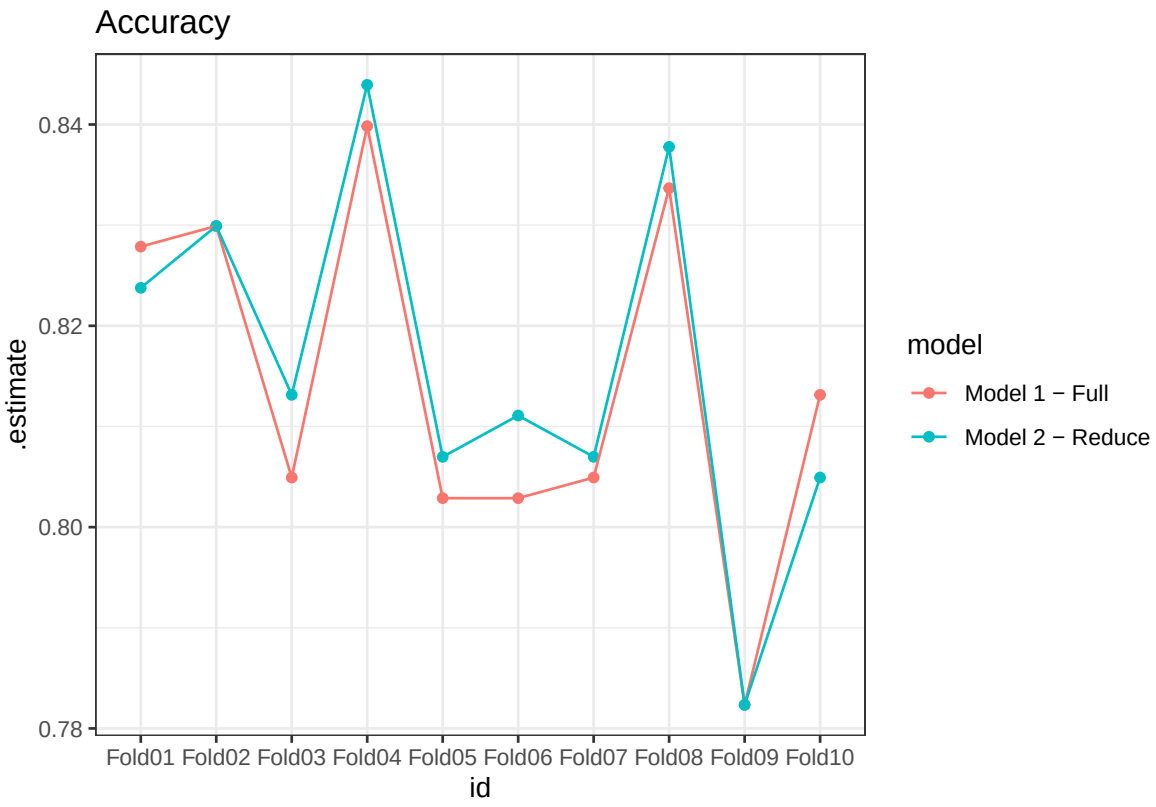
```
set.seed(345)
folds <- vfold_cv(wine_train, v = 10)
```

```
wine_full_rs <- wine_wflow_full %>%
  fit_resamples(folds)

wine_AIC_rs <- wine_wflow_reduce %>%
  fit_resamples(folds)
```

```
metrics_full <- collect_metrics(wine_full_rs, summarize = FALSE) %>% mutate(model = "Model 1
metrics_reduce <- collect_metrics(wine_AIC_rs, summarize = FALSE) %>% mutate(model = "Model 2
```

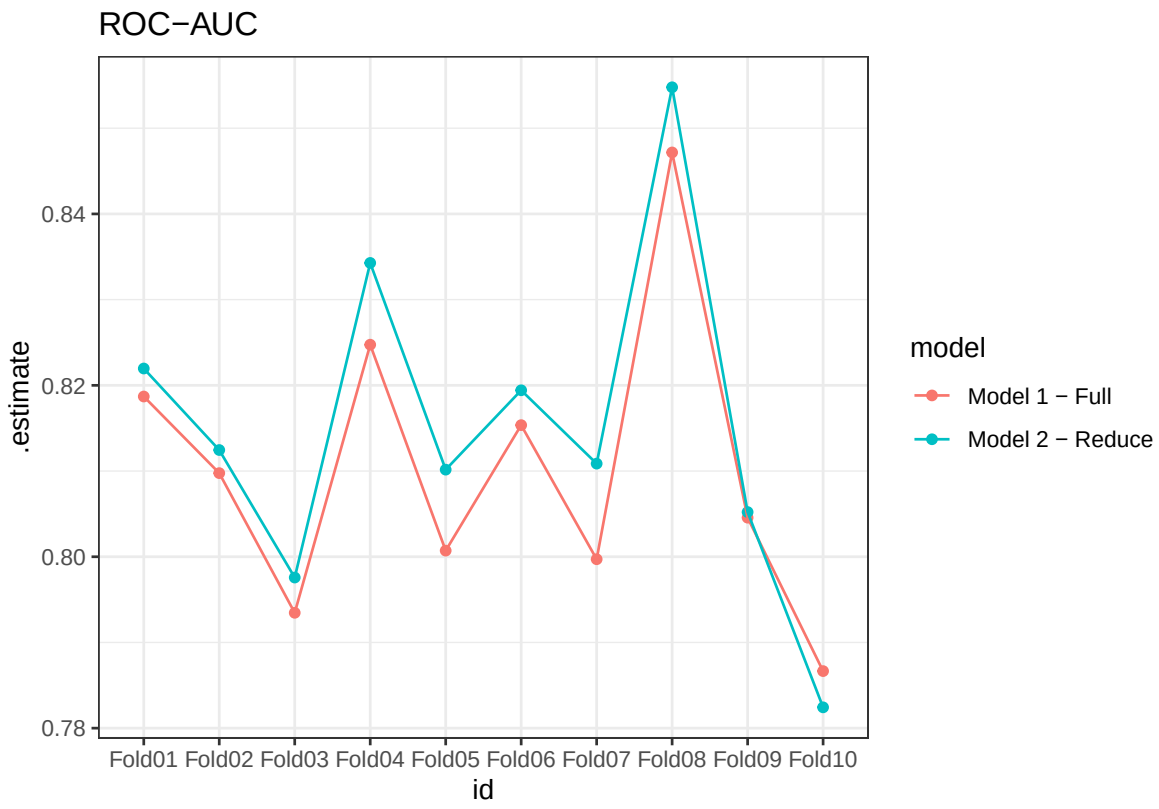
Cross Validation Graph for metrics ::: {.cell}

```
metrics <- bind_rows(metrics_full, metrics_reduce) %>%
arrange(.metric)
ggplot(metrics %>% filter(.metric == "accuracy"),
aes(x = id, y = .estimate,
group = model, color = model)) +
  geom_point() +
  geom_line() +
  labs(title = "Accuracy") +
  theme_bw()
```



::: ::: {.cell}

```
ggplot(metrics %>% filter(.metric == "roc_auc"),

aes(x = id, y = .estimate,
    group = model, color = model)) +
  geom_point() +
  geom_line() +
  labs(title = "ROC-AUC") +
  theme_bw()
```



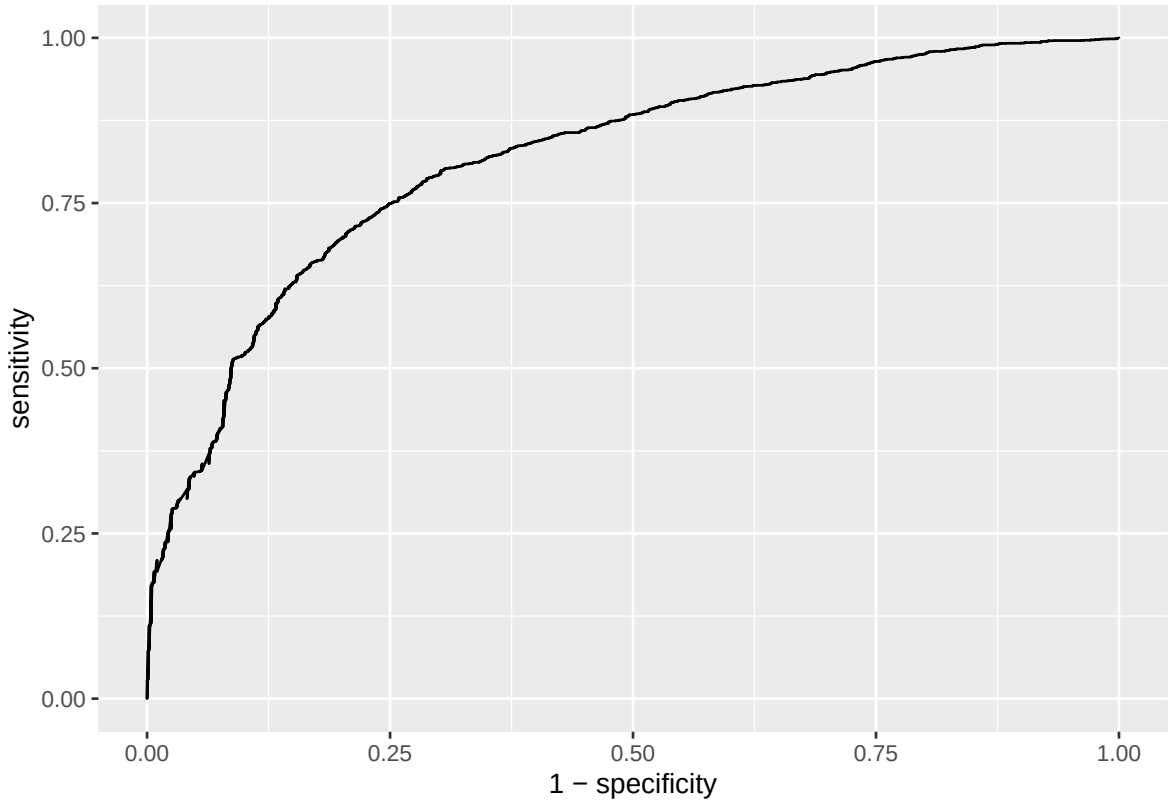
::: The two models have similar accuracy and the roc-auc for 10 folds, so due to the principles of parsimonious, we prefer the reduce one.

The AOC curve of the reduce model's prediction: ::: {.cell}

```
best_AIC_pred <- predict(best_AIC_model, wine_test) %>% bind_cols(wine_test)
```

```
New names:
* `` -> ...1
```

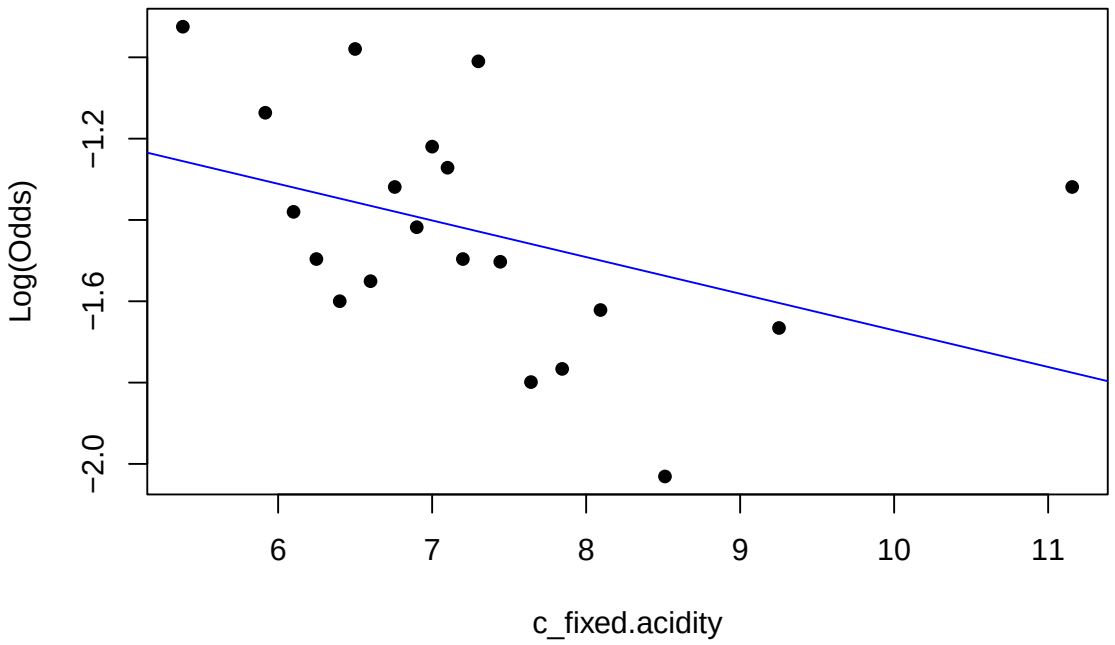
```
best_AIC_pred <- best_AIC_fit %>% mutate(.pred_1 = exp(...1) / (1 + exp(...1))) %>% mutate(.p
best_AIC_pred %>% roc_curve(truth = as.factor(good_wine), .pred_0) %>%
  ggplot(aes(x = 1 - specificity, y = sensitivity)) +
  geom_line()
```



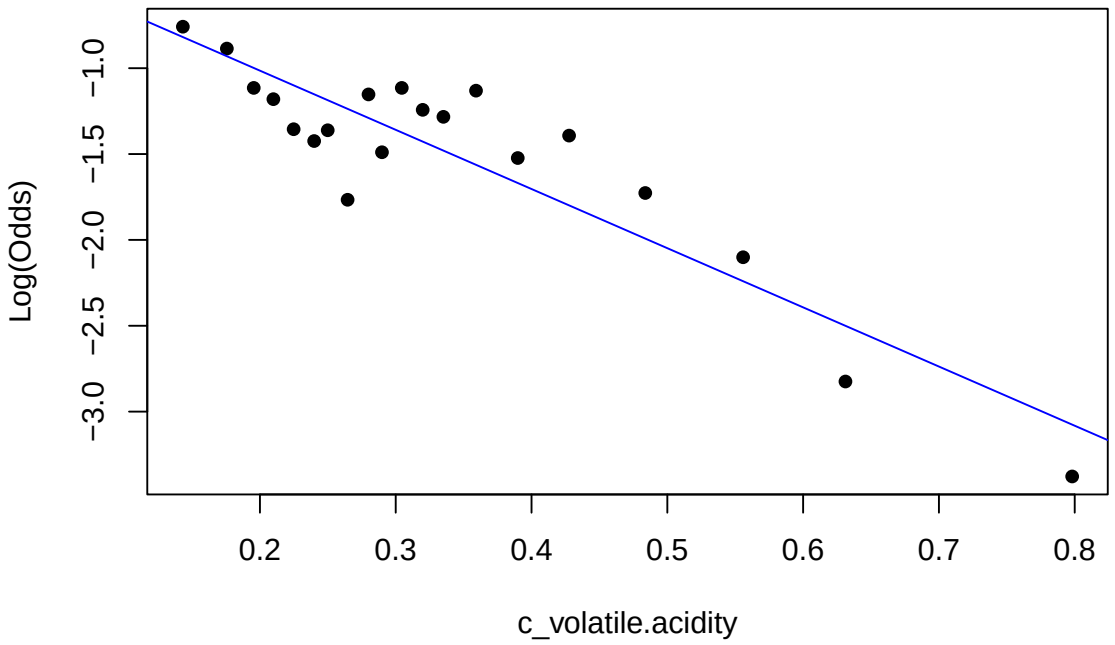
:::

Conditions Logistic Model: linearity ::: {.cell}

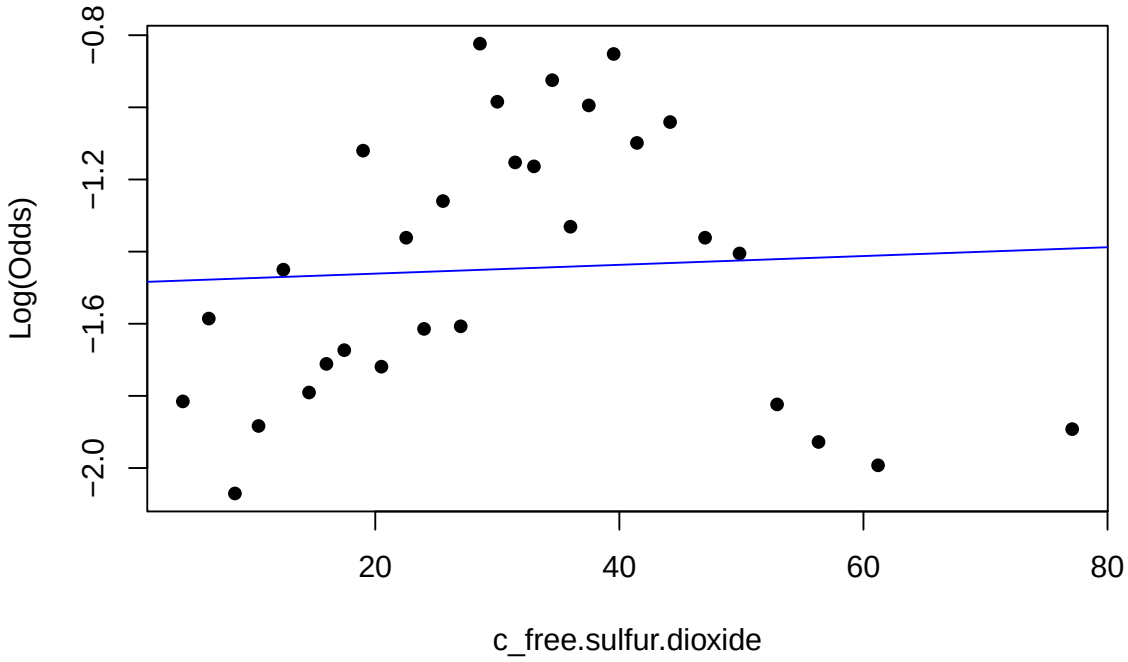
```
emplogitplot1(good_wine ~ c_fixed.acidity,
              data = wine,
              ngroups = 20)
```



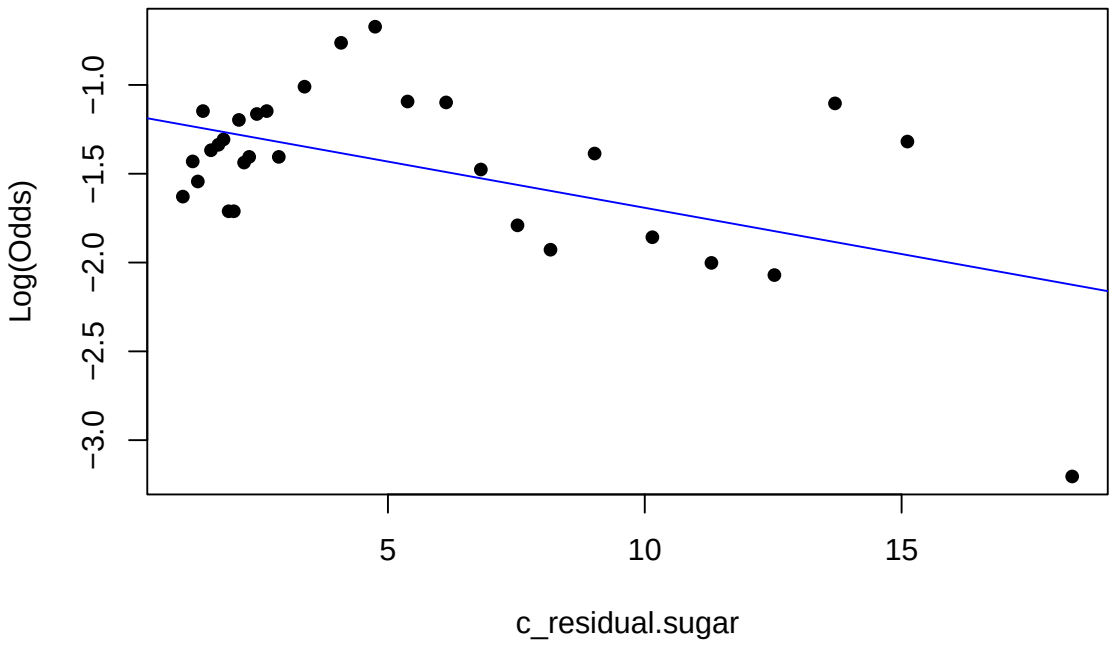

```
emplogitplot1(good_wine ~ c_volatile.acidity,
              data = wine,
              ngroups = 20)
```



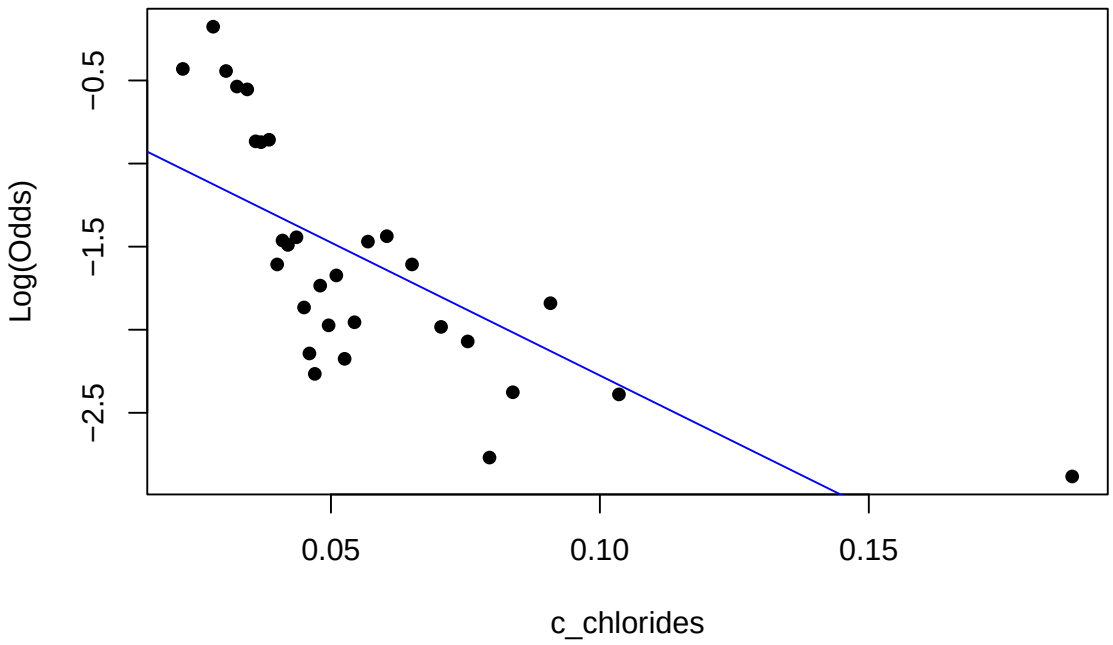

```
emplogitplot1(good_wine ~ c_free.sulfur.dioxide,
              data = wine,
              ngroups = 30)
```



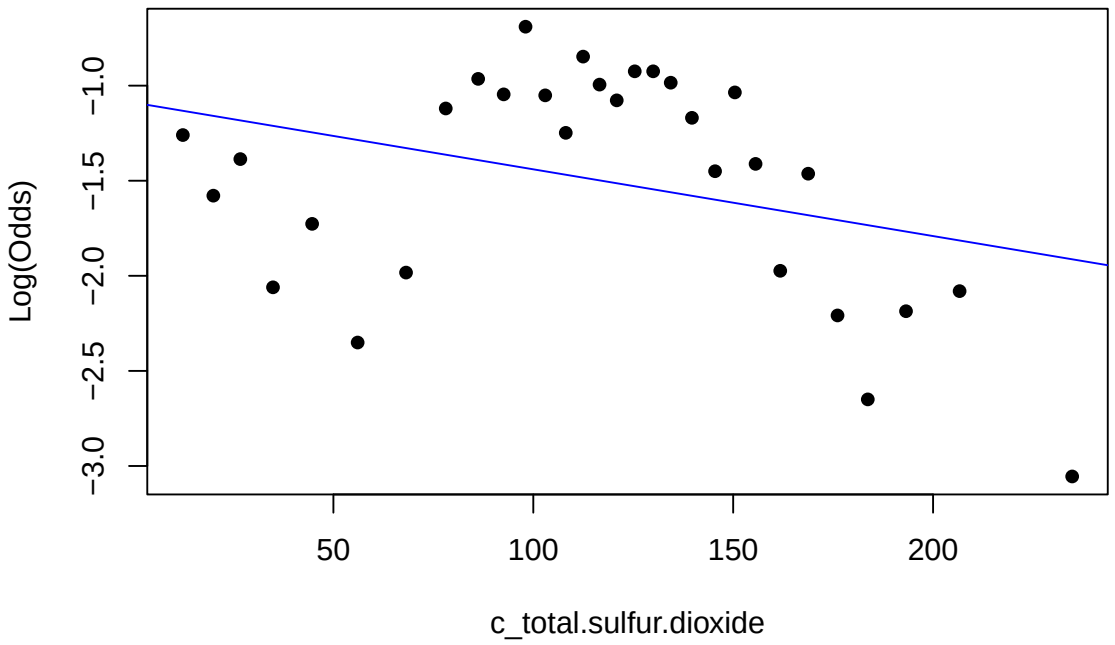

```
emplogitplot1(good_wine ~ c_residual.sugar,
              data = wine,
              ngroups = 30)
```



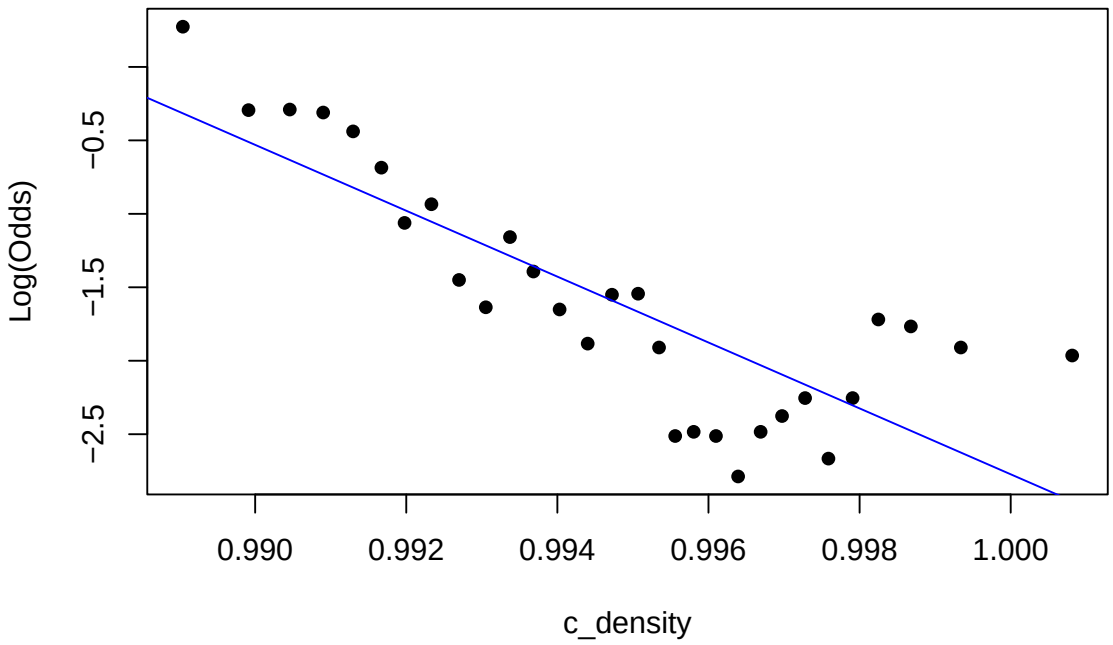

```
emplogitplot1(good_wine ~ c_chlorides,
              data = wine,
              ngroups = 30)
```



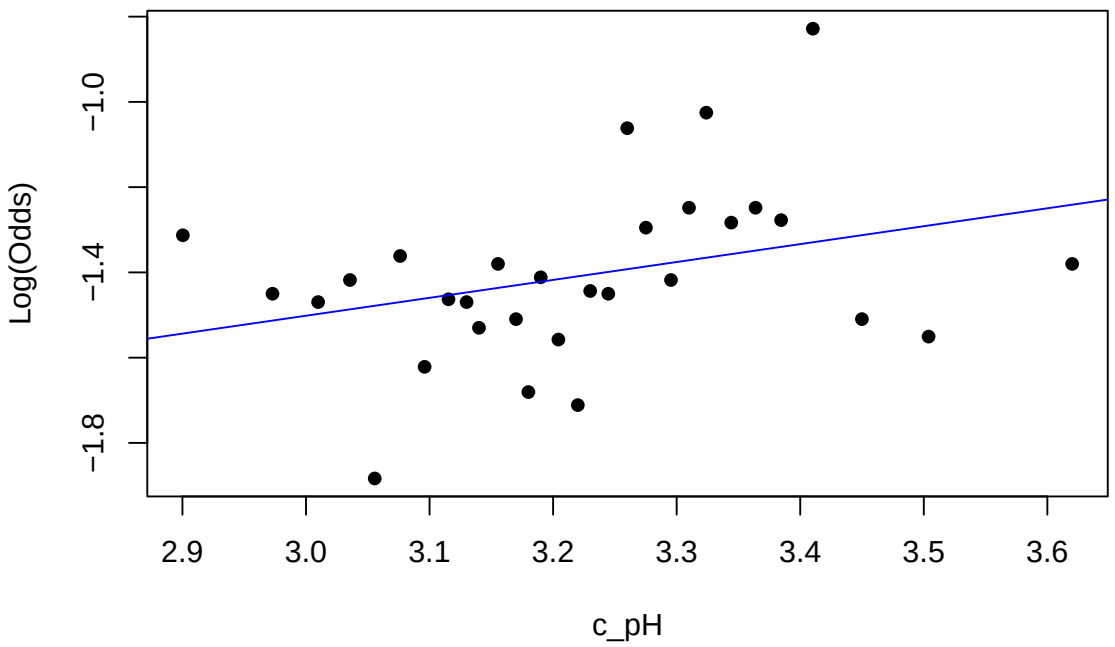

```
emplogitplot1(good_wine ~ c_total.sulfur.dioxide,
              data = wine,
              ngroups = 30)
```



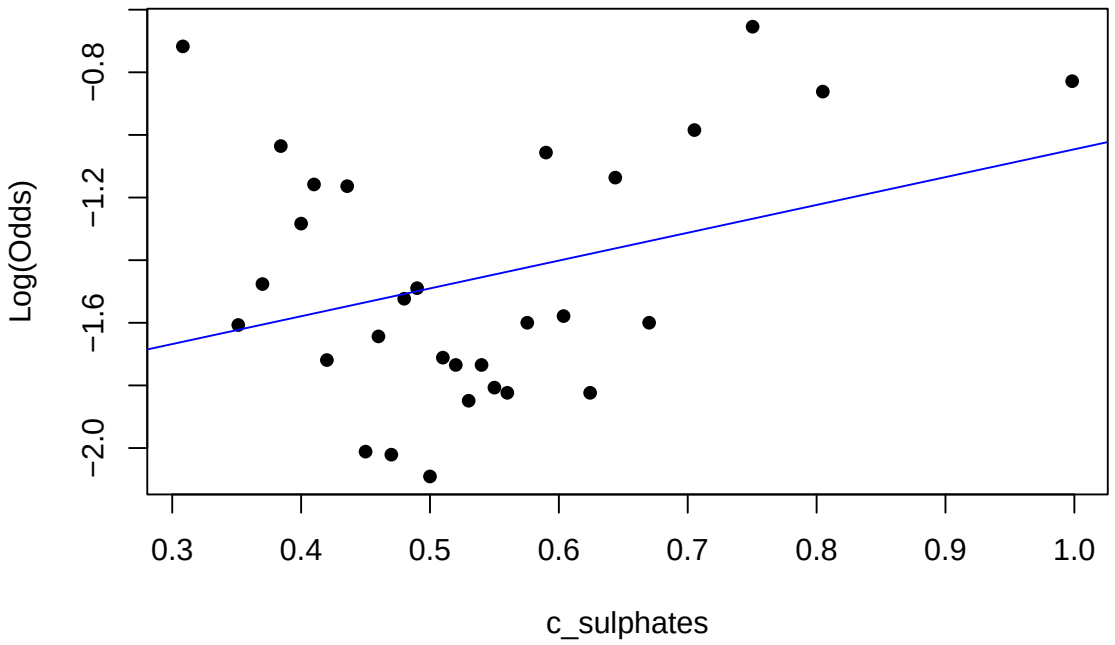

```
emplogitplot1(good_wine ~ c_density,
              data = wine,
              ngroups = 30)
```



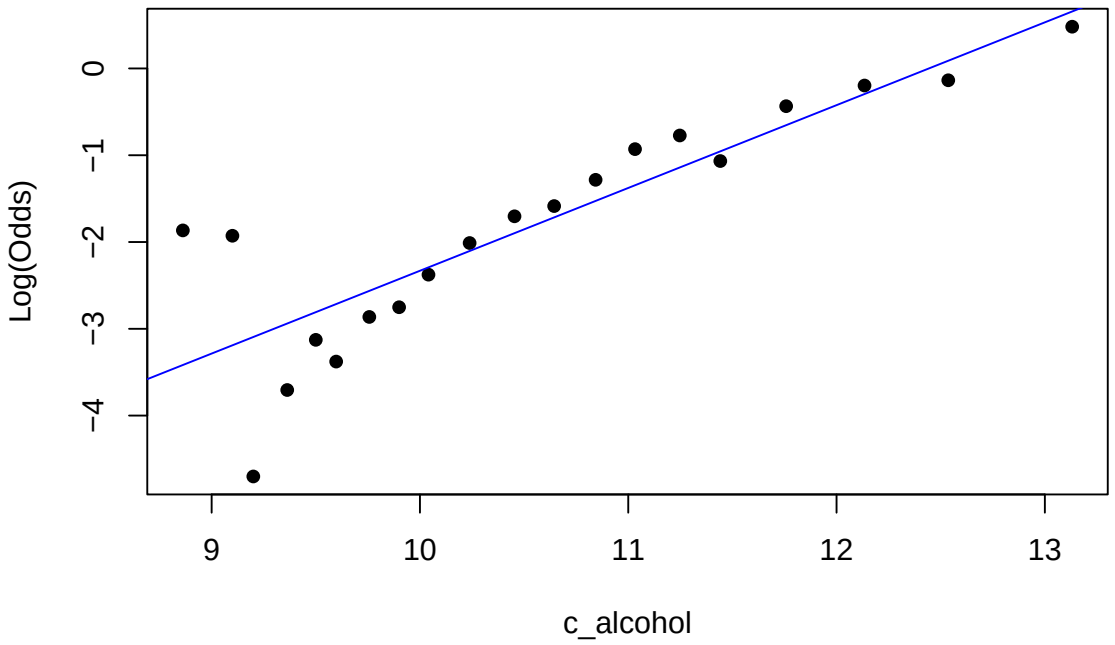

```
emplogitplot1(good_wine ~ c_pH + color,
              data = wine,
              ngroups = 30)
```




```
emplogitplot1(good_wine ~ c_sulphates,
              data = wine,
              ngroups = 30)
```




```
emplogitplot1(good_wine ~ c_alcohol + color,
              data = wine,
              ngroups = 20)
```



:::

Not every variables satisfy the linearity condition. (This can be mentioned in the limitation part of the final report.)

Independence The independence is satisfied. The Vinho Verde region is a vast region spreading 15500 hectareas of vineyards in far-north Portugal, and the data are not collected across time.

Results

- model interpretation
- full model
- best AIC model
- AIC BIC ROC
- model selection: do not include the interactive terms due to parsimonious. because the ROC does not improve much.

Discussion & Conclusion

- what chemical components contribute to wine quality
- future research suggestions
- suggestions to wine valley?

Reference

Paulo Cortez, University of Minho, Guimarães, Portugal, http://www3.dsi.uminho.pt/pcortez
A. Cerdeira, F. Almeida, T. Matos and J. Reis, Viticulture Commission of the Vinho Verde Region(CVRVV), Porto, Portugal, 2009.