Linear Discriminant Analysis using R

Dr. James. Ready

January 14, 2020

Linear Discriminant Analysis (LDA) is one of several statistical methods used in analyzing data when the dependent variable is categorical and the independent variables are interval. LDA looks for linear combinations of the independent variables to best explain the data and predict the different classes (Malhotra, 2019). Discriminant scores are calculated for each observation for each class based on these linear combinations. LDA is often used in marketing and political science.

There are five objectives of a discriminant analysis -

- Develop the linear discriminant functions
- Examine whether significant differences exist between or among the groups
- Determine which predictive variables contribute the most to the intergroup differences
- Classify cases to one or more groups based on the values of the predictive variables
- Evaluate the accuracy of the classification

When a dependent variable has two levels or outcomes, the LDA is known as a *two-group descriminant analysis*. When the dependent variable has three or more levels, the LDA is called a *multiple discriminant analysis* (Malhotra, 2019). This document describes the process to perform both using R.

Preparatory Steps

Before beginning, three packages need to be loaded into the R environment -

- tidyverse (Wickham, 2019). {tidyverse} is a set of packages that work in harmony because they share common data representations and 'API' design. For this instruction, only one package in the tidyverse is needed, {readxl}; however, the entire tidyverse can be used in future applications, so we'll load them all.
- MASS (Ripley, Venables, Bates, Hornick, Gebhardt, & Firth, 2019). {MASS} contains functions and datasets supporting Venables and Ripley's "Modern Applied Statistics with S" (4th edition, 2002). These functions have been ported from the S language to the R language by Ripley and others.
- rrcov (Todorov, 2019). {rrcov} contains classical and robust statistics for Multiple Analysis of Variances (MANOVA). In this document, one function from this library will be used.

The following code will check to see if these packages are installed on your computer. If not, the packages will be downloaded from the Comprehensive R Archive Network (CRAN) cloud-based mirror site (https://cloud.r-project.org/) and installed. Once downloaded and installed, the package functions will be loaded into the current R environment for access.

```r
if(!require(tidyverse)){
    install.packages("tidyverse")
}
```

```
## Loading required package: tidyverse
```

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

```
## <U+2713> ggplot2 3.2.1     <U+2713> purrr   0.3.3
## <U+2713> tibble  2.1.3     <U+2713> dplyr   0.8.3
## <U+2713> tidyr   1.0.0     <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1     <U+2713> forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------- tidyverse
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
if(!require(MASS)){
    install.packages("MASS")
}
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
if(!require(rrcov)){
    install.packages("rrcov")
}
```

```
## Loading required package: rrcov
```

```
## Loading required package: robustbase
```

```
## Scalable Robust Estimators with High Breakdown Point (version 1.4-9)
```

```r
# load the packages into the R environment so functions can be accessed
library(tidyverse)
library(MASS)
library(rrcov)
```

Two-group Discriminant Analysis

To understand how to perform a two-group discriminant analysis using R, let's use the Malhotra (2019) Chapter 18 dataset. First, let's start with a research question -

RQ: Which characteristics of a family explain visiting a specific resort in the past two years?

The associated null hypothesis (H_0) would be that no characteristics would explain visiting a specific resort. The alternative hypothesis (H_A) would be there would be characteristics explaining visiting a specific resort.

To begin, read the associated Microsoft Excel file containing the data using the *read_xls* function from the {readxl} package. Once loaded into the R Environment, display some records to confirm the import. The function *file.choose()* allows a user to navigate to where the file was stored when downloaded by the user. The function *head()* displays the first six records for each variable.

```
# import the training data from Malhotra (2019), Table 18.2 (Information on
# Resort Visits) to the dataframe "resortvisits.data"
resortsvisit.data <- readxl::read_xls(file.choose(),
                                      sheet = 1)
# list the first 6 records along with the column headings
head(resortsvisit.data)
```

```
## # A tibble: 6 x 8
##      No Visit Income Travel Vacation Hhsize   Age Amount
##   <dbl> <dbl>  <dbl>  <dbl>    <dbl>  <dbl> <dbl>  <dbl>
## 1     1     1   50.2      5        8      3    43      2
## 2     2     1   70.3      6        7      4    61      3
## 3     3     1   62.9      7        5      6    52      3
## 4     4     1   48.5      7        5      5    36      1
## 5     5     1   52.7      6        6      4    55      3
## 6     6     1   75        8        7      5    68      3
```

In our example, the dependent variable is Visit (1 = Visited; 2 = Did not visit). The independent variables are Income, Travel, Vacation, Household Size (Hhsize), and Age (Malhotra, 2019). Once data is loaded, a user must specify the linear discriminant model by using the *lda* function from the {MASS} package (Ripley, Venables, Bates, Hornick, Gebhardt, & Firth, 2019). The item on the left side of "~" is the dependent variable, and the items on right side of the "~" are the independent variables. A user would place a "+" between the variables since a linear model assumes the independent variables are additive.

```
# Create the model and store the results to the lda.model1 variable. By storing
# the model to a variable, it can be called from memory as long as the R
# environment is active.
lda.model1 <- MASS::lda(Visit ~ Income + Travel + Vacation + Hhsize + Age,
                        data = resortsvisit.data)
# Note: the "MASS:" can be removed if the package is loaded. It's listed here
# for illustration purposes
print(lda.model1)
```

```
## Call:
## lda(Visit ~ Income + Travel + Vacation + Hhsize + Age, data = resortsvisit.data)
##
## Prior probabilities of groups:
##   1   2
## 0.5 0.5
```

```
##
## Group means:
##     Income   Travel Vacation   Hhsize      Age
## 1 60.52000 5.400000 5.800000 4.333333 53.73333
## 2 41.91333 4.333333 4.066667 2.800000 50.13333
##
## Coefficients of linear discriminants:
##                 LD1
## Income   -0.08476710
## Travel   -0.04964455
## Vacation -0.12028129
## Hhsize   -0.42738931
## Age      -0.02454380
```

```
# Note: Omitting the print() function and just typing the model name
# will output the same results.
```

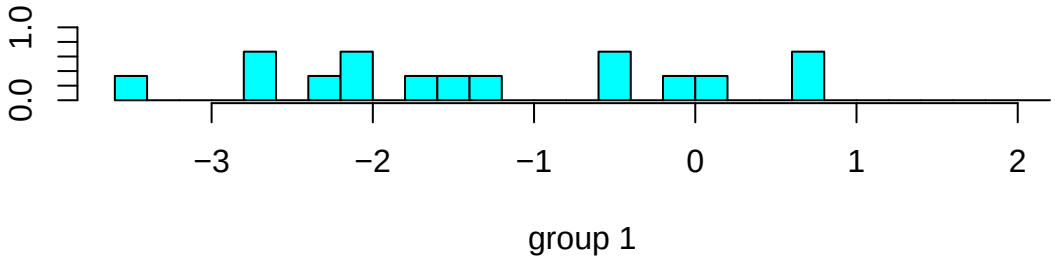
The system-generated output from the *lda* function reports four pieces of information for a two-group discriminant model: the prior probabilities, the group means, and the coefficients of linear discriminants.

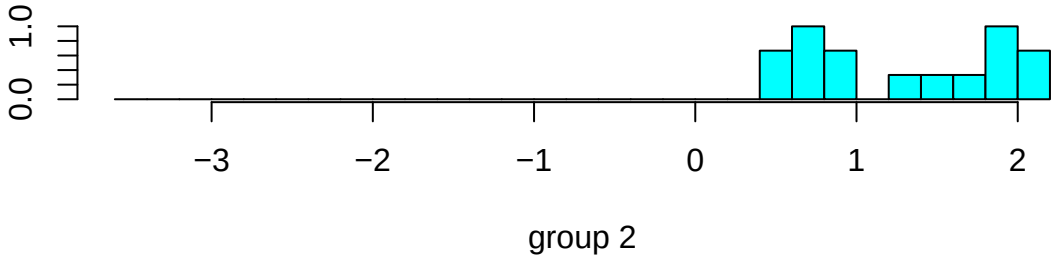
- Prior probability of groups: The probability of randomly selecting an observation from the total dataset. Since we know that 50% of the cases are coded as 1, and 50% of the cases are coded as 2, this output reports the same data.
- Group means: The groups means are the means for each independent variable based upon the dependent variable.
- Coefficients of linear discriminants: These are the coefficient values for each independent variable or discriminant. The larger the coefficient, the more influence the variable has on the classification. For visualization purpose, the formula would be stated (rounded to four digits) -

$(Income * -0.0848) + (Travel * -0.0486) + (Vacation * -0.1203) + (Hhsize * -0.4274) + (Age * -.0245)$

A useful way to understand an LDA model is to *plot* the spacing between the groups -

```
plot(lda.model1)
```





Note how the group 1 is widely dispersed and overlaps a portion of group 2. This overlap can mean that the prediction accuracy of group 1 may not be as strong as the prediction accuracy of group 2, which is more closely aligned and separated from group 1.

Next, examine how well the independent variables contribute to the model via the Wilks Lambda (λ) test statistic from the {rrcov} package. The Wilks scale ranges from 0 to 1, where 0 means total discrimination, and 1 means no discrimination.

```
rrcov::Wilks.test(Visit ~ Income + Travel + Vacation + Hhsize + Age,
                  data = resortsvisit.data)
```

```
##
##  One-way MANOVA (Bartlett Chi2)
##
## data:  x
## Wilks' Lambda = 0.35891, Chi2-Value = 26.13, DF = 5.00, p-value =
## 8.422e-05
## sample estimates:
##     Income   Travel Vacation   Hhsize      Age
## 1 60.52000 5.400000 5.800000 4.333333 53.73333
## 2 41.91333 4.333333 4.066667 2.800000 50.13333
```

The Wilks test statistic is generally reported in the narrative of a study so a reader can be informed about the size and significance of the effect. The Wilks test can be reported as -

> The Wilks Lambda test of the linear discriminant model was significant, $\lambda = 0.359$, $\chi^2(5) = 26.130$, $p < .001$.

Next, examine the prediction ability of the model -

```
predict(lda.model1)
```

```
## $class
##  [1] 2 1 1 1 1 1 2 1 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## Levels: 1 2
##
## $posterior
##              1            2
## 1  0.390660585 0.6093394155
## 2  0.997570081 0.0024299189
## 3  0.996094704 0.0039052958
## 4  0.567482451 0.4325175489
## 5  0.813655349 0.1863446513
## 6  0.999856644 0.0001433564
## 7  0.158564955 0.8414350447
## 8  0.962130719 0.0378692805
## 9  0.980425223 0.0195747767
## 10 0.995692386 0.0043076135
## 11 0.998797621 0.0012023793
## 12 0.998819052 0.0011809478
## 13 0.987435962 0.0125640380
## 14 0.140704302 0.8592956976
## 15 0.796457288 0.2035427118
## 16 0.009028136 0.9909718643
## 17 0.003936652 0.9960633477
## 18 0.029040229 0.9709597706
## 19 0.176385915 0.8236140850
## 20 0.088167063 0.9118329375
## 21 0.010778433 0.9892215674
## 22 0.120614912 0.8793850878
## 23 0.116787874 0.8832121259
## 24 0.206247123 0.7937528769
## 25 0.103151882 0.8968481177
## 26 0.025740607 0.9742593929
## 27 0.158182931 0.8418170689
## 28 0.009434962 0.9905650381
## 29 0.006511104 0.9934888962
## 30 0.004636046 0.9953639536
##
## $x
##           LD1
## 1   0.1721432
## 2  -2.3302165
## 3  -2.1459063
## 4  -0.1051699
## 5  -0.5707714
## 6  -3.4271069
## 7   0.6462858
## 8  -1.2527326
## 9  -1.5155672
## 10 -2.1077806
## 11 -2.6031391
```

```
## 12 -2.6101120
## 13 -1.6900309
## 14  0.7006965
## 15 -0.5283137
## 16  1.8193958
## 17  2.1427973
## 18  1.3590665
## 19  0.5967511
## 20  0.9046844
## 21  1.7500914
## 22  0.7693032
## 23  0.7834709
## 24  0.5218853
## 25  0.8374826
## 26  1.4070864
## 27  0.6473957
## 28  1.8021686
## 29  1.9469432
## 30  2.0791991
```

Note three groups of output are provided: class, posterior, and x values. The class value is the prediction of Visit based on the coefficients calculated. The posterior output shows the probability for the classes. For example, record 1 shows that the model would classify this record as a 61% probability of Visit = 2 versus a 39% probability of Visit = 1. Finally, the $x values are the scores of test cases (Ripley et al., 2019).

Next, compare the actual Visit results with the predicted Visit results by creating a contingency table with the predicted class and actual values.

```
table(predict(lda.model1)$class,
      resortsvisit.data$Visit,
      dnn = c("Prediction", "Actual"))
```

```
##            Actual
## Prediction  1  2
##          1 12  0
##          2  3 15
```

Note the model achieved a 90% *hit ratio* (Malhotra, 2019, p. 560), with 100% accuracy in predicting Visit = 2 (15/15) and 80% accuracy in predicting Visit = 1 (12/15).

Finally, let's perform a cross-validation of the model by importing the holdout sample (Malhotra, 2019, p. 553) to validate the model, and report the results via table.

```
validation.data <- readxl::read_xls(file.choose(),
                                    sheet = 2)
table(predict(lda.model1, newdata = validation.data)$class,
      validation.data$Visit,
      dnn = c("Prediction", "Actual"))
```

```
##            Actual
## Prediction 1 2
##          1 4 0
##          2 2 6
```

Applying the model to the holdout sample resulted in an 80% hit ratio, with a 66.7% accuracy for Visit = 1 (4/6) and 100% accuracy for Visit = 2 (6/6).

Multiple Discriminant Analysis

Multiple Discriminant Analysis (MDA) is similar to LDA; however, instead of two categorical outcomes in a dependent variable, there can be more than two. Using the same Malhotra (2019) data, change the dependent variable from Visit to Amount. Note that Amount has three levels: High, Medium, and Low.

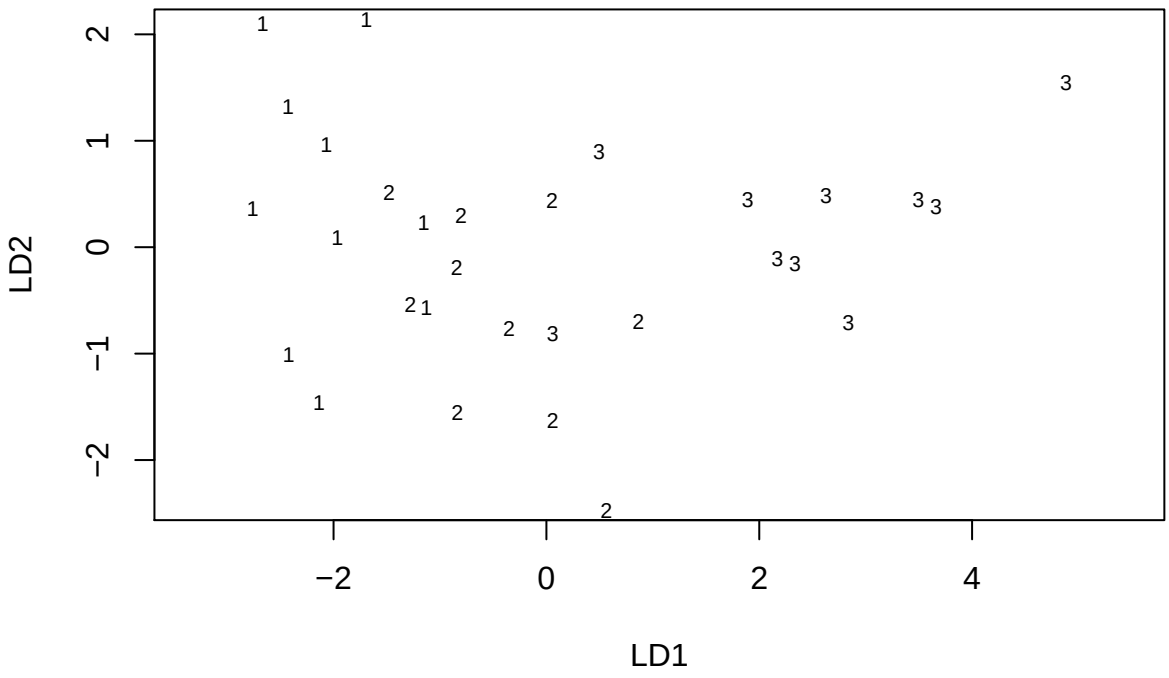
```
lda.model2 <- MASS::lda(Amount ~ Income + Travel + Vacation + Hhsize + Age,
                        data = resortsvisit.data)
lda.model2
```

```
## Call:
## lda(Amount ~ Income + Travel + Vacation + Hhsize + Age, data = resortsvisit.data)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3333333 0.3333333 0.3333333
##
## Group means:
##   Income Travel Vacation Hhsize  Age
## 1  38.57    4.5      4.7    3.1 50.3
## 2  50.11    4.0      4.2    3.4 49.5
## 3  64.97    6.1      5.9    4.2 56.0
##
## Coefficients of linear discriminants:
##                  LD1         LD2
## Income     0.15426584 -0.06197148
## Travel     0.18679766  0.42234297
## Vacation -0.06952264  0.26126524
## Hhsize   -0.12653341  0.10027963
## Age       0.05928055  0.06284206
##
## Proportion of trace:
##    LD1    LD2
## 0.9393 0.0607
```

Similar to the two-group example, the *lda* function outputs prior probabilities of groups, group means, and coefficients of linear discriminants. A fourth output item is also listed: the *proportion of trace*. Proportion of trace identifies the two discriminant functions under the rule *k - 1*. Note that the first discriminant function, LD1, accounts for 93.93% of the explained variance, while the second discriminant function, LD2, accounts for only 6.07% of the explained variance.

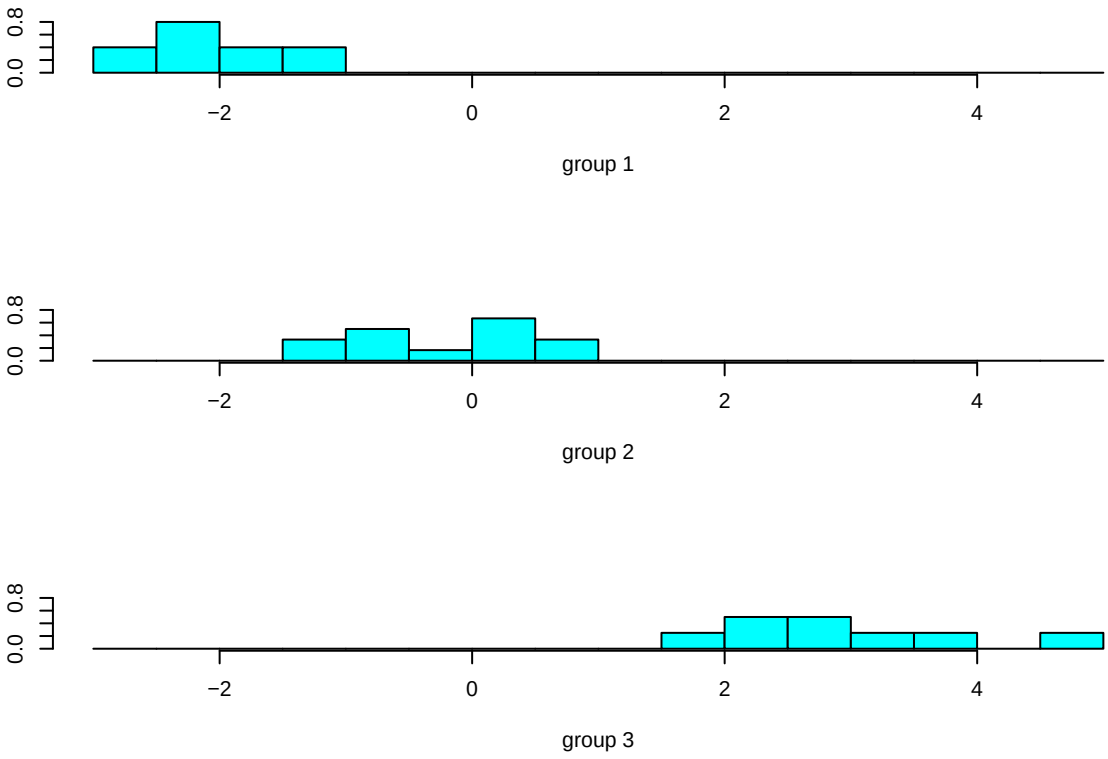
Next, plot the results of the model taking note of the grouping of the dependent variable. The standard output of an MDA is a scatterplot of all groups on function LD1 and LD2.

```
plot(lda.model2)
```



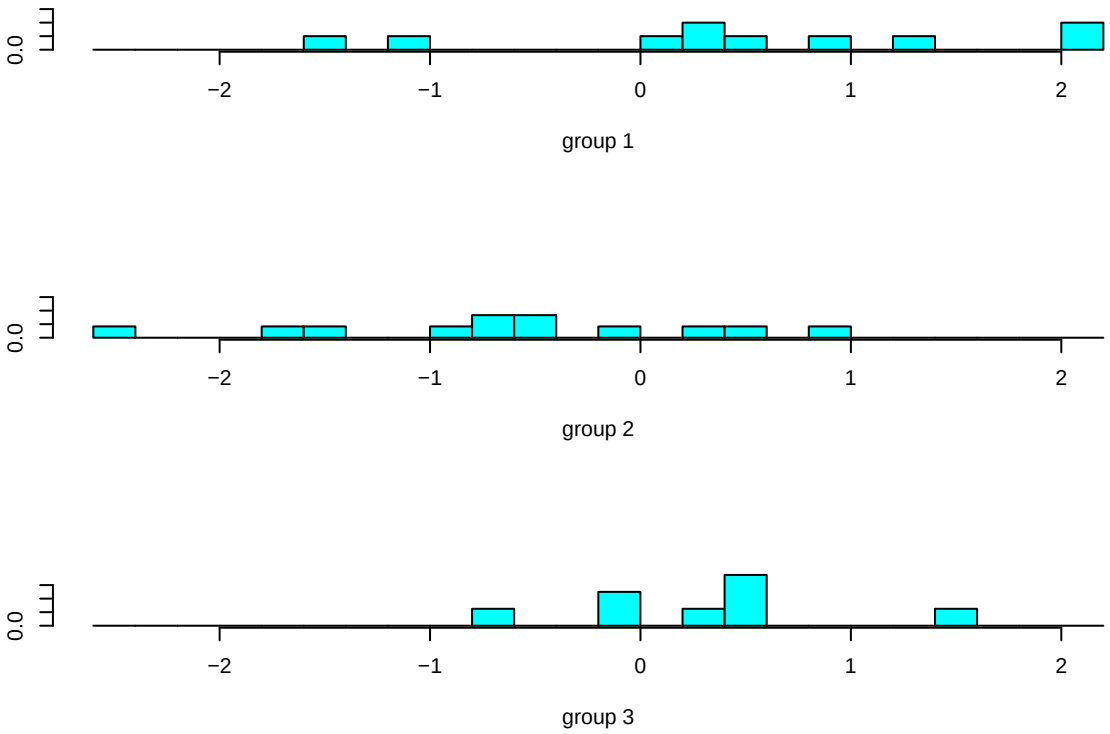
However, to recreate a histogram-type output used in two-group discriminant analysis, a user must use the predict function. First, a histogram of the first discriminant function.

```
# use the *predict* function and the output from $x to plot against the $class
# grouping
ldahist(predict(lda.model2)$x[,1], g = predict(lda.model2)$class)
```



Note how group 1 and group 2 slightly overlap, but group 1 and group 2 do not overlap with group 3. This visually represents the near 94% variance explanation. Doing the same for the second discriminant function depicts the much lower explanation power.

```
ldahist(predict(lda.model2)$x[,2], g = predict(lda.model2)$class)
```



Note how all three groups overlap in some way based on the second discriminant function. Next, examine the results of the Wilks Lambda test -

```
rrcov::Wilks.test(Amount ~ Income + Travel + Vacation + Hhsize + Age,
                  data = resortsvisit.data)
```

```
##
##  One-way MANOVA (Bartlett Chi2)
##
## data:  x
## Wilks' Lambda = 0.16642, Chi2-Value = 44.831, DF = 10.000, p-value =
## 2.333e-06
## sample estimates:
##   Income Travel Vacation Hhsize  Age
## 1  38.57    4.5      4.7    3.1 50.3
## 2  50.11    4.0      4.2    3.4 49.5
## 3  64.97    6.1      5.9    4.2 56.0
```

The model is significant, and can be reported in narrative form as -

> The Wilks Lambda test of the multiple discriminant model was significant, $\lambda = 0.166$, $\chi^2(10) = 44.831$, $p < .001$.

Examine a contingency table with predicted class and actual values -

```
table(predict(lda.model2)$class,
      resortsvisit.data$Amount,
      dnn = c("Prediction", "Actual"))
```

```
##           Actual
## Prediction 1 2 3
##          1 9 1 0
##          2 1 9 2
##          3 0 0 8
```

As depicted by the table, the model achieved a *hit ratio* of 86.7%, with 90% accuracy in predicting Amount = 1 and Amount = 2, and an 80% accuracy in predicting Amount = 3.

Finally, let's examine the predictive accuracy of the MDA by applying it to the holdout sample -

```
table(predict(lda.model2, newdata = validation.data)$class,
      validation.data$Amount,
      dnn = c("Prediction", "Actual"))
```

```
##           Actual
## Prediction 1 2 3
##          1 3 0 1
##          2 1 3 0
##          3 0 1 3
```

Note the individual and group 75% hit ratio.

References:

Malhotra, N. K. (2019). *Marketing research: An applied orientation* (7th ed.). Pearson.

Ripley, B., Venables, B., Bates, D. M., Hornick, K., Gebhardt, A., & Firth, D. (2019, December 20). *Support functions and datasets for Venables and Ripley's MASS*. Retrieved from https://cran.r-project.org/web/packages/MASS/MASS.pdf

Todorov, V. (2019, November 11). *rrcov: Scalable robust estimators with high breakdown point*. Retrieved from https://cran.r-project.org/web/packages/rrcov/rrcov.pdf

Wickham, H. (2019, November 21). *tidyverse: Easily install and load the 'Tidyverse'*. Retrieved from https://cran.r-project.org/web/packages/tidyverse/tidyverse.pdf