R Notebook

This notebook runs through some examples to emphasize important material for Exam 1. The notebook uses a wine data set that was edited from the wine data sets on the UCI repository.

Load the Data

First we read in the data and take a look.

```
wine <- read.csv("wine_all.csv", header=TRUE)
str(wine)
```

```
## 'data.frame':    6497 obs. of  13 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 ...
##  $ type                : Factor w/ 2 levels "red","white": 1 1 1 1 1 1 1 1 1 1 ...
```

- Which columns are quantitative?
- Which columns are qualitative?
- What is a factor?

Divide data into train and test

- Why do we set a seed?
- What does sample do?
- How did we subset the data?
- Why do we divide into train and test sets?

```
set.seed(1234)
i <- sample(1:nrow(wine), 0.8*nrow(wine), replace=FALSE)
train <- wine[i,]
test <- wine[-i,]
```

Build a linear regression model

We will try to predict quality from all other factors.

- Explain the lm() function
- Explain the output of summary including:
- What is the formula?
- Is the intercept considered a "predictor"? Why or why not?
- What are residuals?

- Which predictors seem good, and why?
- What is RSE?
- What is R-squared?
- What is the F-statistic?
- Are these metrics for the train or test set?
- What is a dummy variable? Do you see one below?

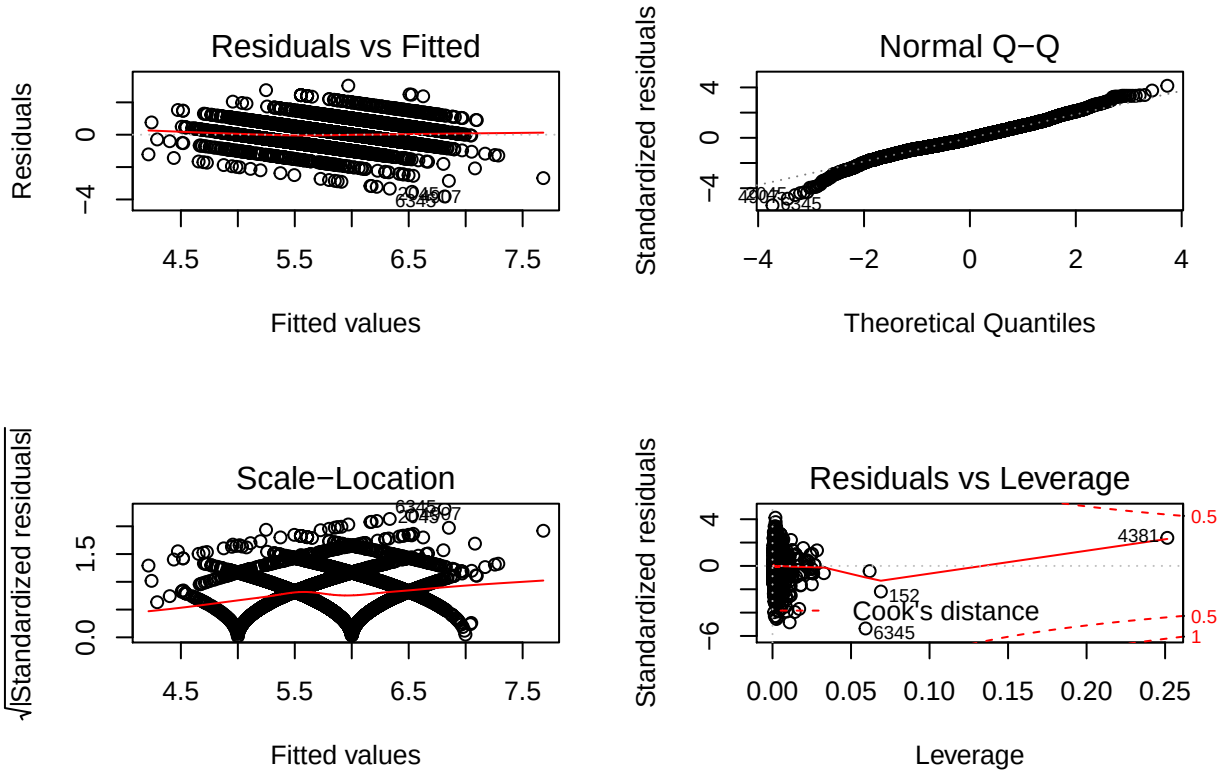
```
lm1 <- lm(quality~., data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = quality ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8153 -0.4695 -0.0387  0.4550  3.0302
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           9.907e+01  1.543e+01   6.422 1.46e-10 ***
## fixed_acidity         7.889e-02  1.736e-02   4.544 5.65e-06 ***
## volatile_acidity     -1.441e+00  9.091e-02 -15.853  < 2e-16 ***
## citric_acid          -4.947e-02  8.899e-02  -0.556 0.578287
## residual_sugar        6.084e-02  6.569e-03   9.263  < 2e-16 ***
## chlorides            -7.755e-01  3.671e-01  -2.113 0.034690 *
## free_sulfur_dioxide   4.919e-03  8.573e-04   5.737 1.02e-08 ***
## total_sulfur_dioxide -1.223e-03  3.626e-04  -3.373 0.000749 ***
## density              -9.812e+01  1.565e+01  -6.270 3.90e-10 ***
## pH                    4.559e-01  1.007e-01   4.529 6.06e-06 ***
## sulphates             7.061e-01  8.403e-02   8.402  < 2e-16 ***
## alcohol               2.305e-01  1.974e-02  11.677  < 2e-16 ***
## typewhite            -3.690e-01  6.264e-02  -5.891 4.08e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7341 on 5184 degrees of freedom
## Multiple R-squared:  0.2908, Adjusted R-squared:  0.2891
## F-statistic: 177.1 on 12 and 5184 DF,  p-value: < 2.2e-16
```

Plot the Residuals

- What do we hope to see?
- What are outliers?
- What are leverage points?

```
par(mfrow=c(2,2))
plot(lm1)
```



Evaluate on the Test Data

- What does pred contain?
- What is correlation? Did you get a good correlation?
- What is mse?
- How do we compute mse for the test set? For the train set?
- Why might rmse be easier to interpret?
- Look at some predictions vs. actual values.

```
pred <- predict(lm1, newdata=test)
cor_lm <- cor(pred, test$quality)
mse_lm <- mean((pred-test$quality)^2)
mse_train <- mean(lm1$residuals^2)
rmse_lm <- sqrt(mse_lm)
print(cbind(head(pred, n=10), head(test$quality, n=10)))
```

```
##        [,1] [,2]
## 1  4.975544    5
## 17 6.035371    7
## 26 5.432663    5
## 28 5.576527    5
## 32 5.400780    6
## 33 5.172980    5
## 34 5.484764    6
## 36 5.198096    6
## 38 5.672396    7
## 40 5.857307    5
```

Build another Linear Regression model

- What is anova?
- How do we interpret the results?
- If a linear regression model has predictors with low p-values, should we take them out? Why or why not?

```
lm2 <- lm(quality~volatile_acidity+residual_sugar+alcohol+sulphates+type+type*alcohol, data=train)
anova(lm1, lm2)
```

```
## Analysis of Variance Table
##
## Model 1: quality ~ fixed_acidity + volatile_acidity + citric_acid + residual_sugar +
##     chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     density + pH + sulphates + alcohol + type
## Model 2: quality ~ volatile_acidity + residual_sugar + alcohol + sulphates +
##     type + type * alcohol
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   5184 2794.0
## 2   5190 2840.6 -6   -46.584 14.405 2.307e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Linear Regression

- supervised or unsupervised?
- classification or regression?
- parametric or non-parametric?
- bias (strong assumptions about the shape of the data) and variance?
- what are some assumptions of the linear model?
- what is a confounding variable? How can we detect them?
- how do we model interaction effects?
- can a linear model tell us about correlation or causation or both between x and y?
- does a linear model have to be a straight line? Why or why not?

Build a Logistic Regression Model

We want to predict red/white given all other predictors.

- Why do we need "family=binomial"
- What is a glm?
- Can you interpret the coefficients in logistic regression the same as we did for linear regression?
- Explain null deviance v. residual deviance.
- What does AIC tell us?

```
glm1 <- glm(type~., data=train, family=binomial)
summary(glm1)
```

```
##
## Call:
## glm(formula = type ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -5.5961   0.0007   0.0167   0.0579   6.3018
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.875e+03  2.098e+02   8.936  < 2e-16 ***
## fixed_acidity         3.311e-01  2.560e-01   1.293 0.195968
## volatile_acidity     -6.320e+00  1.105e+00  -5.721 1.06e-08 ***
## citric_acid           3.197e+00  1.306e+00   2.449 0.014326 *
## residual_sugar        1.012e+00  1.138e-01   8.894  < 2e-16 ***
## chlorides            -2.440e+01  4.846e+00  -5.035 4.79e-07 ***
## free_sulfur_dioxide  -6.144e-02  1.594e-02  -3.855 0.000116 ***
## total_sulfur_dioxide  5.085e-02  5.403e-03   9.412  < 2e-16 ***
## density              -1.872e+03  2.138e+02  -8.758  < 2e-16 ***
## pH                    2.149e+00  1.566e+00   1.372 0.170085
## sulphates            -2.476e+00  1.351e+00  -1.833 0.066825 .
## alcohol              -2.016e+00  3.150e-01  -6.398 1.57e-10 ***
## quality              -3.144e-01  2.187e-01  -1.438 0.150501
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5811.16  on 5196  degrees of freedom
## Residual deviance:  354.38  on 5184  degrees of freedom
## AIC: 380.38
##
## Number of Fisher Scoring iterations: 9
```

Check the Train and Test

- What is this code doing, and why?

```
summary(train$type)
```

```
##   red white
##  1284  3913
```

```
summary(test$type)
```

```
##   red white
##   315   985
```

Evaluate on the Test Data

- What does probs look like?
- What would it look like without type="response"?
- How do we translate probabilities into predictions?
- How do we know what integers 'type' was coded in?
- What does the table show us?
- Why do we use accuracy as a metric instead of cor or mse?

```
probs <- predict(glm1, newdata=test, type="response")
pred_glm <- ifelse(probs>0.5, 2, 1)
table(pred_glm, test$type)
```

```
##
## pred_glm red white
##        1 311     3
##        2   4   982
```

```
acc_glm <- mean(pred_glm==as.integer(test$type))
```

Logistic Regression

- supervised or unsupervised?
- classification or regression?
- parametric or non-parametric?
- bias and variance?
- are the coefficients computed directly or by optimization methods?

kNN

First try kNN classification with unscaled data.

- How did this compare to logistic regression?
- What does k mean in kNN?

kNN Classification

```
library(class)
knn_pred <- knn(train=train[,1:12], test=test[,1:12], cl=train$type, k=3)
table(knn_pred, test$type)
```

```
##
## knn_pred red white
##    red   272    32
##    white  43   953
```

```
acc_knn <- mean(knn_pred==test$type)
```

Now scale the data and try again.

- Compare to the previous results.
- Should we try different values of k? Why or why not?

```
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
train_norm <- as.data.frame(lapply(train[,1:12], normalize))
test_norm <- as.data.frame(lapply(test[,1:12], normalize))
knn_pred2 <- knn(train=train_norm, test=test_norm, cl=train$type, k=3)
table(knn_pred2, test$type)
```

```
##
## knn_pred2 red white
##     red   311    22
##     white   4   963
```

```
acc_knn_scaled <- mean(knn_pred2==test$type)
```

kNN Regression

- Why did we leave out type?

```
train_reg <- train_norm[,1:11]
test_reg <- test_norm[,1:11]
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.3
```

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

```
## Loading required package: ggplot2
```

```
fit <- knnreg(train_reg, train$quality, k=3)
predictions <- predict(fit, test_reg)
cor_knn <- cor(predictions, test$quality)
mse_knn <- mean((predictions - test$quality)^2)
```

try different values of k

- What is the best k?
- How does it compare to the other models?

```
test_mse_knn <- rep(0, 40)
test_cor_knn <- rep(0, 40)
for (i in 1:40){
  fit <- knnreg(train_reg, train$quality, k=i)
  pred <- predict(fit, newdata=test_reg)
  test_cor_knn[i] <- cor(pred, test$quality)
  test_mse_knn[i] <- mean((pred - test$quality)^2)
}
which.min(test_mse_knn)
```

```
## [1] 22
```

```
which.max(test_cor_knn)
```

```
## [1] 22
```

```
test_mse_knn[22]  # .547
```

```
## [1] 0.5470226
```

```
test_cor_knn[22]  # .548
```

```
## [1] 0.5478941
```

kNN

- supervised or unsupervised?
- classification or regression?
- parametric or non-parametric?
- bias and variance?
- advantages?
- disadvantages?

k-means

```
set.seed(1234)
df <- wine[]
df <- as.data.frame(scale(df[,-13]))  # remove type
fit.km <- kmeans(df, 2, nstart=20)
fit.km
```

```
## K-means clustering with 2 clusters of sizes 1668, 4829
##
## Cluster means:
##   fixed_acidity volatile_acidity citric_acid residual_sugar  chlorides
## 1     0.8203503        1.1587448  -0.3368131     -0.5896682  0.9070517
## 2    -0.2833598       -0.4002457   0.1163397      0.2036791 -0.3133076
##   free_sulfur_dioxide total_sulfur_dioxide    density         pH
## 1          -0.8323898           -1.1742444  0.6706333  0.5554976
## 2           0.2875184            0.4055994 -0.2316456 -0.1918762
##    sulphates     alcohol     quality
## 1  0.8242456 -0.08641804 -0.28082994
## 2 -0.2847052  0.02984993  0.09700235
##
## Clustering vector:
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [35] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [69] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [103] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [137] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [171] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [205] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
##  [239] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [273] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [307] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [341] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [375] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [409] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [443] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [477] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [511] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [545] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [579] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [613] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [647] 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [681] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [715] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [749] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [783] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [817] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1
##  [851] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [885] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [919] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [953] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [987] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1
## [1021] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1055] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1
```

```
## [1089] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1
## [1123] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1157] 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1191] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1225] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1259] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1
## [1293] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1327] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1361] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
## [1395] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1429] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1
## [1463] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1
## [1497] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1531] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1565] 1 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1599] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [1633] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1667] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1701] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1735] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1769] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1803] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [1837] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1871] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
## [1905] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1939] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [1973] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2007] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [2041] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2075] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [2109] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2143] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2177] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [2211] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2245] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2279] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2313] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2347] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [2381] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2415] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2449] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2483] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2
## [2517] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
## [2551] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2585] 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2619] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2653] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2687] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [2721] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [2755] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2789] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [2823] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2857] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2891] 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
## [2925] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2959] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [2993] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3027] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3061] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3095] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3129] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3163] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3197] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3231] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3265] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3299] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3333] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3367] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3401] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [3435] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2
## [3469] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3503] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2
## [3537] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3571] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3605] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2
## [3639] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3673] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3707] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3741] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
## [3775] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3809] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3843] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3877] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3911] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3945] 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [3979] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4013] 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4047] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [4081] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4115] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4149] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4183] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4217] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [4251] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4285] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4319] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4353] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [4387] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4421] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
## [4455] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4489] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4523] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4557] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4591] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
## [4625] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4659] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4693] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4727] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
## [4761] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4795] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4829] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4863] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4897] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4931] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4965] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4999] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5033] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5067] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5101] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
## [5135] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5169] 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5203] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5237] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [5271] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5305] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5339] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5373] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5407] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5441] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5475] 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
## [5509] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5543] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
## [5577] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5611] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
## [5645] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5679] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5713] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5747] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5781] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
## [5815] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5849] 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5883] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [5917] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5951] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [5985] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6019] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6053] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6087] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6121] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6155] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6189] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6223] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2
## [6257] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6291] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6325] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6359] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [6393] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6427] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6461] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [6495] 2 2 2
##
## Within cluster sum of squares by cluster:
```

```
## [1] 20237.60 42240.26
##  (between_SS / total_SS =  19.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

Interpreting clusters

The within-ss for the two clusters are large. This is probably due to the large number of predictors.

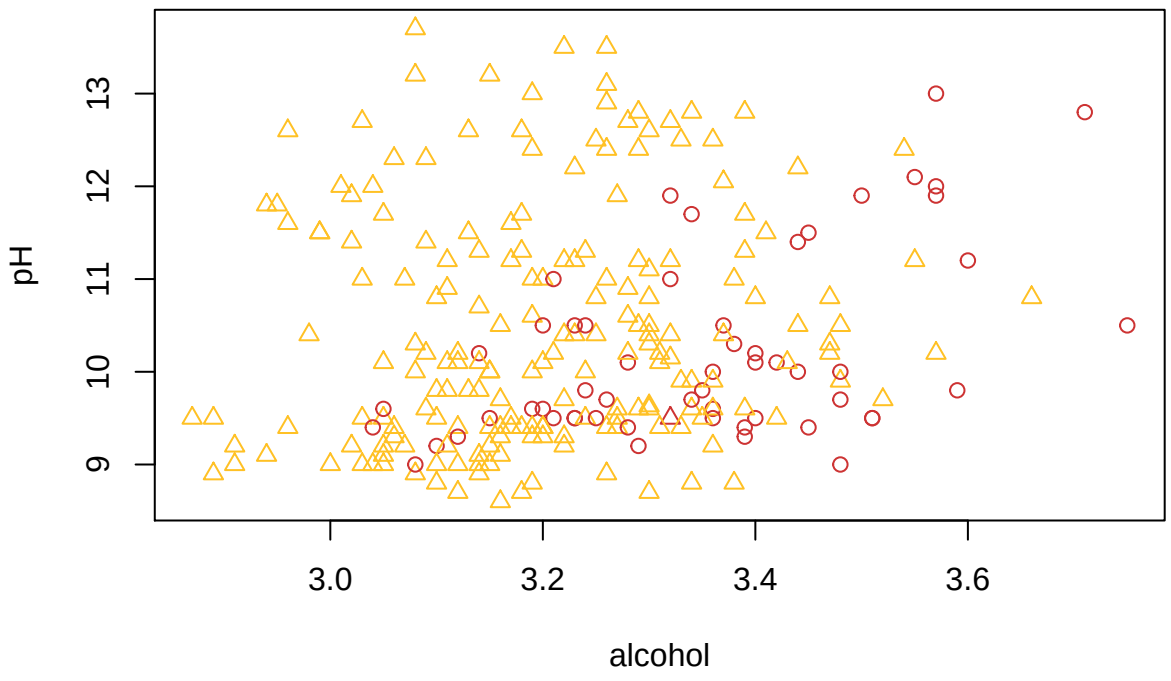
Looking at the correlation of the clusters and the type red/white, we seem to have found something in the data.

```
cor(fit.km$cluster, abs(1-as.integer(wine$type)))  # flip 1 and 2
```

```
## [1] 0.9517272
```

Plot

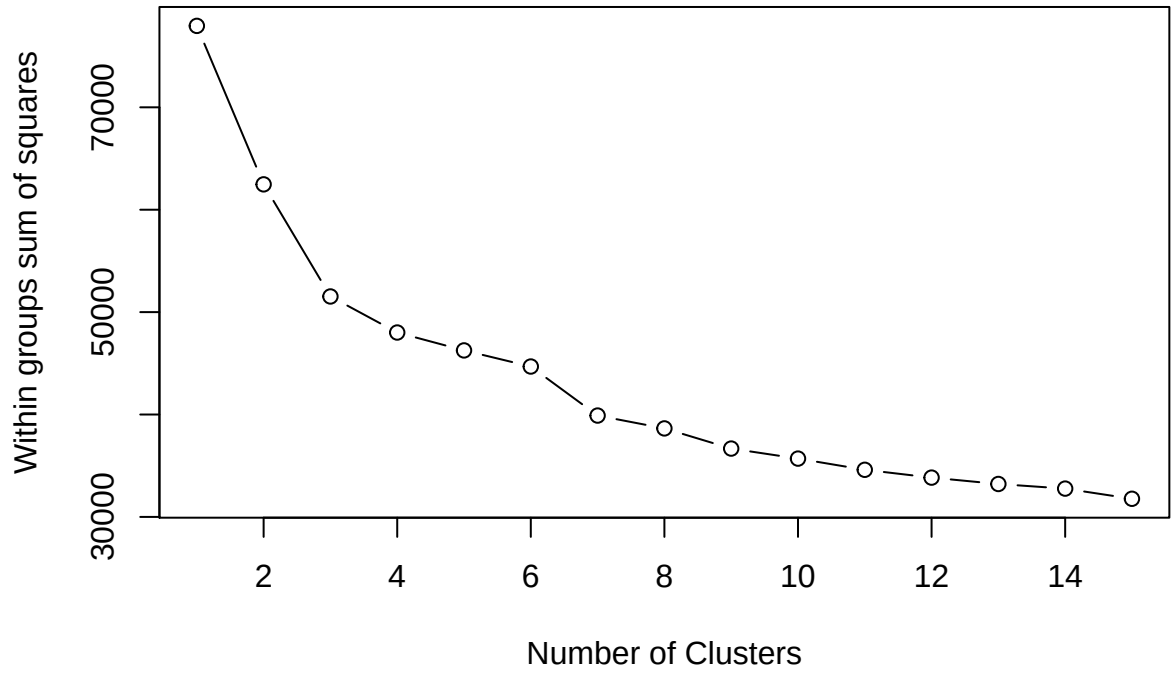
```
# plot is too dense with all data so randomly select 250
j <- sample(1:nrow(wine), 250, replace=FALSE)
# plot - using pH and alcohold to spread out the points
# color = white/red wine
# shape = cluster identified by kmeans
plot(wine$pH[j], wine$alcohol[j], pch=fit.km$cluster[j],
     col=c("brown3","goldenrod1")[as.integer(wine$type[j])],
     xlab="alcohol", ylab="pH")
```



```
# only plotting in 2 dimensions, in 13-d space, white and red wines must be "close"
```

Try Various K

```
wsplot <- function(data, nc=15, seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data,centers=i)$withinss)
  }
  plot(1:nc, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
}
wsplot(df)  # elbow at 3
```




```
fit.km3 <- kmeans(df, 3, nstart=20)
fit.km3
```

```
## K-means clustering with 3 clusters of sizes 1619, 1970, 2908
##
## Cluster means:
##   fixed_acidity volatile_acidity  citric_acid residual_sugar   chlorides
## 1     0.8632594        1.1715622 -0.319621269     -0.6051692  0.92557709
## 2    -0.1873765       -0.3409975  0.250501325      1.1327664 -0.09407513
## 3    -0.3536744       -0.4212497  0.008245951     -0.4304611 -0.45157541
##   free_sulfur_dioxide total_sulfur_dioxide    density          pH
## 1         -0.83732110          -1.18605389  0.6949275  0.53134628
## 2          0.80759493           0.94385295  0.7241666 -0.38069887
## 3         -0.08092818           0.02091848 -0.8774745 -0.03792051
##    sulphates    alcohol    quality
## 1  0.8220545 -0.1390949 -0.2913803
## 2 -0.2655267 -0.8028659 -0.3047144
```

```
## 3 -0.2777918  0.6213344  0.3686492
##
## Clustering vector:
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [35] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [69] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
##  [103] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1
##  [137] 1 1 1 1 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [171] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [205] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1
##  [239] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [273] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [307] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [341] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [375] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
##  [409] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [443] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [477] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [511] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [545] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [579] 1 1 1 1 1 1 1 1 1 1 3 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [613] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [647] 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [681] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [715] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [749] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [783] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 1 1 1 1 1 1 1 1
##  [817] 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 1 1
##  [851] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [885] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 3 1 1 1
##  [919] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1
##  [953] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1
##  [987] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 1 1 1 1 3 1 1 1 1 1 1 3 3 1
## [1021] 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1
## [1055] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 2 1 1 1 1 1 3
## [1089] 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 3 1 1 1 3 1 1 1
## [1123] 3 1 1 3 3 1 1 1 1 3 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1
## [1157] 3 3 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1191] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1
## [1225] 1 1 1 1 3 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1259] 1 1 1 1 1 1 1 1 1 1 1 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
## [1293] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1327] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1361] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1395] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1429] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
## [1463] 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1
## [1497] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1531] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1565] 1 1 3 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1599] 3 2 3 3 2 2 3 2 2 3 3 3 3 3 3 2 3 3 3 3 2 3 3 3 1 3 2 3 2 3 3 2 3 3
## [1633] 3 2 3 3 3 2 2 2 2 2 3 3 3 2 2 2 2 3 3 3 3 2 3 2 2 3 3 2 2 3 3 3 3 3
## [1667] 2 3 2 2 2 2 3 3 2 3 3 2 2 3 2 2 2 2 2 2 2 2 2 2 2 3 3 3 2 2 3 2 2 2
## [1701] 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 1 3 3 2 2 3 2 2 3 3 3 3 2 3 3 3 2 2 2 2
```

```
## [1735] 2 3 2 3 3 3 2 3 3 3 3 3 1 3 3 3 2 3 3 1 2 2 3 3 3 3 2 3 2 2 2 2 3 2
## [1769] 2 3 3 3 3 2 3 3 2 1 2 2 2 2 2 2 2 2 3 3 2 2 2 3 3 2 2 2 2 2 2 2 2 2
## [1803] 3 3 2 2 2 1 3 3 3 2 3 3 2 2 2 2 2 2 2 3 3 3 2 3 2 3 2 2 2 2 2 2 2 2
## [1837] 2 3 2 2 3 3 2 2 3 3 3 3 2 2 2 2 3 3 3 3 3 3 3 3 2 3 2 3 2 2 2 3 3 2
## [1871] 2 2 2 2 2 2 3 3 3 3 3 2 2 2 2 2 2 2 2 2 3 2 3 2 3 2 3 2 3 3 3 3 3 2
## [1905] 2 2 2 3 3 3 3 3 2 2 2 3 3 3 3 3 3 3 2 3 2 2 3 2 3 3 3 3 2 3 3 3 2 3
## [1939] 3 2 3 2 3 3 3 3 3 2 2 2 3 3 3 3 2 2 2 3 3 3 2 3 3 2 3 3 3 3 3 3 3 1
## [1973] 3 3 3 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 2 3 3 3 2 2 3 3 2 3 3 2 3 2 3 2 3
## [2007] 3 3 3 2 2 3 3 2 2 3 2 3 3 3 2 2 2 2 2 2 2 3 2 2 3 2 1 3 3 3 3 2 3 3
## [2041] 3 3 2 2 3 3 2 2 3 2 3 3 3 3 2 3 2 3 2 2 2 2 3 2 2 2 3 2 2 2 2 3 3 3
## [2075] 2 3 3 3 3 2 3 3 2 2 2 3 3 3 3 2 3 3 2 3 3 3 2 3 3 3 2 2 2 3 3 2 2 1 3 3
## [2109] 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 3 2 2 2 3
## [2143] 2 2 3 2 3 3 2 2 3 3 2 3 3 2 3 2 3 2 3 2 3 2 2 3 2 3 2 2 3 2 3 3 3 3
## [2177] 3 3 3 2 3 2 2 3 2 2 3 3 3 2 3 2 3 3 3 2 3 3 3 2 3 3 3 3 3 2 2 2 1 3
## [2211] 3 2 2 2 2 3 3 3 3 2 2 3 3 3 3 2 2 2 3 2 2 3 2 3 3 3 2 2 2 3 2 2 2 2
## [2245] 2 3 2 2 2 2 2 2 2 3 3 3 3 3 2 3 3 2 3 2 3 3 2 3 2 2 3 3 3 2 2 2 3 3
## [2279] 3 3 3 2 2 3 2 3 1 2 3 3 2 2 2 2 2 3 2 2 2 2 3 3 3 3 3 2 3 2 3 2 3 3
## [2313] 2 3 3 2 2 3 3 2 3 3 3 3 3 3 2 3 1 2 2 3 2 2 3 2 2 3 3 3 3 3 2 2 3 2
## [2347] 3 2 3 3 2 2 2 3 3 2 2 3 3 2 2 2 2 2 3 2 3 2 2 3 3 2 2 3 3 3 2 2 2 3
## [2381] 2 2 2 2 2 2 2 3 2 2 3 2 3 2 2 2 2 3 3 2 2 2 2 3 2 2 2 2 2 2 3 3 2 2
## [2415] 3 3 2 3 2 3 2 2 3 3 2 2 3 3 3 1 3 3 3 1 3 3 3 3 3 3 2 2 3 3 3 2 3 2
## [2449] 3 3 2 3 2 3 3 2 2 2 2 3 2 3 3 3 3 3 2 2 3 2 2 3 3 3 3 3 3 3 3 3 3 3
## [2483] 3 2 3 3 3 3 3 3 3 3 3 3 2 3 3 2 3 2 2 2 3 3 1 3 2 2 3 3 3 3 1 3 3
## [2517] 3 3 3 2 2 2 2 3 3 3 3 2 2 3 2 2 2 2 2 3 2 2 2 2 2 3 3 2 3 2 3 1 3 3
## [2551] 2 3 3 2 3 3 3 3 2 2 3 2 3 2 3 3 2 3 3 3 3 2 3 3 2 3 2 3 3 3 3 3 3 3
## [2585] 3 2 2 3 3 3 1 2 3 3 3 3 2 2 3 3 2 2 3 3 3 3 3 3 3 3 3 3 2 2 3 2 3 3
## [2619] 2 3 3 3 2 3 3 3 1 3 2 3 2 2 2 2 3 1 1 3 3 1 3 1 2 3 3 3 3 3 3 2 2 3
## [2653] 1 3 3 3 2 3 2 3 2 2 2 3 2 2 3 3 3 3 2 3 2 2 3 2 3 2 2 3 2 2 2 3 2 3
## [2687] 3 2 3 2 2 3 3 2 3 2 3 2 3 2 3 3 3 2 3 3 3 3 2 3 3 2 3 1 3 3 2 3 2 2
## [2721] 3 3 3 3 2 3 3 3 3 3 3 2 3 3 2 3 3 2 3 3 2 2 3 3 2 3 3 3 2 2 2 1 3 2
## [2755] 3 2 2 2 2 3 2 3 2 3 3 3 3 3 3 3 2 3 3 2 2 2 3 2 2 2 2 3 3 3 2 3 3 3
## [2789] 3 3 3 2 2 2 2 3 2 2 3 3 3 2 3 3 2 2 2 3 3 3 2 3 3 3 2 3 1 3 3 3 3 3
## [2823] 2 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 2 2 3 3 3 3 2 3 3 3 2 3 3 3 3 2 3 3
## [2857] 2 2 2 2 3 3 2 3 3 3 2 2 2 2 3 2 3 2 3 2 2 3 3 3 3 2 3 3 3 3 3 3 3 3
## [2891] 2 3 3 3 2 3 3 3 3 2 2 2 2 2 3 3 2 3 3 2 3 3 2 2 2 3 3 3 2 3 3 3 3 3
## [2925] 3 2 3 3 3 3 2 2 3 3 3 3 2 2 2 3 3 2 2 3 3 3 3 3 3 3 3 3 2 2 3 2 2 3
## [2959] 3 3 3 3 3 3 3 3 2 2 2 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 2 3 3 3
## [2993] 3 3 3 3 2 2 3 2 2 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 2 3 3 2 3 3 2 3 3 3
## [3027] 3 3 3 3 3 3 3 3 2 2 3 2 2 2 3 3 3 2 3 3 3 3 3 2 3 2 2 2 2 3 3 3 3 3
## [3061] 3 3 3 3 3 3 3 3 3 2 2 3 3 3 3 2 3 3 3 3 3 3 3 2 3 3 2 2 2 3 3 2 3 3
## [3095] 3 1 3 3 3 3 3 3 3 3 3 2 2 2 3 3 3 3 2 3 3 2 2 3 3 3 3 3 2 2 2 3 2 3
## [3129] 2 2 3 2 3 2 3 3 3 3 3 2 3 3 3 3 3 2 2 3 3 2 2 3 3 2 3 2 2 2 3 3 1 3
## [3163] 3 1 3 2 3 2 2 2 2 2 3 2 2 3 1 2 2 2 3 3 2 2 2 2 2 2 3 3 2 3 3 2 2 3
## [3197] 2 3 3 2 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 2 2 2 3 2 3
## [3231] 3 3 3 2 3 2 3 2 3 2 2 2 2 2 2 2 3 3 2 3 2 3 2 2 3 3 2 2 2 2 2 3 2 2
## [3265] 3 3 3 3 3 2 3 2 2 2 2 3 3 3 3 2 2 2 2 2 2 2 2 2 3 1 2 3 2 2 3 2 3 3
## [3299] 3 3 2 2 2 2 3 3 2 1 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2 3 2 2 3 3 2 3
## [3333] 3 2 3 3 2 3 3 2 3 2 2 2 2 3 2 3 3 2 3 2 3 3 2 2 2 3 2 2 3 3 2 2 2 2
## [3367] 2 3 3 2 3 2 2 3 2 2 3 3 3 2 3 2 1 3 2 3 3 3 3 2 3 3 2 3 2 3 3 2 2 2
## [3401] 3 2 2 2 2 3 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 3 2 2 2 2 2 2 2 2 3
## [3435] 2 2 3 2 2 3 3 2 2 3 2 2 2 2 3 3 3 3 1 3 2 1 3 3 2 3 2 3 2 3 1 2 2 3
## [3469] 2 3 3 2 2 2 2 3 3 2 2 2 2 2 2 2 2 3 2 2 3 2 2 2 2 2 2 2 2 1 3 2 3 2
## [3503] 3 2 2 3 3 3 2 2 3 3 3 3 3 2 3 2 2 2 2 2 3 3 1 1 3 3 2 2 2 1 2 3 2 2
## [3537] 3 3 2 2 2 2 2 2 2 2 2 3 2 2 1 3 2 2 2 3 2 2 3 3 2 3 2 2 2 3 2 3 3 3
```

```
## [3571] 3 2 2 2 2 2 2 3 2 2 2 2 2 2 2 3 2 2 3 1 2 3 3 2 2 2 2 2 2 3 3 3 3 3
## [3605] 2 2 3 3 3 3 3 3 2 3 3 3 3 3 2 3 3 2 3 2 2 1 2 2 2 3 2 3 2 3 3 3 2 3
## [3639] 3 2 3 3 3 2 3 2 2 2 2 2 2 2 3 3 3 2 3 2 2 2 3 3 3 2 3 3 3 3 3 3 3 2
## [3673] 2 2 3 1 3 3 2 1 2 2 2 3 3 3 2 3 3 2 2 1 2 2 2 2 2 2 2 3 2 3 2 3 3 2 2
## [3707] 2 2 2 2 2 2 3 2 2 3 3 3 3 2 3 2 2 3 2 2 2 3 3 3 2 2 3 3 2 3 2 3 2 2 2
## [3741] 2 2 3 2 2 3 2 2 3 3 2 3 3 1 2 3 3 3 3 3 3 1 3 2 3 3 2 2 2 2 2 2 2 2 2
## [3775] 2 2 3 2 3 2 3 2 2 2 2 1 3 2 2 3 2 2 2 2 3 3 3 2 2 2 3 2 3 3 2 2 3 3
## [3809] 3 3 3 3 3 3 2 1 3 3 3 3 2 3 2 3 2 2 2 2 3 2 3 3 3 1 2 2 2 3 2 2 2 2
## [3843] 2 2 3 2 3 3 2 2 3 2 2 3 3 3 1 2 1 2 3 3 2 2 2 2 2 2 2 3 2 3 1 3 2 2
## [3877] 3 2 2 3 3 2 2 2 2 2 2 2 3 3 3 3 3 3 2 3 2 3 3 3 2 2 2 3 3 3 2 2 3 2
## [3911] 3 2 3 3 2 3 2 2 3 3 3 3 3 2 3 2 3 2 2 2 3 2 3 2 2 2 2 2 2 3 3 2 3 3 3
## [3945] 2 2 3 3 2 2 2 3 3 3 3 3 3 2 1 2 1 3 2 2 3 2 2 2 3 3 2 3 1 3 3 3 3 2
## [3979] 2 2 3 3 3 3 2 3 3 3 3 3 2 3 3 2 2 2 3 3 3 1 3 2 3 2 2 3 3 2 3 2 2 2
## [4013] 3 2 3 2 2 3 2 3 2 2 3 2 3 3 2 3 2 2 2 2 2 2 2 2 3 2 2 2 3 2 2 2 2 2
## [4047] 2 3 3 2 2 3 3 2 2 2 2 2 2 2 3 3 2 3 2 2 3 3 3 2 3 3 2 3 1 2 3 2 2 3
## [4081] 2 2 2 2 2 2 3 3 2 1 2 2 2 2 3 2 2 2 3 2 2 1 1 2 2 2 3 2 2 2 2 3 3 3
## [4115] 3 3 2 3 2 2 3 3 3 2 3 3 3 3 2 3 2 2 2 3 3 2 2 2 3 2 3 2 2 3 3 2 2 2
## [4149] 2 2 2 2 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 2 3 3 3 3 3 2 3 2 2 2 3 2 2 2 2
## [4183] 2 2 2 2 3 2 1 2 3 3 2 3 3 2 3 2 2 3 2 3 3 3 3 3 2 2 3 2 2 3 2 3 2 2
## [4217] 2 3 2 2 2 3 2 3 2 3 3 2 3 3 3 2 2 2 3 2 2 3 3 3 2 1 3 2 3 3 3 2 2 2
## [4251] 3 3 2 2 2 2 3 2 3 2 3 3 3 3 2 3 3 1 2 3 3 2 3 3 3 3 3 3 2 3 2 3 3 3
## [4285] 3 3 2 2 2 3 3 3 3 3 3 2 3 3 3 3 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 3
## [4319] 3 3 2 3 3 3 2 3 3 2 3 2 2 3 3 2 2 3 3 2 3 2 2 2 3 3 3 3 2 3 2 3 3 3
## [4353] 3 3 2 2 3 3 3 3 2 3 3 2 3 3 2 3 3 2 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 3
## [4387] 2 2 2 2 2 2 3 2 3 3 2 3 3 2 2 3 3 3 3 2 2 2 3 3 3 3 3 3 3 3 3 2 3 2
## [4421] 2 2 3 2 3 2 3 3 2 2 2 3 3 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 2 3 3 3
## [4455] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 3 3 3 3 2 3 3 3 3 3 3
## [4489] 3 3 3 2 3 3 3 2 2 3 2 3 3 2 3 3 2 2 3 3 3 2 2 3 2 3 3 3 3 3 2 2 3 3
## [4523] 3 2 2 3 3 3 2 3 3 2 3 3 3 3 2 3 3 3 3 2 3 3 3 3 3 2 2 3 3 3 3 3 3 3
## [4557] 3 3 3 2 3 2 3 3 3 2 2 3 3 3 3 3 2 2 3 3 2 2 3 3 3 2 3 3 3 3 3 3 3 3
## [4591] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 3 3 2 2 3 2 3 3 3 3 3 1 2 2
## [4625] 3 3 2 3 3 3 2 3 2 2 3 3 2 3 2 2 2 2 2 2 3 3 3 3 2 2 2 2 3 3 3 3 3 2
## [4659] 3 2 2 3 2 2 2 2 3 3 3 3 3 2 3 2 3 2 2 3 3 3 2 3 3 3 3 3 2 3 3 3 2 3
## [4693] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 3 3 3 2 3 3 3 3 3 3 2 3 3 3 2 2 3
## [4727] 3 3 3 3 2 3 3 3 2 3 3 3 2 2 3 3 3 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3
## [4761] 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [4795] 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 2 2 2 3 2 3 3 3 2 1 3 3 3 3 2 3 3 3 2 2
## [4829] 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 2 2 2 2 2 2 3 2
## [4863] 3 2 2 3 3 3 2 3 3 3 3 3 3 3 3 2 3 3 3 3 2 3 3 3 3 2 2 3 3 2 3 3 2 2
## [4897] 2 3 3 3 3 3 3 3 3 3 2 2 3 3 3 3 3 2 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 2
## [4931] 2 3 3 3 2 2 2 3 3 3 3 3 2 2 2 2 3 3 2 3 3 3 3 3 3 3 2 3 3 3 3 2 3 3 3
## [4965] 3 3 3 3 3 3 3 3 3 3 2 3 3 3 2 3 3 3 3 2 3 3 2 3 3 3 3 3 3 3 2 2 2 3 2
## [4999] 3 3 3 3 3 3 3 3 3 2 2 3 3 3 2 2 3 3 2 3 2 3 2 3 3 2 3 2 2 2 3 2 2 2 3
## [5033] 3 3 3 3 3 2 2 2 3 3 3 3 2 3 2 2 3 3 3 3 2 3 3 3 3 3 3 2 2 3 3 3 2 3 3
## [5067] 2 3 2 3 2 3 3 3 2 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3
## [5101] 3 2 3 3 3 3 3 3 2 2 2 3 3 3 3 3 3 3 3 3 2 3 2 2 3 3 3 2 1 3 2 2 3 2 2
## [5135] 2 3 3 3 3 3 3 3 2 3 3 2 2 2 3 3 3 2 2 2 3 3 3 3 2 3 3 2 3 2 3 3 3 3 3
## [5169] 3 3 1 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 2 2 2 3 3 3 3 3 3 3 2 3 2 2 3 3
## [5203] 3 3 2 2 3 2 2 2 3 3 2 2 3 2 3 2 2 2 2 3 2 3 2 3 2 2 2 3 3 2 3 2 3 3
## [5237] 3 3 3 3 2 2 3 3 3 3 3 3 3 3 2 3 2 3 3 3 2 3 3 3 3 1 3 3 3 2 3 3 3 2
## [5271] 3 3 3 3 3 3 3 3 2 3 2 3 3 2 3 2 2 2 3 3 3 3 3 3 2 3 3 2 3 2 3 2 2 2 2
## [5305] 2 3 3 2 3 3 2 2 2 3 2 3 3 2 2 3 3 3 3 3 3 3 3 3 3 2 2 3 3 2 3 3 2 3
## [5339] 2 2 2 2 2 2 2 3 2 2 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 3 2 2
## [5373] 3 2 3 3 3 3 3 2 3 3 2 2 3 2 2 2 2 2 2 2 2 3 2 2 2 3 3 3 3 3 3 3 3 2
```

```
## [5407] 3 3 3 2 3 3 3 2 3 2 3 3 3 3 2 3 2 3 3 3 3 3 3 3 3 2 2 2 3 3 2 2 3
## [5441] 2 3 3 3 2 3 2 1 1 3 3 3 3 3 3 3 3 3 2 2 2 2 2 3 3 3 2 2 2 2 2 2 2 2
## [5475] 3 2 3 2 1 2 2 3 3 2 3 3 3 2 2 3 3 2 3 3 3 2 2 3 2 3 3 3 3 3 3 3 3 3
## [5509] 3 3 2 3 3 3 3 3 2 3 3 3 2 3 3 3 3 2 3 2 2 3 3 3 3 3 2 3 2 3 3 2 3 3
## [5543] 3 3 3 3 3 3 2 3 2 3 3 2 3 3 3 3 3 2 2 3 3 2 3 3 2 2 3 2 3 1 2 2 3 3
## [5577] 3 3 3 2 3 3 3 3 2 2 2 3 2 2 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 2 3 3
## [5611] 3 3 2 2 3 2 3 2 3 2 3 2 2 3 3 2 3 3 3 3 3 2 3 2 3 3 3 3 1 2 2 2 2 2
## [5645] 2 2 2 2 3 2 3 2 2 2 3 2 3 2 3 3 3 3 3 3 3 3 2 3 2 2 3 2 3 2 2 2 3 3
## [5679] 3 2 2 3 3 3 3 3 3 3 3 2 3 3 2 3 2 3 3 3 3 2 2 2 3 3 3 3 2 3 3 3 3 3
## [5713] 3 3 3 2 2 3 3 2 2 2 3 3 2 2 2 2 3 3 2 2 3 3 3 3 2 3 3 2 2 3 2 2 2 2
## [5747] 3 2 3 2 3 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 2 3 3 2 2 3 2 3 3 2 2
## [5781] 3 2 3 3 2 3 3 3 3 3 3 3 3 3 3 3 2 2 3 3 3 3 3 2 3 2 3 3 3 3 2 2 1 2
## [5815] 2 2 2 3 2 3 3 2 3 3 2 2 3 2 3 3 3 3 3 3 2 2 3 2 3 2 3 3 3 3 2 2 3 3 3
## [5849] 3 2 3 2 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 3 2 3 2 2 2 2
## [5883] 3 3 3 3 3 3 2 2 2 3 2 3 2 3 3 3 2 2 2 2 3 3 3 2 3 3 3 2 3 3 3 3 3 1
## [5917] 3 3 3 2 2 3 3 3 2 2 2 3 2 2 2 2 2 2 2 2 2 2 3 2 2 3 3 2 3 2 3 2 2 3
## [5951] 3 3 3 3 3 3 3 3 2 2 3 3 3 2 3 2 3 2 3 3 3 3 3 3 2 3 3 3 2 3 3 3 3 3 2
## [5985] 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 2 3 3 2 3 2 3 3 3 3 2 2 3 2
## [6019] 2 2 2 3 3 2 3 3 2 3 3 2 3 3 2 3 3 3 3 3 3 2 3 2 3 3 3 3 3 3 2 2 2 2
## [6053] 3 2 2 2 2 3 3 2 3 3 3 2 2 3 2 3 3 3 3 3 1 2 3 3 2 2 3 2 2 3 3 3 3 3
## [6087] 3 3 3 3 3 3 2 3 2 3 2 3 3 3 2 3 3 3 2 3 3 3 3 3 3 3 3 2 3 3 3 3 2 2
## [6121] 2 3 2 2 2 2 2 3 3 2 2 2 3 2 3 2 2 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3
## [6155] 2 3 2 3 3 3 3 3 3 3 2 2 2 3 3 3 3 3 3 3 3 3 2 3 2 2 3 3 2 3 3 3 3 3 3
## [6189] 3 3 2 3 2 3 3 3 2 2 3 3 3 2 3 3 3 3 2 3 3 3 3 2 2 2 3 3 3 3 2 3 3 3
## [6223] 3 3 3 2 3 3 3 2 3 2 3 2 2 3 3 3 2 3 3 3 2 3 3 3 3 2 1 1 3 2 3 2 2 2
## [6257] 2 3 3 3 3 3 3 3 3 2 3 3 3 2 3 3 3 3 3 2 3 2 3 3 3 3 3 3 2 3 2 2 2 2
## [6291] 2 3 2 2 3 3 3 2 2 2 3 3 3 2 3 3 2 3 3 3 2 3 3 3 3 3 3 3 3 3 3 2 2 3
## [6325] 2 3 2 2 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 2 2 3 2 2 3 2 3 3 2 3 3 3 3
## [6359] 2 2 2 3 3 3 3 3 3 2 2 2 2 2 2 3 3 3 3 3 2 3 2 2 2 2 2 3 3 3 2 3 3 2 3
## [6393] 2 2 3 3 3 2 3 3 3 3 3 3 3 2 3 3 3 2 3 3 2 3 3 2 3 3 3 2 3 3 3 2 3 3
## [6427] 3 2 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 1 3 3 3 2 3 2 3 3 3 2 2 3 2 3 3
## [6461] 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 2 2 2 3 3 2 2 3 3 3 2 3 3 3 3 2
## [6495] 3 3 3
##
## Within cluster sum of squares by cluster:
## [1] 18770.18 13782.01 18976.59
##  (between_SS / total_SS =  33.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
fit.km$withinss
```

```
## [1] 20237.60 42240.26
```

```
fit.km3$withinss  # lower
```

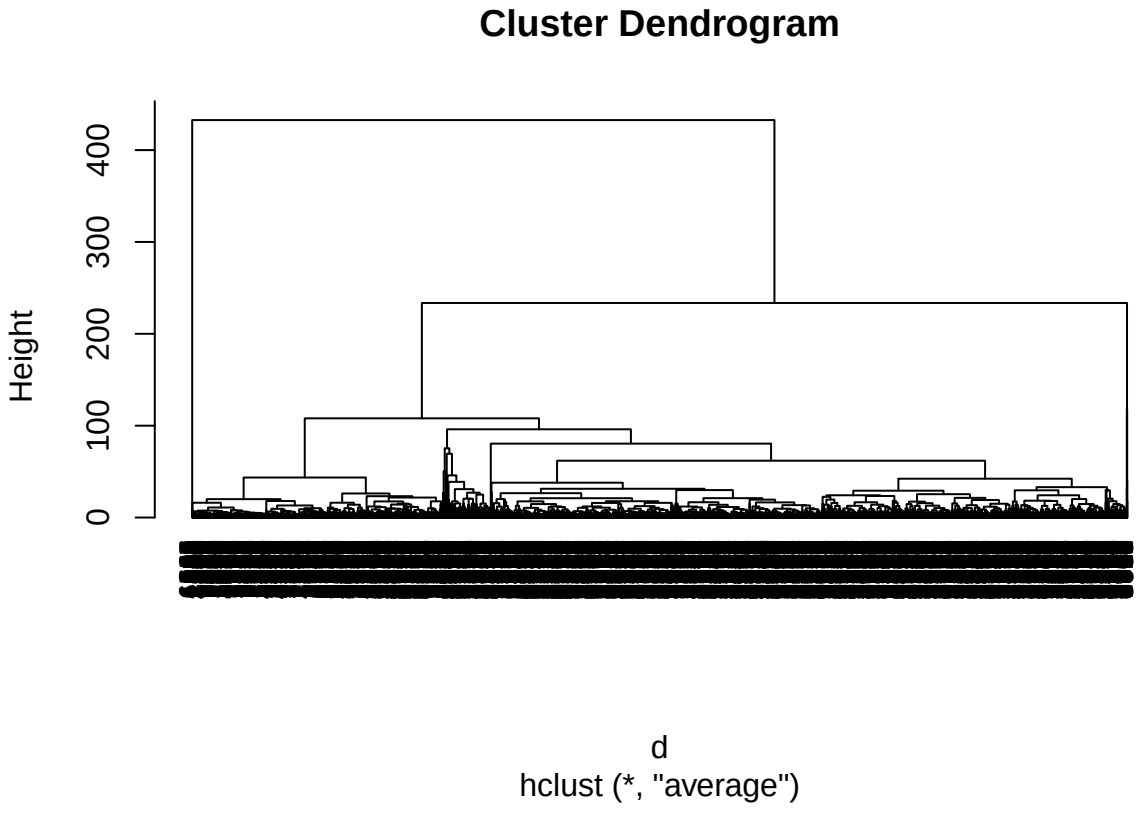
```
## [1] 18770.18 13782.01 18976.59
```

Hierarchical clustering

```
d <- dist(wine)  # data was normalized above
```

```
## Warning in dist(wine): NAs introduced by coercion
```

```
fit.average <- hclust(d, method="averag")
plot(fit.average, hang=-1, cex=.8)
```

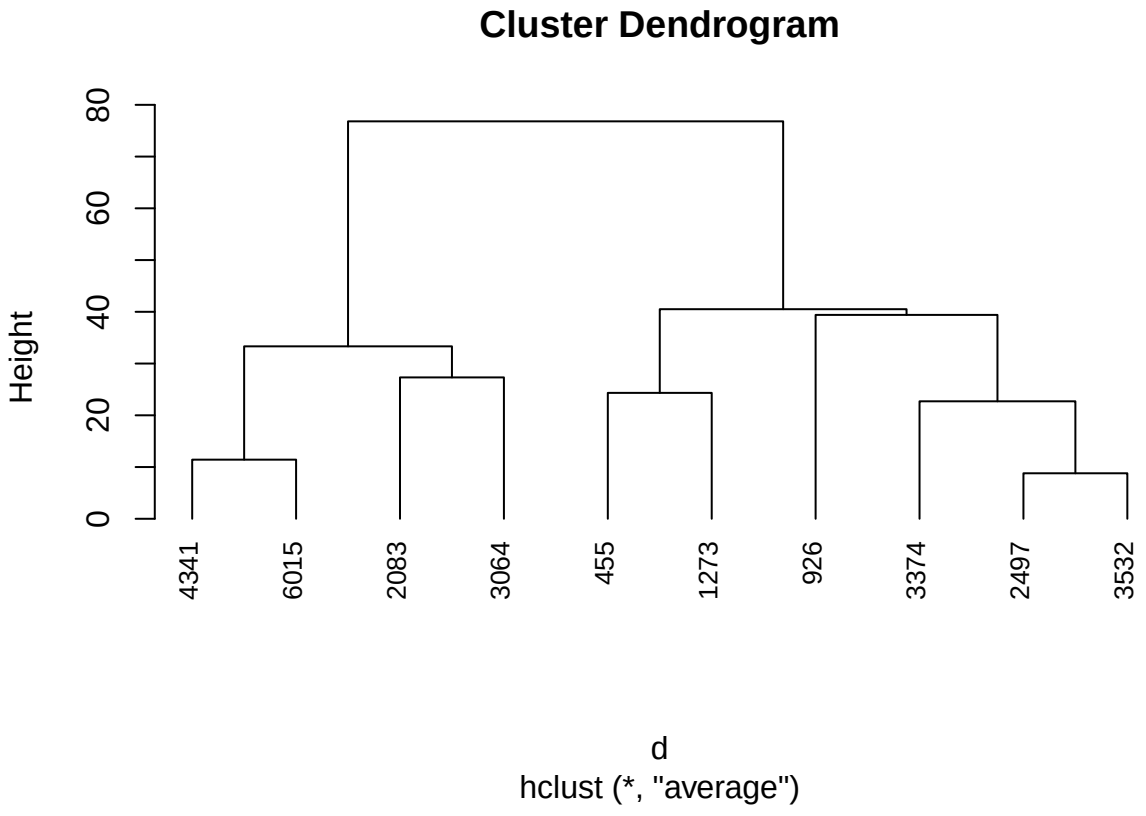


Try fewer examples so we can see the data.

```
j <- sample(1:nrow(wine), 10, replace=FALSE)
d <- dist(wine[j,])  # data was normalized above
```

```
## Warning in dist(wine[j, ]): NAs introduced by coercion
```

```
fit.small <- hclust(d, method="averag")
plot(fit.small, hang=-1, cex=.8)
```



Compare some values.

```
# compare 5698 and 4705 == -1.9
sum(df[5698,] - df[4705,])
```

```
## [1] -1.947792
```

```
# compare 3555 and 6127 == 6.84
sum(df[3555,] - df[6127,])  # diff = -1.07, 3 times as much
```

```
## [1] 6.841612
```

General ML Questions

- What is an indication that you have overfit the data?
- What is an indication that you have underfit the data?
- What is the relation between underfitting/underfitting and bias/variance?

Other Things to Study

- Loss and cost functions for linear regression, logistic regression
- How these functions are used for linear regression, logistic regression
- Matrix notation for data
- gradient descent
- k-fold cross validation
- parameter versus hyper-parameter
- a couple of questions about k-means, hierarchical clustering