Data Science 2: Midterm Exam

March, 2018

Please include at the top of your exam whether you are a 109b/121b student, or a 209b student

I am a 209b student.

This exam involves exploring a data set on red wine quality, and how quality relates to physio-chemical features of wine. A description of the study can be downloaded from the **publisher's web site.** The following questions will focus on a subset of data consisting of 1599 red wines. The data are contained in the file `winequality-red.csv`. For each bottle of wine, the following features are measured.

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol
12. quality (score between 0 and 10)

The main goal of this analysis is to predict red wine quality from the physio-chemical features.

Problem 1 [30 points]

Fit an additive model of quality on the physio-chemical variables on all 1599 wines in the data set. Use smoothing splines to fit each predictor variable. No need to explicitly perform cross-validation - please use the default smoothing selections.

```r
library(ggplot2)
library(dplyr)
library(tidyr)
library(gridExtra)
library(splines)
library(gam)
library(cluster)
library(factoextra)
library(matrixStats)
library(mclust)
library(NbClust)
library(reshape2)
library(devtools)
library(ggfortify)
library(corrplot)
library(rstan)
```

```r
# read data
wine_data = read.csv("winequality-red.csv")
head(wine_data)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

```r
# check data type
sapply(wine_data, class)
```

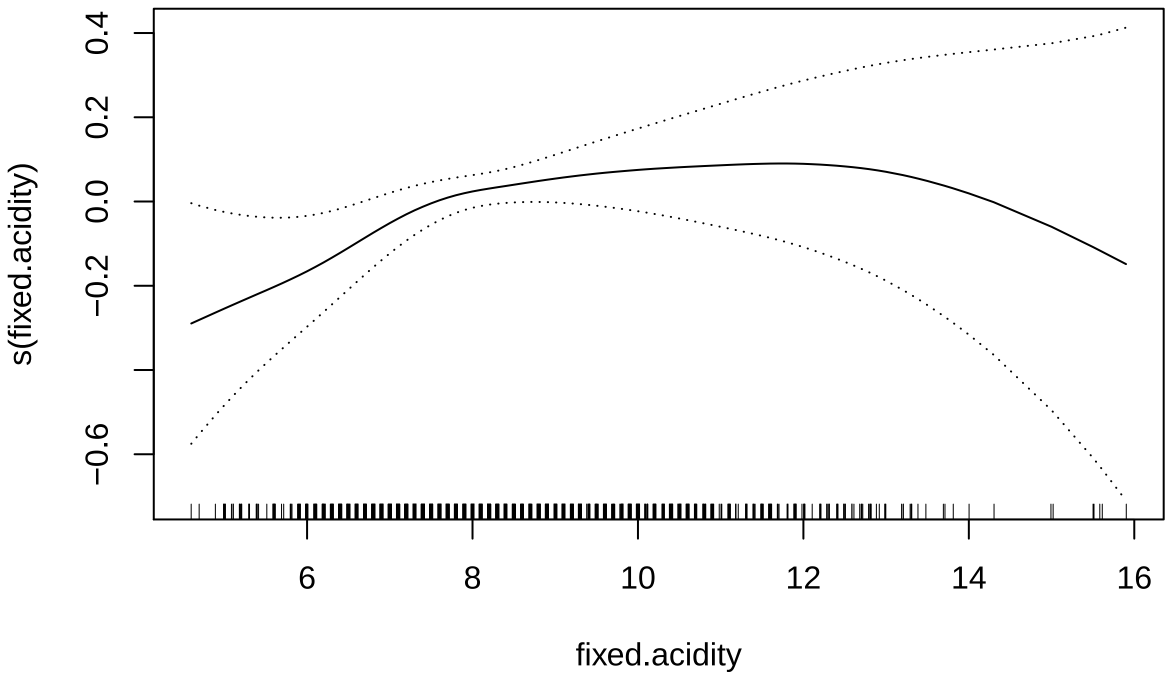
```
##        fixed.acidity     volatile.acidity          citric.acid
##            "numeric"            "numeric"            "numeric"
##       residual.sugar            chlorides  free.sulfur.dioxide
##            "numeric"            "numeric"            "numeric"
## total.sulfur.dioxide              density                   pH
##              "numeric"              "numeric"              "numeric"
##              sulphates                alcohol                quality
##              "numeric"              "numeric"              "integer"
```

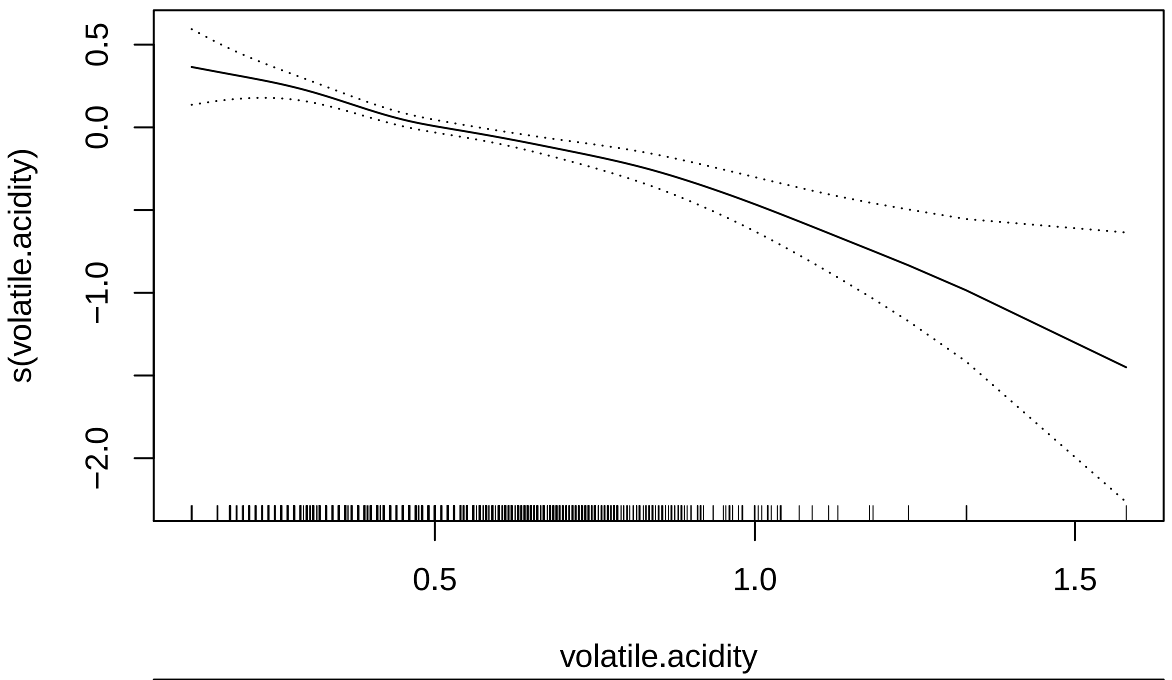
```
feature_names <- colnames(wine_data)[-12]
```

```
gam.s_formula <- as.formula(paste0("quality ~ ",
                                   paste0("s(", feature_names, ")",
                                          collapse = "+")))
model.gam.s <- gam(gam.s_formula, data=wine_data)
```

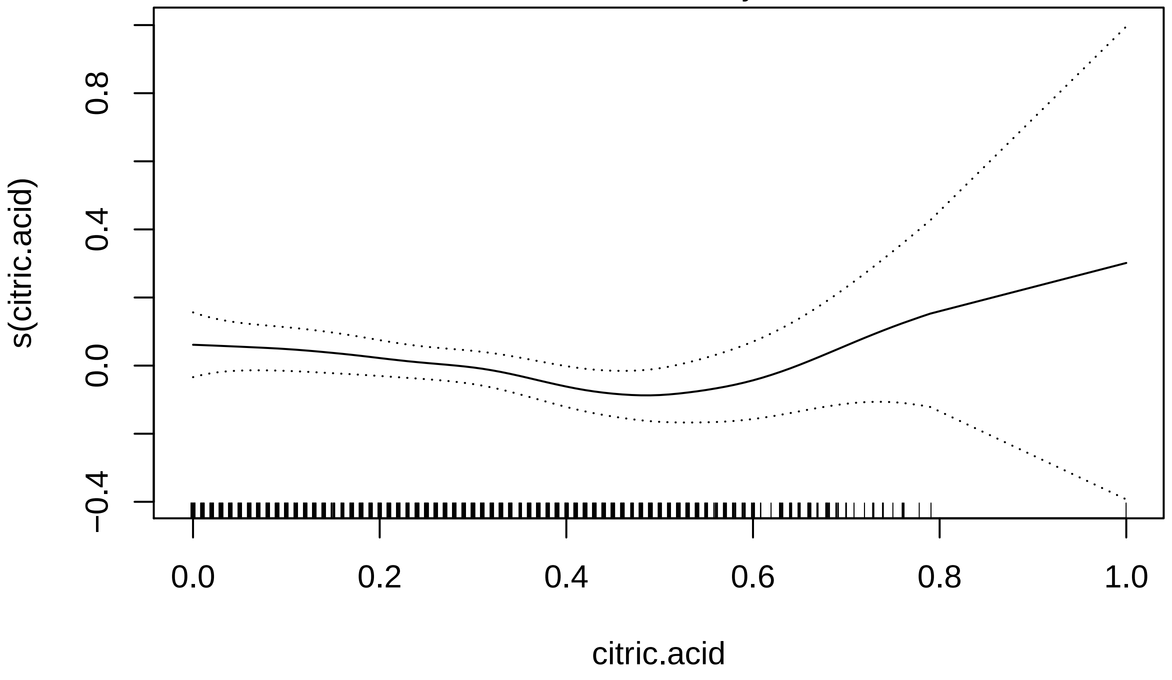
(a) [5 points] Plot the smooth of each predictor variable with standard error bands. Which variables seem to have a non-linear contribution to mean quality?

```
plot(model.gam.s, se = TRUE)
```

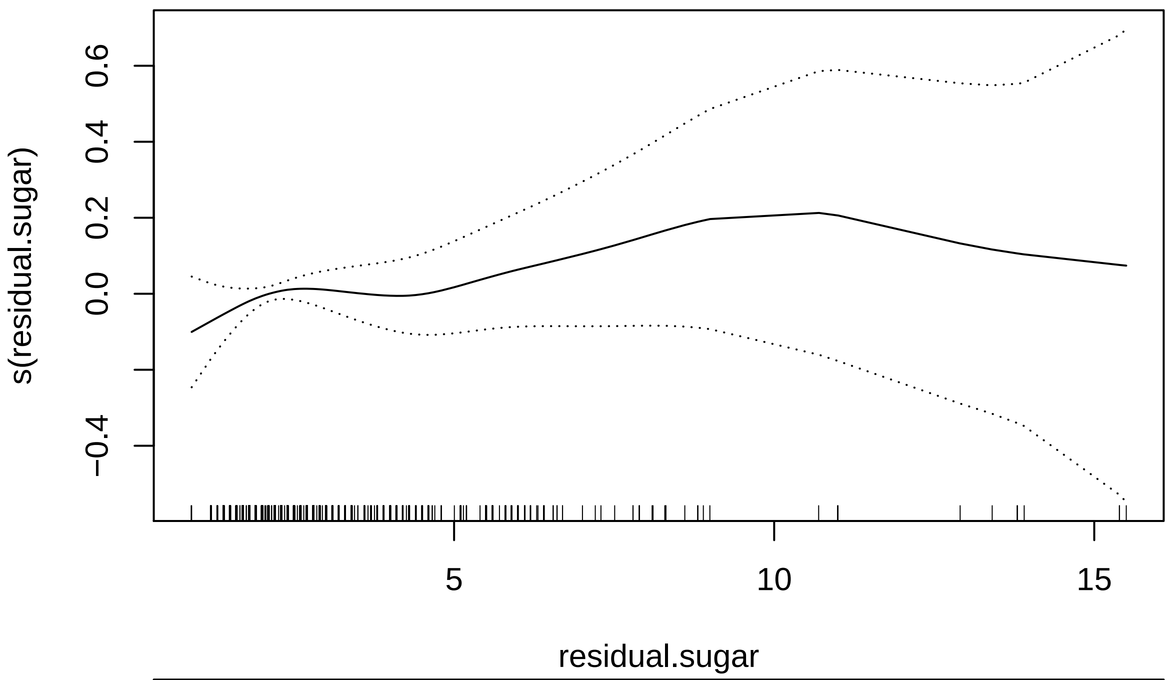




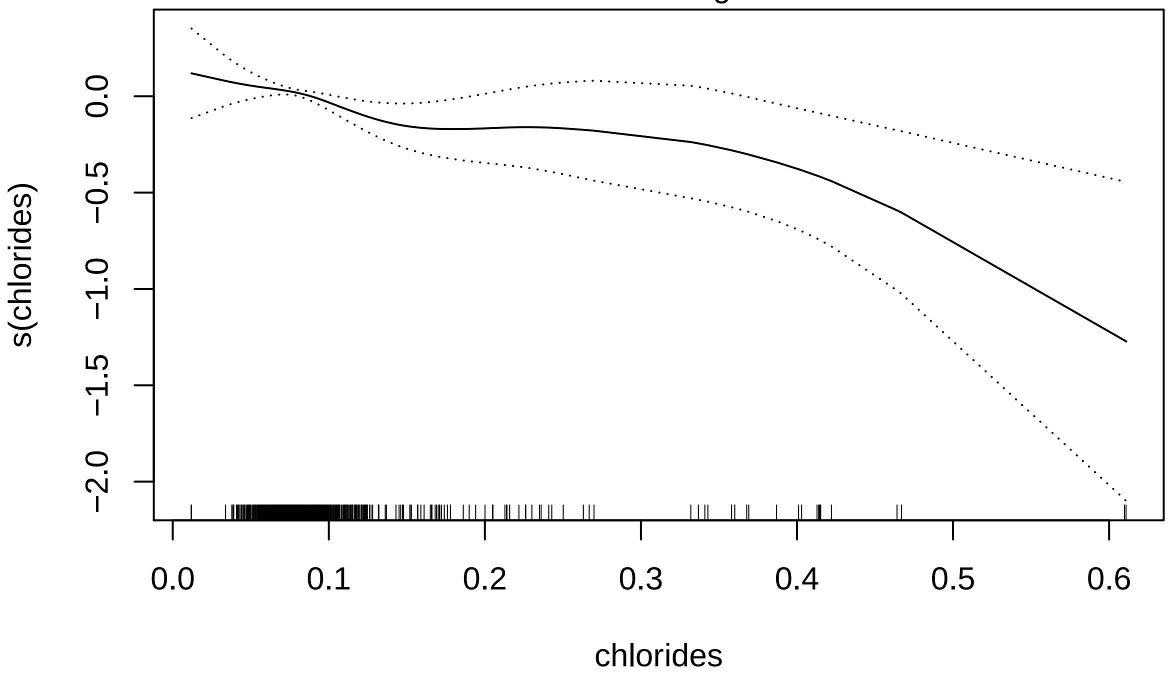
s(volatile.acidity)
0.5
0.0
−1.0
−2.0
0.5
1.0
1.5
volatile.acidity



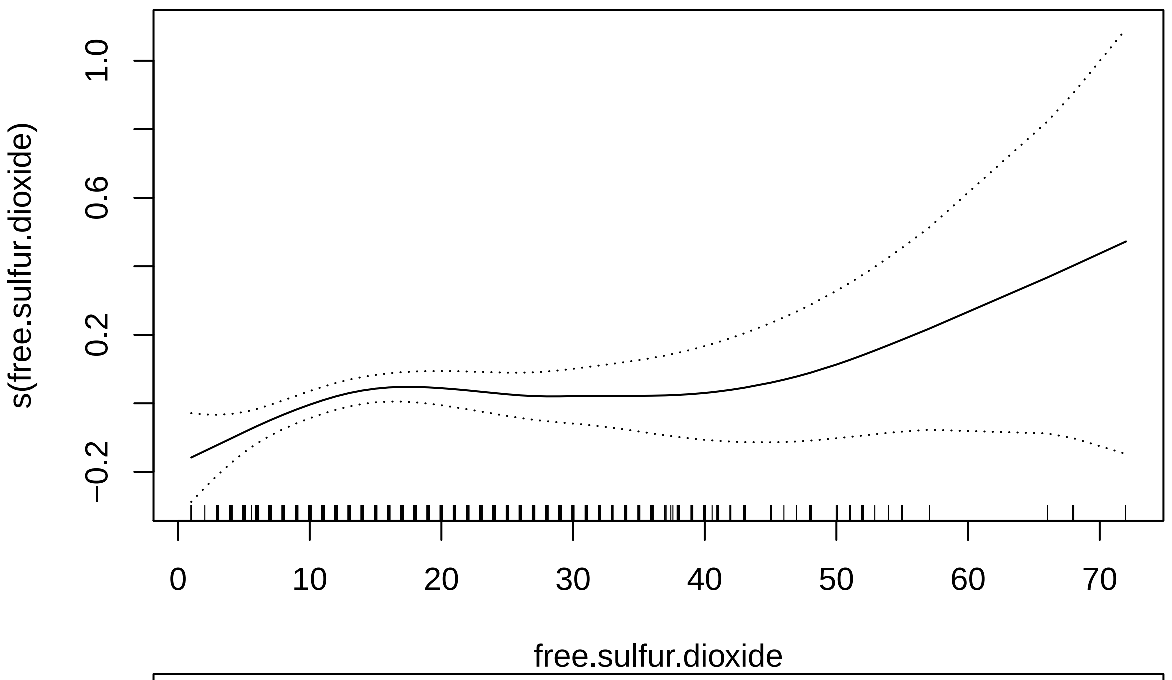
s(citric.acid)
0.8
0.4
0.0
−0.4
0.0
0.2
0.4
0.6
0.8
1.0
citric.acid

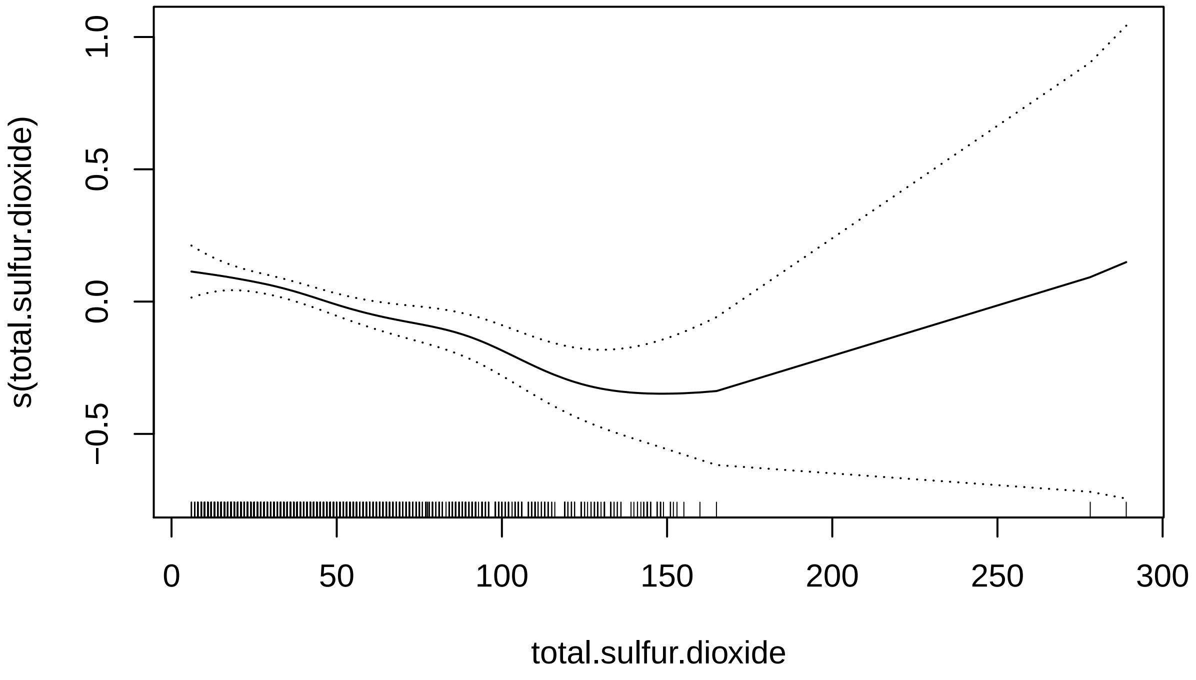


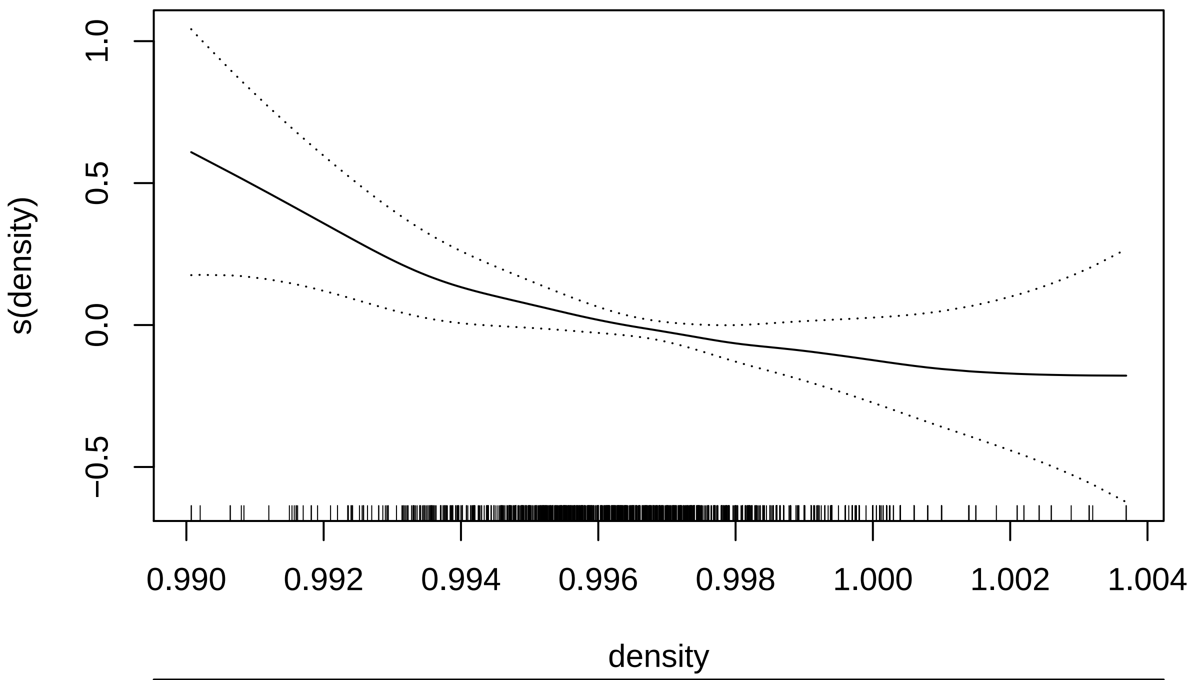
s(residual.sugar)
0.6
0.4
0.2
0.0
−0.4
5
10
15
residual.sugar

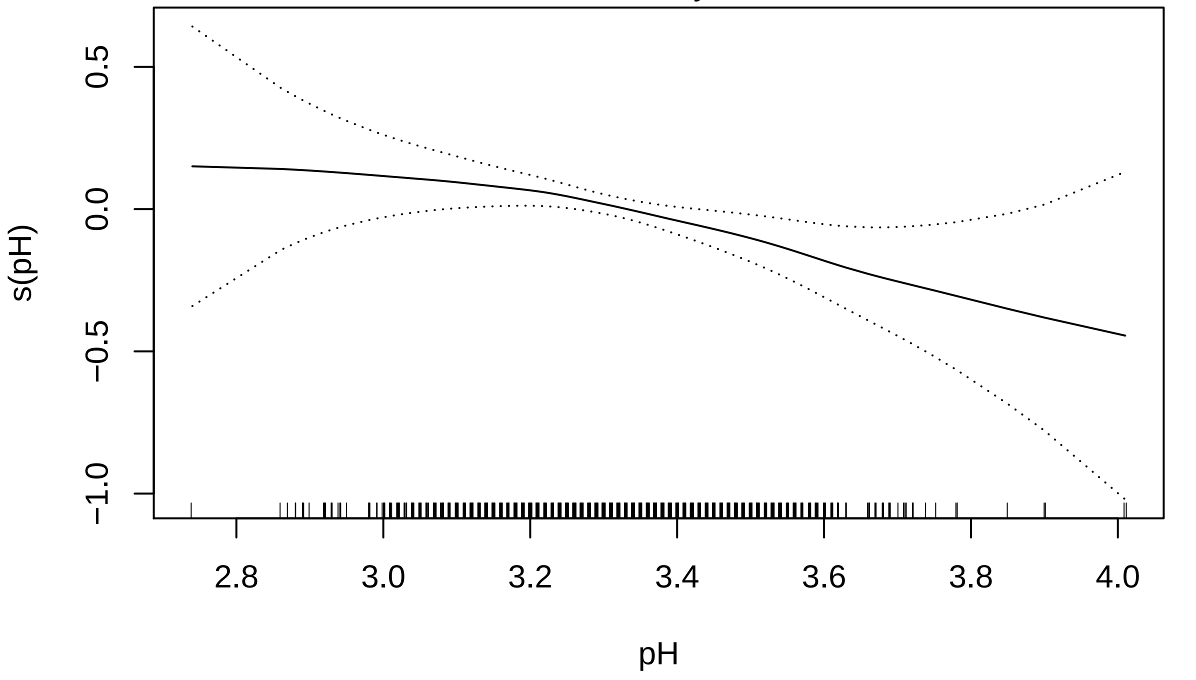


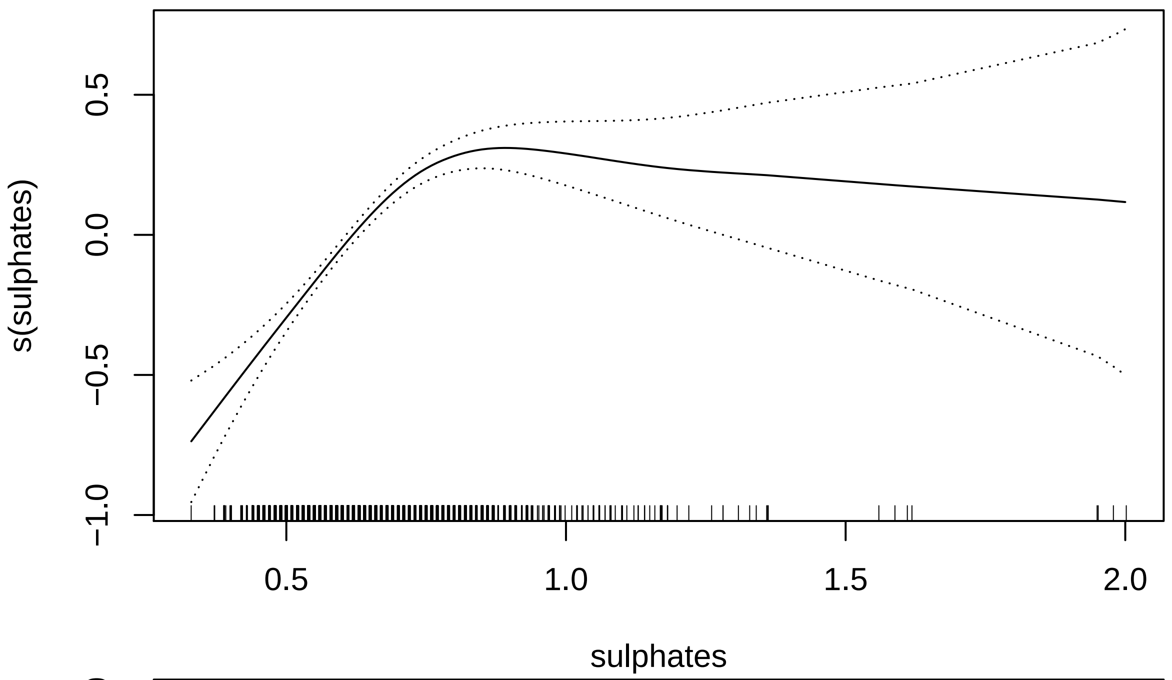
s(chlorides)
0.0
−0.5
−1.0
−1.5
−2.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
chlorides

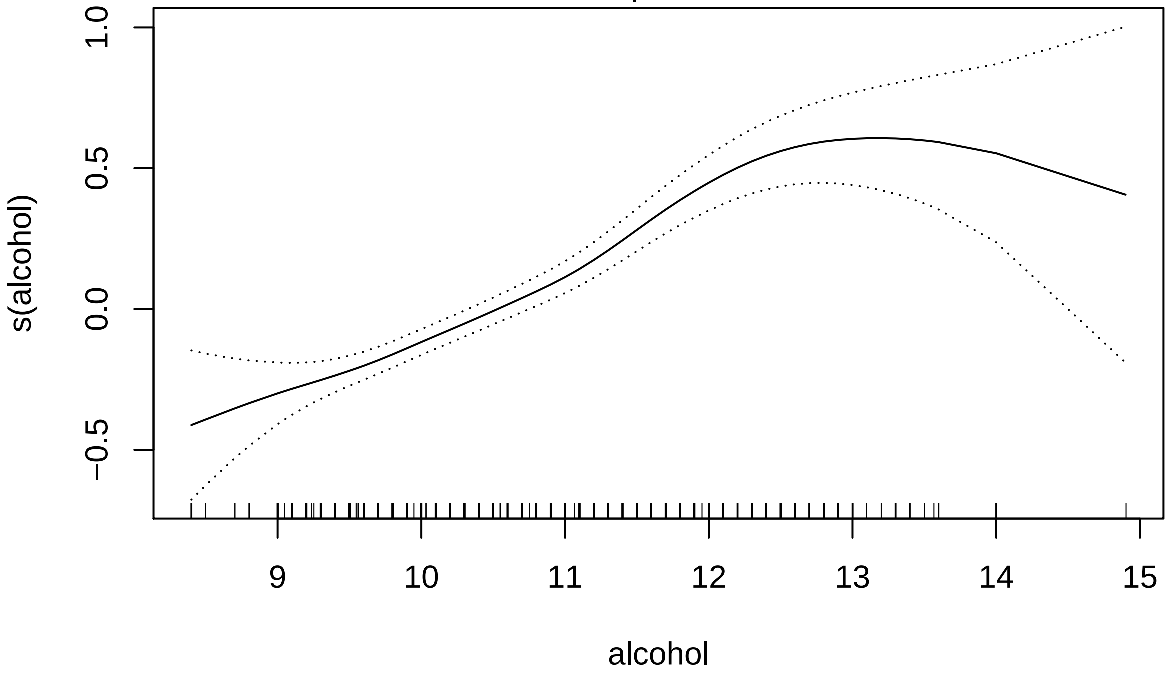

s(free.sulfur.dioxide)
−0.2
0.2
0.6
1.0
0
10
20
30
40
50
60
70
free.sulfur.dioxide


s(total.sulfur.dioxide)
−0.5
0.0
0.5
1.0
0
50
100
150
200
250
300
total.sulfur.dioxide


s(density)
1.0
0.5
0.0
−0.5
0.990
0.992
0.994
0.996
0.998
1.000
1.002
1.004
density


s(pH)
0.5
0.0
−0.5
−1.0
2.8
3.0
3.2
3.4
3.6
3.8
4.0
pH





ANSWER (Problem 1. (a)):
Based on the plots of each predictor variable's smooth with standard error bands, 'fixed.acidity', 'citric.acid', 'residual.sugar', 'chlorides', 'free.sulfur.dioxide', 'total.sulfur.dioxide', 'sulphates' and 'alcohol' seem to have a non-linear contribution to mean quality.

(b) [10 points] Is the overall non-linearity evidenced in the variable-specific smooths statistically significant? Justify your answer with a likelihood ratio test comparing the additive model to a model that includes the features linearly.

```r
# check variable-specific smooths' significance
summary(model.gam.s)
```

```
##
## Call: gam(formula = gam.s_formula, data = wine_data)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.59210 -0.39451 -0.01445  0.41169  2.01386
##
## (Dispersion Parameter for gaussian family taken to be 0.3913)
##
##     Null Deviance: 1042.165 on 1598 degrees of freedom
## Residual Deviance: 608.1472 on 1554 degrees of freedom
## AIC: 3083.986
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                            Df Sum Sq Mean Sq  F value    Pr(>F)
## s(fixed.acidity)            1  11.98  11.977  30.6059 3.704e-08 ***
## s(volatile.acidity)         1 120.14 120.136 306.9848 < 2.2e-16 ***
## s(citric.acid)              1   0.00   0.002   0.0051  0.943113
## s(residual.sugar)           1   0.09   0.090   0.2299  0.631661
## s(chlorides)                1   6.66   6.663  17.0254 3.884e-05 ***
## s(free.sulfur.dioxide)      1   2.71   2.711   6.9287  0.008567 **
## s(total.sulfur.dioxide)     1  26.12  26.121  66.7468 6.328e-16 ***
## s(density)                  1  71.26  71.258 182.0868 < 2.2e-16 ***
## s(pH)                       1   2.92   2.915   7.4493  0.006418 **
## s(sulphates)                1  60.58  60.578 154.7958 < 2.2e-16 ***
## s(alcohol)                  1  33.92  33.919  86.6728 < 2.2e-16 ***
## Residuals                1554 608.15   0.391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                         Npar Df  Npar F     Pr(F)
## (Intercept)
## s(fixed.acidity)              3  3.7217  0.011048 *
## s(volatile.acidity)           3  1.9932  0.113083
## s(citric.acid)                3  2.8406  0.036714 *
## s(residual.sugar)             3  1.6058  0.186125
## s(chlorides)                  3  2.3446  0.071258 .
## s(free.sulfur.dioxide)        3  3.0062  0.029351 *
## s(total.sulfur.dioxide)       3  3.2367  0.021460 *
## s(density)                        3  1.9323  0.122401
## s(pH)                             3  0.6279  0.597019
## s(sulphates)                      3 22.9579 1.532e-14 ***
## s(alcohol)                        3  4.8324  0.002369 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# model with linear features
gam.lm_formula <- as.formula(paste0("quality ~ ",
                                    paste0(feature_names,
                                           collapse = "+")))
model.gam.lm <- gam(gam.lm_formula, data=wine_data)

# anova test to compare models
anova(model.gam.lm, model.gam.s, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
## Model 2: quality ~ s(fixed.acidity) + s(volatile.acidity) + s(citric.acid) +
##     s(residual.sugar) + s(chlorides) + s(free.sulfur.dioxide) +
##     s(total.sulfur.dioxide) + s(density) + s(pH) + s(sulphates) +
##     s(alcohol)
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      1587     666.41
## 2      1554     608.15 33   58.263 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANSWER (Problem 1. (b)):
Based on the anova test's result on nonparametric effects of variable-specific smooths, the non-linearity is most significantly evidenced in 'sulphates' and 'alcohol', and also evidenced in 'fixed.acidity', 'citric.acid', 'free.sulfur.dioxide' and 'total.sulfur.dioxide'. The significant non-linearity from 'sulphates' and 'alcohol' could also be seen in the their plots with standard error bands in part 1a - there is an interval on the x-coordinate where the y range value (s(x)) does not contain 0.

The anova Chi test between the GAM model with spline terms and the model with only linear terms gave a very small p value (less than 2.2e-16). This suggests the GAM model with spline terms is statistically more significant than the linear model.

(c) [10 points] We now want to investigate how to produce the best expected wine quality based on the physio-chemical content.

- [109b/121b students only] Based on the additive model fit, how might you **approxi-**

mately optimize the physio-chemical composition to produce the highest expected wine quality? Use the results from part (a) to answer this question. What is the resulting estimated wine quality? *Hint: For the latter part, use the* `predict` *function.*

- [209b students only] Based on the GAM fit, determine **numerically** the optimal combination of physio-chemical features to produce the highest expected wine quality. What are the values of the physio-chemical features corresponding to the optimal wine, and what is the quality rating for the optimized wine? *Hint: The* `preplot` *function applied to a GAM fit produces a list containing as many components as there are smoothed predictors, and each component is a list itself containing the* `x` *and* `y` *values that produce the variable-specific smooths.* It may help that the intercept is the sample mean of the quality scores. Or you can use the values as part of the `predict` function to obtain the estimated value of the optimized wine.

```
# get the x and y values of each variable-specific spline
gam.results <- preplot(model.gam.s)
var_splines <- names(gam.results)
opt_feature_values <- rep(0, length(var_splines))
for (i in 1:length(var_splines)) {
  opt_feature_idx <- which.max(gam.results[[var_splines[i]]]$y)
  opt_feature_values[i] <- gam.results[[var_splines[i]]]$x[opt_feature_idx]
}

# predict wine quality using this optimal feature values set
names(opt_feature_values) <- feature_names
pred_opt_features <- predict(model.gam.s,
                             newdata=data.frame(t(as.matrix(opt_feature_values))))
print("The feature values corresponding to the optimal wine are:")
```

```
## [1] "The feature values corresponding to the optimal wine are:"
```

```
print(opt_feature_values)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##             11.70000              0.12000              1.00000
##       residual.sugar            chlorides  free.sulfur.dioxide
##             10.70000              0.01200             72.00000
## total.sulfur.dioxide              density                   pH
##            289.00000              0.99007              2.74000
##            sulphates              alcohol
##              0.89000             13.20000
```

```
print(paste0("The predicted optimal wine quality score: ",
             pred_opt_features))
```

```
## [1] "The predicted optimal wine quality score: 9.02275231142911"
```

(d) [5 points] What might be a concern or limitation with optimizing the physio-

chemical composition of wine based on the additive model fit? Your answer should be connected to the assumptions underlying additive models. **ANSWER (Problem 1. (d)):**
The limitation with optimizing the physio-chemical composition of wine based on the additive model fit is that we need to eliminate colinearity between predictors (or their smoothing spline terms), which is the assumption underlying additive models.

The concern for the wine data is that predictors can be correlated. For instance, ‘free.sulfur.dioxide‘ and ‘total.sulfur.dioxide‘ are highly correlated, and various types of acidity are definetly correlated with ‘pH‘.

Problem 2 [40 points]

Rather than fit a single model to all of the wines, we will fit different models to different subsets of the data (in Problem 3). In preparation, this problem will involve partitioning the data into different clusters/subsets.

(a) [5 points] Explain a reason we might expect different relationships between quality and physio-chemical wine composition by different subsets of the data identified in Problem 1.

ANSWER (Problem 2. (a)):
Because the wine evalutation scores can be based on other factors than just physio-chemical composition, the same physio-chemical composition of some wine could receive very different evaluations. For instance, people from place A prefer sour wine while people from place B prefer sweet wine. Therefore, we can expect the same type of some sour wine would receive overall higher quality score in place A than in place B.

(b) [5 points] Prior to performing clustering, you will center each column, and also scale each column so that each transformed feature has a standard deviation of 1.0. Briefly justify the decision to scale the data in this manner. Be specific to the context of this problem.

```r
feature.mean.before <- colMeans(as.matrix(wine_data[, feature_names]))
feature.sd.before <- colSds(as.matrix(wine_data[, feature_names]))
print("[Before scaling] Feature Mean: ")
```

```
## [1] "[Before scaling] Feature Mean: "
```

```r
print(feature.mean.before)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##           8.31963727           0.52782051           0.27097561
##       residual.sugar            chlorides  free.sulfur.dioxide
##           2.53880550           0.08746654          15.87492183
## total.sulfur.dioxide              density                   pH
##          46.46779237           0.99674668           3.31111320
##           sulphates             alcohol
##          0.65814884         10.42298311
```

```
print("[Before scaling] Feature Standard Deviation: ")
```

```
## [1] "[Before scaling] Feature Standard Deviation: "
```

```
print(feature.sd.before)
```

```
##  [1]  1.741096318  0.179059704  0.194801137  1.409928060  0.047065302
##  [6] 10.460156970 32.895324478  0.001887334  0.154386465  0.169506980
## [11]  1.065667582
```

```
features.scaled <- scale(wine_data[, feature_names])

print("[After scaling] Feature Mean: ")
```

```
## [1] "[After scaling] Feature Mean: "
```

```
colMeans(features.scaled)
```

```
##       fixed.acidity    volatile.acidity         citric.acid
##        3.547819e-16        1.841085e-16       -9.381176e-17
##      residual.sugar           chlorides free.sulfur.dioxide
##       -1.155094e-16        8.713432e-17       -5.578020e-17
## total.sulfur.dioxide            density                  pH
##        3.901880e-17        2.365659e-14       -2.436207e-17
##           sulphates             alcohol
##        2.016684e-17        8.615685e-17
```

```
print("[After scaling] Feature Standard Deviation: ")
```

```
## [1] "[After scaling] Feature Standard Deviation: "
```

```
colSds(features.scaled)
```

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

ANSWER (Problem 2. (b)):
In general, feature standardization is important to clustering so that the meassure of pair-wise data point similarity is not dominated by differnet feature scales; and it is important to PCA so that the first several principle components explain the total data variance with a fair consideration over all features rather than one or two features that have larger scale than others.

Specific to the context of this problem, before standardization, the standard deviation of ‘free.sulfur.dioxide‘ (10.460157) and ‘total.sulfur.dioxide‘ (32.8953245) are much larger than the other features, which could result in clusters dominated by these 2 features.

(c) [10 points] Suppose we decide to perform partitioning-around-medoids clustering of the observations based only on the physio-chemical features but not using quality. To determine the best number of clusters, optimize based on the gap statistic in the following manner:

1. Set the random number seed to 123 (`set.seed(123)`). Now select a random sample of 200 wines (*hint: use the* `sample` *function*).

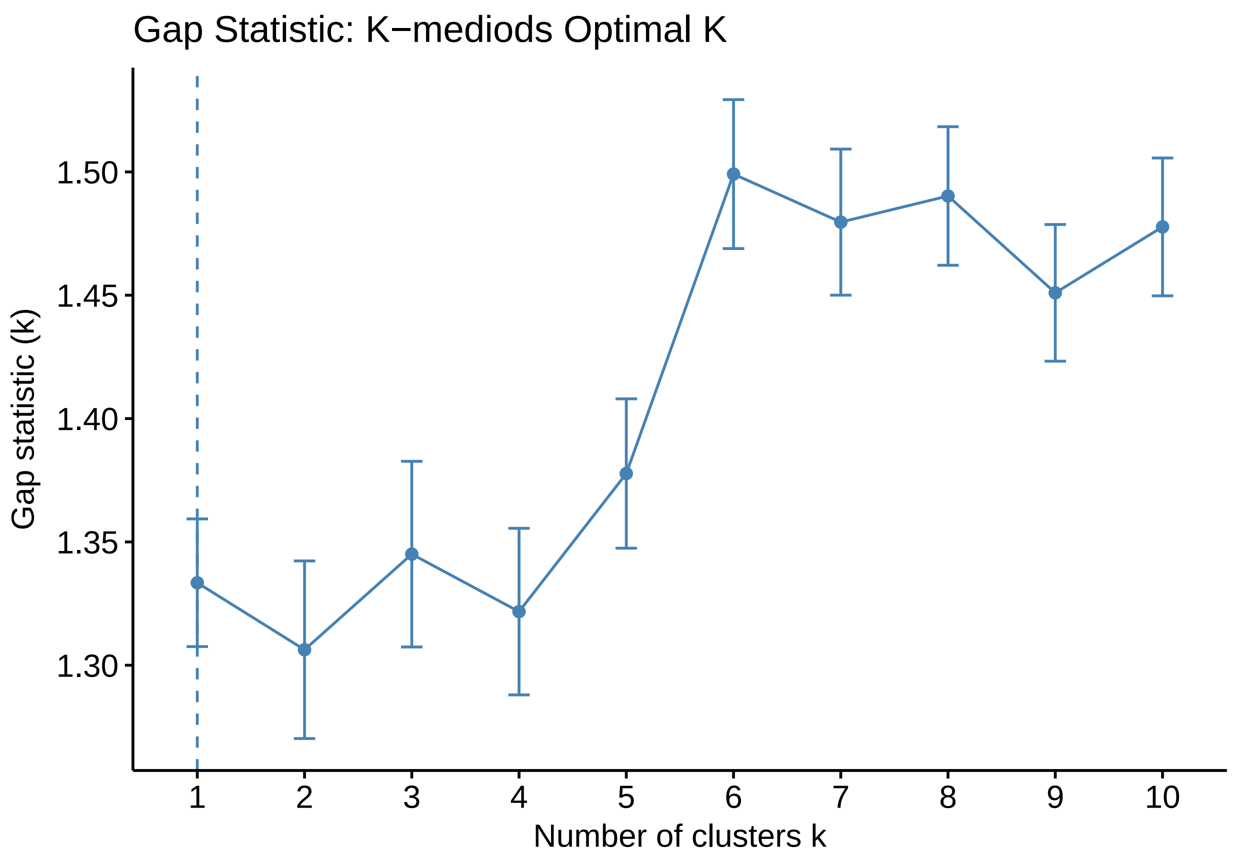
2. Set the random number seed to 321 (`set.seed(321)`). Optimize the gap statistic using the method described by Tibshirani (2001) based on the standard error rule, using `d.power=2`.

   Explain how 1 cluster is the optimal number of clusters according to Tibshirani's rule, even though 6 clusters would be chosen if one were to use the maximum gap statistic.

```
# sample 200 wines
set.seed(123)
samp <- sample(x=1:nrow(features.scaled), size=200, replace=F)
features.scaled.samp <- features.scaled[samp, ]
quality.samp <- wine_data$quality[samp]

# gap stat to choose number of clusters for k-mediods clustering
set.seed(321)
gapstat = clusGap(features.scaled.samp, FUN=pam, d.power=2, K.max=10, B=500)
print(gapstat, method="Tibs2001SEmax")
```

```
## Clustering Gap statistic ["clusGap"] from call:
## clusGap(x = features.scaled.samp, FUNcluster = pam, K.max = 10,    B = 500, d.power = 2)
## B=500 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
##  --> Number of clusters (method 'Tibs2001SEmax', SE.factor=1): 1
##           logW   E.logW      gap     SE.sim
##  [1,] 6.972844 8.306292 1.333448 0.02588282
##  [2,] 6.813394 8.119683 1.306289 0.03599735
##  [3,] 6.626267 7.971310 1.345042 0.03763001
##  [4,] 6.551881 7.873632 1.321751 0.03376585
##  [5,] 6.423769 7.801493 1.377724 0.03027421
##  [6,] 6.244083 7.743181 1.499099 0.03019518
##  [7,] 6.213207 7.692842 1.479635 0.02960796
##  [8,] 6.159896 7.650126 1.490230 0.02808761
##  [9,] 6.159868 7.610831 1.450962 0.02770733
## [10,] 6.099090 7.576785 1.477695 0.02793793
```

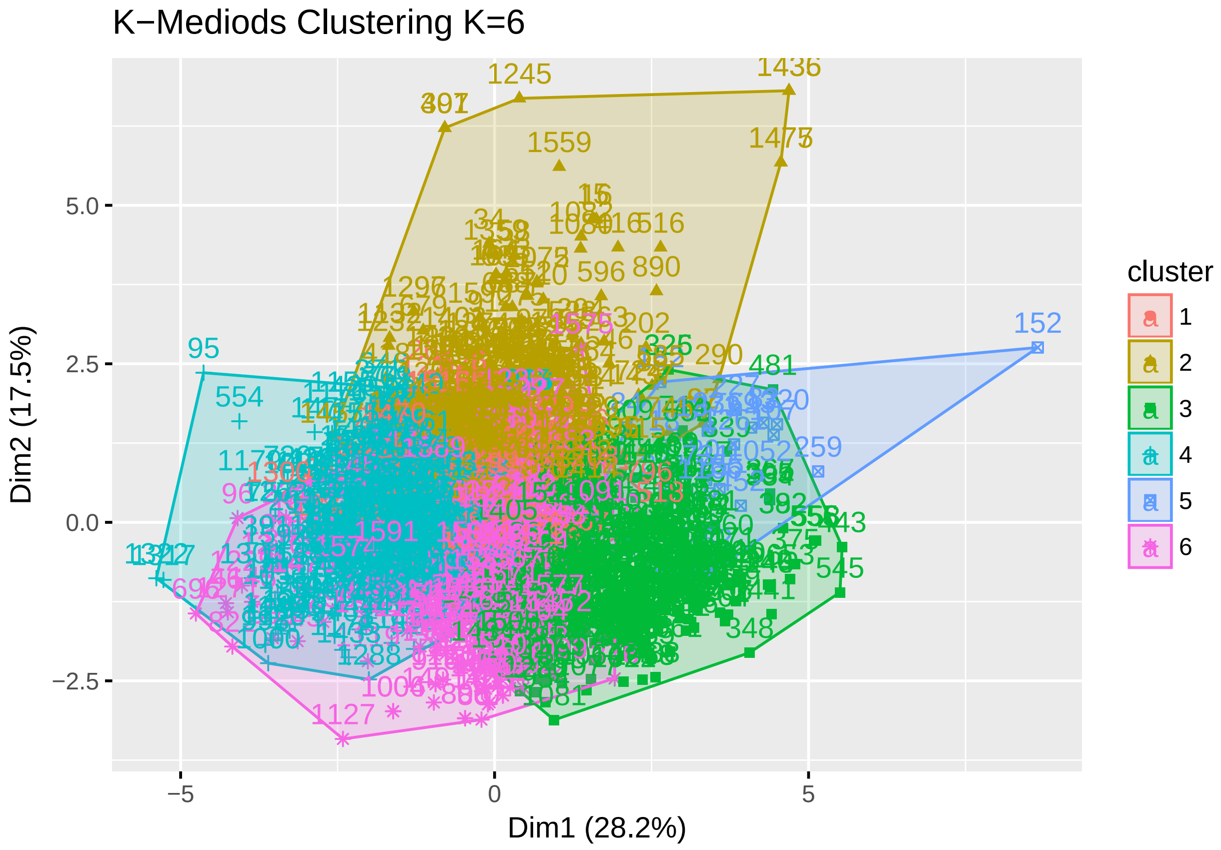
```
fviz_gap_stat(gapstat,
  maxSE=list(method="Tibs2001SEmax", SE.factor=1)) +
  ggtitle("Gap Statistic: K-mediods Optimal K")
```



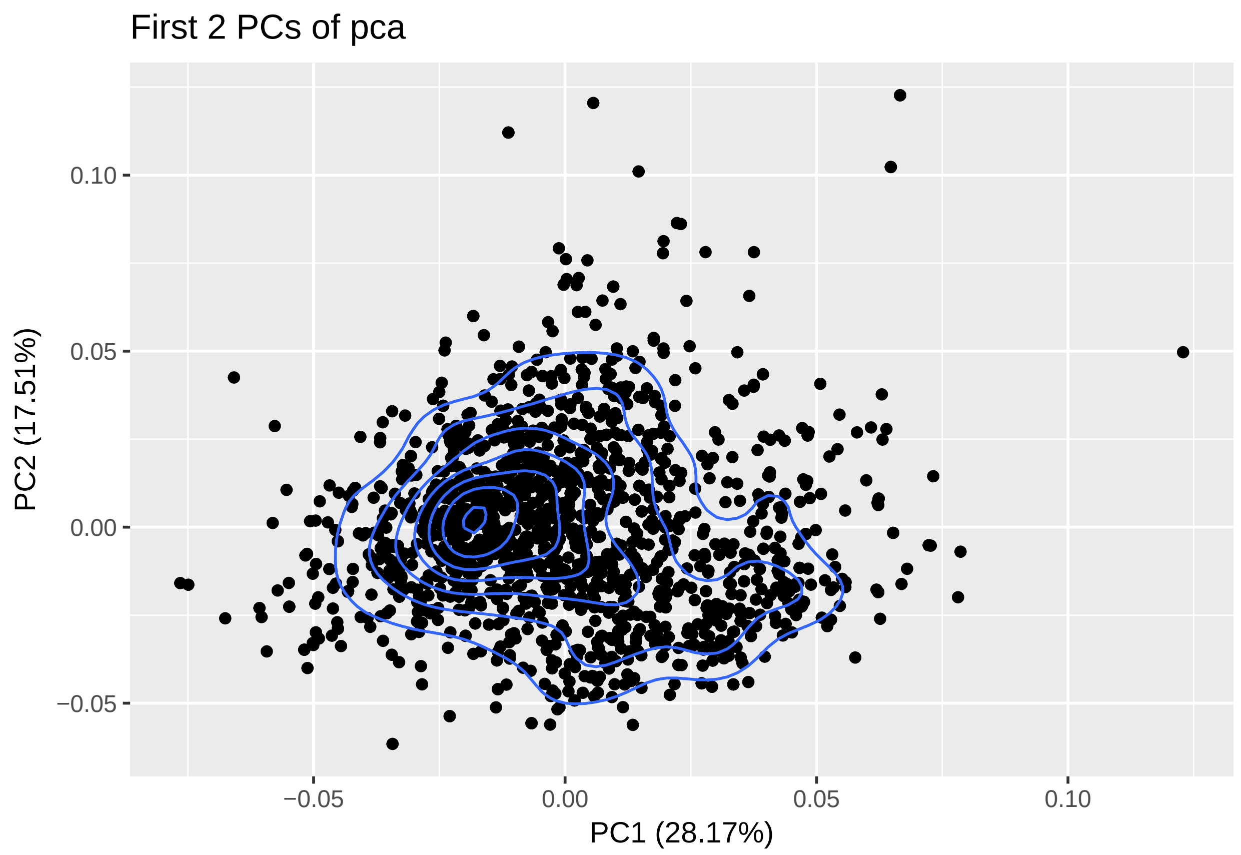
ANSWER (Problem 2. (c)):
The gap statistic measures how significantly better it is to cluster the data into K groups than not clustering at all. According to Tibshirani's rule, the optimal number of clusters is the smallest K such that $Gap(K) \geq Gap(K+1) - s_{K+1}$. Since $Gap(K=1) > Gap(K=2)$, which means 2-cluster segmentation is already not necessary, Tibshirani's rule would select 1 as the optimal number of clusters.

(d) [10 points] Partition the full data into six clusters via partitioning-around-medoids on the scaled version of the data. Save the cluster identifiers as a new column in the original data frame (*Hint: the* `clustering` *component of the resulting cluster object contains the IDs*). Plot the first two principal components of the scaled data and visually show the cluster memberships. Show that the proportion of variance in the original data represented by the principal component plot is 45.7%. Use the output of `prcomp` to demonstrate this.

```
# 5-mediods clustering
results.pam <- pam(features.scaled, k=6)
fviz_cluster(results.pam, data=wine_data, main='K-Mediods Clustering K=6')
```



```
wine_data$clusters <- results.pam$clustering

# PCA
pca <- prcomp(features.scaled, center=T)
autoplot(pca) + geom_density2d() + ggtitle('First 2 PCs of pca')
```



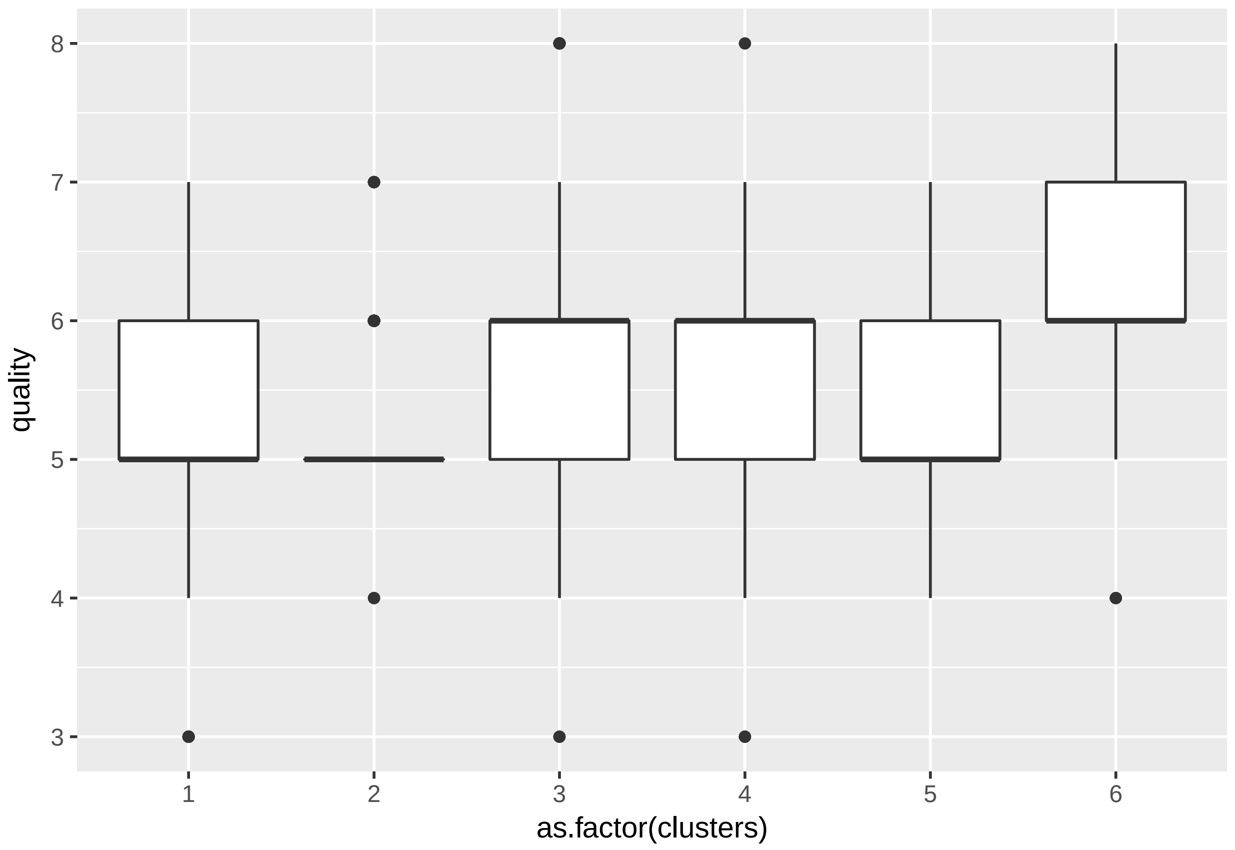
```r
pc.vars <- cumsum(pca$sdev^2/sum(pca$sdev^2))
print(paste0("Variance Explained by the first 2 principal components = ", pc.vars[2]))
```

```
## [1] "Variance Explained by the first 2 principal components = 0.45682201184294"
```

ANSWER (Problem 2. (d)):
Clusters shown in the plot, and the variance Explained by the first 2 principal components is 0.456822 ($\sim$ 0.457).

(e) [10 points] Create a side-by-side boxplot of quality scores by cluster (*Hint: If using* ggplot *you should use the* geom_boxplot *function – do not forget to make the cluster ID variable a factor in R*). Does the distribution of quality scores differ visually by the clusters you determined? Would you have expected the distribution of quality scores to differ?

```r
ggplot(aes(y = quality, x = as.factor(clusters)), data = wine_data) + geom_boxplot()
```



ANSWER (Problem 2. (e)):
As shown in the boxplot, the distribution of quality scores differ by clusters (eg. cluster 2 and 6 are obviously different than the others). This matches our expectation that there could be different relationships between quality and physio-chemical features by different subsets of the data.

Problem 3 [30 points]

We will now fit a normal hierarchical linear model for quality scores against the physio-chemical predictors nested in the formed clusters from the previous problem.

(a) [10 points] Implement a normal hierarchical linear model in Stan (called from R) to fit the model. Make sure you let all the linear model coefficients vary by cluster. *Hint: You may find the Stan code supplied with the lecture notes helpful.* You may assume that the intercepts across the six clusters have a normal prior distribution with a mean which is the average of the quality scores across the whole data set, and with an unknown standard deviation. The 11 physio-chemical coefficients across the six clusters can be assumed to be normally distributed centered at 0 with different standard deviations. Finally, you may assume that all the standard deviation parameters have a prior uniform distribution with a minimum of 0 and maximum of 100 (which is sufficiently large).

```
# create list
stan_list <- list()
stan_list$N <- nrow(wine_data) # number of observations
stan_list$M <- length(feature_names) # number of features
stan_list$J <- 6 # number of cluster
stan_list$clusters <- wine_data$clusters
stan_list$fixed_acidity <- wine_data$fixed.acidity
stan_list$volatile_acidity <- wine_data$volatile.acidity
stan_list$citric_acidity <- wine_data$citric.acid
stan_list$residual_suger <- wine_data$residual.sugar
stan_list$chlorides <- wine_data$chlorides
stan_list$free_sulfur_dioxide <- wine_data$free.sulfur.dioxide
stan_list$total_sulfur_dioxide <- wine_data$total.sulfur.dioxide
stan_list$density <- wine_data$density
stan_list$pH <- wine_data$pH
stan_list$sulphates <- wine_data$sulphates
stan_list$alcohol <- wine_data$alcohol

stan_list$quality <- wine_data$quality
stan_list$mean_quality <- mean(wine_data$quality)

# stan code
stan_code <- c("
data {
  int N; // # observations
  int M; // # features = 11
  int J; // Number of clusters = 6
  int clusters[N];
  real<lower=0> fixed_acidity[N];
  real<lower=0> volatile_acidity[N];
  real<lower=0> citric_acidity[N];
  real<lower=0> residual_suger[N];
  real<lower=0> chlorides[N];
  real<lower=0> free_sulfur_dioxide[N];
  real<lower=0> total_sulfur_dioxide[N];
  real<lower=0> density[N];
  real<lower=0> pH[N];
  real<lower=0> sulphates[N];
  real<lower=0> alcohol[N];
  int<lower=0> quality[N]; // response
  real mean_quality; // mean(quality)
}

parameters {
  real<lower=0> sigma;
  real<lower=0> sigma_0;
  real<lower=0> sigma_m[M];
  real beta_0[J];
  real beta_ij[M, J];
}

model {
  // Prior
  sigma ~ uniform(0, 100);
  sigma_0 ~ uniform(0, 100);
  for (i in 1:M) {
    sigma_m[i] ~ uniform(0, 100);
  }

  for (j in 1:J) {
    beta_0[j] ~ normal(mean_quality, sigma_0);
  }
  for (i in 1:M) {
    for (j in 1:J) {
      beta_ij[i, j] ~ normal(0, sigma_m[i]);
    }
  }

  // Likelihood
  for (n in 1:N) {
    int cluster = clusters[n];
    quality[n] ~ normal(beta_0[cluster] +
    beta_ij[1, cluster] * fixed_acidity[n] +
    beta_ij[2, cluster] * volatile_acidity[n] +
    beta_ij[3, cluster] * citric_acidity[n] +
    beta_ij[4, cluster] * residual_suger[n] +
    beta_ij[5, cluster] * chlorides[n] +
    beta_ij[6, cluster] * free_sulfur_dioxide[n] +
    beta_ij[7, cluster] * total_sulfur_dioxide[n] +
    beta_ij[8, cluster] * density[n] +
    beta_ij[9, cluster] * pH[n] +
    beta_ij[10, cluster] * sulphates[n] +
    beta_ij[11, cluster] * alcohol[n], sigma);
  }
}

generated quantities {
  real quality_rep[N]; // Draws from posterior predictive dist
  for (n in 1:N) {
    int cluster = clusters[n];
    quality_rep[n] = beta_0[cluster] +
    beta_ij[1, cluster] * fixed_acidity[n] +
    beta_ij[2, cluster] * volatile_acidity[n] +
    beta_ij[3, cluster] * citric_acidity[n] +
    beta_ij[4, cluster] * residual_suger[n] +
    beta_ij[5, cluster] * chlorides[n] +
    beta_ij[6, cluster] * free_sulfur_dioxide[n] +
    beta_ij[7, cluster] * total_sulfur_dioxide[n] +
    beta_ij[8, cluster] * density[n] +
    beta_ij[9, cluster] * pH[n] +
    beta_ij[10, cluster] * sulphates[n] +
    beta_ij[11, cluster] * alcohol[n];
  }
}

")
```

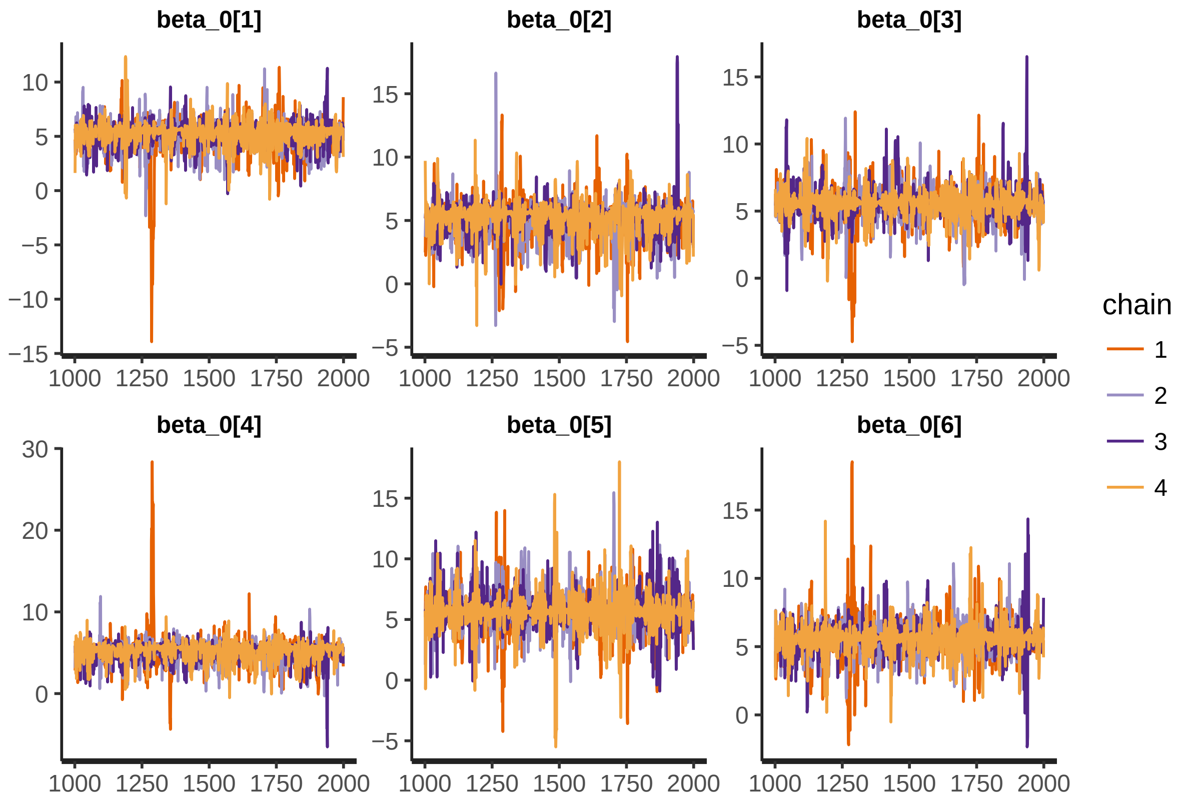
```
options(mc.cores = parallel::detectCores())
wine.fit <- stan(model_code = stan_code,
            data = stan_list,
            iter = 2000,
            chains = 4,
            seed = 46,
            refresh = FALSE)
```

```
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/i
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/i
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/include/boost/config/c
## #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##            ^
## <command line>:6:9: note: previous definition is here
## #define BOOST_NO_CXX11_RVALUE_REFERENCES 1
##          ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                             ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                             ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                             ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                              ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                              ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                              ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                              ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                              ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                              ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                              ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                               ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                               ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Rcpp
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/RcppEigen/include/Eigen/s
##     #pragma clang diagnostic pop
##                               ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/StanHeaders/include/stan/
##     static void set_zero_all_adjoints() {
##                 ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/StanHeaders/include/stan/
##     static void set_zero_all_adjoints_nested() {
##                 ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/StanHeaders/include/stan/
##       size_t fft_next_good_size(size_t N) {
##               ^
## In file included from file13d94174c9ce.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/Stan
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/i
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/i
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/i
## In file included from /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/i
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/include/boost/multi_ar
##       typedef typename Array::index_range index_range;
##                                           ^
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/include/boost/multi_ar
##       typedef typename Array::index index;
##                                     ^
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/include/boost/multi_ar
##       typedef typename Array::index_range index_range;
##                                           ^
## /Library/Frameworks/R.framework/Versions/3.4/Resources/library/BH/include/boost/multi_ar
##       typedef typename Array::index index;
##                                     ^
## 21 warnings generated.

## Warning: There were 52 divergent transitions after warmup. Increasing adapt_delta above 0.8
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: There were 42 transitions after warmup that exceeded the maximum treedepth. Increas
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded

## Warning: Examine the pairs() plot to diagnose sampling problems
```

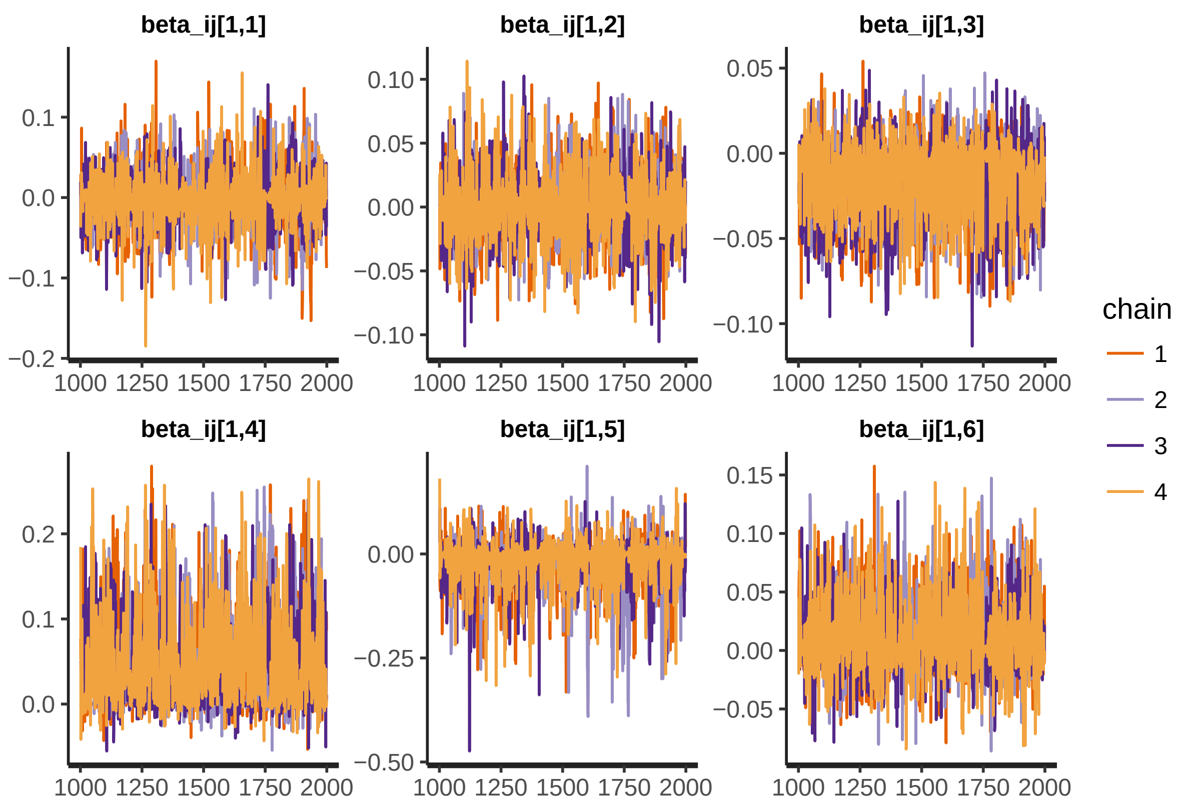
(b) [10 points] Briefly report on the details of your model implementation (number of iterations of burn-in and the number of iterations of saved parameters, number of parallel samplers, and any assurances that the sampler converged). (*Hint: If you saved the Stan fit of your model in the R object* `wine.fit`, *you can access the matrix of model summaries from* `summary(wine.fit)$summary`.) Do not be concerned about warnings of divergent transitions after warm-up if you have evidence that the sampler converged for the feature coefficients.

```
p1 <- plot(wine.fit, plotfun="trace",
     pars=c('beta_0[1]', 'beta_0[2]',
            'beta_0[3]', 'beta_0[4]',
            'beta_0[5]', 'beta_0[6]'))
```

```r
print(p1)
```



```r
p2 <- plot(wine.fit, plotfun="trace",
     pars=c('beta_ij[1,1]', 'beta_ij[1,2]',
            'beta_ij[1,3]', 'beta_ij[1,4]',
            'beta_ij[1,5]', 'beta_ij[1,6]'))
print(p2)
```



```r
head(summary(wine.fit)$summary[,"Rhat"], 20)
```

```
##        sigma      sigma_0   sigma_m[1]   sigma_m[2]   sigma_m[3]
##    1.0000458    1.0024903    1.0082819    1.0020913    1.0037682
##   sigma_m[4]   sigma_m[5]   sigma_m[6]   sigma_m[7]   sigma_m[8]
##    1.0042579    1.0051272    1.0035796    1.0005971    1.0033903
##   sigma_m[9]  sigma_m[10]  sigma_m[11]    beta_0[1]    beta_0[2]
##    0.9997120    1.0008188    1.0005634    1.0052546    1.0032561
##    beta_0[3]    beta_0[4]    beta_0[5]    beta_0[6] beta_ij[1,1]
##    1.0028410    1.0015559    1.0027442    1.0002846    0.9998023
```

ANSWER (Problem 3. (b)):

Prior distribution:

$\sigma \sim \text{Uniform}(0, 100)$
$\sigma_0 \sim \text{Uniform}(0, 100)$
$\sigma_j \sim \text{Uniform}(0, 100)$
$\beta_{0j} \sim N(mean(quality), \sigma_0)$, with $\sigma_0 \sim \text{Uniform}(0, 100)$
$\beta_{mj} \sim N(0, \sigma_j)$, with $\sigma_j \sim \text{Uniform}(0, 100)$

Model for data:

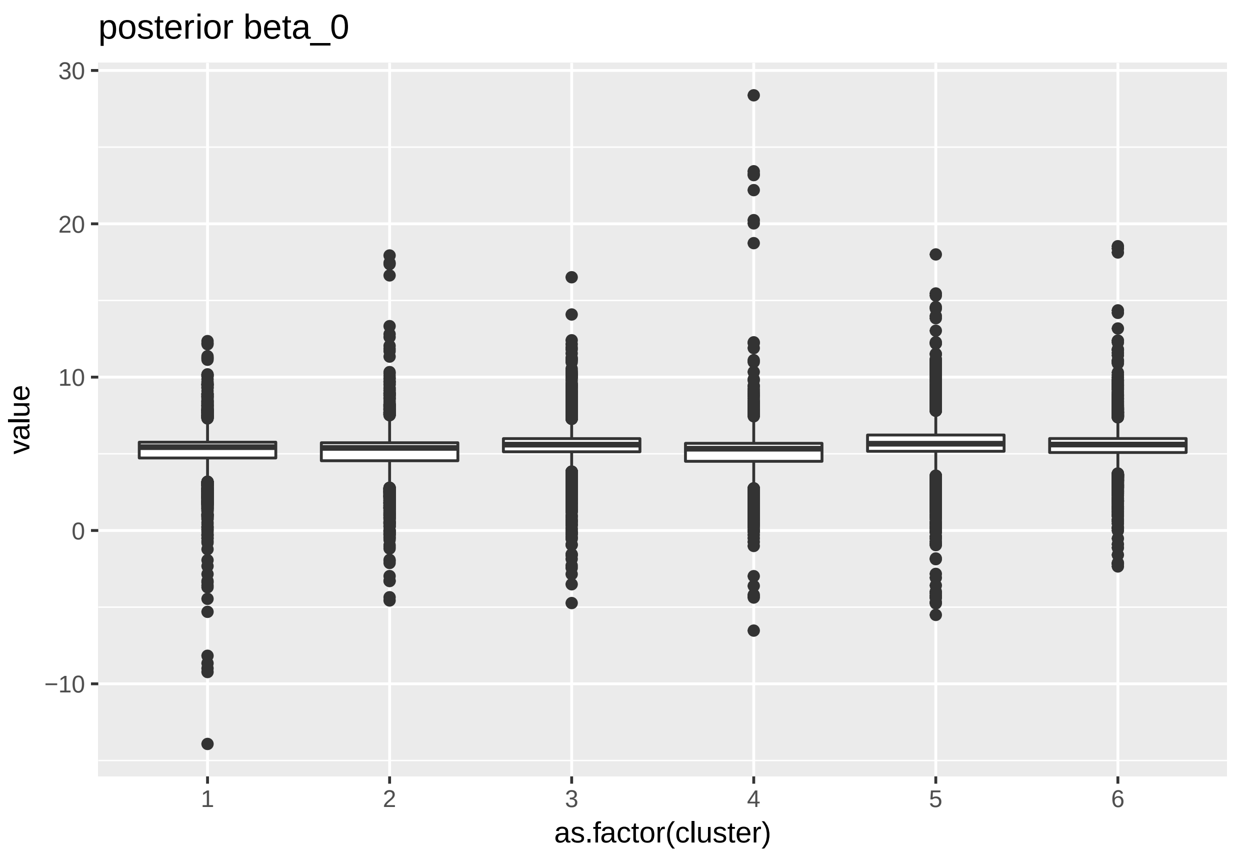
$quality_{nj} \sim N(\beta_{0j} + \sum_{m=1}^{11} \beta_{mj} feature_m, \sigma)$ with $\sigma \sim \text{Uniform}(0, 100)$,

$n = 1, \ldots, N$, the number of observations in the data
$m = 1, \ldots, M$, the number of features (11)
$j = 1, \ldots, J$, the number of clusters (6)

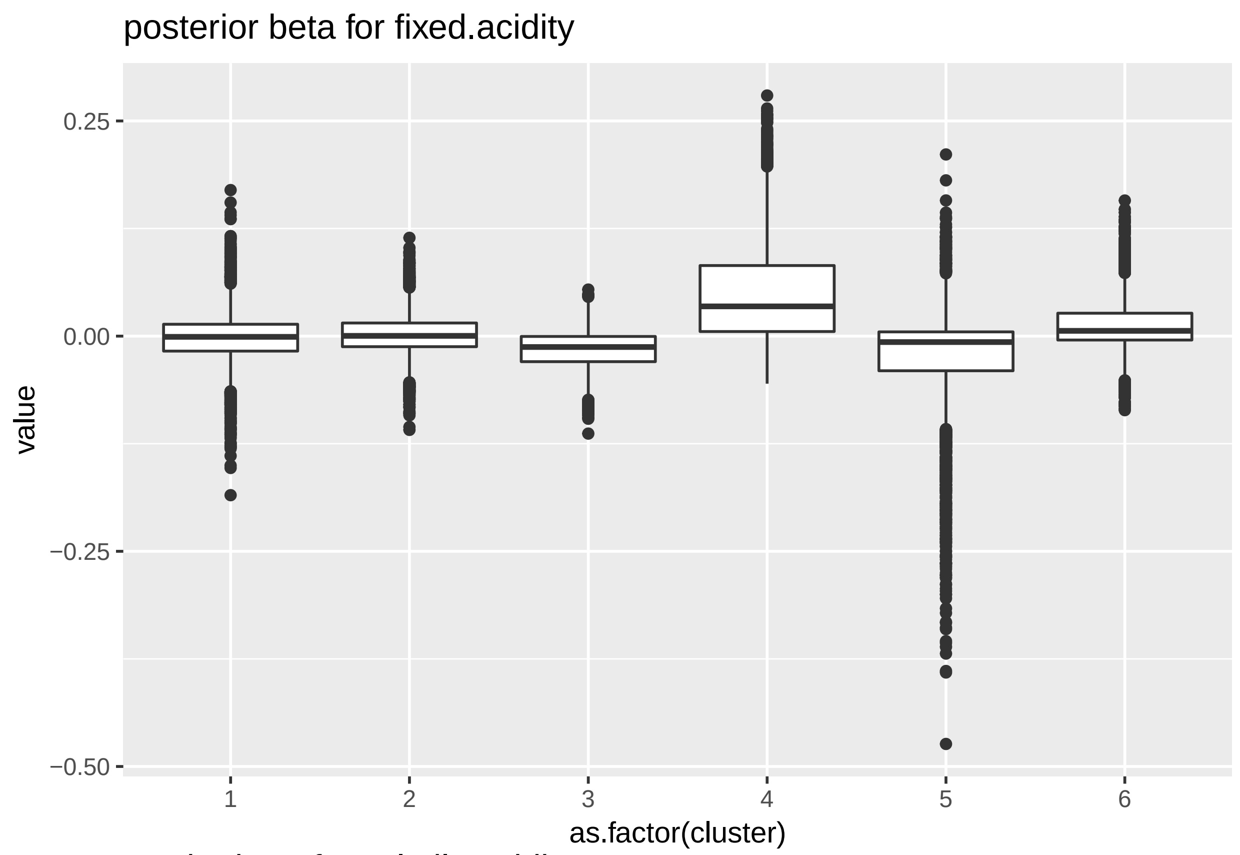
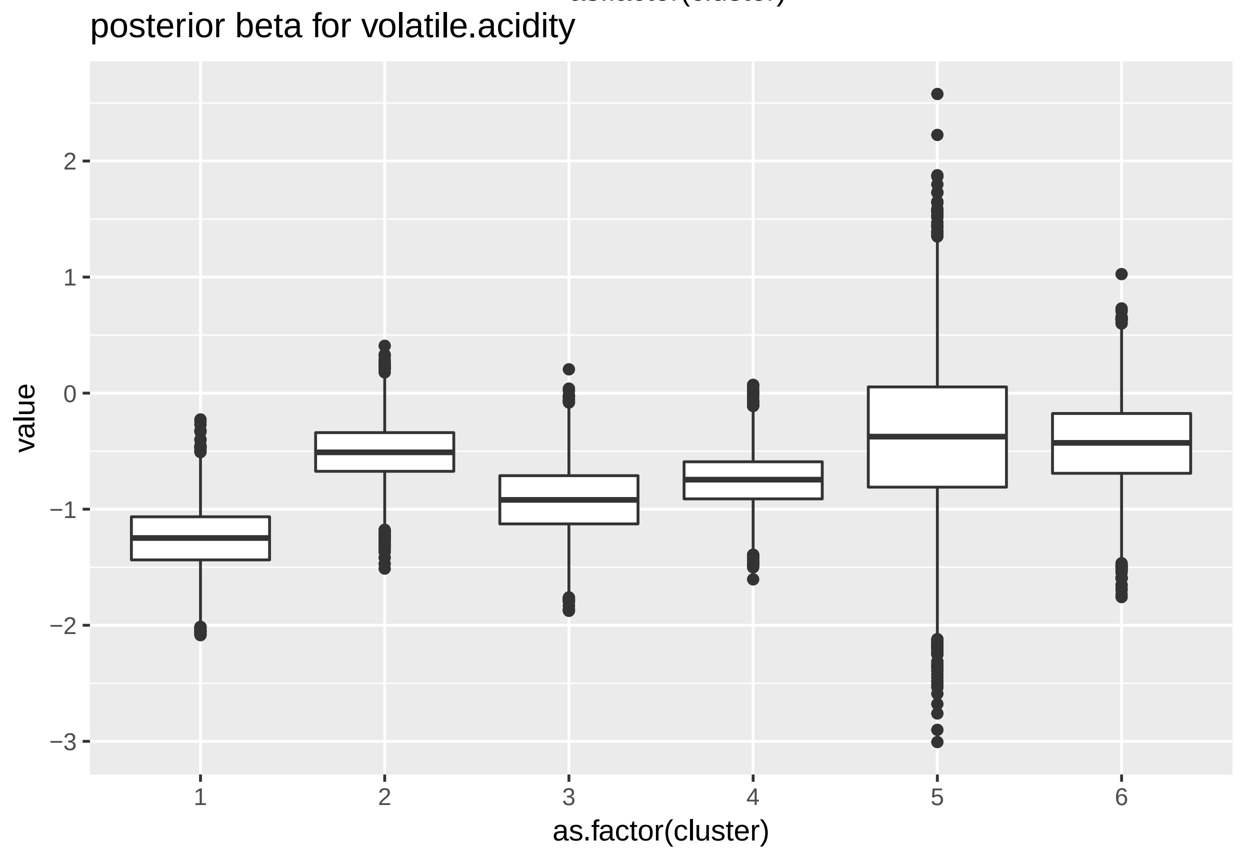
Number of total iterations: 2000.
Number warm up iterations: (default) floor(iteration/2) = 1000.
Number of parallel samplers: determined by number of cores
Assurances that the sampler converged: 1) The trace plots of β_0 and some of the coefficients by cluster show convergence. 2) The R-hat statistics (the degree of convergence of a random Markov Chain) of each parameter's posterior draws are all smaller than 1.1, which also indicates convergence.

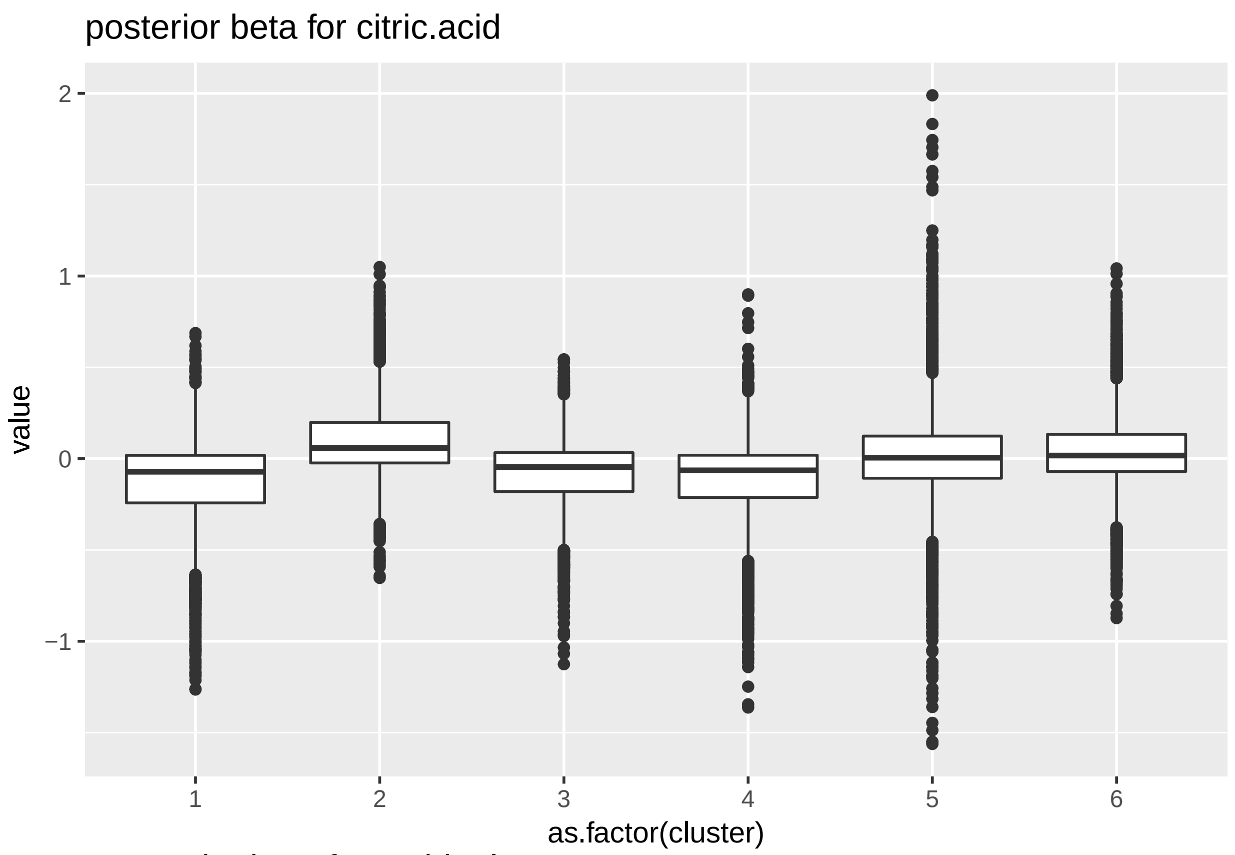
(c) [10 points] Create a visualization that demonstrates the variation of coefficients across clusters. One natural way would be to display side-by-side boxplots of the posterior simulated draws for the relevant coefficients. (*Hint: Use the* `extract` *function applied to the fitted Stan model to obtain simulated coefficient values.*) Based on these results, do you think that the hierarchical model by formed clusters was helpful in explaining the variation in quality scores? Briefly justify.

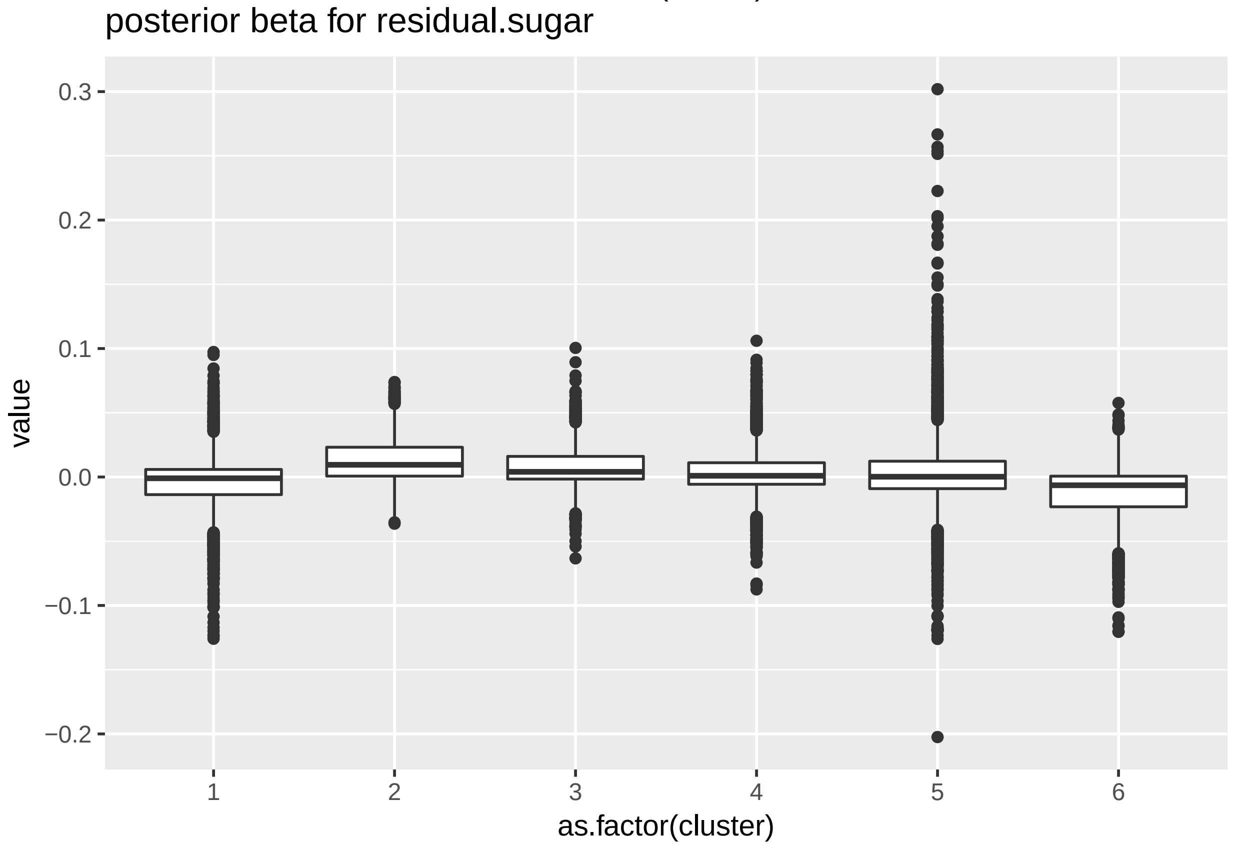
```r
wine.fit.draws <- extract(wine.fit)
beta_0.draws <- wine.fit.draws$beta_0
beta_ij.draws <- wine.fit.draws$beta_ij

nrows <- nrow(beta_ij.draws)

beta_0.draws.matrix <- matrix(rep(0, nrows*6*2), ncol = 2)
for (i in 1:6) {
  start_row <- (i-1)*nrows + 1
  end_row <- i*nrows
  beta_0.draws.matrix[start_row:end_row, 1] <- beta_0.draws[, i] # values
  beta_0.draws.matrix[start_row:end_row, 2] <- rep(i, nrows) # cluster label
}
df.beta_0.draws.matrix = as.data.frame(beta_0.draws.matrix)
names(df.beta_0.draws.matrix) <- c("value", "cluster")
beta_0_plot <- ggplot(aes(y = value, x = as.factor(cluster)),
                      data = df.beta_0.draws.matrix) +
  geom_boxplot() +
  ggtitle("posterior beta_0")
print(beta_0_plot)
```



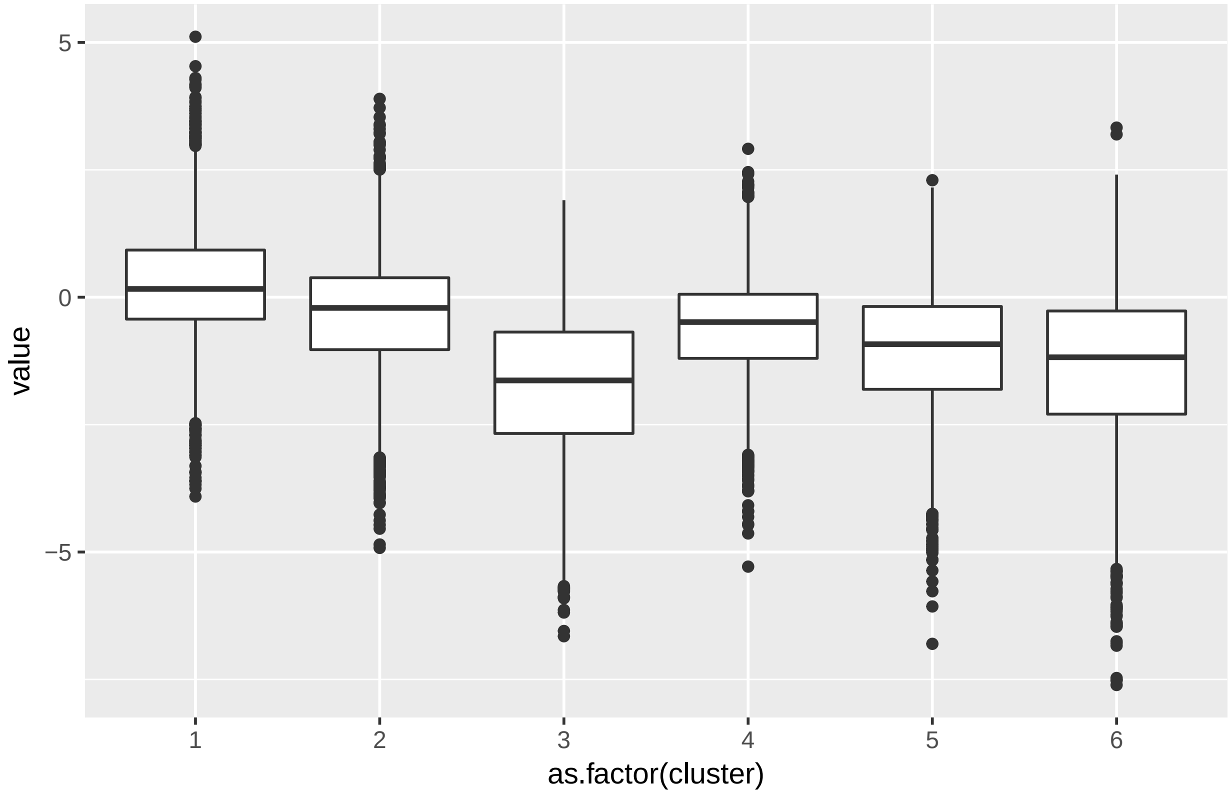
```r
for (m in 1:length(feature_names)) {
  beta_m.draws.matrix <- matrix(rep(0, nrows*6*2), ncol = 2)
  for (i in 1:6) {
    start_row <- (i-1)*nrows + 1
    end_row <- i*nrows
    beta_m.draws.matrix[start_row:end_row, 1] <- beta_ij.draws[, m, i]
    beta_m.draws.matrix[start_row:end_row, 2] <- rep(i, nrows)
  }
  df.beta_m.draws.matrix = as.data.frame(beta_m.draws.matrix)
  names(df.beta_m.draws.matrix) <- c("value", "cluster")

  beta_m_plot <- ggplot(aes(y = value, x = as.factor(cluster)),
                        data = df.beta_m.draws.matrix) +
    geom_boxplot() +
    ggtitle(paste0("posterior beta for ", feature_names[m]))
  print(beta_m_plot)
}
```


posterior beta for fixed.acidity
value
0.25
0.00
−0.25
−0.50
1
2
3
4
5
6
as.factor(cluster)

posterior beta for volatile.acidity
value
2
1
0
−1
−2
−3
1
2
3
4
5
6
as.factor(cluster)

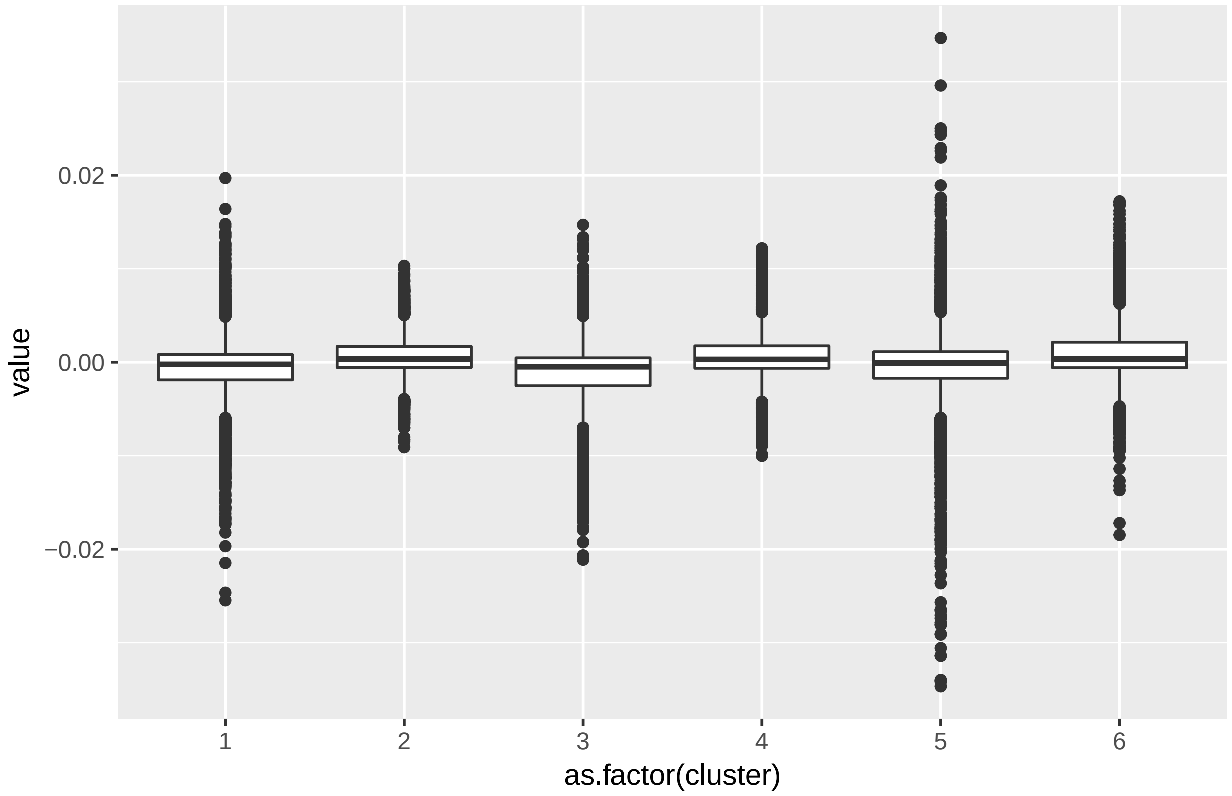

posterior beta for citric.acid
value
2
1
0
−1
1
2
3
4
5
6
as.factor(cluster)

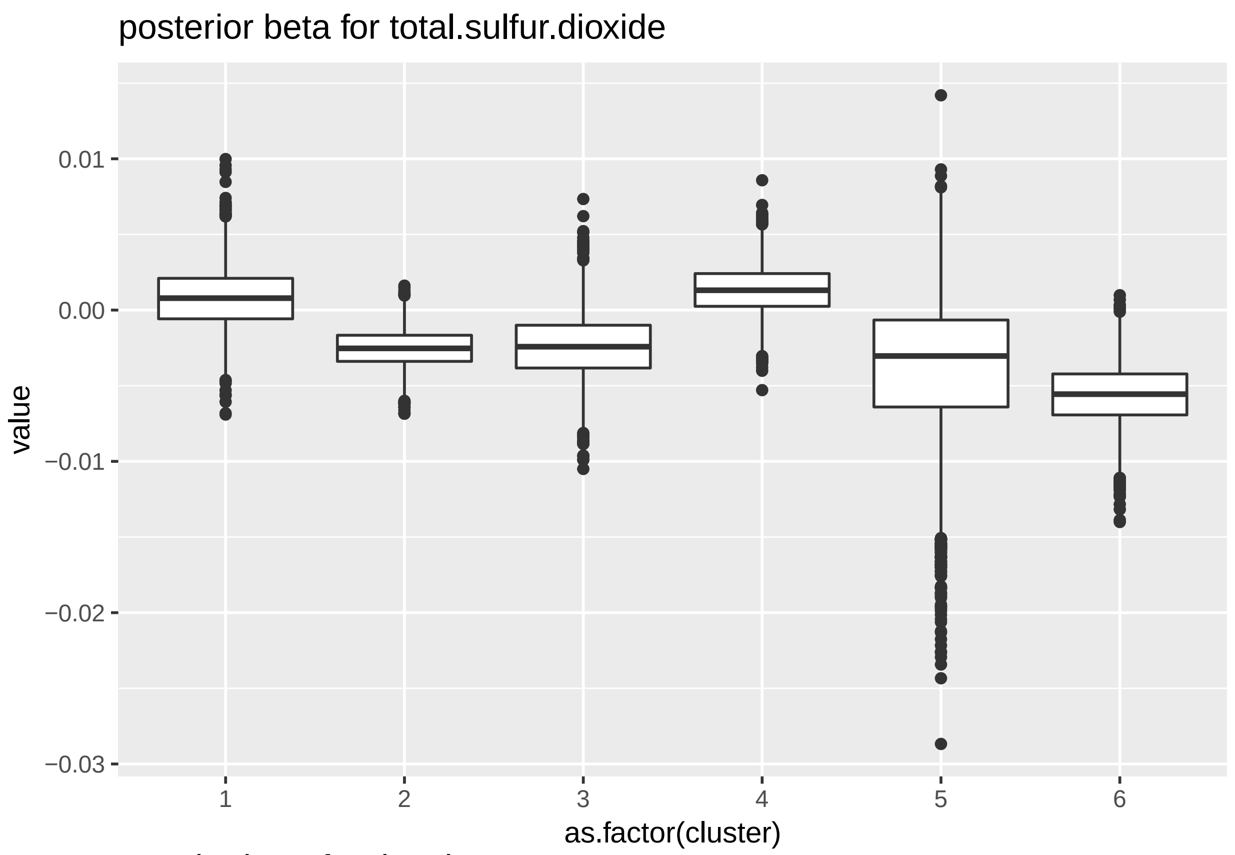

posterior beta for residual.sugar
value
0.3
0.2
0.1
0.0
−0.1
−0.2
1
2
3
4
5
6
as.factor(cluster)

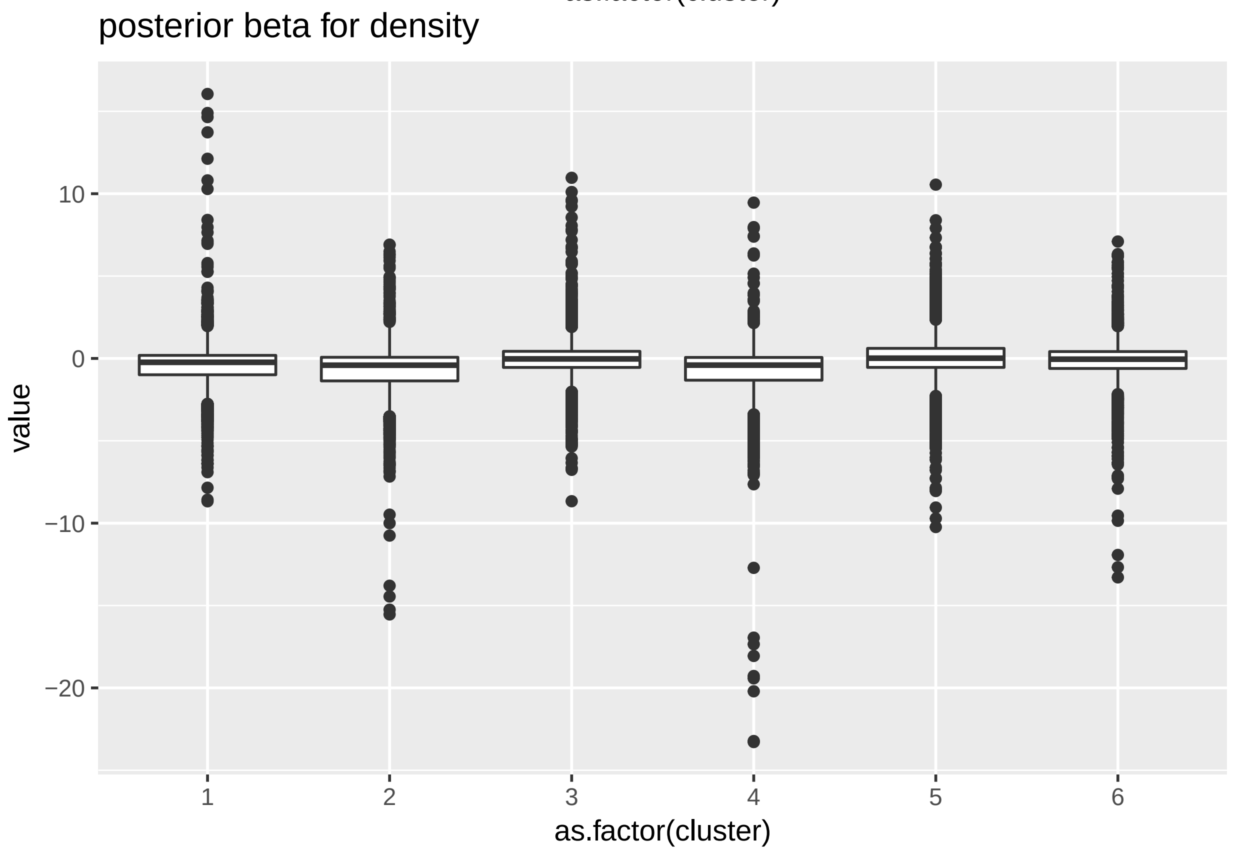
posterior beta for chlorides



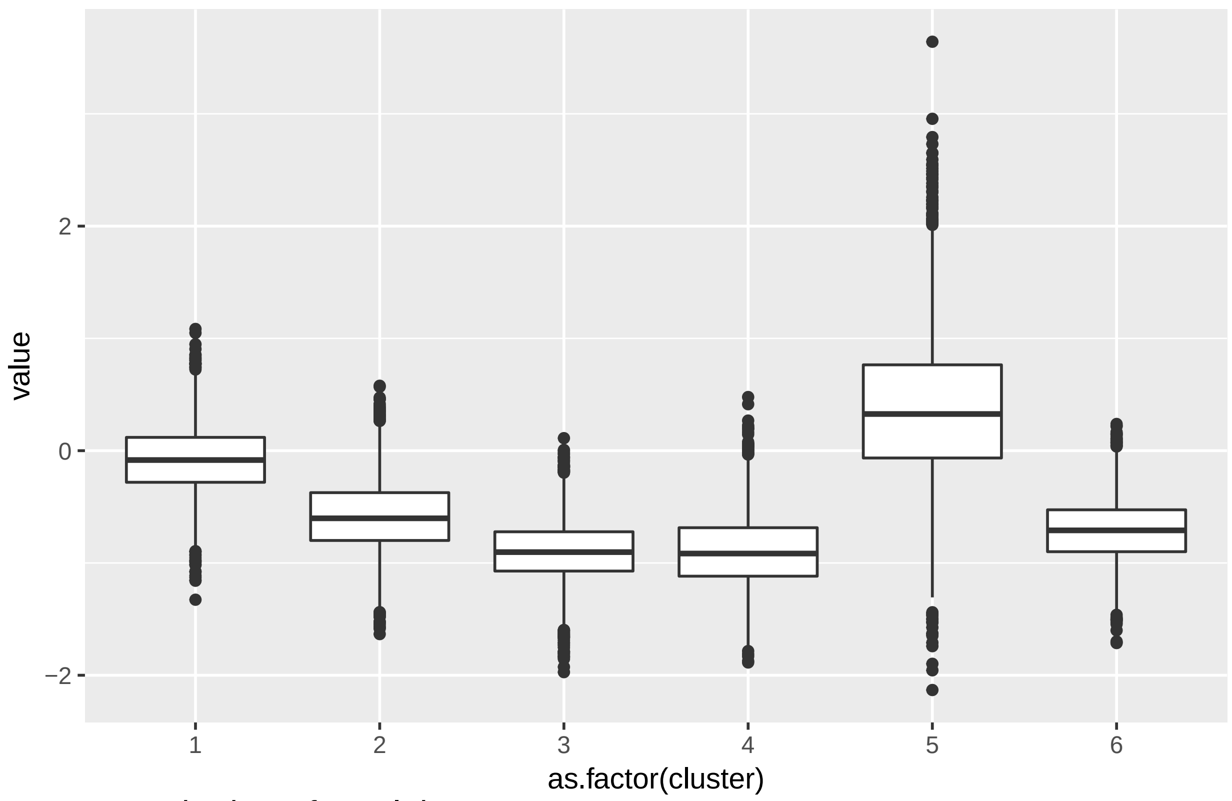
posterior beta for free.sulfur.dioxide



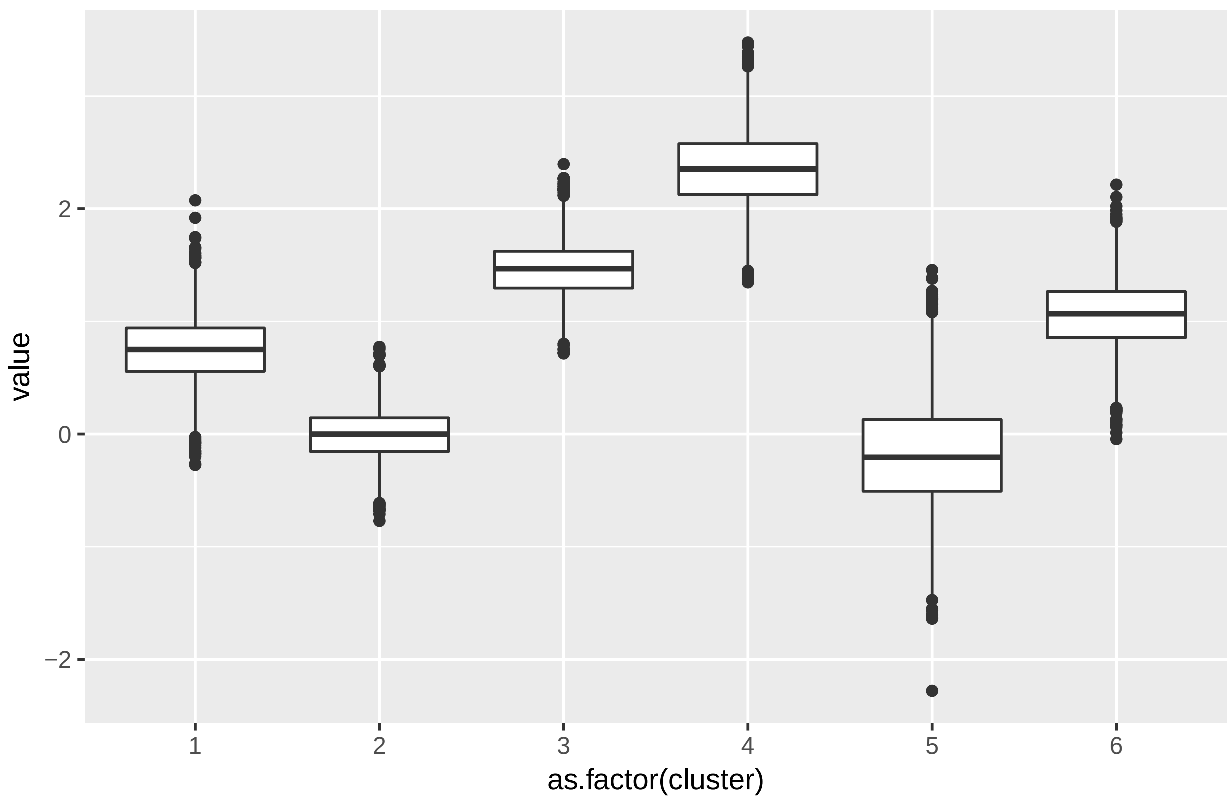

posterior beta for total.sulfur.dioxide
value
0.01
0.00
−0.01
−0.02
−0.03
1
2
3
4
5
6
as.factor(cluster)

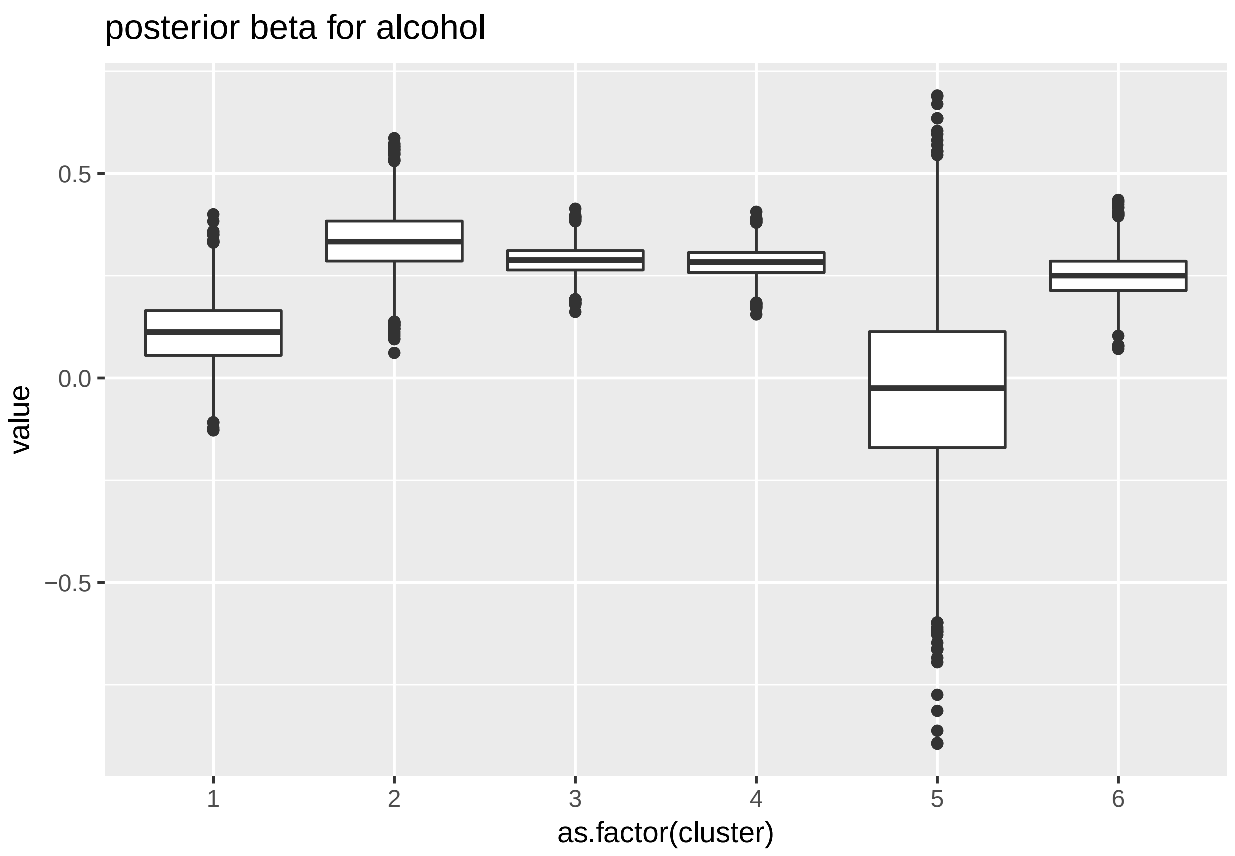

posterior beta for density
value
10
0
−10
−20
1
2
3
4
5
6
as.factor(cluster)

posterior beta for pH



posterior beta for sulphates





ANSWER (Problem 3. (c)):
Based on the side-by-side boxplots of the posterior simulated draws, the coefficients for ‘fixed.acidity‘, ‘volatile.acidity‘ ‘chlorides‘, ‘total.sulfur.dioxide‘, ‘pH‘, ‘sulphates‘ and ‘alcohol‘ do vary by clusters. Therefore, I think the hierarchical model differenciating coefficients by formed clusters is helpful in explaining the variation in quality scores.