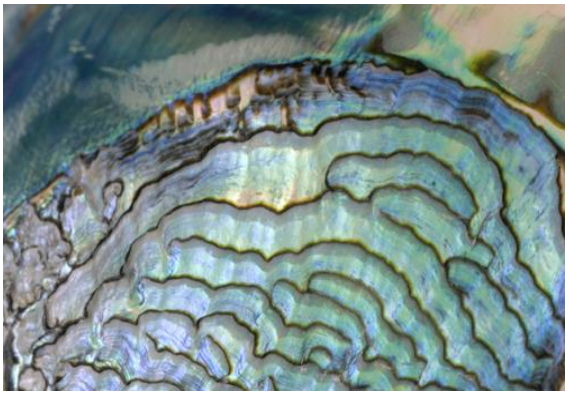
Data Mining Project Report

Gender prediction of Australia Abalone



Professor: Dr. Chris Rump

Students: Prakash Thummisetti, Jing-Yi Wu, Connor Delong

Executive Summary

Dataset and source:

For our project we decided to use data about Australian Abalones which was collected in Victoria Australia and hosted by the University of California, Irvine. The data set was originally used to help predict Abalones' ages due to the cumbersome nature of determining it. However, we decided to use this data to create several models that could be used to predict sex.

Data exploration and modification:

In order to create our models, we used Length, Diameter, Height, Whole weight, Shucked weight, viscera weight, shell weight, and Rings as our predictor variables. We decided to remove one of the categories of sex (Infant) which reduced the total number of observations we could use. We split the remaining data into training, validation, and test data.

There was not much data exploration needed to create our models. Only one technique's assumptions needed to be checked, lack of multicollinearity.

Models:

The four data mining procedures we used were: logistic regression, classification/regression trees, k-nearest neighbor, and Naïve Bayes. We then determined the best models of each method (relatively minimal error without over-fitting the model) and tested the accuracy of each.

Results and conclusions:

Lastly, we used cross-validation to determine the best model of the four different methods we used. Through this technique, we found that the logistic regression model was the best.

1. Introduction:

The dataset considered for this project is that of a species of abalone (edible sea snail). The dataset has 9 variables explained below in detail:

Sex	nominal	--	M, F, and I (infant)
Length	continuous	mm	Longest shell measurement
Diameter	continuous	mm	perpendicular to length
Height	continuous	mm	with meat in shell
Whole weight	continuous	grams	whole abalone
Shucked weight	continuous	grams	weight of meat
Viscera weight	continuous	grams	gut weight (after bleeding)
Shell weight	continuous	grams	after being dried
Rings	integer	--	+1.5 gives the age in years

The source of the data is below, University of California, Irvine, Machine learning repository:

http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.names

2. Data Modification:

The original dataset predicts the number of rings of the abalone which when adding 1.5 gives the age of the abalone in years. The dataset has been used for age prediction in the past.

(I) Response variables:

We, however, want to use Length, Diameter, Height, Whole weight, Shucked weight, viscera weight, and shell weight, and Rings to predict **Sex**, which is our response variable.

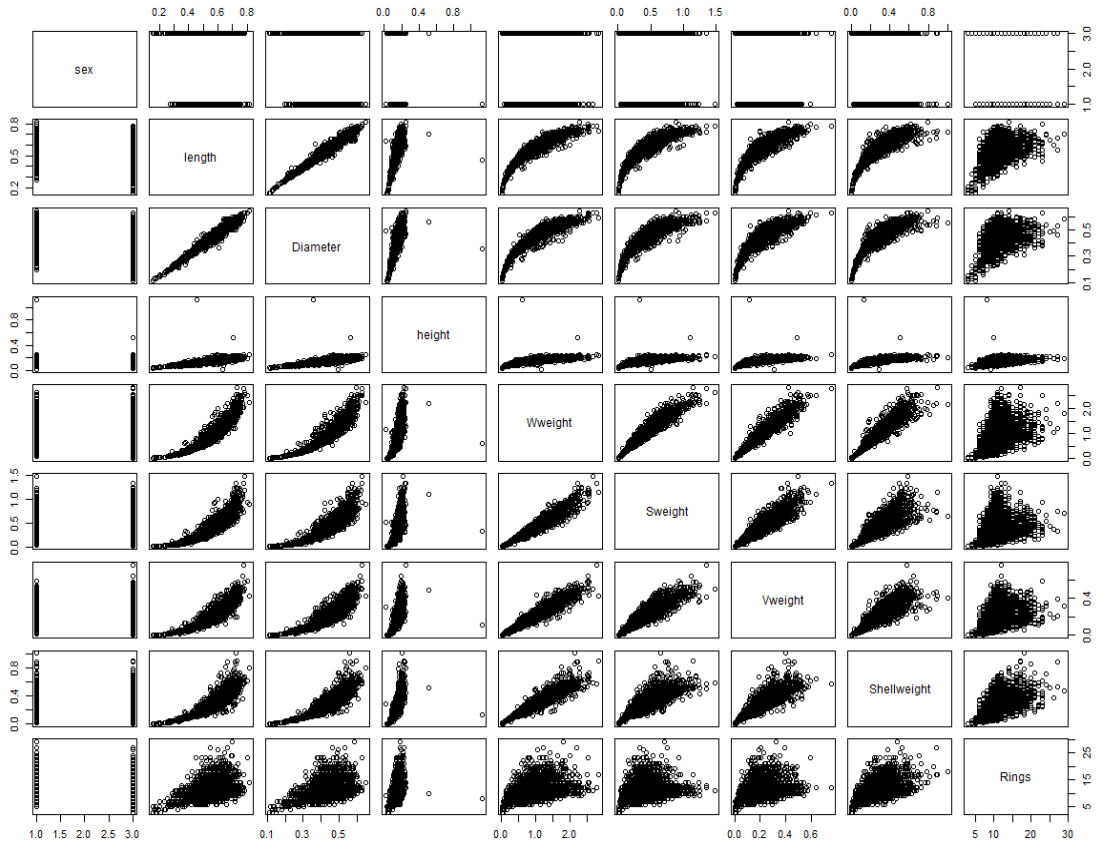
To be able to accommodate the data mining technique of logistic regression, we only considered the Male and the Female observations from the Gender column (dropping the *infant* class). After all the above mentioned steps we have the number of observations at 2835.

(II) Sequence of observations & Selection of test data:

In order to get a better or fair test data from original data, we randomized the sequence of all observations and assigned first 60% of data as training data, 20 % of data as validation data and the rest of data as test data. The portion is 60%, 20% and 20% for training data, validation data and test data, respectively.

This data has been split into training, test and validation data.

3. Data Exploration:



(The scatterplot will full variables and response variables)

Since all no methods require the normality assumption, we don't need to transform variables or normalize data. However, the Naive Bayes method requires independence between predictors. We created a scatterplot for the original data to see any multicolinearity in the predictor variables. Length, Diameter, Whole weight, Viscera weight and Shell weight are highly correlated. We decided to delete those variables and use the rest (just for Naïve Bayes).

4. Model:

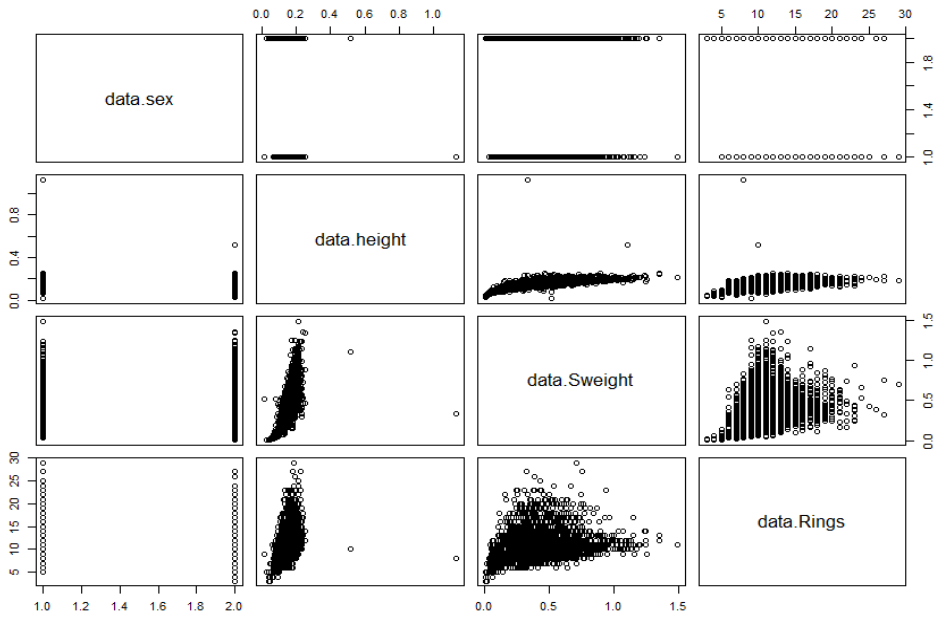
In order to maintain independence from the time order (this is not time series data anyway), we have shuffled all the observations before making the training, test and validation data splits. After that we ran 4 different models on the training data, namely:

A. Naïve Bayes **B**. Logistic Regression **C**. CaRT **D**. K-Nearest Neighbor

For each method, R has package to implement each method for us. We use R to do all the procedures.

A. **NaïveBayes:**

First, we deleted highly correlated variables. We keep height, Shucked weight, Rings.



(The correlation plot after we deleted some variables.)

We use 60% of data as training data, and 20 % of data is validation data.

Column - True, Row -predicted	F	M	Total
F	59	63	*122*
M	205	244	*449*
Total	*264*	*307*	*571*

(Table A)

The accuracy of using this method is 0.530648. NaiveBayes is good at predicting at Male cases and bad at female case.

B. **Logistic regression:**

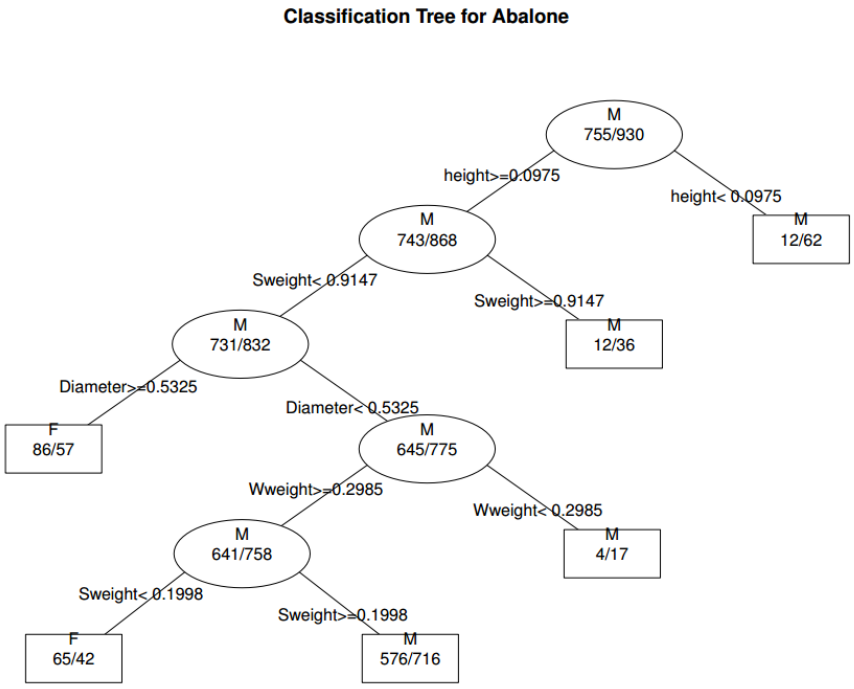
There is no statistical assumption under Logistic regression, so, we did not change the variables.

Column - True, Row -predicted	F	M	Total
F	59	52	*111*
M	205	255	*460*
Total	*264*	*307*	*571*

(Table B)

The accuracy is 0.5492228, best cut off is at probability 0.5.

C. **Decision Tree**



(The full tree from training data)

We use the Training data to plot the tree. The package called rpart in R plotted for us. It generated the best pruned tree.

When we put the Validation data into this tree. The accuracy rate 0.5376532.

Column - True, Row - Predicted	F	M	Total
F	53	45	*98*
M	211	262	*473*
Total	*264*	*307*	*571*

(Table C)

D. **K-Nearest Neighbor. (K is 7)**

We used Euclidean distance to calculate. We didn't delete any of the variables. The accuracy is 0.5288967

Column - True, Row -predicted	F	M	Total
F	109	114	*223*
M	155	193	*348*
Total	*264*	*307*	*571*

(Table D)

5. Results:

Naive	Tree	Logistic	K-NN
0.530648	0.5376532	0.5499124	0.588967

The table concludes all the results from 4 methods. Logistic regression gave us highest rate, and K-nearest neighbor gave the least result.

We found out most methods are good at predicting males . K-nearest neighbor performed evenly on predicting males and females. In this case, there are more male observations than female observations. We think that this is related to the performance of each method.

6. Conclusions:

The original dataset was used for age prediction in the abalone which was based on the number of rings (as opposed to gender prediction that was done in this project). The past prediction models used on the data had an accuracy of 65.61% (cascade-correlation). So, we think we got very decent prediction.

The 4 methods used were giving an accuracy in same ball park of 51-55% and we employed a cross validation method to choose the best model.

Cross-validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.

We decided to test 10 combinations of training data set and validation data set. For example, the training data is 20% of data and validation data is 60% of data. Then, we averaged ten power rates to see which method generates the highest accuracy and stable result. We found out that Logistic regression is the best model. The averaged accuracy is 0.56.

Naïve power	**Logistic Power**	**Tree power**	**K-NN power**
0.535529	0.5600468	0.542756	0.516107

If we use Logistic Regression model to test the test data, the accuracy rate is 0.549228.

7. Appendices:

Please see the attached file for R code .

Table A.

```
Naive Bayes Classifier for Discrete Predictors
```

```
Call:
naiveBayes.default(x = trainData[, 2:4], y = trainData[, 1])

A-priori probabilities:
trainData[, 1]

        F         M
0.4480712 0.5519288

Conditional probabilities:

              data.height
trainData[, 1]      [,1]       [,2]
             F 0.1576159 0.02895067
             M 0.1524570 0.03482404

              data.Sweight
trainData[, 1]      [,1]      [,2]
             F 0.4497828 0.1996874
             M 0.4387710 0.2249638

              data.Rings
trainData[, 1]     [,1]     [,2]
             F 11.19735 3.115716
             M 10.67849 2.853976
```

***

tables *A list of tables, one for each predictor variable. For each categorical variable a table giving, for each attribute level, the conditional probabilities given the target class. For each numeric variable, a table giving, for each target class, mean and standard deviation of the (sub-)variable.*

Table B.

```
Call:  glm(formula = sex ~ ., family = binomial(logit), data =
trainData)

Coefficients:
(Intercept)       length     Diameter       height      Wweight
Sweight
    2.79073     -2.38368     -3.82736     -0.91203      0.38191
2.24023
    Vweight  Shellweight        Rings
   -1.61620     -0.67096     -0.01648

Degrees of Freedom: 1684 Total (i.e. Null);  1676 Residual
Null Deviance:     2318
Residual Deviance: 2277       AIC: 2295
```

Table C.

```
n= 1685

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1685 755 M (0.4480712 0.5519288)
   2) height>=0.0975 1611 743 M (0.4612042 0.5387958)
     4) Sweight< 0.91475 1563 731 M (0.4676903 0.5323097)
       8) Diameter>=0.5325 143  57 F (0.6013986 0.3986014) *
       9) Diameter< 0.5325 1420 645 M (0.4542254 0.5457746)
        18) Wweight>=0.2985 1399 641 M (0.4581844 0.5418156)
          36) Sweight< 0.19975 107  42 F (0.6074766 0.3925234) *
          37) Sweight>=0.19975 1292 576 M (0.4458204 0.5541796) *
        19) Wweight< 0.2985 21   4 M (0.1904762 0.8095238) *
     5) Sweight>=0.91475 48  12 M (0.2500000 0.7500000) *
   3) height< 0.0975 74  12 M (0.1621622 0.8378378) *
```

Pruned tree

```
n= 1685

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 1685 755 M (0.4480712 0.5519288) *
```

Table D.

Prediction:

```
 [1] M F M M M M M F F M F M M F M F M M M F F M M M F M M M M M M F M F M F F M M M
 [41] M F M M M M M M M F M M F M F M M F M F F M M F F F M M M F M M F M F F F
 [81] F F M F F M M F M F M M M F M M M M F M M M M F M M M F M M M M M M M M
[121] F M M M M F M M M F F F M F M F F F M M M M M F F M M F F F M M M F F M M M F
[161] F F M M F F M F M M F F M M M M M M M F M M M F M M M F M M M F F F F F F F M F
[201] M M M M F M M F M M M M M M M M M M M M M M M M M M F M M M M F M M F M F F M M
[241] M F M F F M M M M F M F M F M M M M F M M M F F M M M M M M M M F M M M M F M M
[281] M M F F F M M F M M M M F F M M M F M M F M M M F F M M F M F F F F M M M F M M
[321] F M F M M M F M F F F F M F M F M M F F M F F M F M F M M F M F F F F F M F M
[361] F M M F M M F F F F F M M M M M F M M M M M M M M M M F F M F M M M M M F F F M
[401] M M M M M M M M M M F M F M M F M F F M F M M F M F F M F M F M M M M M M F M F
[441] F M M F M F M F M F F F M F M M M F F M M F M F F M M M F M M M F F F M F F M F
[481] F M M F M F F M F M F M M F F F F F F M M M M F M M M F M M M M M M F M F F F M
[521] F F M M F M M M M F F F F M F M F M M M M F F F M F F F F F F M M M M M M F M F
[561] M F M M M F M M F M F
```

```
Probability:
  [1] 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857
  [9] 0.5714286 0.5714286 0.7142857 0.8571429 0.8571429 0.5714286 0.7142857 0.5714286
 [17] 0.5714286 0.7142857 0.5714286 0.5714286 0.5714286 0.7142857 0.7142857 0.7142857
 [25] 0.7142857 0.5714286 0.7142857 0.5714286 0.8571429 0.5714286 0.5714286 0.5714286
 [33] 1.0000000 0.5714286 0.7142857 0.8571429 0.5714286 0.7142857 0.8571429 0.5714286
 [41] 0.8571429 0.5714286 0.5714286 0.5714286 0.7142857 0.5714286 0.5714286 0.7142857
```

```
 [49] 0.7142857 0.8571429 0.8571429 0.7142857 0.7142857 0.7142857 0.5714286 0.8571429
 [57] 0.5714286 0.5714286 0.7142857 0.7142857 0.5714286 0.7142857 0.5714286 0.5714286
 [65] 0.5714286 0.7142857 0.8571429 0.7142857 0.7142857 0.5714286 0.5714286 0.5714286
 [73] 0.7142857 0.5714286 0.8571429 0.5714286 0.8571429 0.7142857 0.7142857 0.5714286
 [81] 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857 0.7142857 0.8571429
 [89] 0.7142857 0.7142857 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857
 [97] 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857 0.7142857 0.8571429
[105] 0.5714286 1.0000000 0.7142857 0.8571429 0.5714286 0.8571429 0.5714286 0.5714286
[113] 0.7142857 0.5714286 1.0000000 0.7142857 0.5714286 0.8571429 0.5714286 0.7142857
[121] 0.5714286 0.7142857 0.7142857 0.7142857 0.7142857 0.5714286 0.7142857 0.5714286
[129] 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857 0.5714286
[137] 0.7142857 0.5714286 0.7142857 0.5714286 0.7142857 0.5714286 0.5714286 0.7142857
[145] 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.8571429 0.5714286 0.5714286
[153] 0.5714286 0.5714286 0.8571429 0.8571429 0.5714286 0.7142857 0.5714286 0.7142857
[161] 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857 0.8571429 0.7142857 0.5714286
[169] 0.8571429 0.7142857 0.7142857 0.5714286 1.0000000 0.8571429 0.5714286 0.7142857
[177] 0.7142857 0.7142857 0.5714286 0.5714286 0.7142857 0.5714286 0.8571429 1.0000000
[185] 0.7142857 0.8571429 0.7142857 0.8571429 0.5714286 0.8571429 0.5714286 0.5714286
[193] 0.5714286 0.5714286 0.8571429 0.7142857 0.7142857 0.5714286 0.7142857 0.5714286
[201] 0.5714286 0.5714286 0.8571429 0.7142857 0.7142857 0.5714286 0.5714286 0.5714286
[209] 0.7142857 0.7142857 0.5714286 0.5714286 0.7142857 0.5714286 0.8571429 0.5714286
[217] 0.5714286 1.0000000 0.8571429 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286
[225] 0.5714286 0.5714286 0.8571429 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286
[233] 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857
[241] 0.5714286 0.5714286 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857
[249] 0.7142857 0.5714286 0.8571429 0.5714286 0.5714286 0.5714286 0.8571429 0.8571429
[257] 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857 0.5714286 0.7142857
[265] 0.7142857 0.7142857 0.5714286 0.5714286 0.8571429 0.5714286 0.5714286 0.5714286
[273] 0.5714286 0.7142857 0.7142857 0.5714286 0.7142857 0.5714286 0.7142857 0.7142857
[281] 0.8571429 0.5714286 0.7142857 0.5714286 0.5714286 0.8571429 0.7142857 0.7142857
[289] 0.5714286 0.7142857 0.7142857 0.7142857 0.5714286 0.7142857 0.5714286 0.7142857
[297] 0.5714286 0.5714286 1.0000000 0.7142857 0.7142857 0.7142857 0.5714286 0.5714286
[305] 0.7142857 0.5714286 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286
[313] 0.7142857 0.7142857 0.8571429 0.7142857 0.7142857 0.7142857 0.7142857 0.5714286
[321] 0.5714286 0.5714286 0.5714286 0.7142857 0.5714286 0.7142857 0.8571429 0.5714286
[329] 0.5714286 0.5714286 0.7142857 0.5714286 0.5714286 0.7142857 0.8571429 0.5714286
[337] 0.5714286 0.8571429 0.5714286 0.8571429 0.7142857 0.5714286 0.5714286 0.7142857
[345] 0.7142857 0.7142857 0.8571429 0.5714286 0.8571429 0.7142857 0.5714286 0.7142857
[353] 0.5714286 0.7142857 0.5714286 0.7142857 0.7142857 0.7142857 0.5714286 0.7142857
[361] 0.5714286 1.0000000 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286
[369] 0.5714286 0.5714286 0.5714286 0.7142857 0.7142857 0.5714286 0.5714286 0.5714286
[377] 0.7142857 0.5714286 0.7142857 0.5714286 0.5714286 0.8571429 0.7142857 0.5714286
[385] 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857 0.5714286 0.5714286
[393] 0.5714286 0.5714286 0.5714286 0.7142857 0.5714286 0.8571429 0.5714286 0.7142857
[401] 0.8571429 0.7142857 0.8571429 0.5714286 0.8571429 0.5714286 0.8571429 0.5714286
[409] 0.8571429 0.7142857 0.5714286 0.5714286 0.5714286 0.7142857 0.8571429 0.5714286
[417] 0.5714286 0.8571429 0.5714286 0.7142857 0.5714286 1.0000000 0.5714286 0.5714286
[425] 0.5714286 0.5714286 0.5714286 0.7142857 0.7142857 0.8571429 0.8571429 0.5714286
[433] 0.7142857 0.5714286 1.0000000 0.8571429 0.5714286 0.5714286 0.5714286 0.5714286
[441] 0.5714286 0.5714286 0.7142857 0.5714286 0.5714286 0.5714286 0.7142857 0.7142857
[449] 0.5714286 0.5714286 0.7142857 0.5714286 0.7142857 0.5714286 0.7142857 0.7142857
[457] 0.5714286 0.5714286 0.5714286 0.7142857 0.7142857 0.5714286 0.7142857 0.5714286
[465] 0.7142857 0.5714286 0.8571429 0.7142857 0.7142857 0.7142857 0.7142857 0.5714286
[473] 0.5714286 0.5714286 0.5714286 0.7142857 0.5714286 0.7142857 0.5714286 0.5714286
[481] 0.8571429 0.7142857 0.5714286 0.8571429 0.5714286 0.5714286 0.5714286 0.5714286
[489] 0.5714286 0.7142857 0.7142857 0.5714286 0.8571429 0.5714286 0.7142857 0.5714286
[497] 0.8571429 0.5714286 0.5714286 0.7142857 0.7142857 0.5714286 0.5714286 0.7142857
[505] 0.7142857 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.7142857
[513] 0.7142857 0.7142857 0.5714286 0.7142857 0.5714286 0.7142857 0.7142857 0.5714286
[521] 0.5714286 0.5714286 0.5714286 0.7142857 0.5714286 0.7142857 0.5714286 0.7142857
[529] 0.7142857 0.7142857 0.5714286 0.7142857 0.5714286 0.5714286 0.7142857 0.5714286
[537] 0.7142857 0.7142857 0.5714286 0.8571429 0.5714286 0.7142857 0.7142857 0.5714286
[545] 0.7142857 0.5714286 0.7142857 0.7142857 0.5714286 0.5714286 0.5714286 0.8571429
[553] 0.7142857 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286 0.5714286
[561] 1.0000000 0.5714286 1.0000000 0.7142857 0.8571429 0.5714286 0.7142857 0.7142857
[569] 0.5714286 0.8571429 0.5714286
```