HW3_Q3

Mimis Jimenez

4/20/2020

Clustering and PCA

Load packages and data

```
library(remotes)
library(rlang)
library(LICORS)
library(foreach)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## <U+2713> ggplot2 3.3.0     <U+2713> purrr   0.3.3
## <U+2713> tibble  2.1.3     <U+2713> dplyr   0.8.4
## <U+2713> tidyr   1.0.0     <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1     <U+2713> forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x purrr::%@%()         masks rlang::%@%()
## x purrr::accumulate()  masks foreach::accumulate()
## x purrr::as_function() masks rlang::as_function()
## x dplyr::filter()      masks stats::filter()
## x purrr::flatten()     masks rlang::flatten()
## x purrr::flatten_chr() masks rlang::flatten_chr()
## x purrr::flatten_dbl() masks rlang::flatten_dbl()
## x purrr::flatten_int() masks rlang::flatten_int()
## x purrr::flatten_lgl() masks rlang::flatten_lgl()
## x purrr::flatten_raw() masks rlang::flatten_raw()
## x purrr::invoke()      masks rlang::invoke()
## x dplyr::lag()         masks stats::lag()
## x purrr::list_along()  masks rlang::list_along()
## x purrr::modify()      masks rlang::modify()
## x purrr::prepend()     masks rlang::prepend()
## x purrr::splice()      masks rlang::splice()
## x purrr::when()        masks foreach::when()
```

```
library(dplyr)
library(mosaic)
```

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

```
## Loading required package: ggformula
```

```
## Loading required package: ggstance
```

```
##
```

```
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh

##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following object is masked from 'package:purrr':
##
##     cross

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
wine<- read.csv("~/GitHub/SDS323_Spring2020/hw3/q3/wine.csv")
```

Clusters

```
#Center and scale the data
winex <- wine[,-(12:13)]
winex <- scale(winex, center = TRUE, scale = TRUE)

#Extract centers and scales
mu <- attr(winex,"scaled:center")
sigma <- attr(winex,"scaled:scale")

#Let's run K-means clustering with K = 5
clust_2 <- kmeans(winex, 2, nstart = 25)
clust_2$center
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
## 1     0.8286464        1.1678795  -0.3378091     -0.5903919  0.9216848
## 2    -0.2804833       -0.3953082   0.1143429      0.1998380 -0.3119753
##   free.sulfur.dioxide total.sulfur.dioxide    density         pH  sulphates
## 1          -0.8316090           -1.1872380  0.6815493  0.5673286  0.8430523
## 2           0.2814861            0.4018607 -0.2306934 -0.1920315 -0.2853595
##       alcohol
## 1 -0.07569241
## 2  0.02562065
```

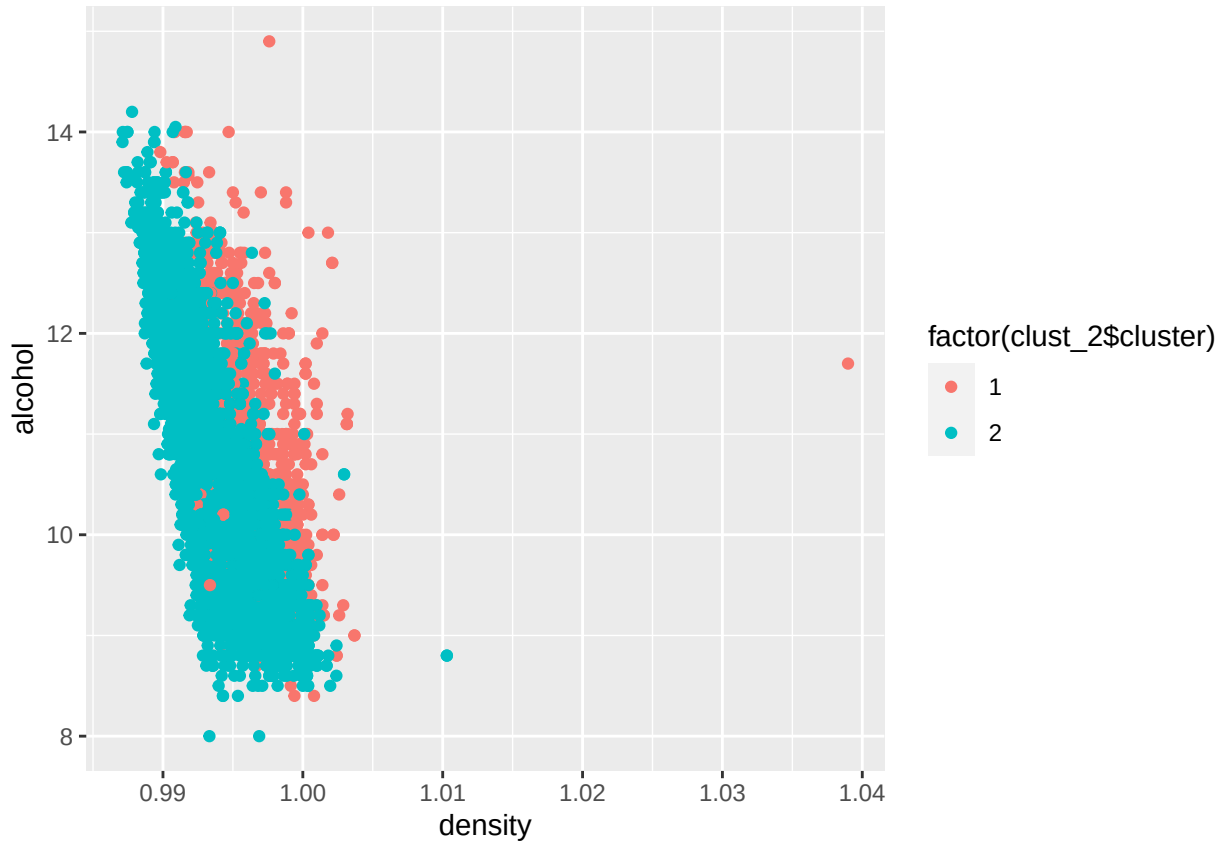
```
clust_2$center[1,]*sigma + mu
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##            8.2895922            0.5319416            0.2695435
##       residual.sugar            chlorides  free.sulfur.dioxide
##            2.6342666            0.0883238           15.7647596
## total.sulfur.dioxide              density                   pH
##           48.6396835            0.9967404            3.3097200
##            sulphates              alcohol
##            0.6567194           10.4015216
```

```
clust_2$center[2,]*sigma + mu
```

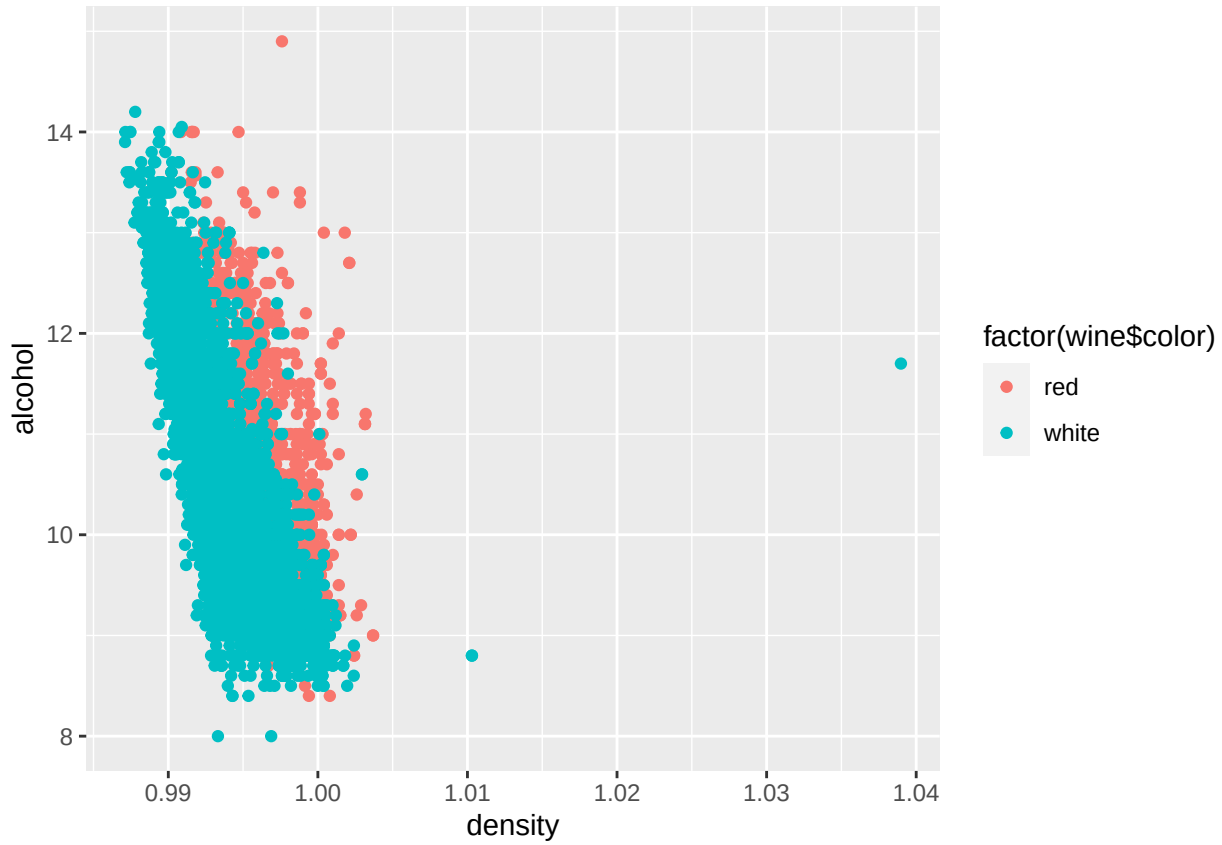
```
##        fixed.acidity     volatile.acidity          citric.acid
##          6.85167903           0.27458385           0.33524928
##       residual.sugar            chlorides  free.sulfur.dioxide
##          6.39402555           0.04510424          35.52152864
## total.sulfur.dioxide              density                   pH
##         138.45848785           0.99400486           3.18762464
##            sulphates              alcohol
##          0.48880511          10.52235888
```

```
#These two plots show that the two clusters did an okay job of separating the wines by color!
qplot(density, alcohol, data = wine, color = factor(clust_2$cluster))
```

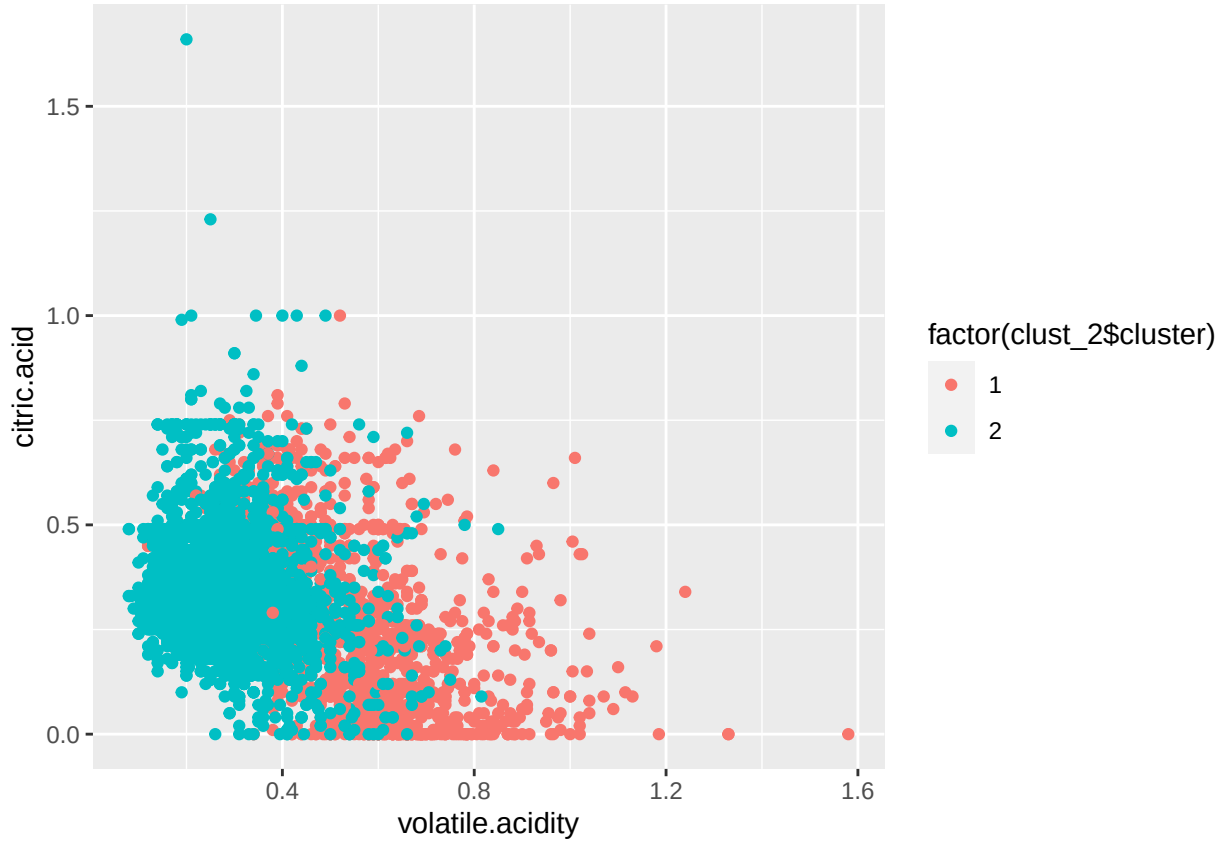



```
qplot(density, alcohol, data = wine, color = factor(wine$color))
```

```
## Warning: Use of `wine$color` is discouraged. Use `color` instead.
```

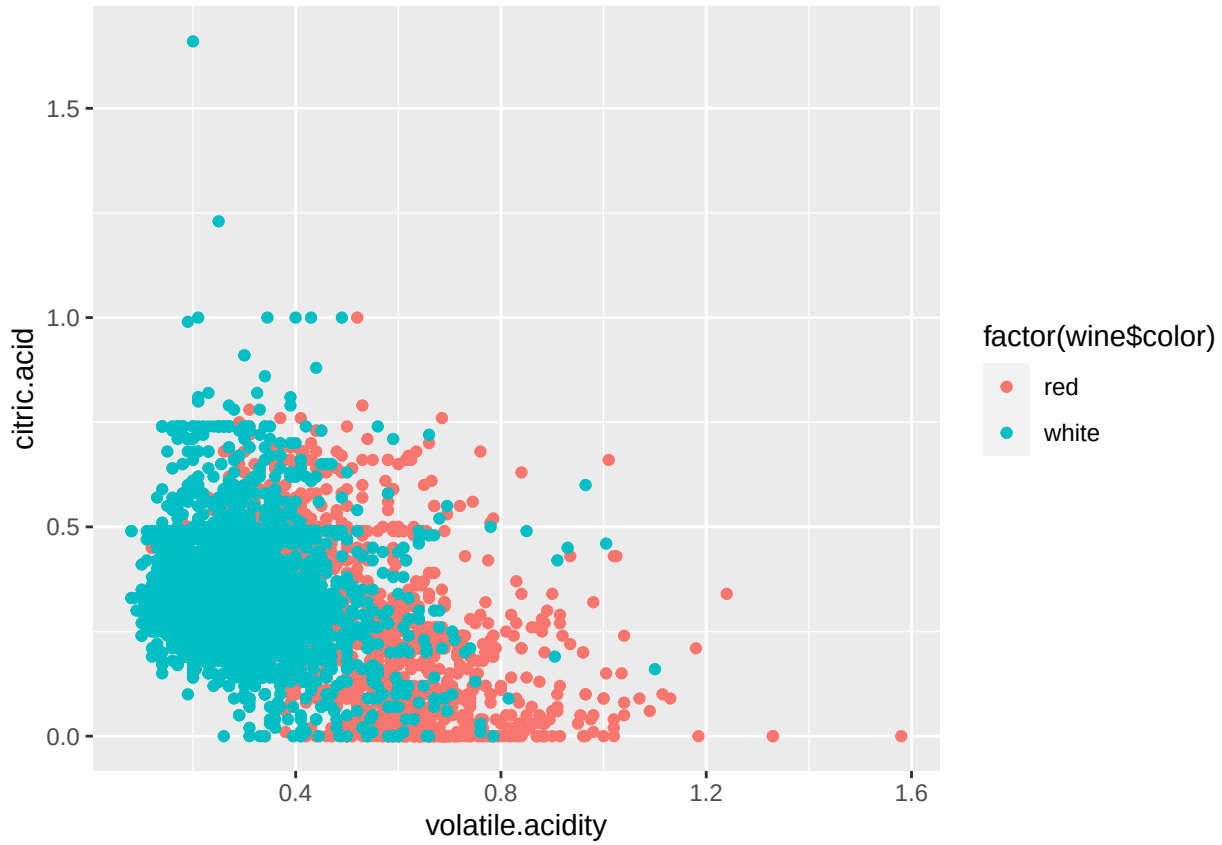



```
qplot(volatile.acidity, citric.acid, data = wine, color = factor(clust_2$cluster))
```




```
qplot(volatile.acidity, citric.acid, data = wine, color = factor(wine$color))
```

```
## Warning: Use of `wine$color` is discouraged. Use `color` instead.
```




```
#Let's see if the same thing can happen with 10 clusters and the 10 levels of quality.
clust_10 <- kmeans(winex, 10, nstart = 25)
```

```
## Warning: did not converge in 10 iterations
```

```
clust_10$center
```

```
##    fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
## 1    -0.34474464     -0.362006095  0.02933959      0.6528430 -0.2488022
## 2    -0.23350276      1.889729993 -1.70704579     -0.6439970  0.5876309
## 3    -0.06128092     -0.386311831  0.48928632      1.9067721 -0.1746695
## 4    -0.65158386     -0.589368007 -0.11148306     -0.4584905 -0.3628869
## 5    -0.23145723      0.004921089  0.93152198     -0.2154650  3.3558108
## 6     0.13435783     -0.445459071  0.12673924     -0.2506792 -0.3579116
## 7     2.48270277      0.283870423  1.19968667     -0.5737053  0.8078938
## 8     0.72030513      1.194337884 -0.25359035     -0.5899214  0.7827595
## 9     0.94594853      1.116858630  1.53479719     -0.7263299  9.8086634
## 10   -0.48847079     -0.256453228  0.02393302     -0.4589816 -0.5924093
##    free.sulfur.dioxide total.sulfur.dioxide     density         pH   sulphates
## 1            1.2654404           1.16623956  0.32560385 -0.1661447 -0.28269979
## 2           -0.8503624          -1.31901537  0.33022215  1.3349140  0.44775298
## 3            0.6190247           0.85543003  1.24688619 -0.6405324 -0.20576109
## 4            0.1203802           0.25249846 -0.51741399  0.9342853  0.09943656
## 5            0.5324614           0.37959522  0.05276444 -0.7381889 -0.21588812
## 6           -0.3539795           0.03175048 -0.43206752 -0.7613086 -0.47435124
## 7           -1.0401020          -1.44473433  1.00256315 -0.1923266  1.41739698
## 8           -0.6913470          -0.89068540  0.78207353  0.4186422  0.58710680
```

```
## 9           -0.8723480          -1.11672865  0.79480696 -0.9572747  4.42958583
## 10          -0.0760769          -0.13546285 -1.37913613 -0.1482919 -0.39605254
##        alcohol
## 1  -0.56772472
## 2   0.02619673
## 3  -1.01701035
## 4   0.06685503
## 5  -0.84991667
## 6  -0.05055325
## 7   0.29856417
## 8  -0.49224024
## 9  -0.89792651
## 10  1.48730360
```

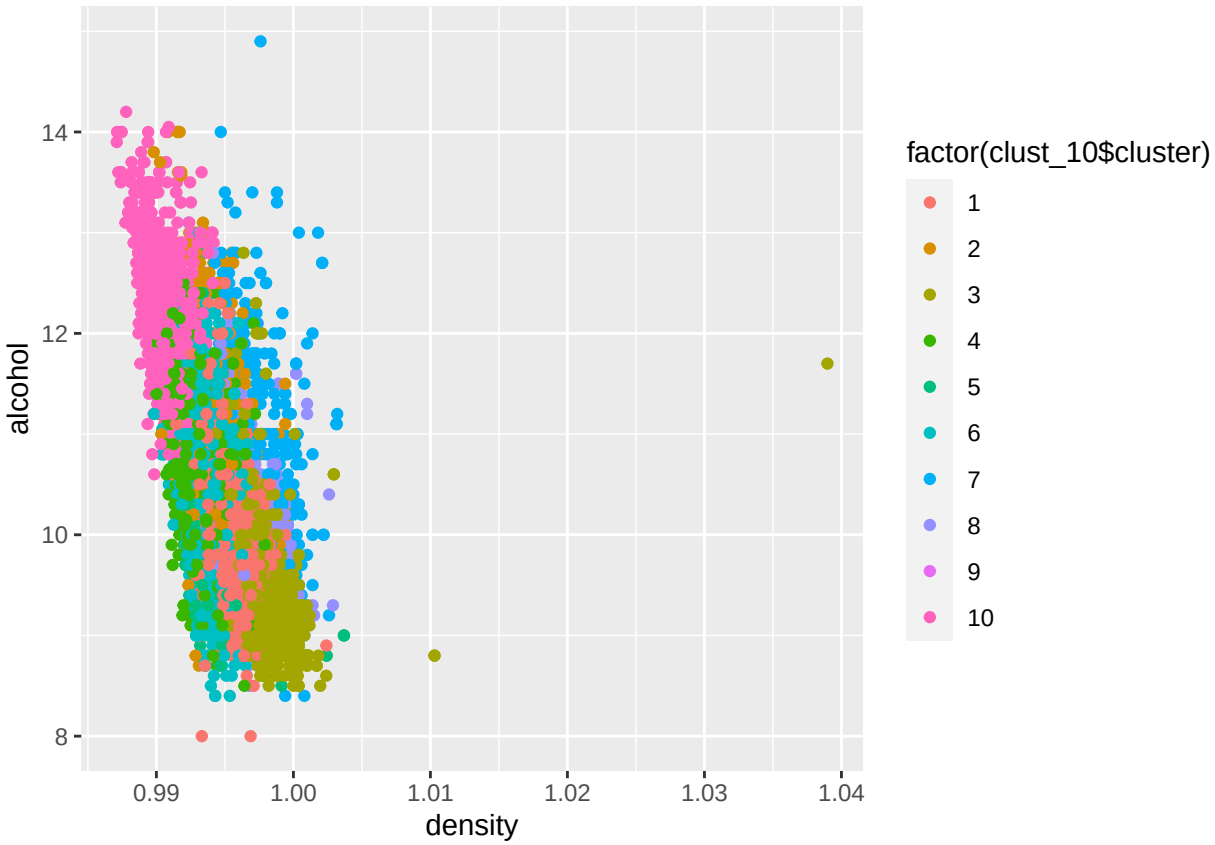
```
clust_10$center[1,]*sigma + mu
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##           6.76836848           0.28006659           0.32289678
##       residual.sugar            chlorides  free.sulfur.dioxide
##           8.54933407           0.04731743          52.98612653
## total.sulfur.dioxide              density                   pH
##         181.66259711           0.99567301           3.19178690
##            sulphates              alcohol
##           0.48920089           9.81466889
```

```
clust_10$center[10,]*sigma + mu
```

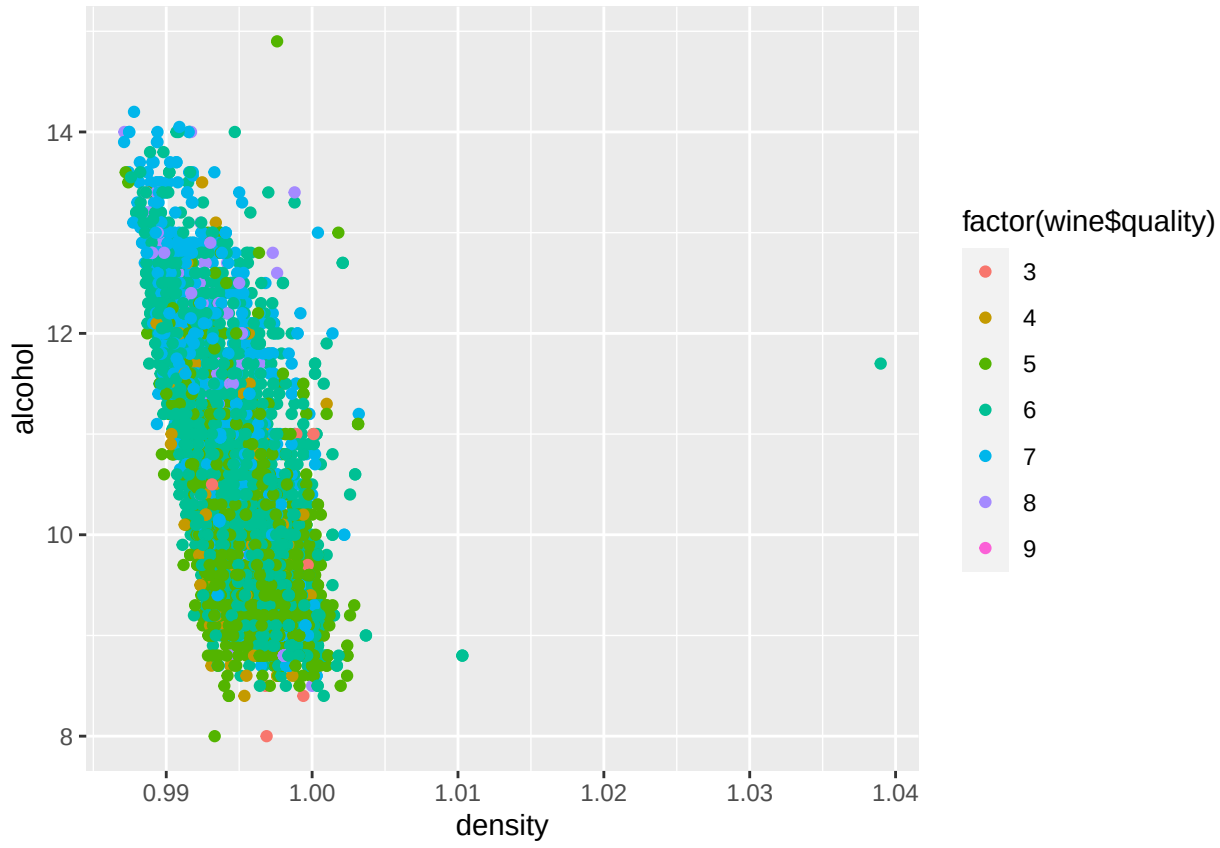
```
##        fixed.acidity     volatile.acidity          citric.acid
##           6.58203704           0.29744444           0.32211111
##       residual.sugar            chlorides  free.sulfur.dioxide
##           3.25949074           0.03527963          29.17500000
## total.sulfur.dioxide              density                   pH
##         108.08796296           0.99056106           3.19465741
##            sulphates              alcohol
##           0.47233333          12.26572531
```

```
#This doesn't really tell us much...
qplot(density, alcohol, data = wine, color = factor(clust_10$cluster))
```

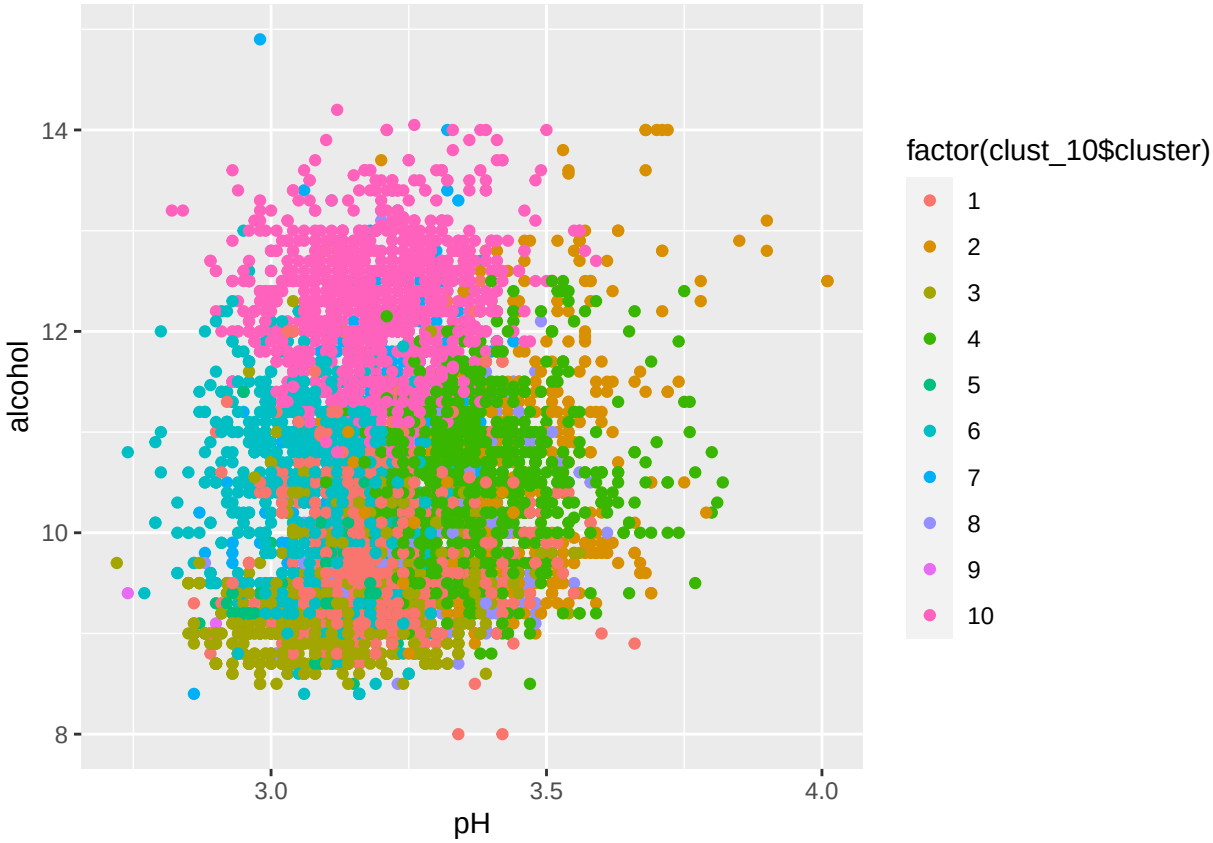



```
qplot(density, alcohol, data = wine, color = factor(wine$quality))
```

```
## Warning: Use of `wine$quality` is discouraged. Use `quality` instead.
```

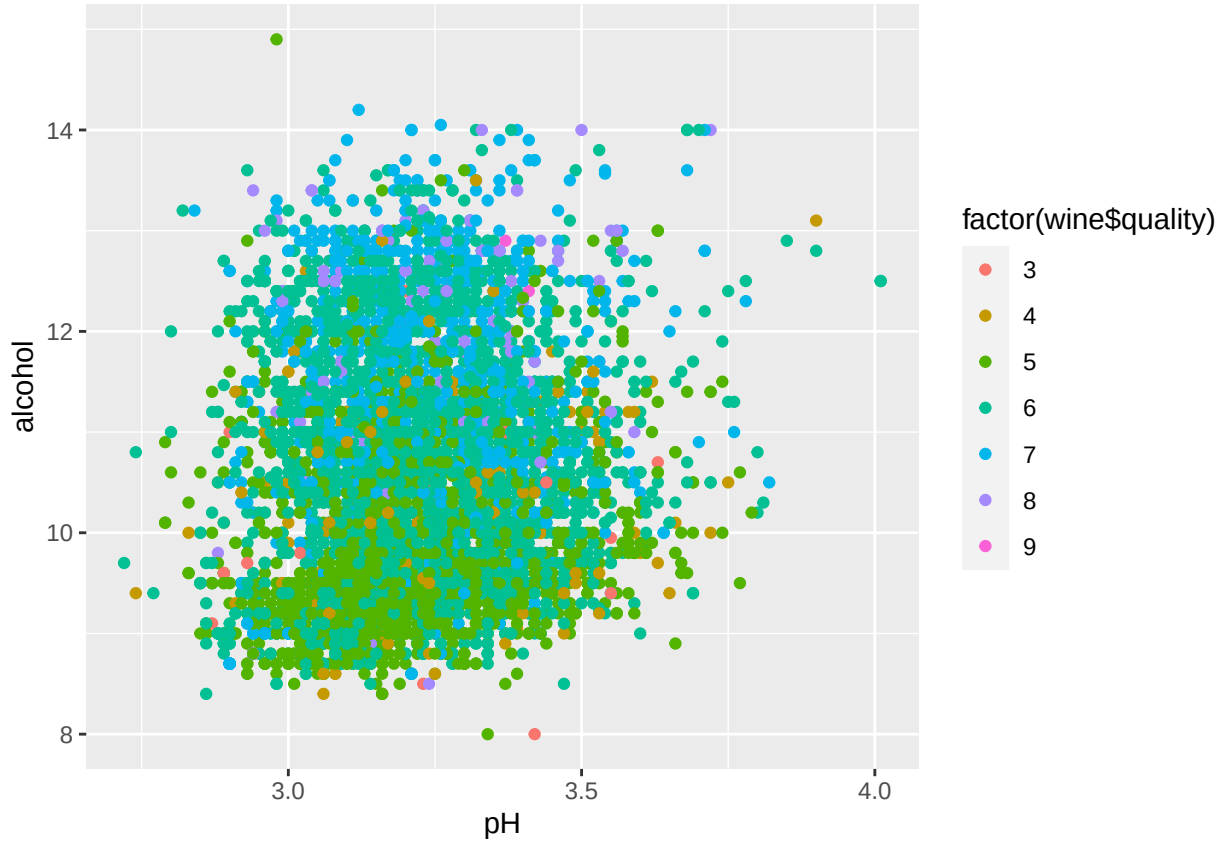



```
#This doesn't help either...
qplot(pH, alcohol, data = wine, color = factor(clust_10$cluster))
```




```
qplot(pH, alcohol, data = wine, color = factor(wine$quality))
```

```
## Warning: Use of `wine$quality` is discouraged. Use `quality` instead.
```




```
#Let's try manipulating the quality variable into different "tiers" of wine quality
wine$quality_tier <- rep(0,nrow(wine))
wine$quality_tier[which(wine$quality<=4)] <- "low"
wine$quality_tier[which(wine$quality==5)] <- "mid"
wine$quality_tier[which(wine$quality==6)] <- "mid"
wine$quality_tier[which(wine$quality==7)] <- "mid"
wine$quality_tier[which(wine$quality>=8)] <- "high"

#Try clusters again with 3
clust_3 <- kmeans(winex, 3, nstart = 25)
clust_3$center
```

```
##   fixed.acidity volatile.acidity  citric.acid residual.sugar   chlorides
## 1    -0.1839863       -0.3534198  0.279108125      1.1998473 -0.08840616
## 2    -0.3493514       -0.4029846 -0.005436896     -0.4362877 -0.44308115
## 3     0.8787523        1.1816140 -0.321751587     -0.6032147  0.94209722
##   free.sulfur.dioxide total.sulfur.dioxide    density          pH  sulphates
## 1          0.84685371           0.95753497  0.7580959 -0.38773319 -0.2580380
## 2         -0.09101054           0.03476467 -0.8518159 -0.03960046 -0.2833150
## 3         -0.83549093          -1.20473049  0.7071704  0.53604029  0.8421072
##      alcohol
## 1 -0.7954480
## 2  0.5690282
## 3 -0.1285825
```

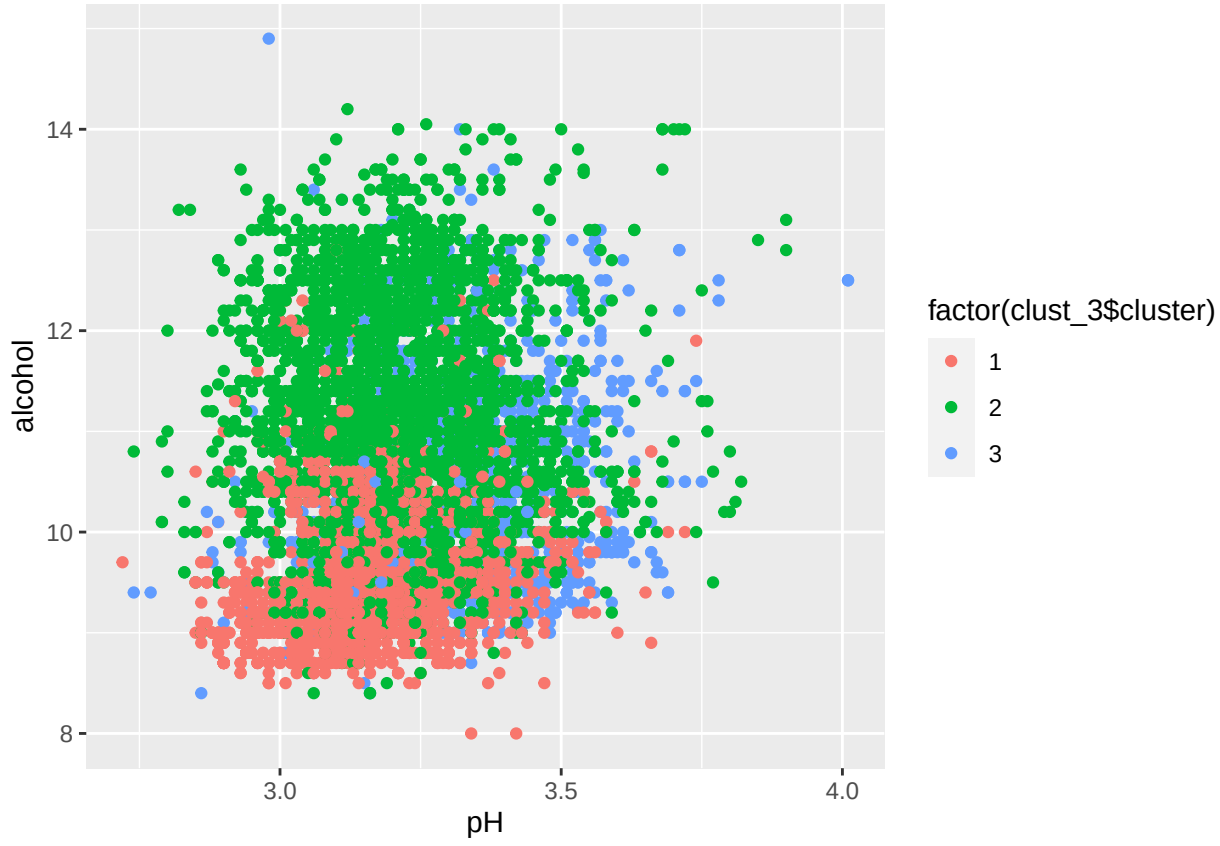
```
clust_3$center[1,]*sigma + mu
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##           6.97678100           0.28148021           0.35919261
##       residual.sugar            chlorides  free.sulfur.dioxide
##          11.15187335           0.05293668          45.55646438
## total.sulfur.dioxide              density                   pH
##         169.86622691           0.99696992           3.15615831
##             sulphates              alcohol
##           0.49287071           9.54306069
```

```
clust_3$center[3,]*sigma + mu
```

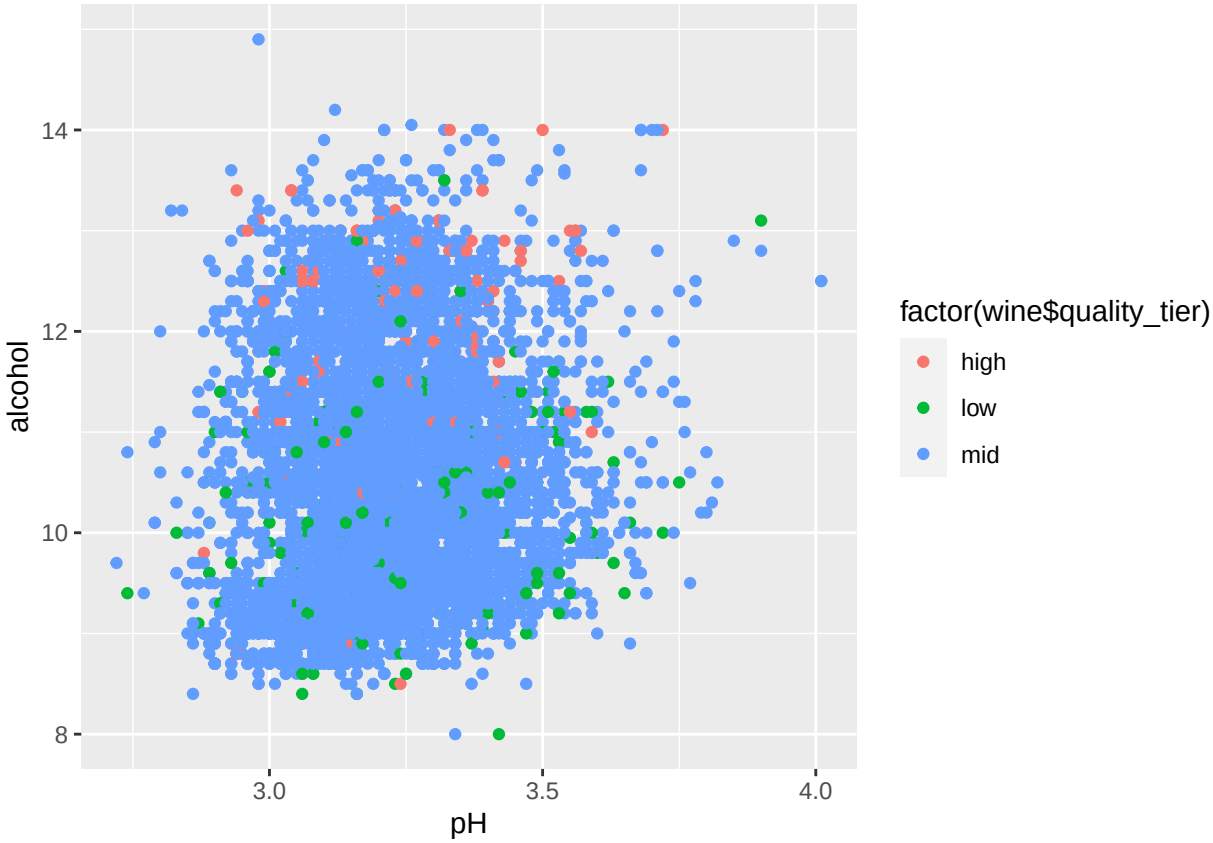
```
##        fixed.acidity     volatile.acidity          citric.acid
##           8.35455116           0.53420276           0.27187696
##       residual.sugar            chlorides  free.sulfur.dioxide
##           2.57325800           0.08903892          15.69585687
## total.sulfur.dioxide              density                   pH
##          47.65097301           0.99681721           3.30468927
##             sulphates              alcohol
##           0.65657878          10.33843900
```

```
#Better, but not a lot of information
qplot(pH, alcohol, data = wine, color = factor(clust_3$cluster))
```

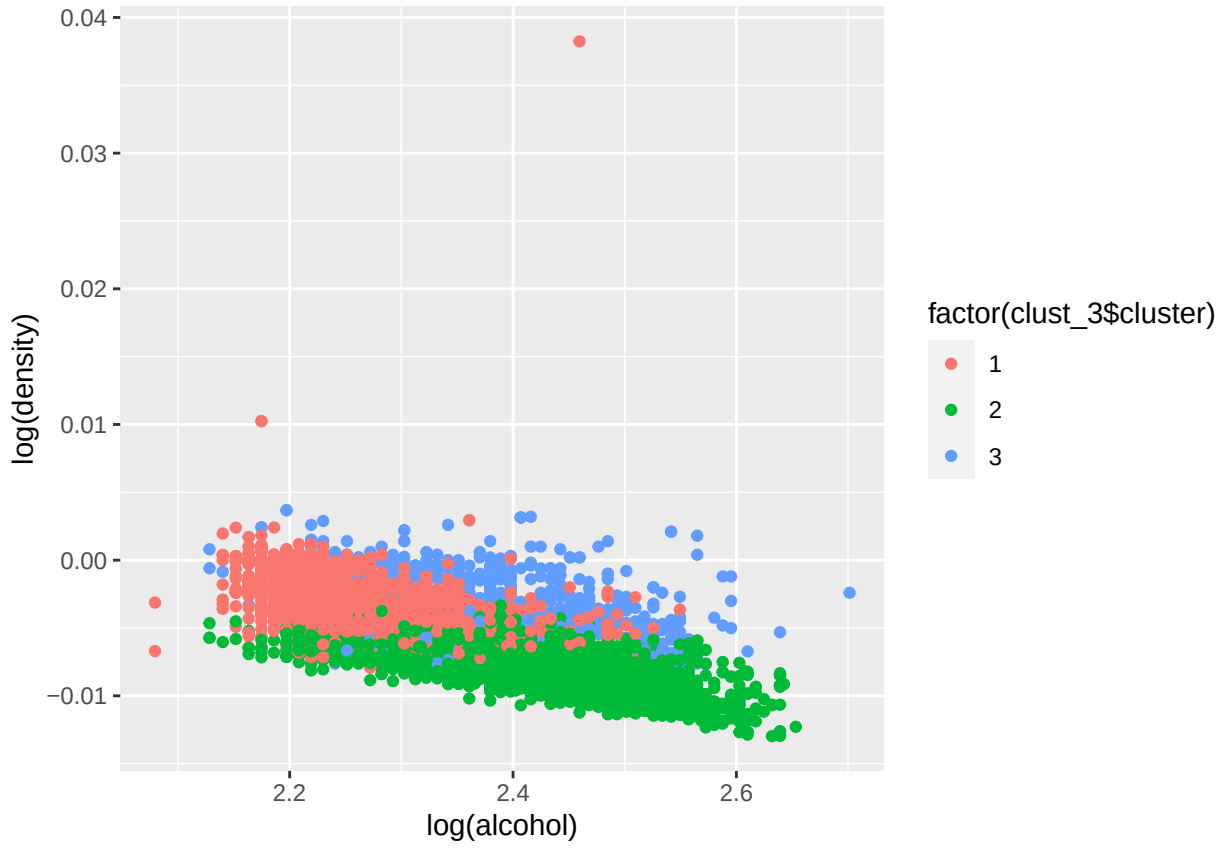



```
qplot(pH, alcohol, data = wine, color = factor(wine$quality_tier))
```

```
## Warning: Use of `wine$quality_tier` is discouraged. Use `quality_tier` instead.
```

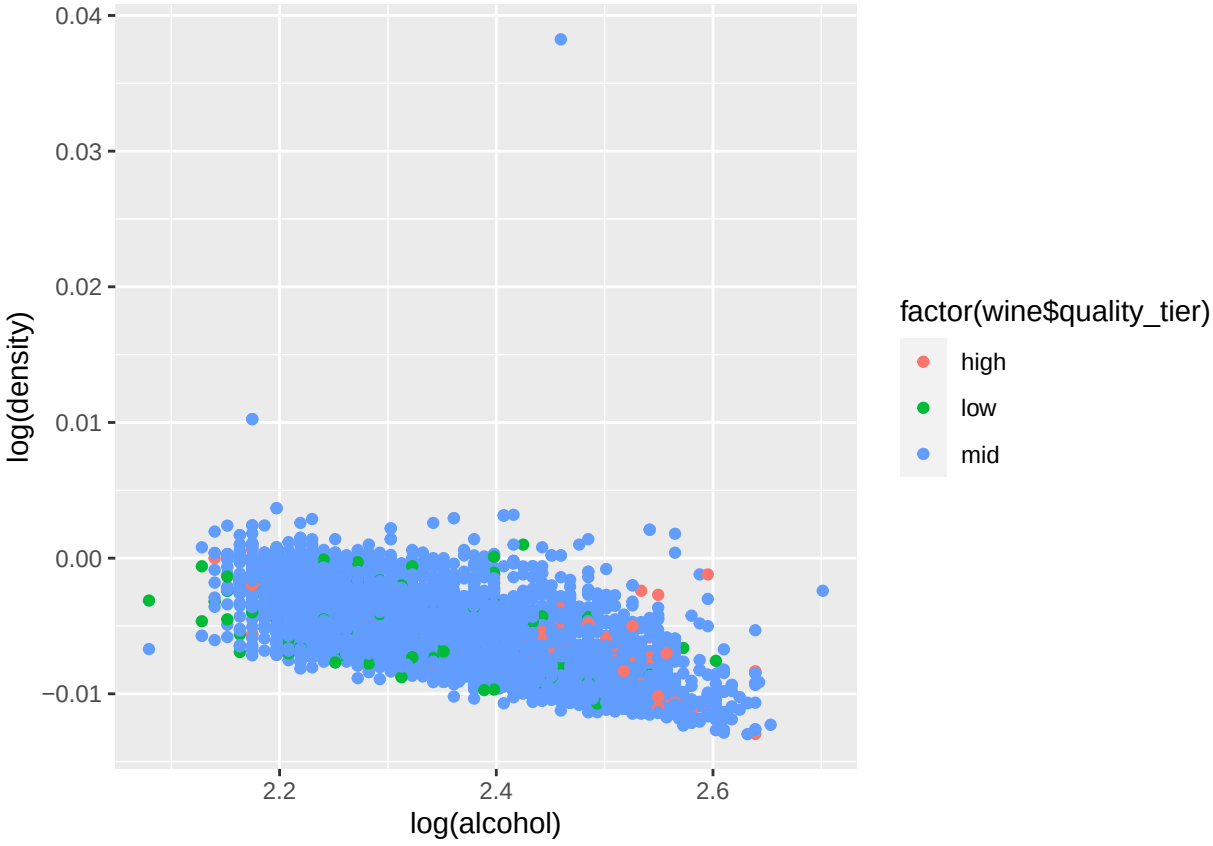



```
#Nope
qplot(log(alcohol), log(density) , data = wine, color = factor(clust_3$cluster))
```

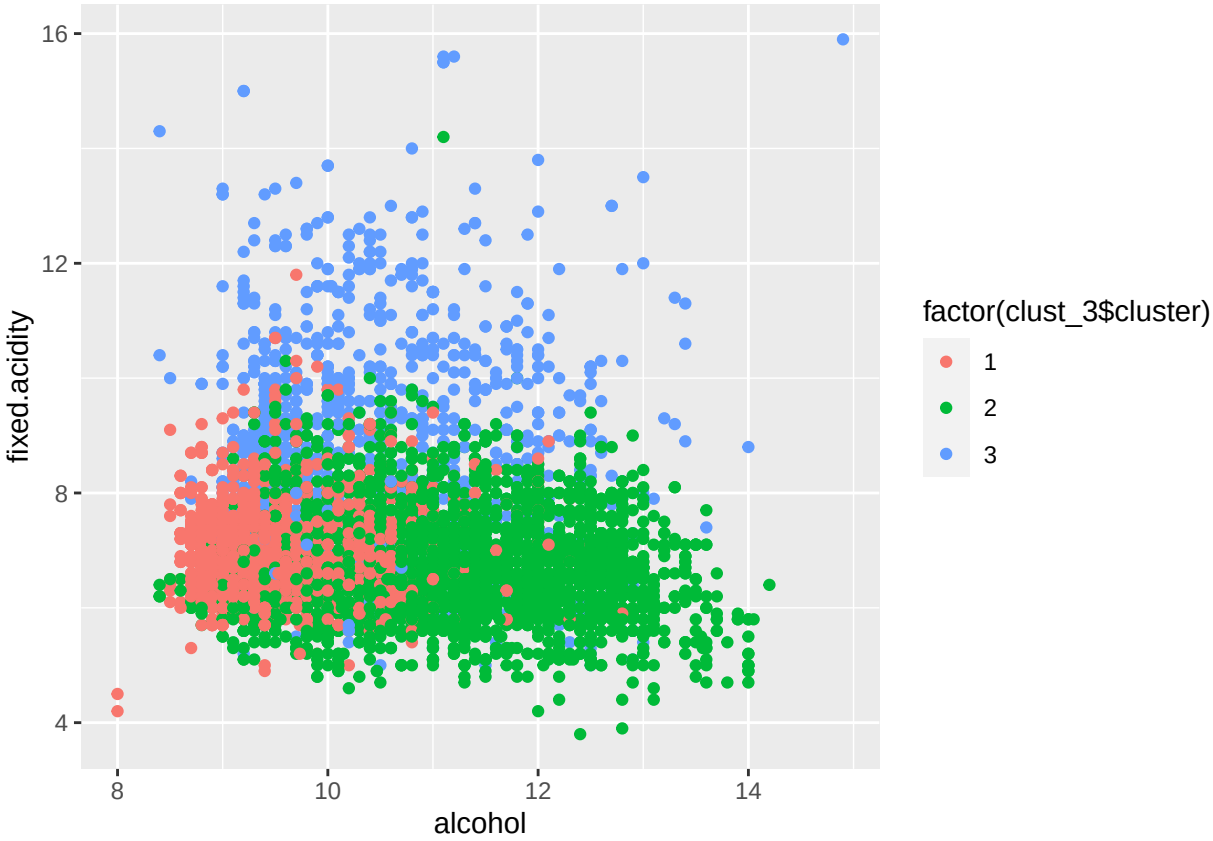



```
qplot(log(alcohol), log(density) , data = wine, color = factor(wine$quality_tier))
```

```
## Warning: Use of `wine$quality_tier` is discouraged. Use `quality_tier` instead.
```

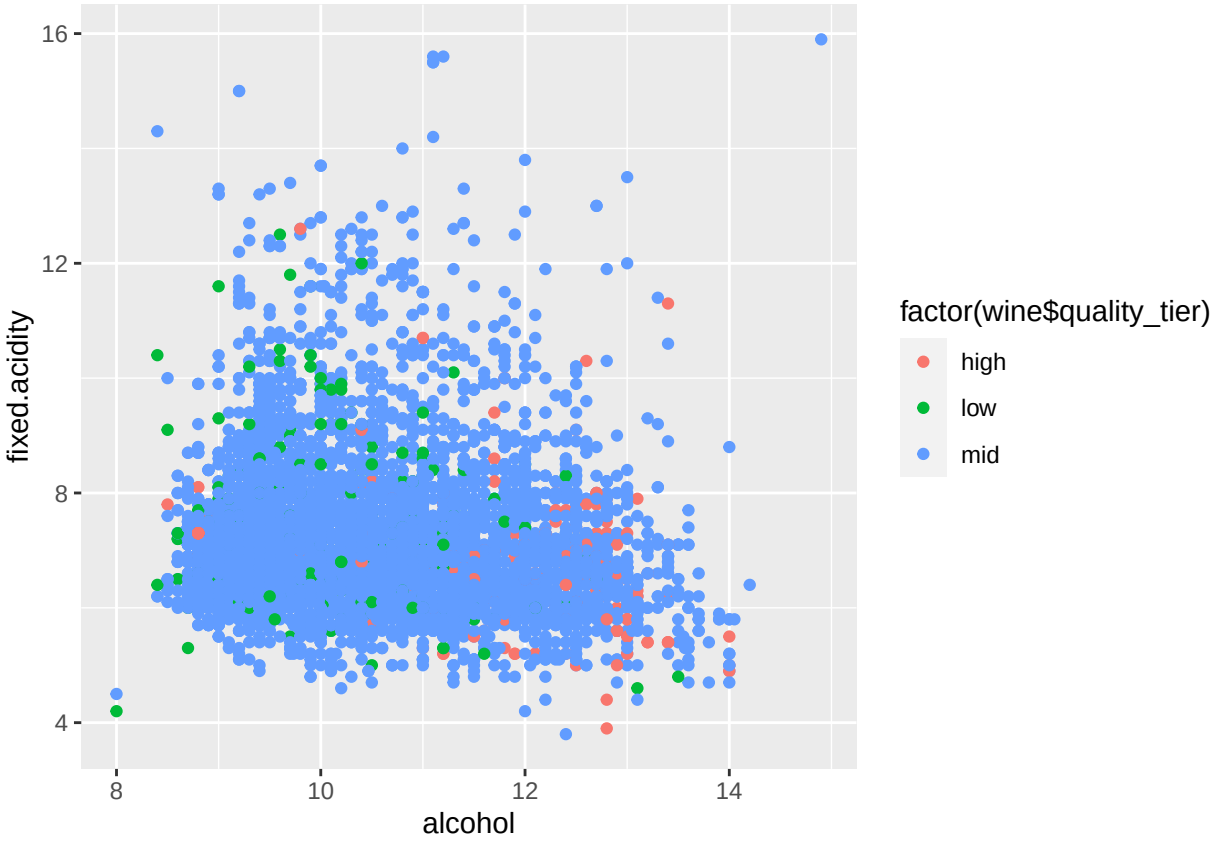



```
#Nope
qplot(alcohol, fixed.acidity , data = wine, color = factor(clust_3$cluster))
```




```
qplot(alcohol, fixed.acidity , data = wine, color = factor(wine$quality_tier))
```

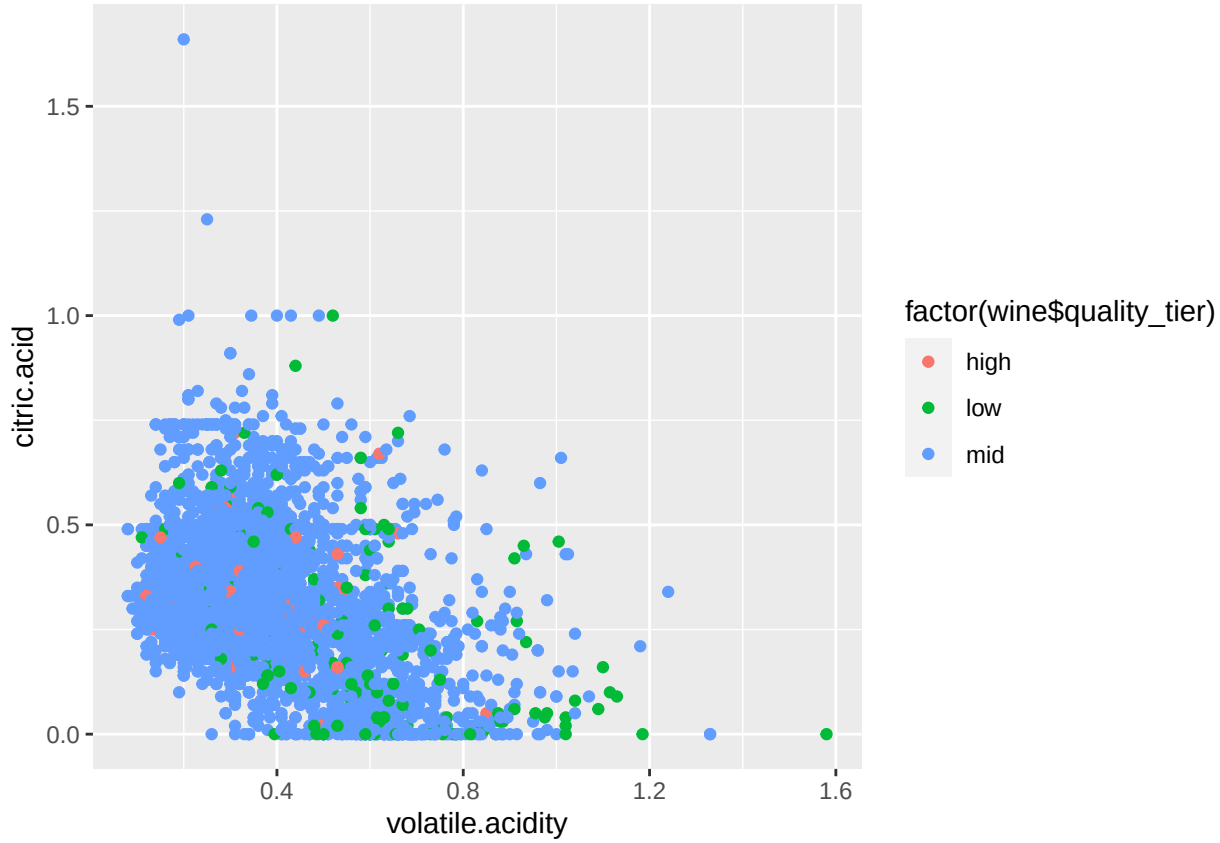
```
## Warning: Use of `wine$quality_tier` is discouraged. Use `quality_tier` instead.
```




```
#Nope
qplot(volatile.acidity, citric.acid , data = wine, color = factor(clust_3$cluster))
```


```
qplot(volatile.acidity, citric.acid , data = wine, color = factor(wine$quality_tier))
```

```
## Warning: Use of `wine$quality_tier` is discouraged. Use `quality_tier` instead.
```



After creating clusters of the wine data, it seems like 2 clusters did a good job of separating the wine by color. I took a few plots of different pairs of variables from the data and first highlighted the color of the clusters and saw how they compared when the color was coded by wine color. As you can see in the plots, the graphs have a similar color pattern when. This shows that the two clusters were successfully able to distinguish the wines by wine color (red or white). However, after creating 10 clusters to see if the clusters could separate the data by the different wine qualities, it did not produce successful results. The graphs color-colored by the 10 clusters were not similar to the graphs color-coded by the 10 different qualities. I thought that maybe it would be hard to distinguish between the 10 different groups of wine quality and it might be better to group the wine qualities into tiers of three levels: low, medium, and high. I ran clusters again, this time with three, to see if the quality tiers could be distinguished. However, this did not seem to work either as the graphs color-coded by the three clusters did not look similar to the graphs color-coded by the quality tiers. Additionally, the graphs color-coded by quality tiers did not have a clear pattern of three distinct groups. All three quality tiers were mixed all over the graph of several pairs of chemical variables. This gives a sign that the quality of a wine is not largely affected by its chemical composition.

PCA

```
#Create PCA
wine_pca<- prcomp(wine[,-(12:14)], center = TRUE, scale. = TRUE)
summary(wine_pca)
```

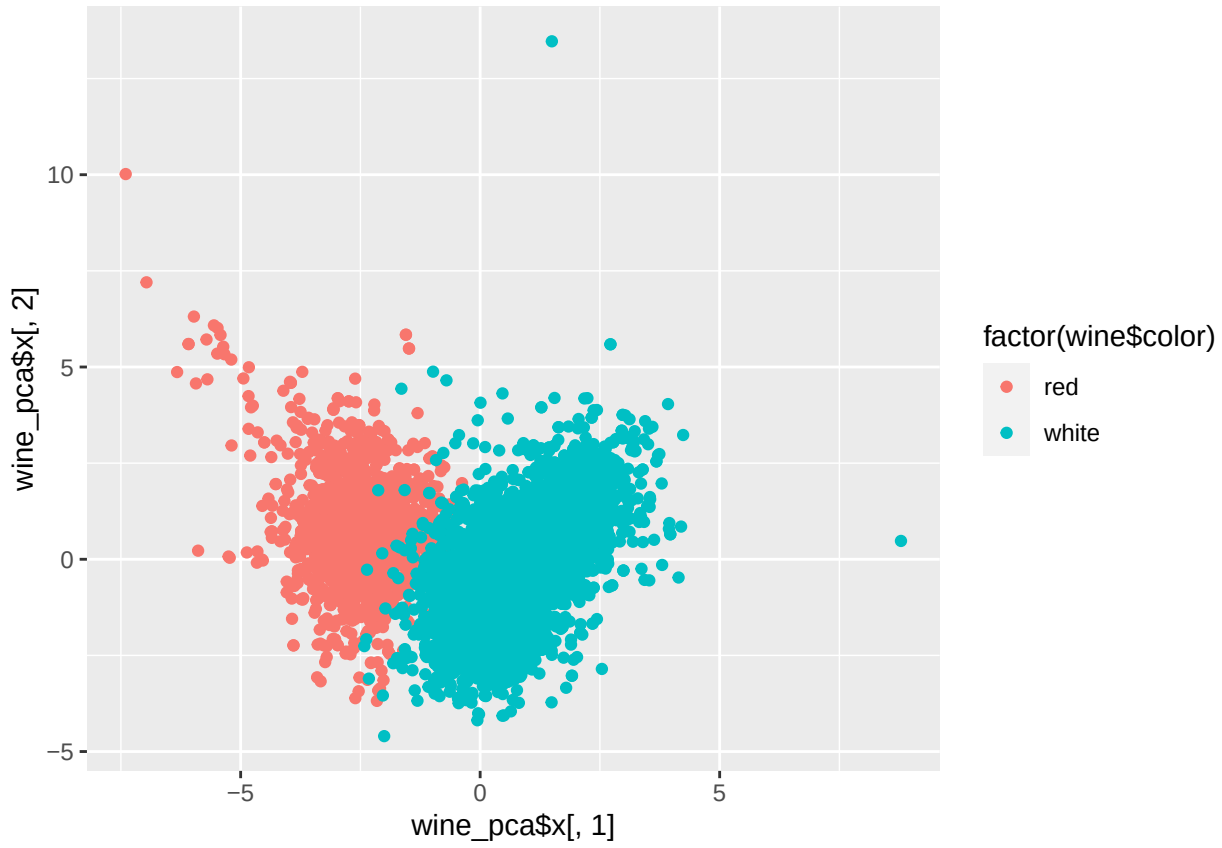
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7407 1.5792 1.2475 0.98517 0.84845 0.77930 0.72330
## Proportion of Variance 0.2754 0.2267 0.1415 0.08823 0.06544 0.05521 0.04756
## Cumulative Proportion  0.2754 0.5021 0.6436 0.73187 0.79732 0.85253 0.90009
```

```
##                            PC8     PC9   PC10    PC11
## Standard deviation     0.70817 0.58054 0.4772 0.18119
## Proportion of Variance 0.04559 0.03064 0.0207 0.00298
## Cumulative Proportion  0.94568 0.97632 0.9970 1.00000
```

```
round(wine_pca$rotation[,1:3],2)
```

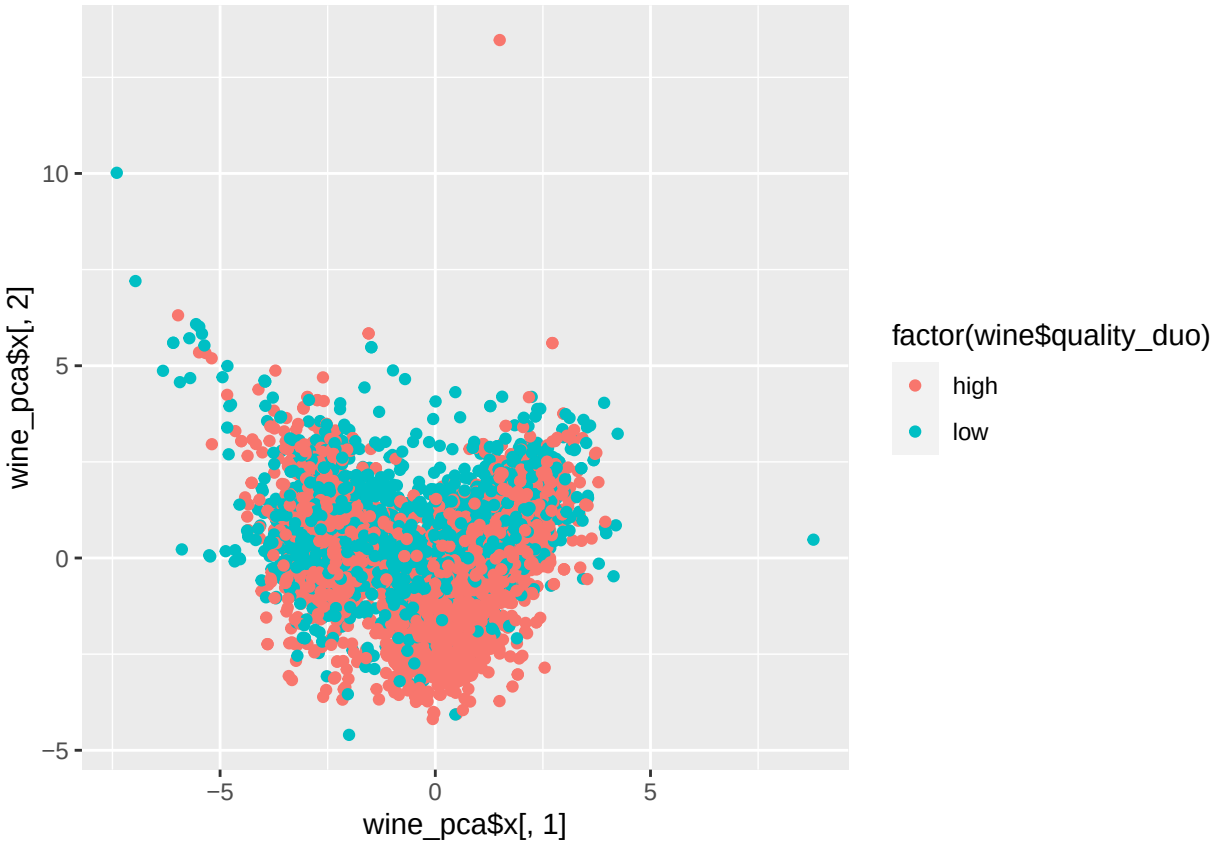
```
##                         PC1   PC2   PC3
## fixed.acidity         -0.24  0.34 -0.43
## volatile.acidity      -0.38  0.12  0.31
## citric.acid            0.15  0.18 -0.59
## residual.sugar         0.35  0.33  0.16
## chlorides             -0.29  0.32  0.02
## free.sulfur.dioxide    0.43  0.07  0.13
## total.sulfur.dioxide   0.49  0.09  0.11
## density               -0.04  0.58  0.18
## pH                    -0.22 -0.16  0.46
## sulphates             -0.29  0.19 -0.07
## alcohol               -0.11 -0.47 -0.26
```

```
qplot(wine_pca$x[,1], wine_pca$x[,2],
      color = factor(wine$color))
```




```
#Rearrange the quality and color variables
wine$quality_duo <- rep(0,nrow(wine))
wine$quality_duo <- ifelse(wine$quality <6, "low", "high")
wine$quality_bi <- ifelse(wine$quality_duo =="high", 1, 0)
wine$color_bi <- ifelse(wine$color =="red", 1, 0)
```

```r
#Plot of two PCA's and wine quality
qplot(wine_pca$x[,1], wine_pca$x[,2],
      color = factor(wine$quality_duo))
```




```r
#Merge data
wine2 <- merge(wine, wine_pca$x[,1:3], by="row.names")
wine2 <- rename(wine2, wine = Row.names)

#Let's see if a logistic regression model can show a relationship between predicting wine color or qual
model1 <- glm(color_bi ~ PC1 + PC2 + PC3, family = binomial, data = wine2)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
model2 <- glm(quality_bi ~ PC1 + PC2 + PC3, family = binomial, data = wine2)
summary(model1)
```

```
##
## Call:
## glm(formula = color_bi ~ PC1 + PC2 + PC3, family = binomial,
##     data = wine2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5188  -0.0565  -0.0143  -0.0004   5.2891
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -4.42158    0.20135 -21.960   <2e-16 ***
## PC1         -3.83718    0.16269 -23.585   <2e-16 ***
## PC2          0.92390    0.07517  12.291   <2e-16 ***
## PC3          0.14920    0.07790   1.915   0.0554 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7250.98  on 6496  degrees of freedom
## Residual deviance:  733.47  on 6493  degrees of freedom
## AIC: 741.47
##
## Number of Fisher Scoring iterations: 9
```

```
summary(model2)
```

```
##
## Call:
## glm(formula = quality_bi ~ PC1 + PC2 + PC3, family = binomial,
##     data = wine2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3373  -1.1186   0.6090   0.9285   3.6464
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.62947    0.02819  22.330  < 2e-16 ***
## PC1          0.08953    0.01508   5.936 2.93e-09 ***
## PC2         -0.42411    0.01890 -22.439  < 2e-16 ***
## PC3         -0.37081    0.02238 -16.570  < 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: 8541.0  on 6496  degrees of freedom
## Residual deviance: 7680.9  on 6493  degrees of freedom
## AIC: 7688.9
##
## Number of Fisher Scoring iterations: 3
```

When PCA is run on the wine data, it produces similar results to above, when creating clusters. PCA was able to distinguish the data by color pretty easily, perhaps even better than clustering. The graph of PC1 vs PC2 and color-coded by wine color shows a clear distinction between red and white wine. However, like clustering, PCA seemed to also have a hard time distinguishing the wine by quality. The graph of PC1 vs PC2 and color-coded by wine quality (separated between low and high) shows the two levels of quality greatly intermingled. I even tried to see what a logistic regression model based on three PC's would fare when predicting color or quality. As expected, the model predicting color had a much lower AIC than the model predicting quality.

From these two methods, I believe that PCA was a better method to distinguish the data based on wine color. Graphing PC1 and PC2 was enough to show the color difference but it also did it in a more efficient way than clustering. PCA was able to condense all of the variables contained in the wine data into a smaller set of data and show the color difference in one graph alone. Clustering required a bit more work and graphs

to show that it was able to show the color difference. However, neither method could accurately distinguish the data by wine quality. This is most likely because there are other factors than chemical composition that pertain to the quality given to a wine by wine snobs.