Module 5 R Activity

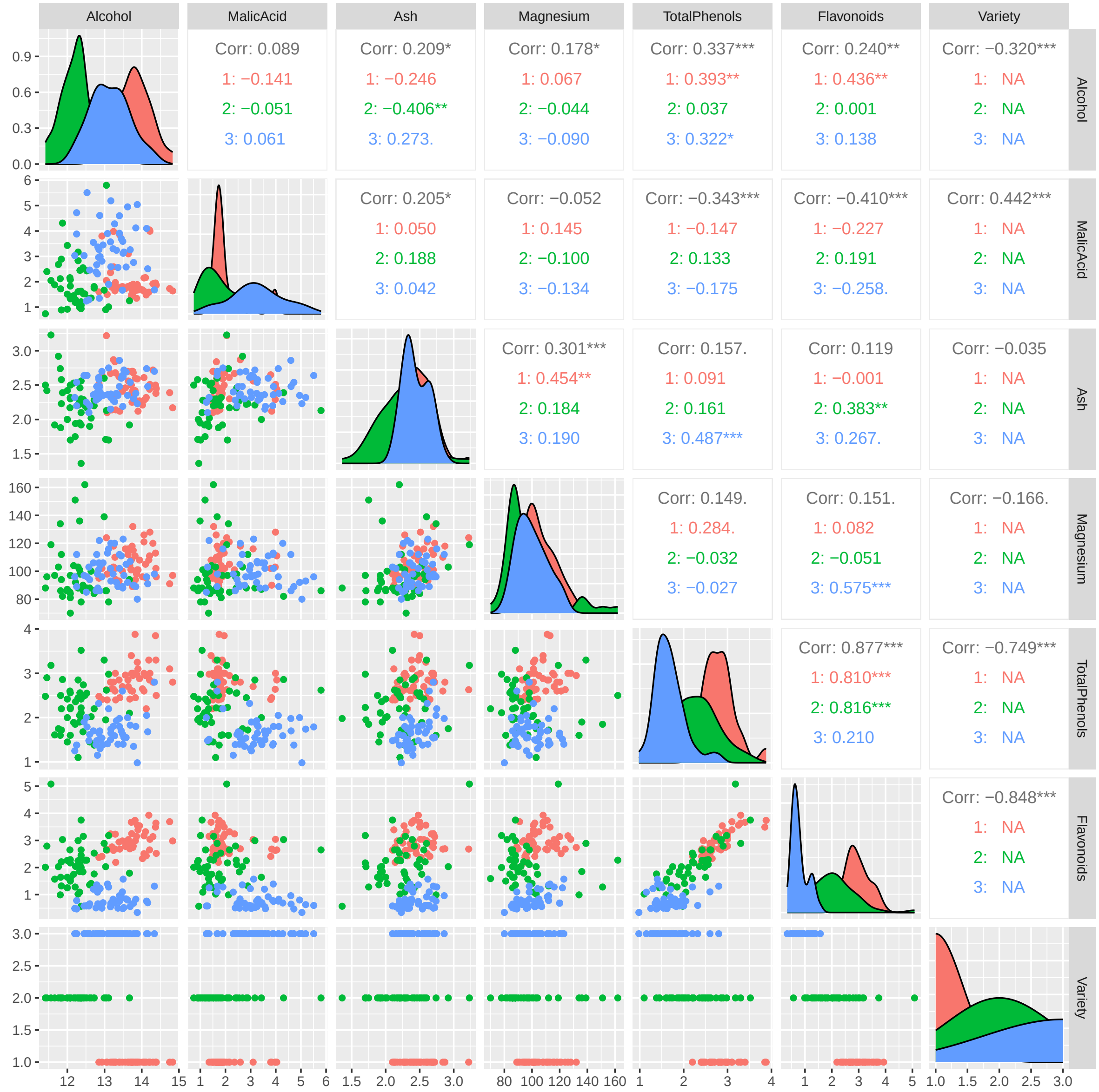
Zahlen Zbinden

Question 1: Read in the BalWineData.csv, which contains measurements on 6 variables for 45. Display the first six rows of data

```
# A tibble: 6 x 7
  Alcohol MalicAcid   Ash Magnesium TotalPhenols Flavonoids Variety
    <dbl>     <dbl> <dbl>     <dbl>        <dbl>      <dbl>   <dbl>
1    13.7      1.67  2.25       118          2.6        2.9       1
2    13.6      1.81  2.7        112          2.85       2.91      1
3    14.2      1.59  2.48       108          3.3        3.93      1
4    12.9      3.8   2.65       102          2.41       2.41      1
5    13.7      1.5   2.7        101          3          3.25      1
6    12.8      1.6   2.52        95          2.48       2.37      1
```

Question 2: Make a pairs plot of the 6 predictor variables of this data set, with points colored by the wine variety. Does it look like there is good separation between the three wine varieties using these six variables?

In some of the variable pairs, like Alcohol and Flavonoids (and a few others) there is seperation, but with a good chunk of the pairs that is not very good seperation at all.



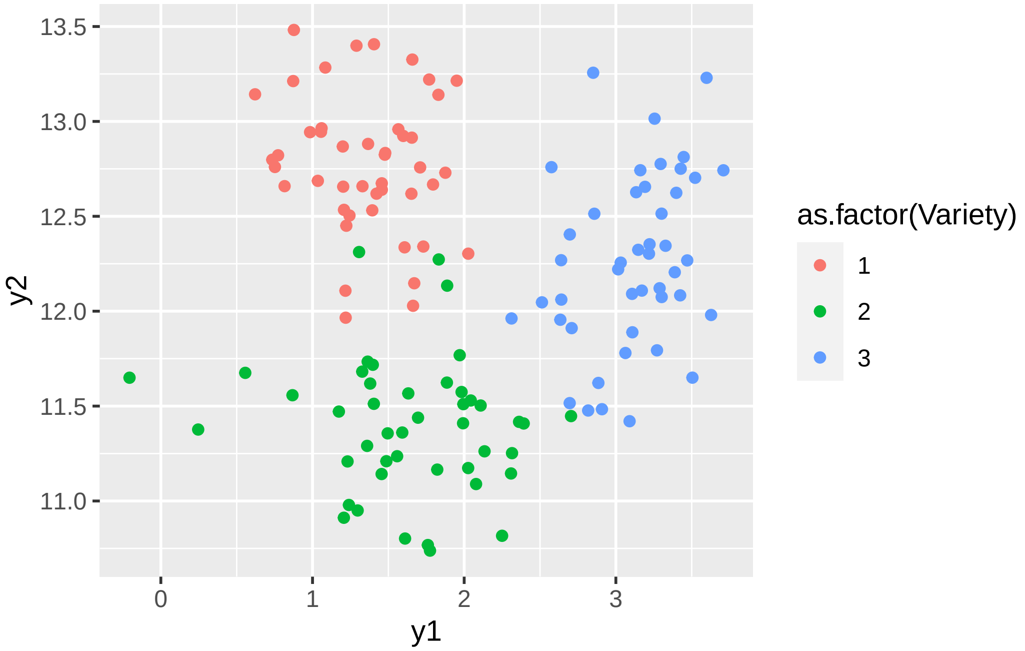
Question 3: Calculate the two linear discriminant function directions for the "wine" data by hand. Give the coefficients for each drection.

```
[1] 3.306977
```

```
[1] 2.100928
```

Question 4: compute the linear discriminatn values for both linear discriminatn directions. Plot the first linear disciminant values vs the second linear discriminaty values and color by variety. Are the Varieties well seperated?

Yes, there is now clear seperation of the variables in respect to the Variety.



Question 5: Use the lda() function to perform linear discriminant analysis on the "wine" data. Based on this output, how much of the total seperation does the first linear discriminant direction explain?

It explains 61.15% of the total seperation.

```
Call:
lda(Variety ~ Alcohol + MalicAcid + Ash + Magnesium + TotalPhenols +
    Flavonoids, data = wine)

Prior probabilities of groups:
        1         2         3
0.3333333 0.3333333 0.3333333

Group means:
   Alcohol MalicAcid      Ash Magnesium TotalPhenols Flavonoids
```

```
1 13.75133  2.050000 2.466222 105.86667     2.829778  2.9717778
2 12.26889  1.852222 2.208889  96.77778     2.235333  2.1100000
3 13.14622  3.248444 2.442000  99.91111     1.673778  0.7891111

Coefficients of linear discriminants:
                    LD1          LD2
Alcohol       0.34168070  1.980871844
MalicAcid     0.34379857  0.122542991
Ash           1.32501818  1.538921529
Magnesium    -0.00337495  0.007281932
TotalPhenols  0.12692385 -0.102187896
Flavonoids   -1.85394568  0.129476315

Proportion of trace:
   LD1    LD2
0.6115 0.3885
```

Question 6: Suppose we have a new wine that has the following values for the six variables...Use LDA to predict which of the three varieties this new ine belongs to.

This new wine is most likely part of variety 1.

Question 7: supposed we have a new wine. Compute the unmodified QDA distances between this new wine and the sample mean of each variety. What are the distances for each variety? Which group would you assign this wine to?

We can see from this analysis that the smallest distance is in type 2, which we would classify this new observation as Variety 2.

```
[1]  46.66499  34.57045 222.83107
```

Question 8: Use the qda() and predict.qda() functions to predict which group this new wine belongs to. What class is this new wine assigned to?

This new wine is still assigned to Variety 2 based on this analysis.

```
$class
[1] 2
Levels: 1 2 3
```

```
$posterior
           1         2            3
1 0.02478883 0.9752112 1.040883e-40
```

Question 9: Set a random seed of 12345, and then randomly partition the wine data into a training set of 90 observations and a test set of 45 observations. Display the first six rows of your training and test sets.

```
# A tibble: 6 x 7
  Alcohol MalicAcid   Ash Magnesium TotalPhenols Flavonoids Variety
    <dbl>     <dbl> <dbl>     <dbl>        <dbl>      <dbl>   <dbl>
1    13.7      1.67  2.25       118         2.6        2.9        1
2    13.6      1.81  2.7        112         2.85       2.91       1
3    14.2      1.59  2.48       108         3.3        3.93       1
4    12.9      3.8   2.65       102         2.41       2.41       1
5    13.7      1.5   2.7        101         3          3.25       1
6    13.9      1.35  2.27        98         2.98       3.15       1
```

```
# A tibble: 6 x 7
  Alcohol MalicAcid   Ash Magnesium TotalPhenols Flavonoids Variety
    <dbl>     <dbl> <dbl>     <dbl>        <dbl>      <dbl>   <dbl>
1    12.8      1.6   2.52        95         2.48       2.37       1
2    13.6      1.73  2.46       116         2.96       2.78       1
3    14.4      1.87  2.38       102         3.3        3.64       1
4    13.8      1.65  2.6         94         2.45       2.99       1
5    13.8      1.53  2.7        132         2.95       2.74       1
6    14.4      1.95  2.5        113         3.85       3.49       1
```

Question 10: fit a knn classifier to the wine data useing k=5 neighbors. what is your estimated classifcation error on the test data.

The estimated classification error is 20%, 9 were misclassified out of 45.

```
[1] 0.2
```

Question 11: fit a knn classified to the wine data using k=15 neighbors. what is your estimated classification error on the test data?

The estimated classification error is 31%.

```
[1] 0.3333333
```

Question 12: using the same training and test set partition of the wine data, fit a CART classifciation model to the training wine data. What is the best (lowest) classification error you can obtain on the test set? Give the code and display the tree that achieves this lowest error rate?

First lets obtain a base model

```
   wine_tree_testpredc1
     1  2  3
  1 11  0  0
  2  4 13  1
  3  0  5 11
```

```
[1] 0.2222222
```

Now lets run test on various hyperparameters

```
predictions  1  2  3
          1 11  4  0
          2  0 12  0
          3  0  2 16
```

I have the best results with 13% error rate with a cp = .01, and default values of minbucket and minsplit.

```
   model_testpredc1
     1  2  3
  1 11  0  0
  2  4 12  2
  3  0  0 16
```

```
[1] 0.1333333
```