Homework 4 – Due Monday October 5, 2015

Cheryl Calhoun

09/29/2015

Add your code blocks to this document and alter the Author name to yours. Your submission will consist of your own R Markdown file plus the compiled HTML version of the document.

I encourage you to submit your completed version as soon as possible. We will grade this quickly so that any opportunities for a regrade can be done quickly.

Assignment

For this assignment we're going to take two data sets with identical structural properties, except for one major difference: one is a data set of red wines, the other of white. Your goal will be to come up with the best predictors of quality for a wine given the scientific characteristics observed. We obtained the data from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Wine+Quality) but pre-processed it a little for the class.

1. Load the data file `wine-tests.RData` into your workspace. Note that one of R's nice features is that you can identify the variables contained in an RData file by wrapping the `load()` function in a `print()` function. What variables are contained within? Confirm that that they are indeed data frames.

```
print(load("wine-tests.RData"))
```

```
## [1] "red"   "white"
```

```
## Determine that these are `red` is a data frame, identify the variables within, and review the summary
is.data.frame(red)
```

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

```
str(red)
```

```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
```

```
summary(red)
```

```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
```

```
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00
##  Median :0.07900   Median :14.00       Median : 38.00
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00
##     density             pH          sulphates         alcohol
##  Min.   :0.9901   Min.   :2.740   Min.   :0.3300   Min.   : 8.40
##  1st Qu.:0.9956   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50
##  Median :0.9968   Median :3.310   Median :0.6200   Median :10.20
##  Mean   :0.9967   Mean   :3.311   Mean   :0.6581   Mean   :10.42
##  3rd Qu.:0.9978   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
##  Max.   :1.0037   Max.   :4.010   Max.   :2.0000   Max.   :14.90
##     quality
##  Min.   :3.000
##  1st Qu.:5.000
##  Median :6.000
##  Mean   :5.636
##  3rd Qu.:6.000
##  Max.   :8.000
```

```
##  Determine that these are `white` is a data frame, identify the variables within, and review the summ
is.data.frame(white)
```

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

```
str(white)
```

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

```
summary(white)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
##  Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200
##  Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391
##  3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900
##  Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   :0.00900   Min.   :  2.00      Min.   :  9.0
```

```
##  1st Qu.:0.03600   1st Qu.: 23.00      1st Qu.:108.0
##  Median :0.04300   Median : 34.00      Median :134.0
##  Mean   :0.04577   Mean   : 35.31      Mean   :138.4
##  3rd Qu.:0.05000   3rd Qu.: 46.00      3rd Qu.:167.0
##  Max.   :0.34600   Max.   :289.00      Max.   :440.0
##     density             pH          sulphates         alcohol
##  Min.   :0.9871   Min.   :2.720   Min.   :0.2200   Min.   : 8.00
##  1st Qu.:0.9917   1st Qu.:3.090   1st Qu.:0.4100   1st Qu.: 9.50
##  Median :0.9937   Median :3.180   Median :0.4700   Median :10.40
##  Mean   :0.9940   Mean   :3.188   Mean   :0.4898   Mean   :10.51
##  3rd Qu.:0.9961   3rd Qu.:3.280   3rd Qu.:0.5500   3rd Qu.:11.40
##  Max.   :1.0390   Max.   :3.820   Max.   :1.0800   Max.   :14.20
##     quality
##  Min.   :3.000
##  1st Qu.:5.000
##  Median :6.000
##  Mean   :5.878
##  3rd Qu.:6.000
##  Max.   :9.000
```

Results: There are two data sets. One for red wine, and one for white wine. There are 1,599 observations in the red wine data set, and 4,898 observations in the white wine data set. Both data sets have 12 variables, one dependent variable: "quality" which is an integer, and 11 independent or predictor variables: fixed.acidity, volatile.acidity, citric.acid, residual.sugar, chlorides, free.sulfur.dioxide, total.sulfur.dioxide, density, pH, sulphates, and alcohol, all of which are numerical values.

total.sulfur.dioxide in the red table has a max value of 289, there could be an outlier here. free.sulfur.dioxide in the white table has a max value of 289, again this could be an outlier.

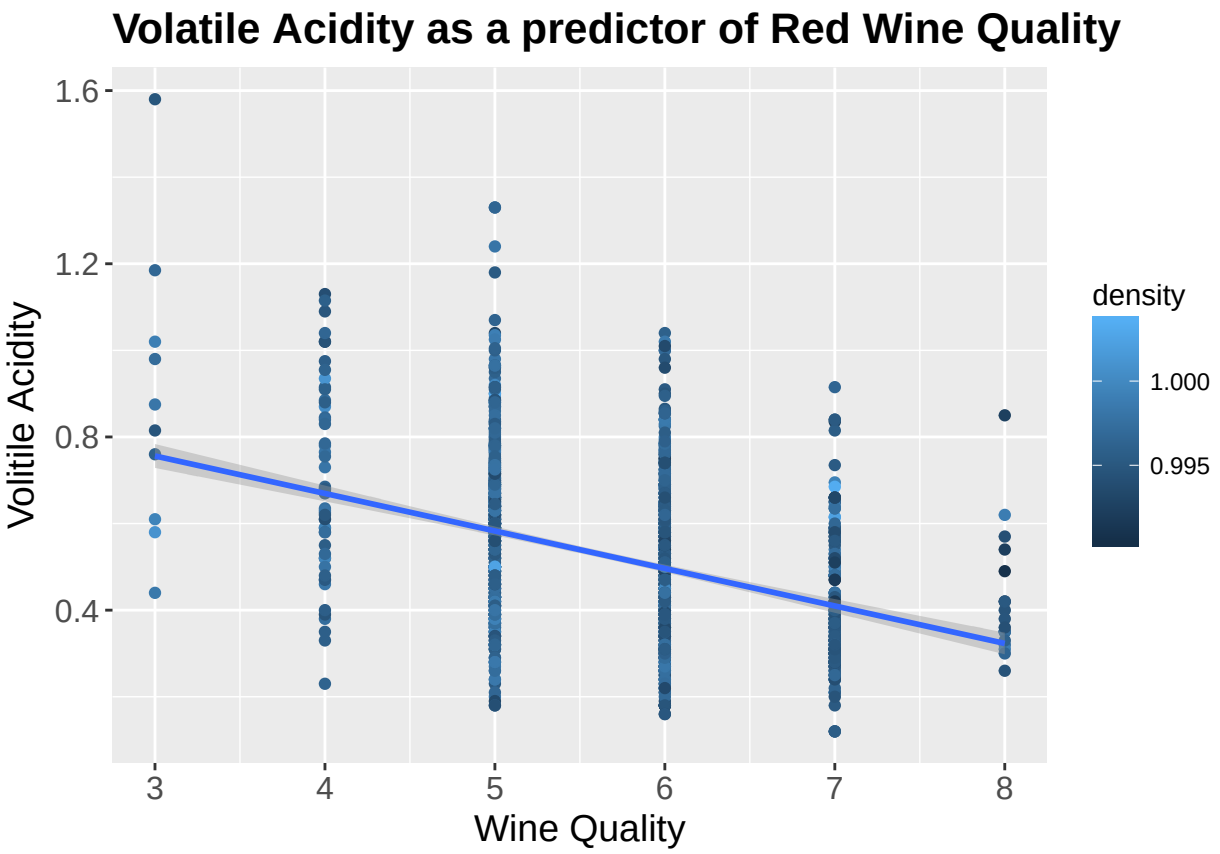
---

2. For each of the wine types (red and white), plot the outcome – `quality` – against some of the predictors. You should have six total plots.

Comment on the structure of your plots. Are there any relations between the variables that you find worth highlighting?

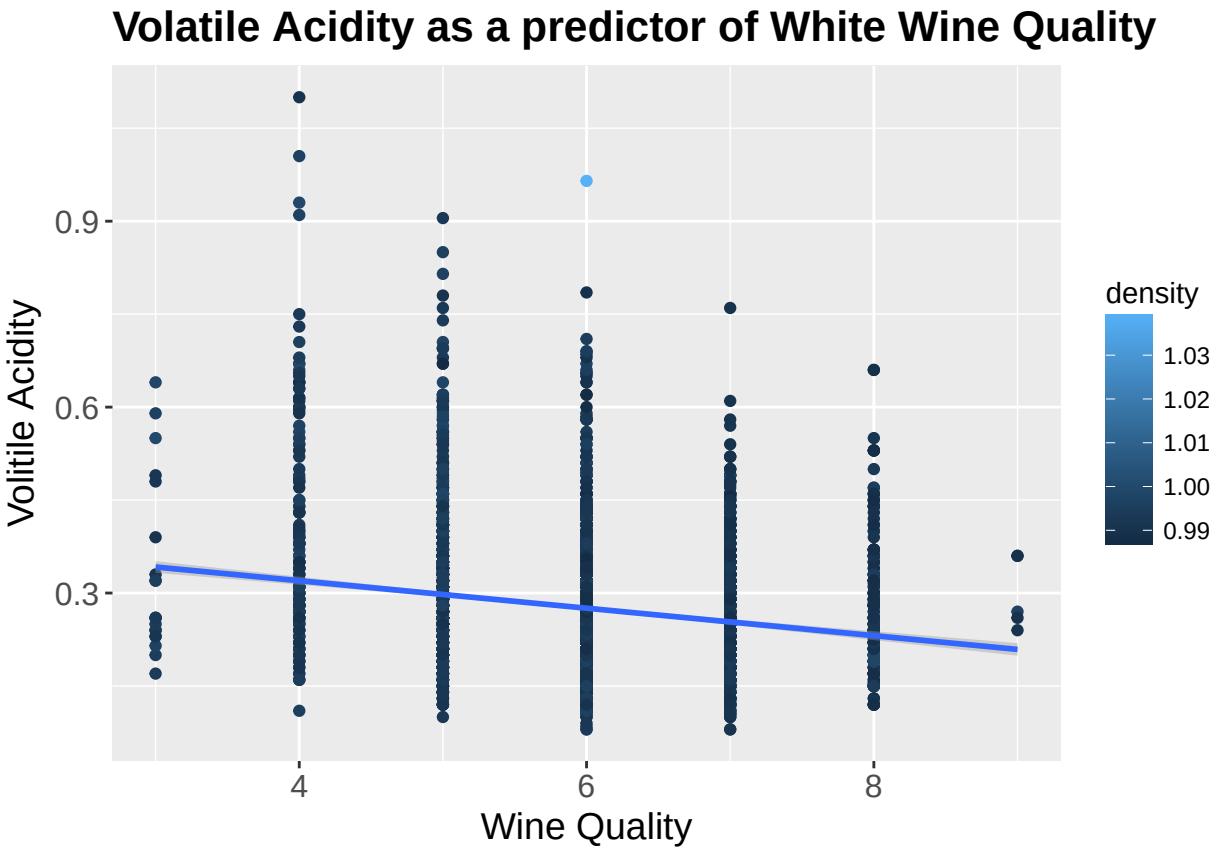
a) Use volatile acidity as the predictor and density as the color.

```
ggplot(red, aes(x=quality, y=volatile.acidity, colour=density)) +
  geom_point() + stat_smooth(method=lm) +
  ggtitle ("Volatile Acidity as a predictor of Red Wine Quality") +
  xlab("Wine Quality") +
  ylab("Volitile Acidity") +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),
        plot.title=element_text(size=16, face="bold"))
```



Results: It appears that as volatile acidity increases, red wine quality decreases. Density seems to be distributed evenly as compared to volitile acidity.

```
ggplot(white, aes(x=quality, y=volatile.acidity, colour=density)) +
  geom_point() + stat_smooth(method=lm) +
  ggtitle ("Volatile Acidity as a predictor of White Wine Quality") +
  xlab("Wine Quality") +
  ylab("Volitile Acidity") +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),
        plot.title=element_text(size=16, face="bold"))
```



Results: It appears that as volatile acidity increases, white wine quality decreases. This affect seems to be more dramatic in red wines versus white wines. It also appears, based on the color of the dots, that white wine may have lower density than red wine.

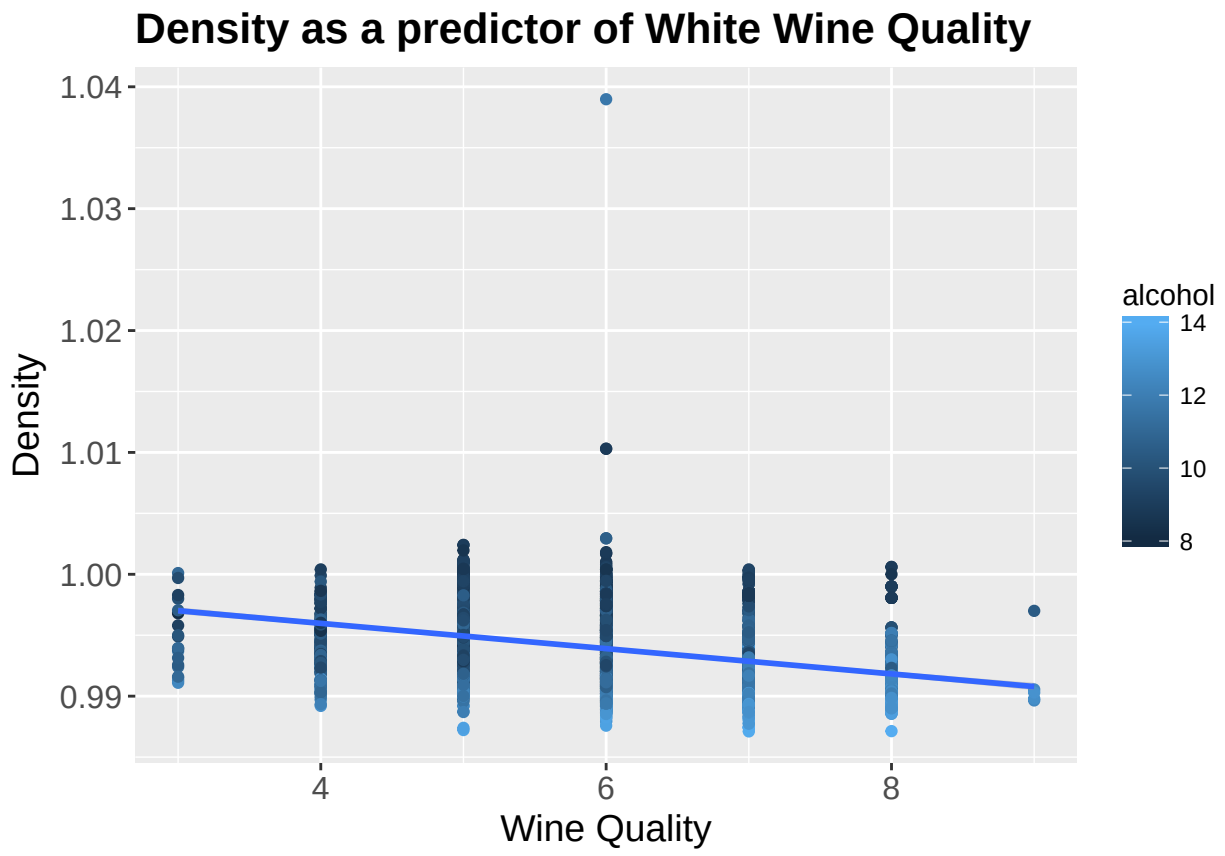
---

b) Use density as the predictor and alcohol as the color.

```
ggplot(red, aes(x=quality, y=density, colour=alcohol)) +
  geom_point() + stat_smooth(method=lm) +
  ggtitle ("Density as a predictor of Red Wine Quality") +
  xlab("Wine Quality") +
  ylab("Density") +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),
        plot.title=element_text(size=16, face="bold"))
```

Results: It appears that as density decreases, wine quality increases. I don't really see any identifiable patterns in the effects of alcohol on this chart.

```
ggplot(white, aes(x=quality, y=density, colour=alcohol)) +
  geom_point() + stat_smooth(method=lm) +
  ggtitle ("Density as a predictor of White Wine Quality") +
  xlab("Wine Quality") +
  ylab("Density") +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),
        plot.title=element_text(size=16, face="bold"))
```

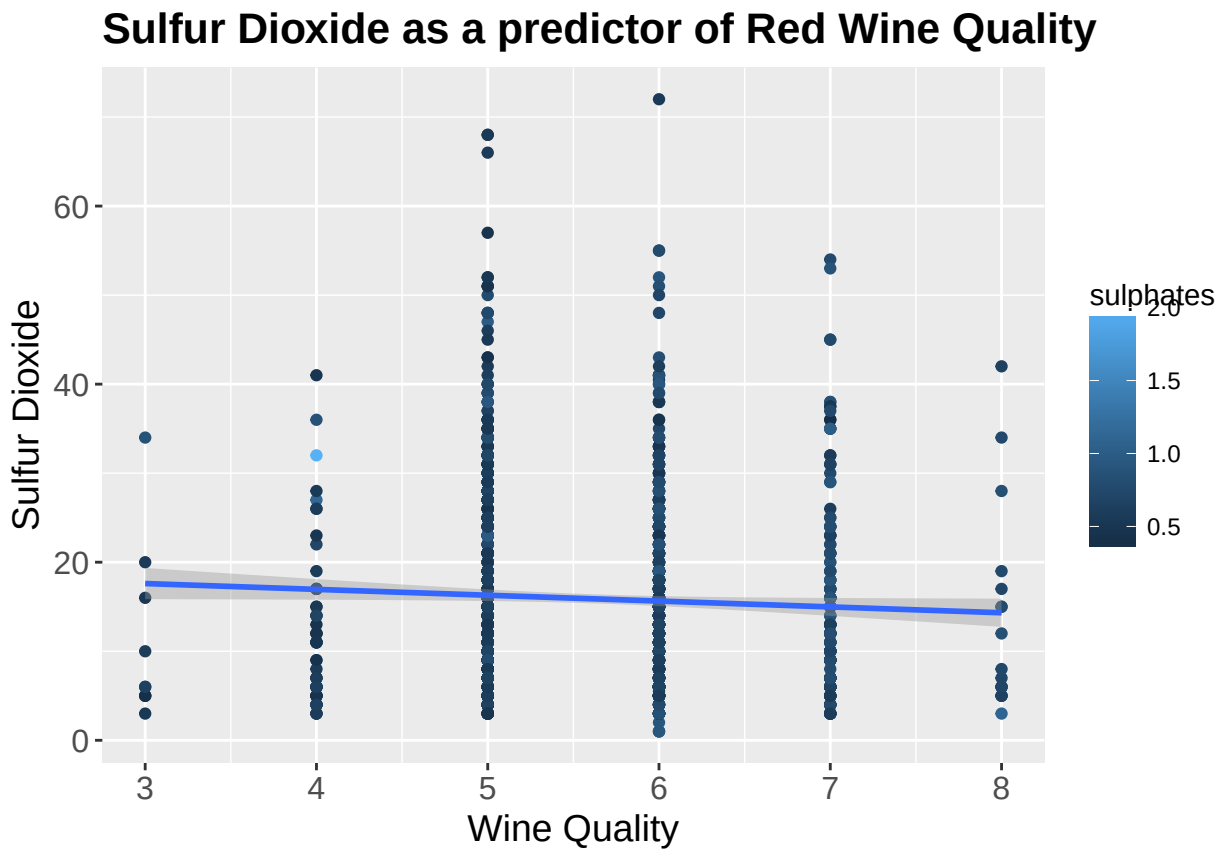


Results: It appears that as density increases, wine quality decrease. White wine has a couple of outliers, one with density at almost 1.04, and another around 1.01, that are skewing the scale of the white wine chart. It might be helpful to find and remove this outlier to see how the remaining data responds.

---

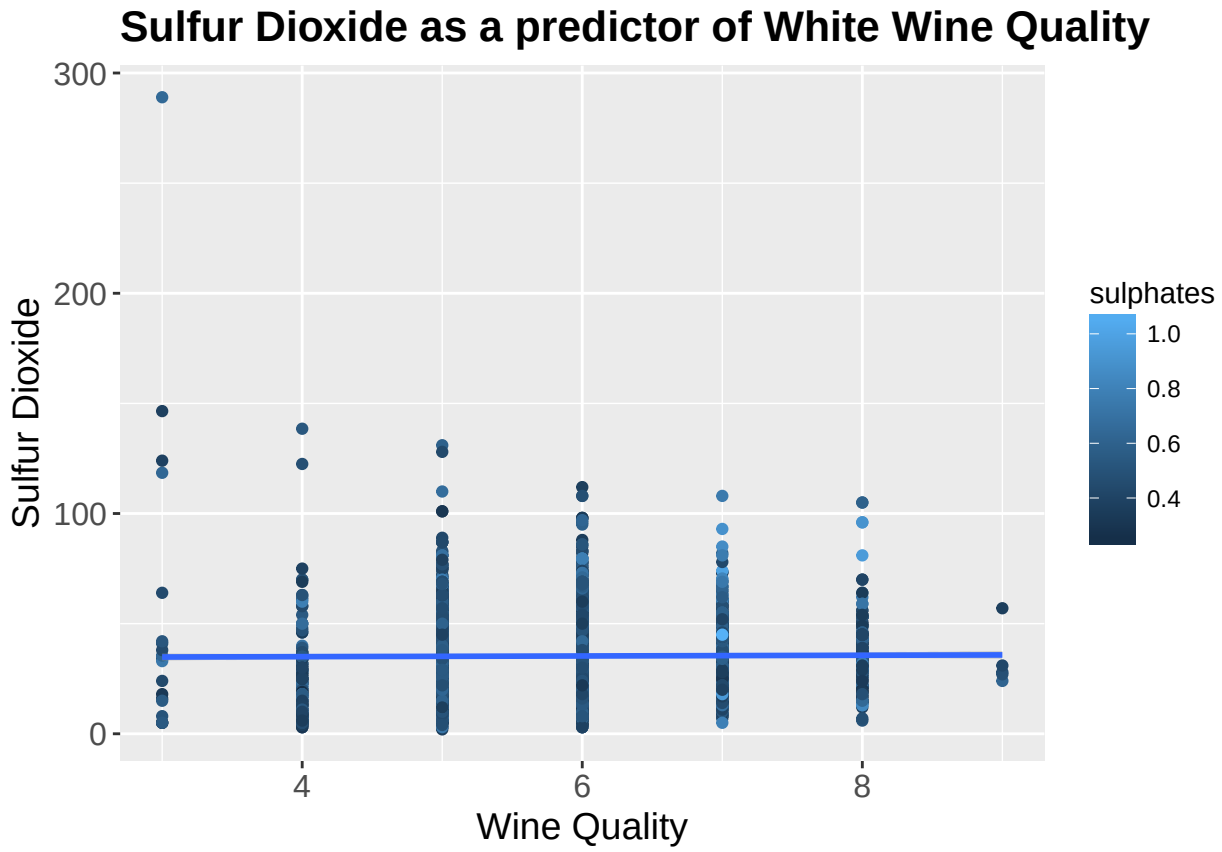
c) Use total sulfur dioxide as the predictor and sulphates as the color.

```
ggplot(red, aes(x=quality, y=free.sulfur.dioxide, colour=sulphates)) +
  geom_point() + stat_smooth(method=lm) +
  ggtitle ("Sulfur Dioxide as a predictor of Red Wine Quality") +
  xlab("Wine Quality") +
  ylab("Sulfur Dioxide") +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),
        plot.title=element_text(size=16, face="bold"))
```



Results: It appears that sulfur dioxide has a moderate impact on red wine quality. I can't really disern any affects of sulphates from this chart.

```
ggplot(white, aes(x=quality, y=free.sulfur.dioxide, colour=sulphates)) +
  geom_point() + stat_smooth(method=lm) +
  ggtitle ("Sulfur Dioxide as a predictor of White Wine Quality") +
  xlab("Wine Quality") +
  ylab("Sulfur Dioxide") +
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),
        plot.title=element_text(size=16, face="bold"))
```



Results: It does not appear that sulfur dioxide has an affect on white wine quality. Again, it appears that white wine has an outlier point that is skewing the chart scale. If the outliers are removed, and the chart re-generated, we may see a larger affect in the white wine chart.

---

3. Now that you have explored the data, perform two linear regressions using `lm()` on quality for both whites and reds separately. Include all variables as predictors, listing them out in sequence for each command statement. Which variable appears to have the most statistically significant outcome?

```
## Linear regression for red wine, using quality as the dependent variable. This model contains all 11 p

red.model <- lm(quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar + chlorides +

## Print the summary of the red.model.
## red.model
summary(red.model)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol - 1, data = red)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.66872 -0.36621 -0.04653  0.45604  2.04187
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## fixed.acidity         0.0041937  0.0164513   0.255  0.79882
## volatile.acidity     -1.0997431  0.1200969  -9.157  < 2e-16 ***
## citric.acid          -0.1841460  0.1471717  -1.251  0.21103
## residual.sugar        0.0070712  0.0120512   0.587  0.55745
## chlorides            -1.9114188  0.4177542  -4.575 5.12e-06 ***
## free.sulfur.dioxide   0.0045478  0.0021639   2.102  0.03574 *
## total.sulfur.dioxide -0.0033186  0.0007269  -4.565 5.37e-06 ***
## density               4.5291462  0.6253297   7.243 6.82e-13 ***
## pH                   -0.5228983  0.1599968  -3.268  0.00111 **
## sulphates             0.8870761  0.1107998   8.006 2.27e-15 ***
## alcohol               0.2970228  0.0172513  17.217  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.648 on 1588 degrees of freedom
## Multiple R-squared:  0.9871, Adjusted R-squared:  0.987
## F-statistic: 1.108e+04 on 11 and 1588 DF,  p-value: < 2.2e-16
```

Results: For red wine, `alcohol` has the most statistically significant outcome with the largest absolute t-value at 17.217, $p < 2e\text{-}16$.

```
## Linear regression for white wine, using quality as the dependent variable.  This model contains all
white.model <- lm(quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar + chlorides

## Print the summary of the white.model.
## white.model
summary(white.model)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol - 1, data = white)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9144 -0.4958 -0.0333  0.4675  3.1762
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## fixed.acidity        -0.0505906  0.0150754  -3.356 0.000797 ***
## volatile.acidity     -1.9585102  0.1138903 -17.196  < 2e-16 ***
## citric.acid          -0.0293492  0.0961648  -0.305 0.760229
## residual.sugar        0.0249884  0.0025917   9.642  < 2e-16 ***
## chlorides            -0.9425824  0.5430204  -1.736 0.082660 .
## free.sulfur.dioxide   0.0047908  0.0008390   5.710 1.20e-08 ***
## total.sulfur.dioxide -0.0008776  0.0003731  -2.352 0.018699 *
```

```
## density               2.0420461  0.3532997   5.780 7.94e-09 ***
## pH                    0.1683951  0.0835957   2.014 0.044022 *
## sulphates             0.4164536  0.0973279   4.279 1.91e-05 ***
## alcohol               0.3656334  0.0111203  32.880  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7562 on 4887 degrees of freedom
## Multiple R-squared:  0.9839, Adjusted R-squared:  0.9838
## F-statistic: 2.707e+04 on 11 and 4887 DF,  p-value: < 2.2e-16
```

Results: For white wine, `alcohol` has the most statistically significant outcome with the largest absolute t-value at 32.880, p < 2e-16.

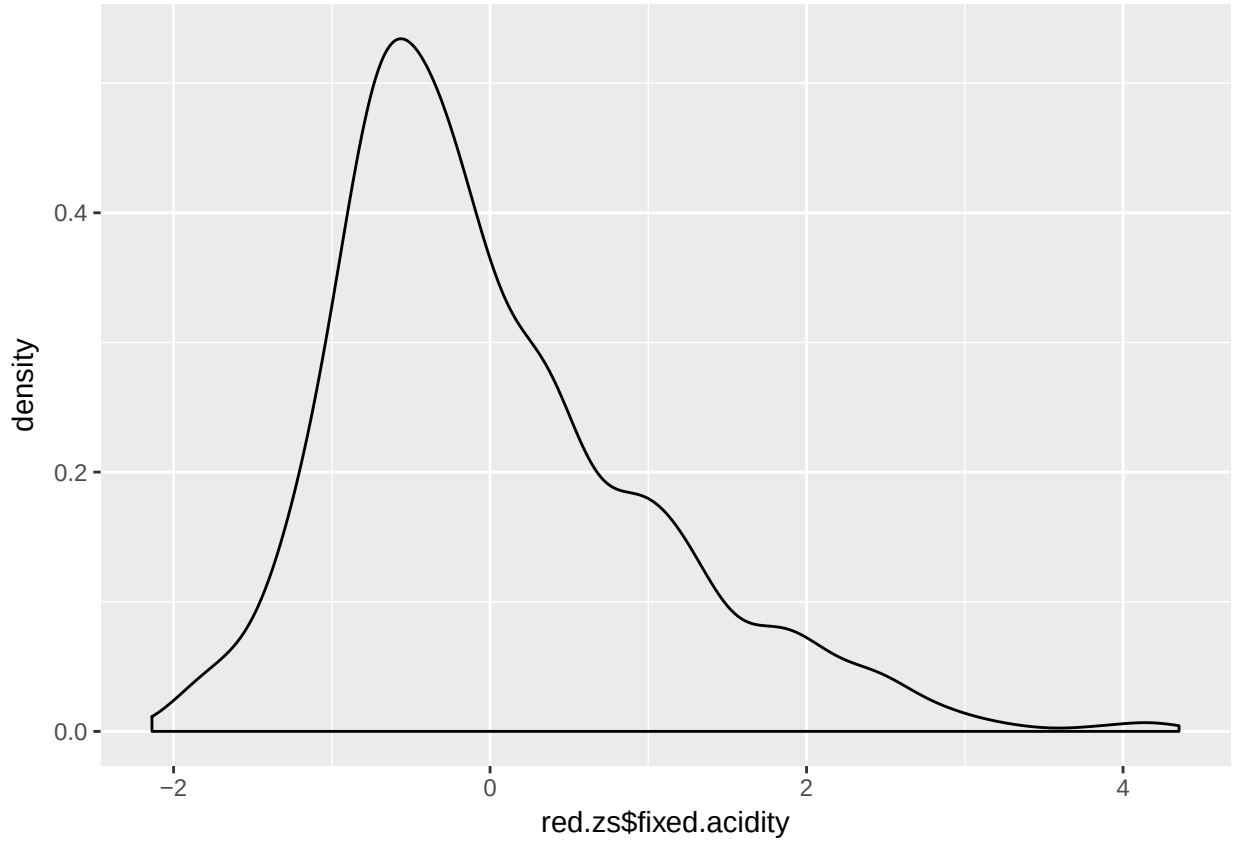
---

4. Create two new data frames for `red` and `white`. Use `mutate()` to standardize the variables in question: subtract their mean and divide by their standard deviation, in that order. Plot two variables from each to confirm that they appear to have the correct distributions.

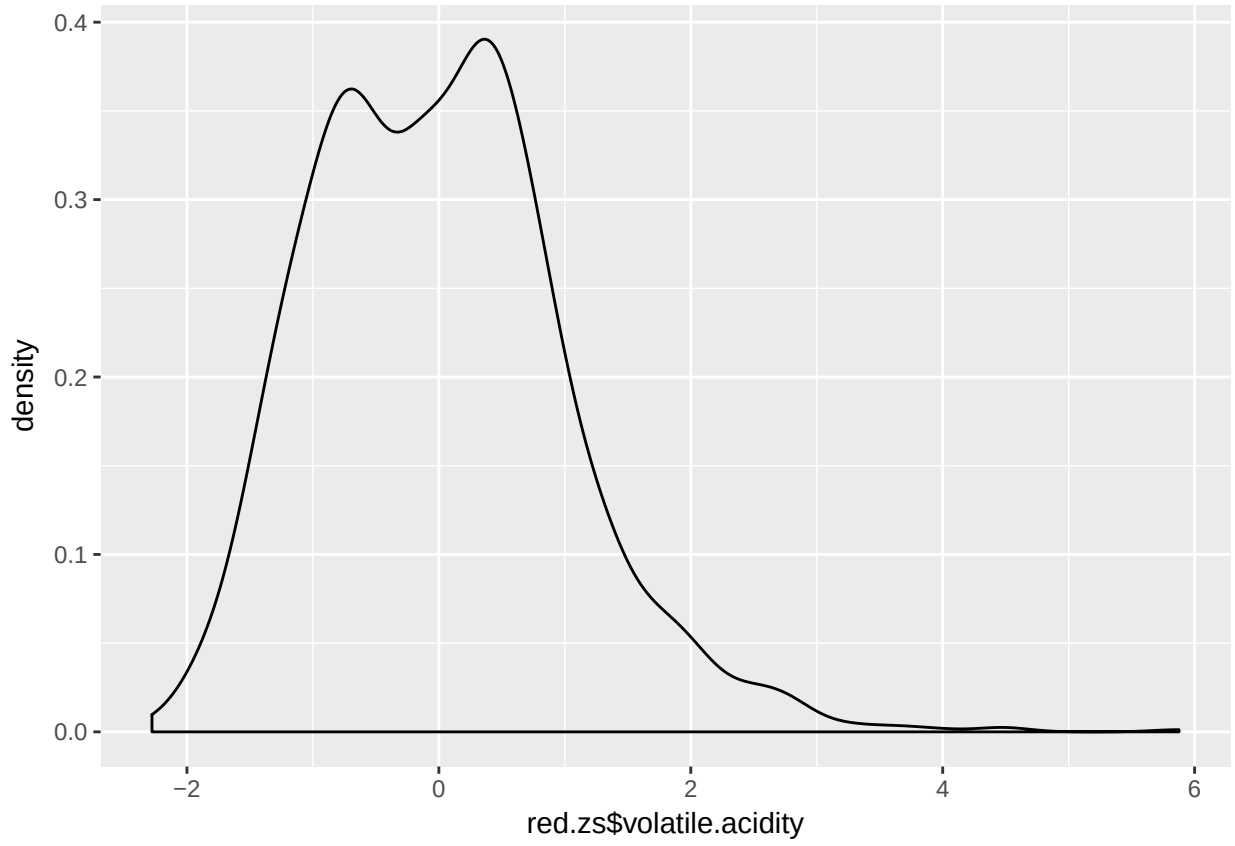
```
## Create a new dataframe with standardized "z" scores for all independent variables. Assign the "z" sc

red.zs <- red %>%
  mutate(fixed.acidity = (fixed.acidity - mean(fixed.acidity)) / sd(fixed.acidity)) %>%
  mutate(volatile.acidity = (volatile.acidity - mean(volatile.acidity)) / sd(volatile.acidity)) %>%
  mutate(citric.acid = (citric.acid - mean(citric.acid)) / sd(citric.acid)) %>%
  mutate(residual.sugar = (residual.sugar - mean(residual.sugar)) / sd(residual.sugar)) %>%
  mutate(chlorides = (chlorides - mean(chlorides)) / sd(chlorides)) %>%
  mutate(free.sulfur.dioxide = (free.sulfur.dioxide - mean(free.sulfur.dioxide)) / sd(free.sulfur.dioxi
  mutate(total.sulfur.dioxide = (total.sulfur.dioxide - mean(total.sulfur.dioxide)) / sd(total.sulfur.di
  mutate(density = (density - mean(density)) / sd(density)) %>%
  mutate(pH = (pH - mean(pH)) / sd(pH)) %>%
  mutate(sulphates = (sulphates - mean(sulphates)) / sd(sulphates)) %>%
  mutate(alcohol = (alcohol - mean(alcohol)) / sd(alcohol)) %>%
  mutate(quality = (quality - mean(quality)) / sd(quality))

## Display the new data frame.

str(red.zs)
```

```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed.acidity       : num  -0.528 -0.298 -0.298 1.654 -0.528 ...
##  $ volatile.acidity    : num  0.962 1.967 1.297 -1.384 0.962 ...
##  $ citric.acid         : num  -1.39 -1.39 -1.19 1.48 -1.39 ...
##  $ residual.sugar      : num  -0.4531 0.0434 -0.1694 -0.4531 -0.4531 ...
##  $ chlorides           : num  -0.2436 0.2238 0.0963 -0.2649 -0.2436 ...
##  $ free.sulfur.dioxide : num  -0.466 0.8724 -0.0836 0.1076 -0.466 ...
##  $ total.sulfur.dioxide: num  -0.379 0.624 0.229 0.411 -0.379 ...
##  $ density             : num  0.5581 0.0283 0.1342 0.6641 0.5581 ...
##  $ pH                  : num  1.288 -0.72 -0.331 -0.979 1.288 ...
##  $ sulphates           : num  -0.579 0.1289 -0.0481 -0.461 -0.579 ...
##  $ alcohol             : num  -0.96 -0.585 -0.585 -0.585 -0.96 ...
##  $ quality             : num  -0.788 -0.788 -0.788 0.451 -0.788 ...
```

```
qplot(red.zs$fixed.acidity, geom="density")
```




```
qplot(red.zs$volatile.acidity, geom="density")
```

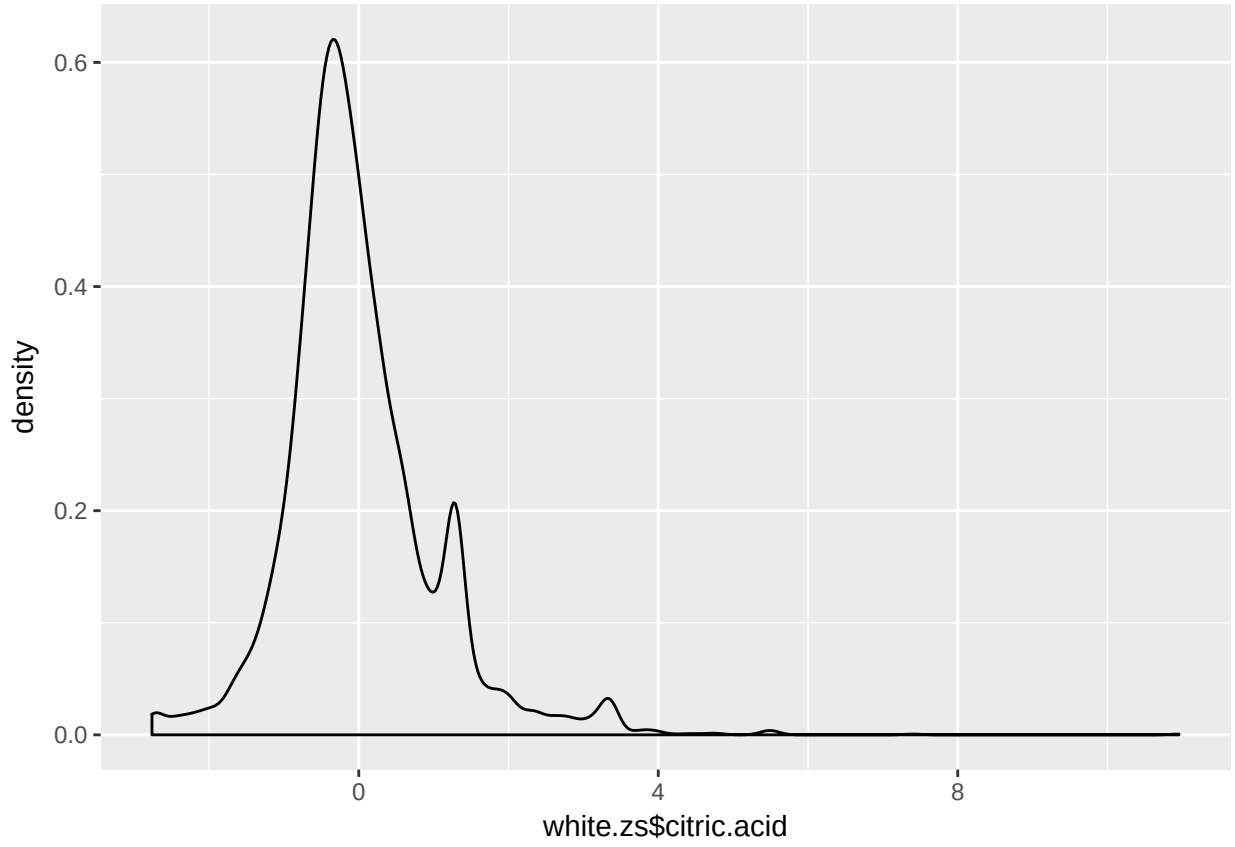


Results: We now have a new data frame, red.zs, which contains the standardized z scores for each predictor. The resulting z scores for `fixed.acidity` and `volatile.acidity` produce plots with distributions clustered around "0", verifying the z scores appear to be correctly calculated.

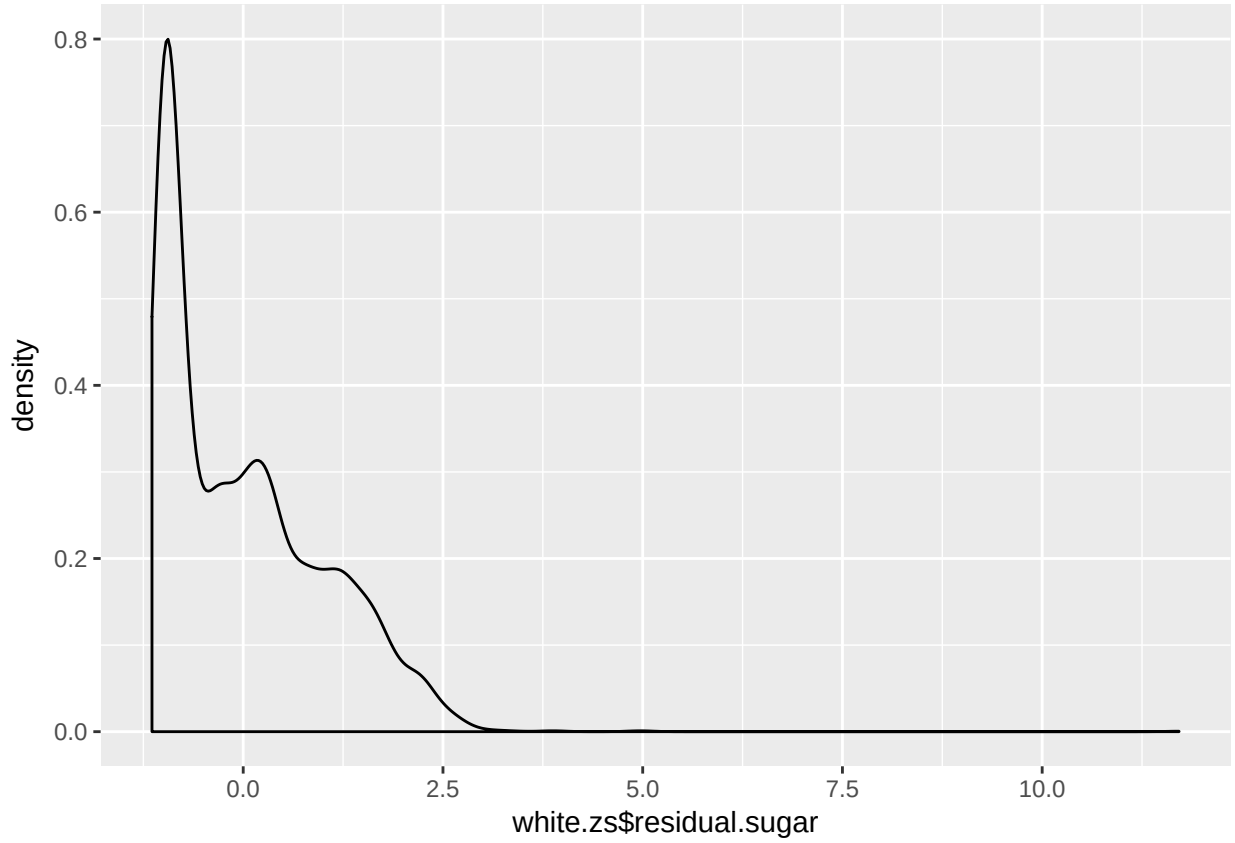
```
## Create the standardized "z" scores for white wine.

white.zs <- white %>%
  mutate(fixed.acidity = (fixed.acidity - mean(fixed.acidity)) / sd(fixed.acidity)) %>%
  mutate(volatile.acidity = (volatile.acidity - mean(volatile.acidity)) / sd(volatile.acidity)) %>%
  mutate(citric.acid = (citric.acid - mean(citric.acid)) / sd(citric.acid)) %>%
  mutate(residual.sugar = (residual.sugar - mean(residual.sugar)) / sd(residual.sugar)) %>%
  mutate(chlorides = (chlorides - mean(chlorides))/sd(chlorides)) %>%
  mutate(free.sulfur.dioxide = (free.sulfur.dioxide - mean(free.sulfur.dioxide))/sd(free.sulfur.dioxide)
  mutate(total.sulfur.dioxide = (total.sulfur.dioxide - mean(total.sulfur.dioxide))/sd(total.sulfur.dio
  mutate(density = (density - mean(density))/sd(density)) %>%
  mutate(pH = (pH - mean(pH))/sd(pH)) %>%
  mutate(sulphates = (sulphates - mean(sulphates))/sd(sulphates)) %>%
  mutate(alcohol = (alcohol - mean(alcohol))/sd(alcohol)) %>%
  mutate(quality = (quality - mean(quality)) / sd(quality))

## Display the data frame.

str(white.zs)
```

```
## 'data.frame':    4898 obs. of  12 variables:
##  $ fixed.acidity       : num  0.172 -0.657 1.476 0.409 0.409 ...
##  $ volatile.acidity    : num  -0.0818 0.2159 0.0175 -0.4786 -0.4786 ...
##  $ citric.acid         : num  0.213 0.048 0.544 -0.117 -0.117 ...
##  $ residual.sugar      : num  2.821 -0.945 0.1 0.416 0.416 ...
##  $ chlorides           : num  -0.0354 0.1477 0.1935 0.5597 0.5597 ...
##  $ free.sulfur.dioxide : num  0.57 -1.253 -0.312 0.687 0.687 ...
##  $ total.sulfur.dioxide: num  0.744 -0.15 -0.973 1.121 1.121 ...
##  $ density             : num  2.33127 -0.00915 0.35863 0.5258 0.5258 ...
##  $ pH                  : num  -1.2468 0.74 0.4751 0.0115 0.0115 ...
##  $ sulphates           : num  -0.34915 0.00134 -0.43677 -0.78726 -0.78726 ...
##  $ alcohol             : num  -1.393 -0.824 -0.337 -0.499 -0.499 ...
##  $ quality             : num  0.138 0.138 0.138 0.138 0.138 ...
```

```
qplot(white.zs$citric.acid, geom="density")
```




```
qplot(white.zs$residual.sugar, geom="density")
```



Results: We now have a new data frame, white.zs, which contains the standardized z scores for each predictor. The resulting z scores for `density` and `residual.sugar` produce plots with distributions clustered around "0", verifying the z scores appear to be correctly calculated.

---

5. Repeat Question 3, but with these standardized variables for predictors instead. Which variables have the greatest effect size in each regression?

```
## Linear regression for red wine based on z scores, using quality as the dependent variable.  This mode

red.zs.model <- lm(quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar + chlorides

## Print summary of red.zs.model.
## red.zs.model
summary(red.zs.model)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol - 1, data = red.zs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3299 -0.4539 -0.0582  0.5597  2.5075
```

```
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## fixed.acidity          0.05388    0.05593   0.963   0.3355
## volatile.acidity      -0.24026    0.02684  -8.951  < 2e-16 ***
## citric.acid           -0.04404    0.03549  -1.241   0.2148
## residual.sugar         0.02851    0.02618   1.089   0.2763
## chlorides             -0.10923    0.02443  -4.471 8.32e-06 ***
## free.sulfur.dioxide    0.05649    0.02812   2.009   0.0447 *
## total.sulfur.dioxide -0.13298    0.02967  -4.481 7.95e-06 ***
## density               -0.04179    0.05054  -0.827   0.4085
## pH                    -0.07908    0.03662  -2.160   0.0309 *
## sulphates              0.19234    0.02399   8.017 2.08e-15 ***
## alcohol                0.36447    0.03494  10.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8022 on 1588 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic:  81.4 on 11 and 1588 DF,  p-value: < 2.2e-16
```

```
## Linear regression for white wine based on z scores, using quality as the dependent variable.  This m

white.zs.model <- lm(quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar + chlorid

## Print summary of white.zs.model.
## white.zs.model
summary(white.zs.model)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol - 1, data = white.zs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3299 -0.5571 -0.0428  0.5235  3.5164
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## fixed.acidity          0.062430   0.019887   3.139   0.0017 **
## volatile.acidity      -0.212048   0.012949 -16.375  < 2e-16 ***
## citric.acid            0.003019   0.013085   0.231   0.8176
## residual.sugar         0.466653   0.043105  10.826  < 2e-16 ***
## chlorides             -0.006100   0.013481  -0.452   0.6509
## free.sulfur.dioxide    0.071681   0.016209   4.422 9.97e-06 ***
## total.sulfur.dioxide -0.013712   0.018140  -0.756   0.4497
## density               -0.507528   0.064410  -7.880 4.02e-15 ***
## pH                     0.117021   0.017965   6.514 8.07e-11 ***
## sulphates              0.081374   0.012935   6.291 3.43e-10 ***
## alcohol                0.268840   0.033653   7.989 1.69e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.8483 on 4887 degrees of freedom
## Multiple R-squared:  0.2819, Adjusted R-squared:  0.2803
## F-statistic: 174.4 on 11 and 4887 DF,  p-value: < 2.2e-16
```

Results: Alcohol has the largest effect size in the red.zs.model, and density has the largest effect size in the white.zs.model with 0.29433, and -0.449486 respectively.

---

6. Produce the design/predictors matrix for each of `red` and `white`. You can use the `select` function from dplyr (as long as you use `as.matrix()` next), the `model.matrix()` function, or some other method of your choosing. Verify the number of columns corresponds to the number of coefficients in your previous `lm()` outputs.

```
## Create quality matrix for red wine
red.quality.matrix <- model.matrix(~ 0 + fixed.acidity + volatile.acidity + citric.acid + residual.sugar
head(red.quality.matrix, 2)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
## 1    -0.5281944        0.9615758   -1.391037    -0.45307667 -0.2436305
## 2    -0.2984541        1.9668271   -1.391037     0.04340257  0.2238052
##   free.sulfur.dioxide total.sulfur.dioxide    density         pH
## 1          -0.4660467           -0.3790141 0.55809987  1.2882399
## 2           0.8723653            0.6241680 0.02825193 -0.7197081
##    sulphates    alcohol
## 1 -0.5790254 -0.9599458
## 2  0.1289101 -0.5845942
```

```
## Check to see if the number of columns in red.quality.matrix corresponds to the number of coefficients
length(red.zs.model$coefficients)
```

```
## [1] 11
```

```
dim(red.quality.matrix)[2]
```

```
## [1] 11
```

Results: The resulting red.quality.matrix has the same number of columns as the number of coefficients in the previous `lm()` output.

```
## Create quality matrix for white wine
white.quality.matrix <- model.matrix(~ 0 + fixed.acidity + volatile.acidity + citric.acid + residual.su
head(white.quality.matrix, 2)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar   chlorides
## 1     0.1720794      -0.08176155  0.21325843      2.8210611 -0.03535139
## 2    -0.6574340       0.21587359  0.04799622     -0.9446688  0.14773200
##   free.sulfur.dioxide total.sulfur.dioxide      density         pH
## 1           0.5698734            0.7444890  2.331273996 -1.2467940
## 2          -1.2528907           -0.1496693 -0.009153237  0.7399531
##     sulphates    alcohol
## 1 -0.34914861 -1.3930102
## 2  0.00134171 -0.8241915
```

```
## Check to see if the number of columns in white.quality.matrix corresponds to the number of coefficie
length(white.zs.model$coefficients)
```

```
## [1] 11
```

```
dim(white.quality.matrix)[2]
```

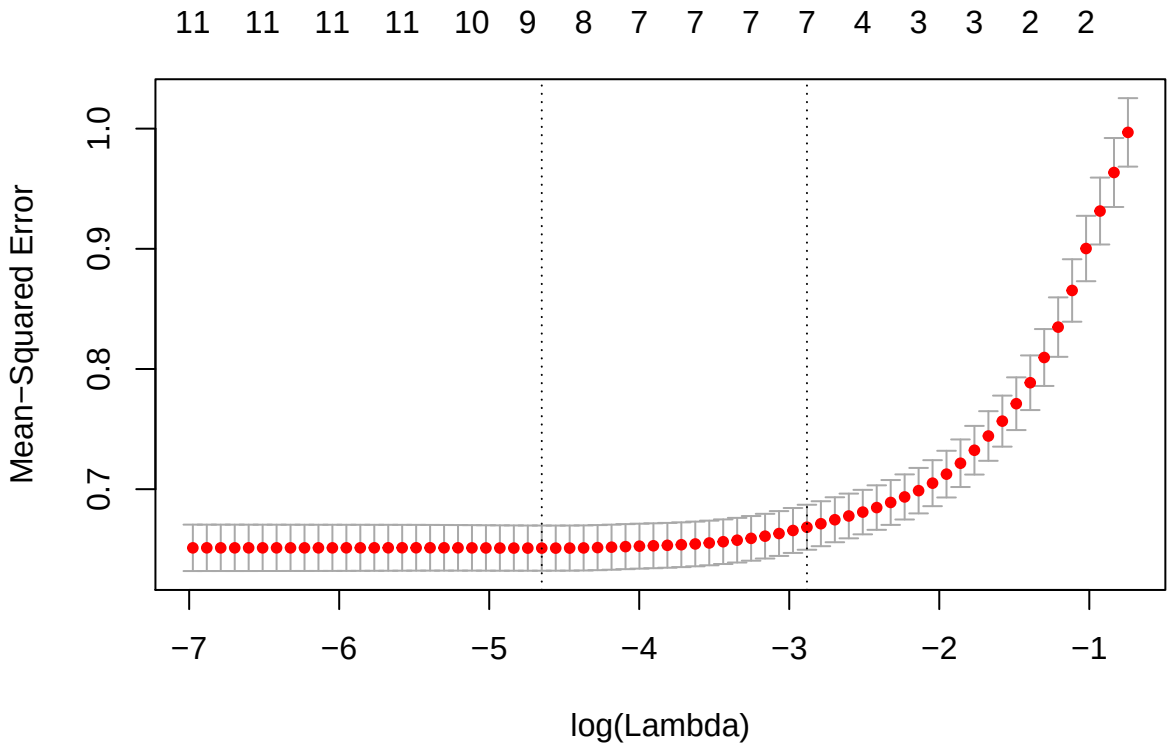
```
## [1] 11
```

Results: The resulting `white.quality.matrix` has the same number of columns (11) as the number of coefficients (11) in the previous `lm()` output.

---

7. Use `cv.glmnet()` with the Lasso (`alpha=1`, by default) to run the penalized linear model for quality as the outcome with all your predictors as previously done, for each of the two data frames. What values for `lambda` produce the smallest cross-validated error in each case? How much of a reduction is this in cross-validated error from the basic model you fit in Question 3 (corresponding to `lambda = 0`)?

```
##cv.glmnet() calculations for red wine
## First Validation Step
red.quality.cv = cv.glmnet (red.quality.matrix, red.zs$quality, alpha=1 )
plot(red.quality.cv)
```



Results: This plot shows the cross-validation mean squared error (MSE) as a function of log(lambda) curve (red dotted line), including the upper and lower standard deviation curves. The dotted lines represent lambda.min and lambda.min plus one standard error. As lambda gets smaller, the curve flattens out. The

numbers across the top of the plot indicate how many non-zero predictors are in the model at each level of lambda.

```
rpicked <- which (red.quality.cv$lambda == red.quality.cv$lambda.min)
rpicked
```

```
## [1] 43
```

```
## The `lambda` that produces the smallest cross-validated error is:
red.quality.cv$lambda.min
```

```
## [1] 0.009564299
```

```
## The cross validated error is:
red.quality.cv$cvm[rpicked]
```

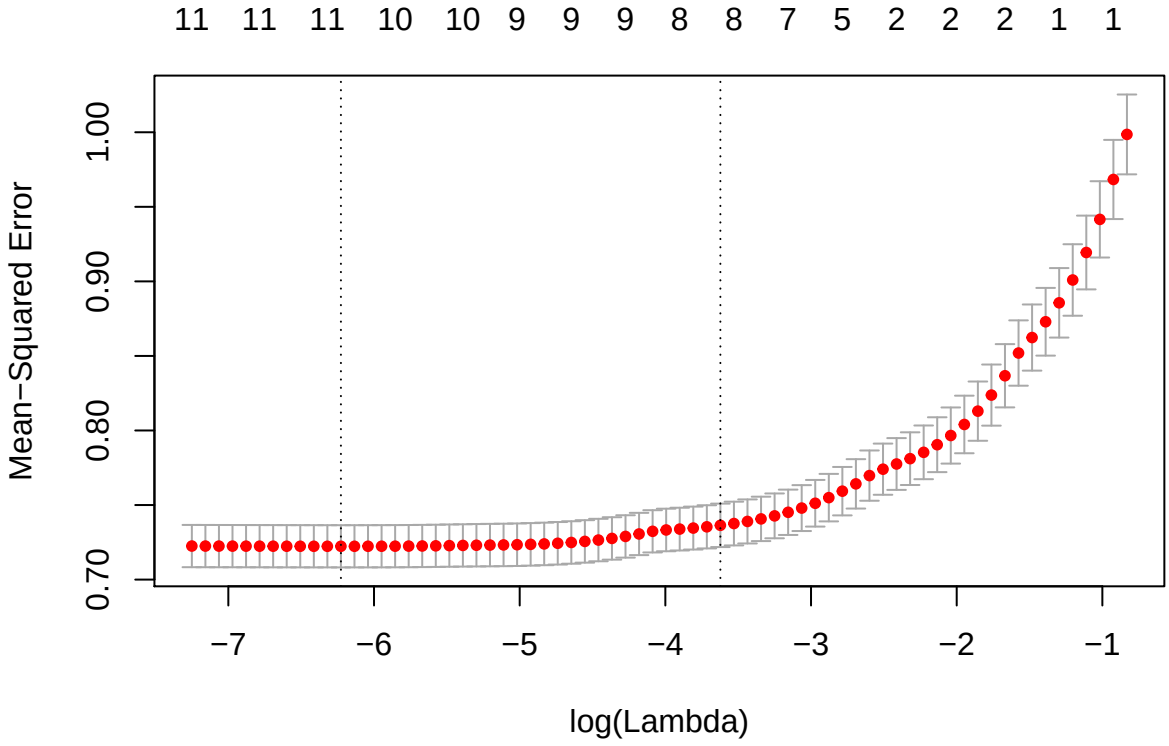
```
## [1] 0.6509618
```

```
## How much of a reduction is this in cross-validated error from the basic fit model in Question 3?
summary(red.zs.model)
```

```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol - 1, data = red.zs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3299 -0.4539 -0.0582  0.5597  2.5075
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## fixed.acidity          0.05388    0.05593   0.963   0.3355
## volatile.acidity      -0.24026    0.02684  -8.951  < 2e-16 ***
## citric.acid           -0.04404    0.03549  -1.241   0.2148
## residual.sugar         0.02851    0.02618   1.089   0.2763
## chlorides             -0.10923    0.02443  -4.471 8.32e-06 ***
## free.sulfur.dioxide    0.05649    0.02812   2.009   0.0447 *
## total.sulfur.dioxide -0.13298    0.02967  -4.481 7.95e-06 ***
## density               -0.04179    0.05054  -0.827   0.4085
## pH                    -0.07908    0.03662  -2.160   0.0309 *
## sulphates              0.19234    0.02399   8.017 2.08e-15 ***
## alcohol                0.36447    0.03494  10.432  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8022 on 1588 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic:  81.4 on 11 and 1588 DF,  p-value: < 2.2e-16
```

Results: The `lambda` that produces the smallest value is [43] = 0.0095643. The cross-validated error for red.quality.cv [43] is 0.6509618. This can be compared to the adjusted r-squared (0.3561) as listed above in the red.zs.model summary.

```
##cv.glmnet() calculations for white wine
## First Validation Step
white.quality.cv = cv.glmnet (white.quality.matrix, white.zs$quality, alpha=1)
plot(white.quality.cv)
```



Results: This plot shows the cross-validation mean squared error (MSE) as a function of log(lambda) curve (red dotted line), including the upper and lower standard deviation curves. The dotted lines represent lambda.min and lambda.min plus one standard error. As lambda gets smaller, the curve flattens out. The numbers across the top of the plot indicate how many non-zero predictors are in the model at each level of lambda.

```
## Find the minium value of lambda.
wpicked <- which (white.quality.cv$lambda == white.quality.cv$lambda.min)
wpicked
```

```
## [1] 59
```

```
##The `lambda` that produces the smallest value is:"
white.quality.cv$lambda.min
```

```
## [1] 0.001975077
```

```
##The cross validated error is:
white.quality.cv$cvm[wpicked]
```

```
## [1] 0.7223316
```

```
## How much of a reduction is this in cross-validated error from the basic fit model in Question 3?
summary(white.zs.model)
```

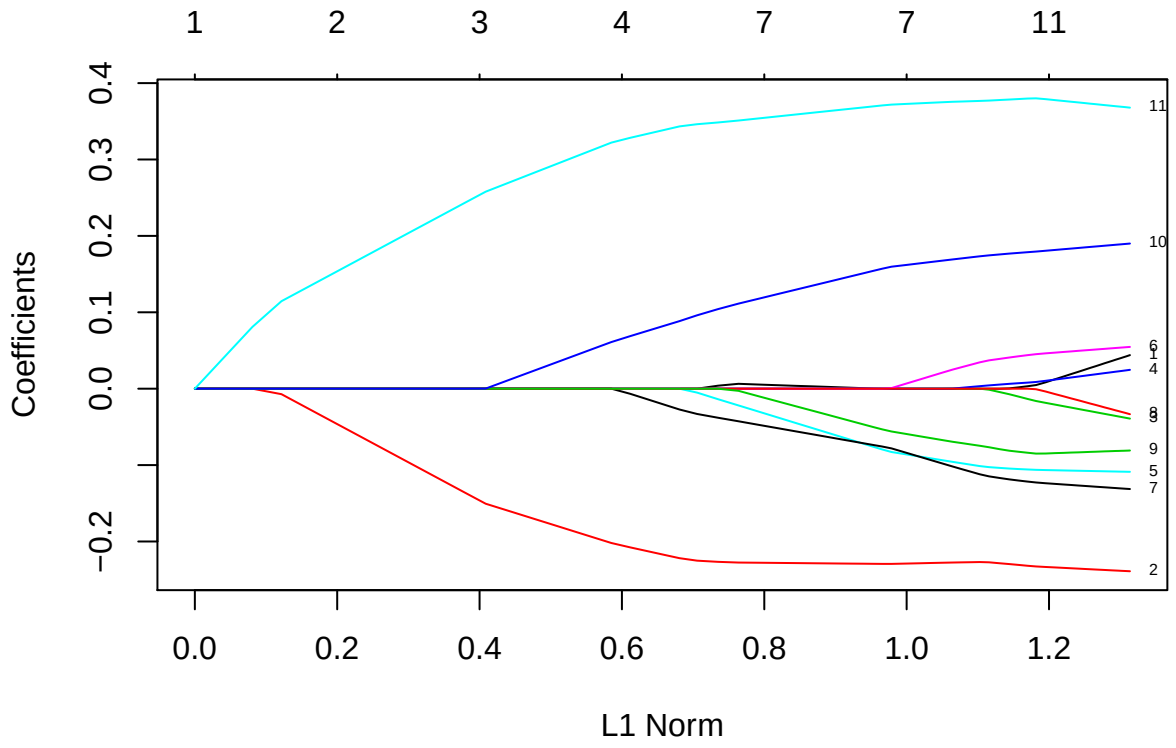
```
##
## Call:
## lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +
##     residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol - 1, data = white.zs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3299 -0.5571 -0.0428  0.5235  3.5164
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## fixed.acidity         0.062430   0.019887   3.139   0.0017 **
## volatile.acidity     -0.212048   0.012949 -16.375  < 2e-16 ***
## citric.acid           0.003019   0.013085   0.231   0.8176
## residual.sugar        0.466653   0.043105  10.826  < 2e-16 ***
## chlorides            -0.006100   0.013481  -0.452   0.6509
## free.sulfur.dioxide   0.071681   0.016209   4.422 9.97e-06 ***
## total.sulfur.dioxide -0.013712   0.018140  -0.756   0.4497
## density              -0.507528   0.064410  -7.880 4.02e-15 ***
## pH                    0.117021   0.017965   6.514 8.07e-11 ***
## sulphates             0.081374   0.012935   6.291 3.43e-10 ***
## alcohol               0.268840   0.033653   7.989 1.69e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8483 on 4887 degrees of freedom
## Multiple R-squared:  0.2819, Adjusted R-squared:  0.2803
## F-statistic: 174.4 on 11 and 4887 DF,  p-value: < 2.2e-16
```

Results: The `lambda` that produces the smallest value is [59] = 0.0095643. The cross-validated error for white.quality.cv [59] is 0.7223316. This can be compared to the adjusted r-squared (0.2803) as listed above in the red.zs.model summary.

---

8. Use `glmnet()` to fit the "shrinkage" model to each data set. Use the same `lambda` series as outputted in the previous steps for each model. Note the column in the `beta` matrix that corresponds to the ideal `lambda`. Do any of the estimates for `beta` in each model get shrunk all the way to zero?

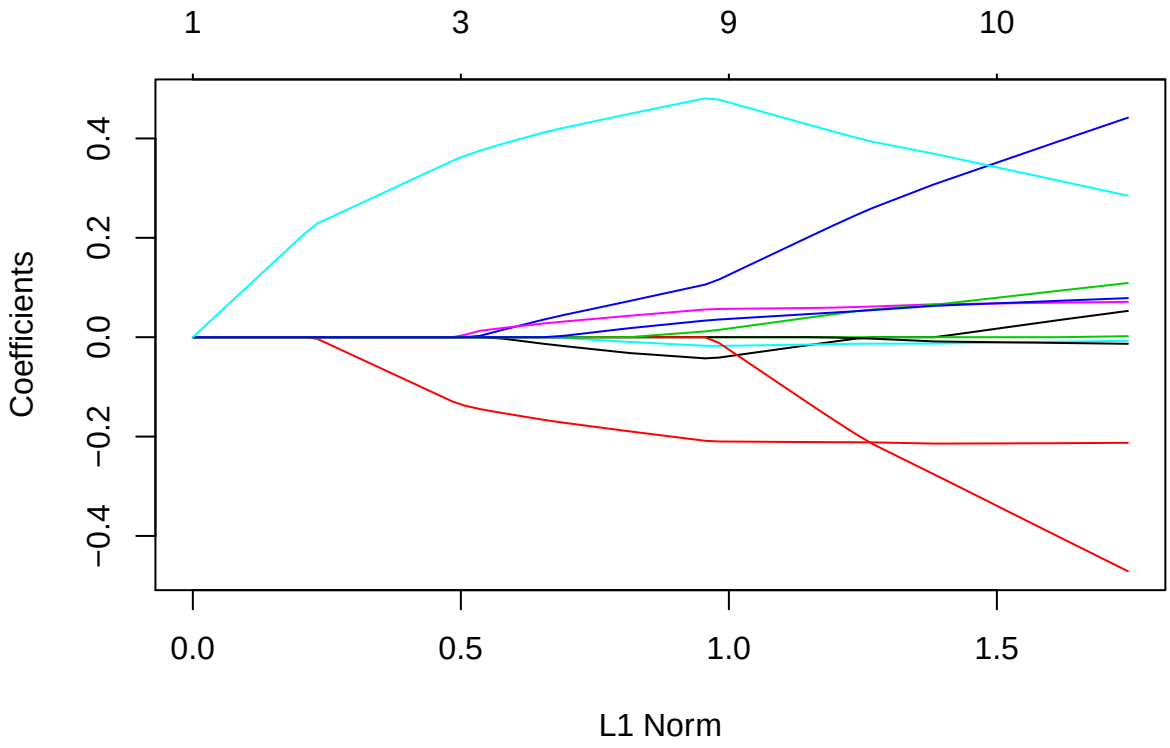
```
## Calculations for red wine.
red.quality = glmnet (red.quality.matrix, red.zs$quality, lambda = red.quality.cv$lambda)
plot(red.quality, label = TRUE)
```




```
red.quality$beta[,rpicked]
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##          0.000000000         -0.226916251          0.000000000
##       residual.sugar            chlorides  free.sulfur.dioxide
##          0.003292569         -0.101293802          0.034498974
## total.sulfur.dioxide              density                   pH
##         -0.112013206          0.000000000         -0.074691338
##            sulphates              alcohol
##          0.173278262          0.376657480
```

```
## Calculations for white wine.
white.quality = glmnet (white.quality.matrix, white.zs$quality, lambda = white.quality.cv$lambda)
plot(white.quality)
```




```
white.quality$beta[,wpicked]
```

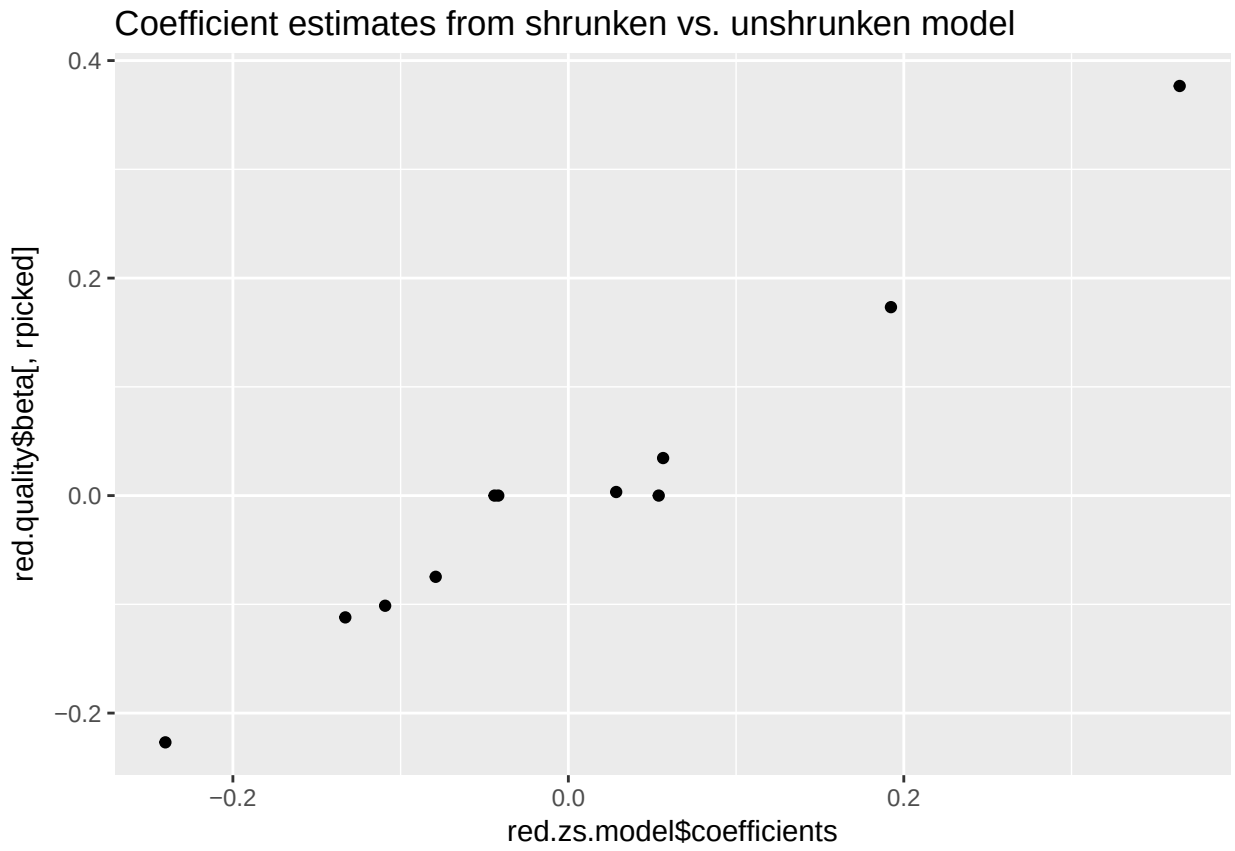
```
##        fixed.acidity     volatile.acidity          citric.acid
##         0.0387961499        -0.2131581415         0.0004668815
##       residual.sugar            chlorides  free.sulfur.dioxide
##         0.4055990886        -0.0085937113         0.0699023860
## total.sulfur.dioxide              density                   pH
##        -0.0119490148        -0.4182313193         0.0971227845
##            sulphates              alcohol
##         0.0745283799         0.3077858715
```

Results: In this step we calculated the model which corresponds to the minimum `lambda` identified in step 7. The resulting plot and betas are shown. The estimates of `red.quality$beta` for `fixed.acidity`, `citric.acid`, and `density` have shrunken to zero. The estimates of `white.quality$beta` for `citric.acid` have shrunken to zero.

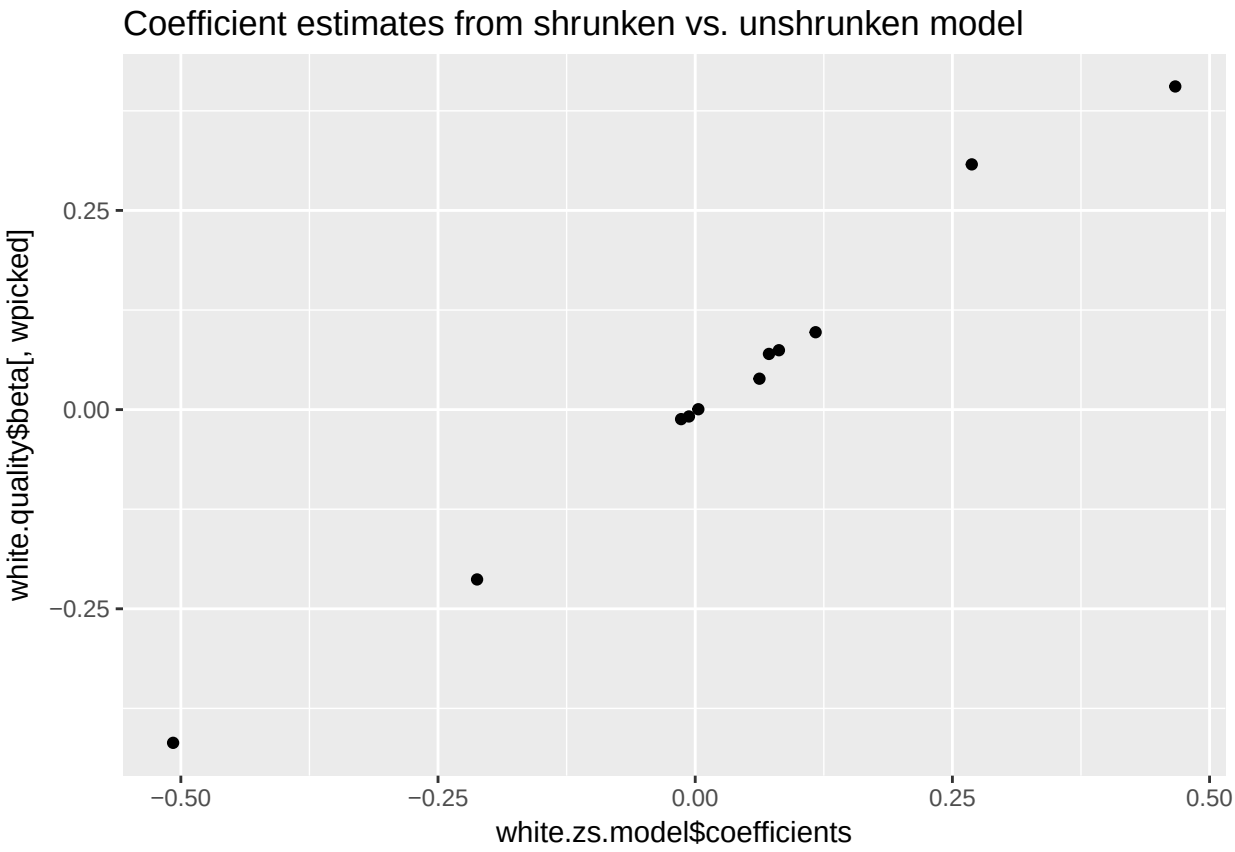
---

9. Plot the coefficient estimates from the unshrunken models (step 3) compared to the ideal shrunken models (step 8) to demonstrate whether this shrunken estimation produced a noticeably different response.

```
qplot (red.zs.model$coefficients, red.quality$beta[,rpicked], main="Coefficient estimates from shrunken
```




```
qplot (white.zs.model$coefficients, white.quality$beta[,wpicked], main="Coefficient estimates from shru
```



Results: In this step we ploted the original coefficient estimates from step 3 to the coefficient estimates from the shrunken model in step 8. There appear to be some differences in coefficients. If there were no differences, the plot would create a straight line.
