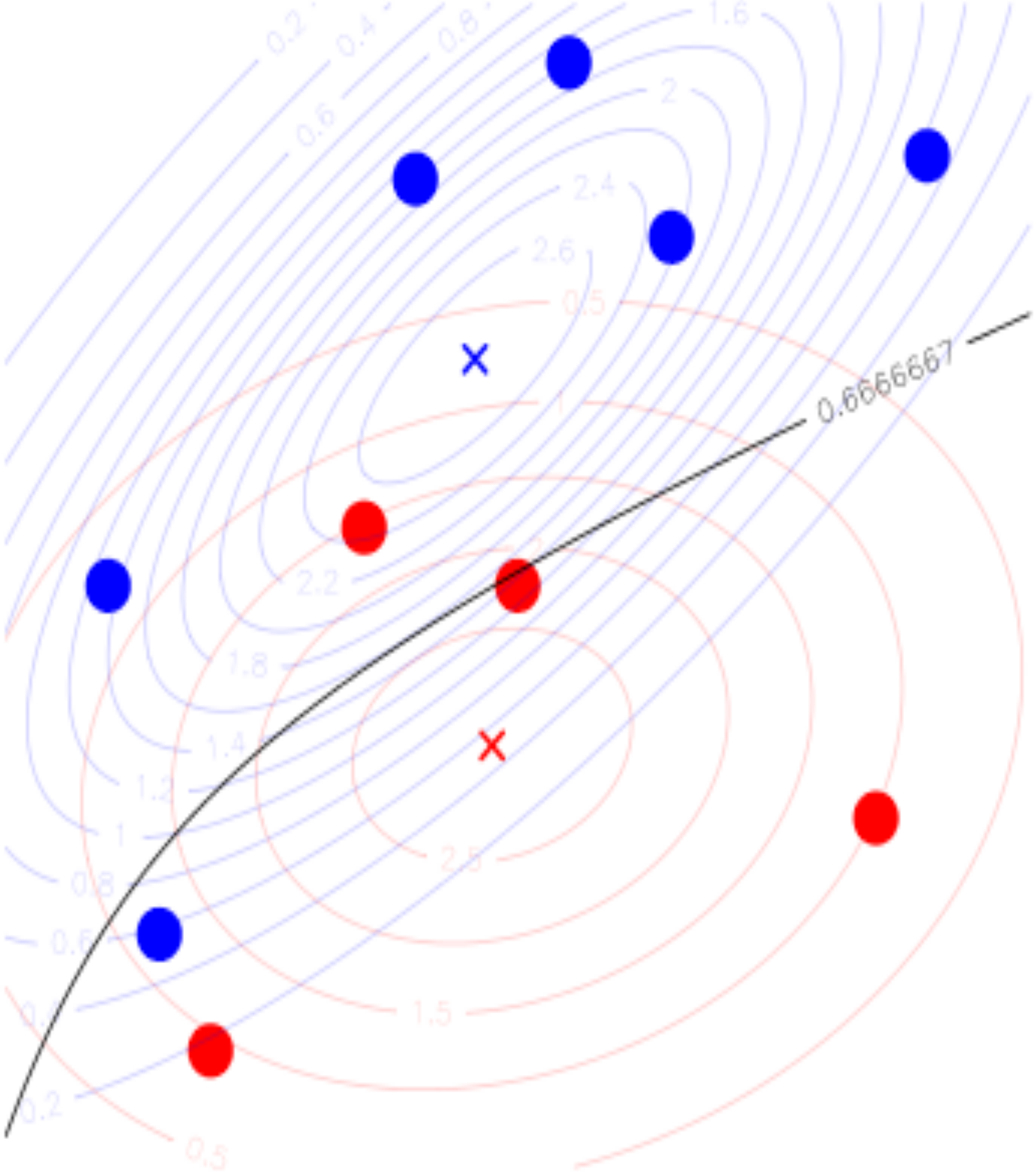
WEEK 9:

INTRODUCTION TO DISCRIMINANT ANALYSIS

Tutorial Week 9

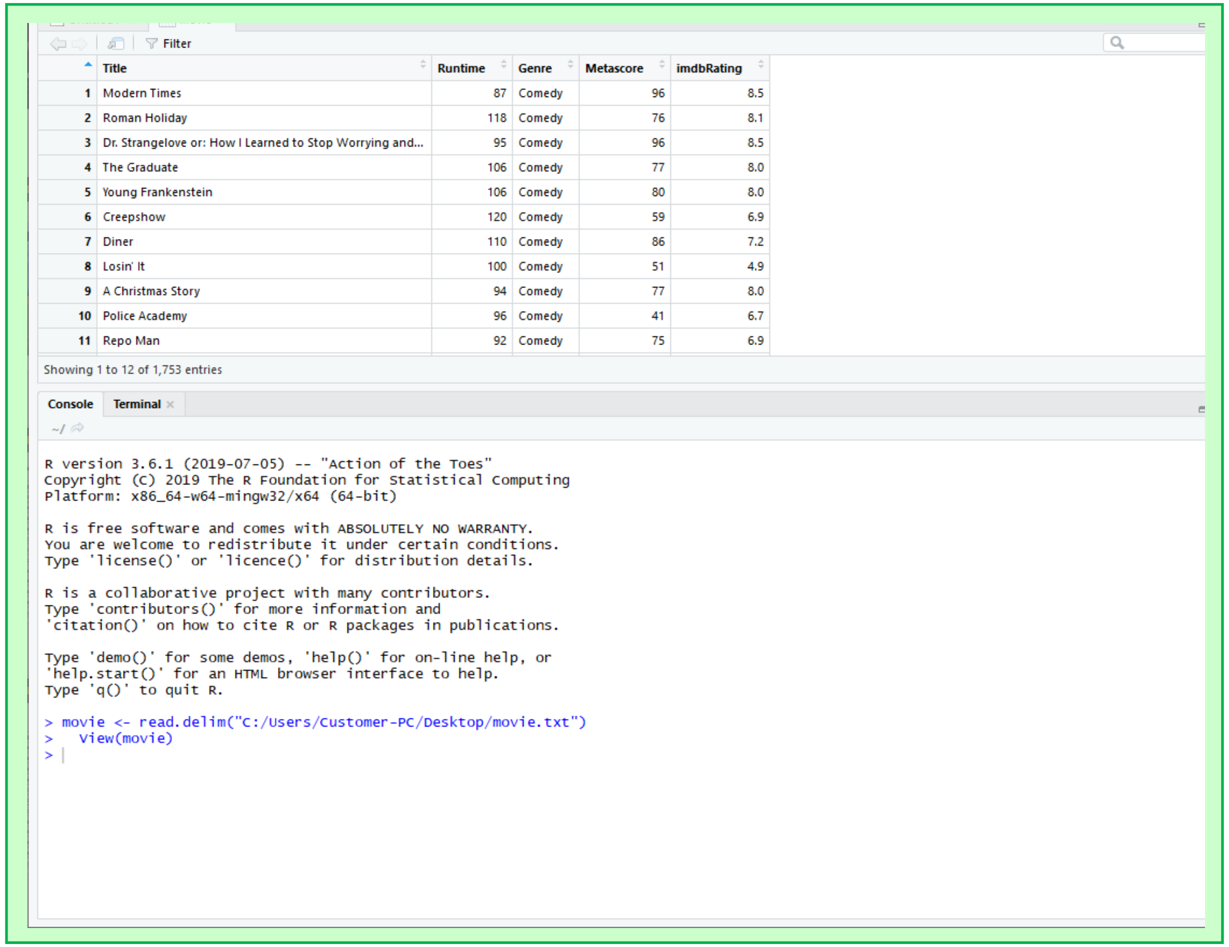
STA30005



DISCRIMINANT ANALYSIS IN R

Data Preparation	Splitting data (train / test)	Applying LDA	Visualisation	Predictions

```
# Load the movie data file (in the tutorial 9 folder)
movie <- read.delim(“LOCATION.txt")

# Show the data
View(movie)

# Install / Load the required packages
install.packages(“tidyverse")
library(tidyverse)
install.packages(“MASS")
library(MASS)
install.packages(“klaR")
library(klaR)

# Set a seed value
set.seed(123)
```



	Title	Runtime	Genre	Metascore	imdbRating
1	Modern Times	87	Comedy	96	8.5
2	Roman Holiday	118	Comedy	76	8.1
3	Dr. Strangelove or: How I Learned to Stop Worrying and...	95	Comedy	96	8.5
4	The Graduate	106	Comedy	77	8.0
5	Young Frankenstein	106	Comedy	80	8.0
6	Creepshow	120	Comedy	59	6.9
7	Diner	110	Comedy	86	7.2
8	Losin' It	100	Comedy	51	4.9
9	A Christmas Story	94	Comedy	77	8.0
10	Police Academy	96	Comedy	41	6.7
11	Repo Man	92	Comedy	75	6.9

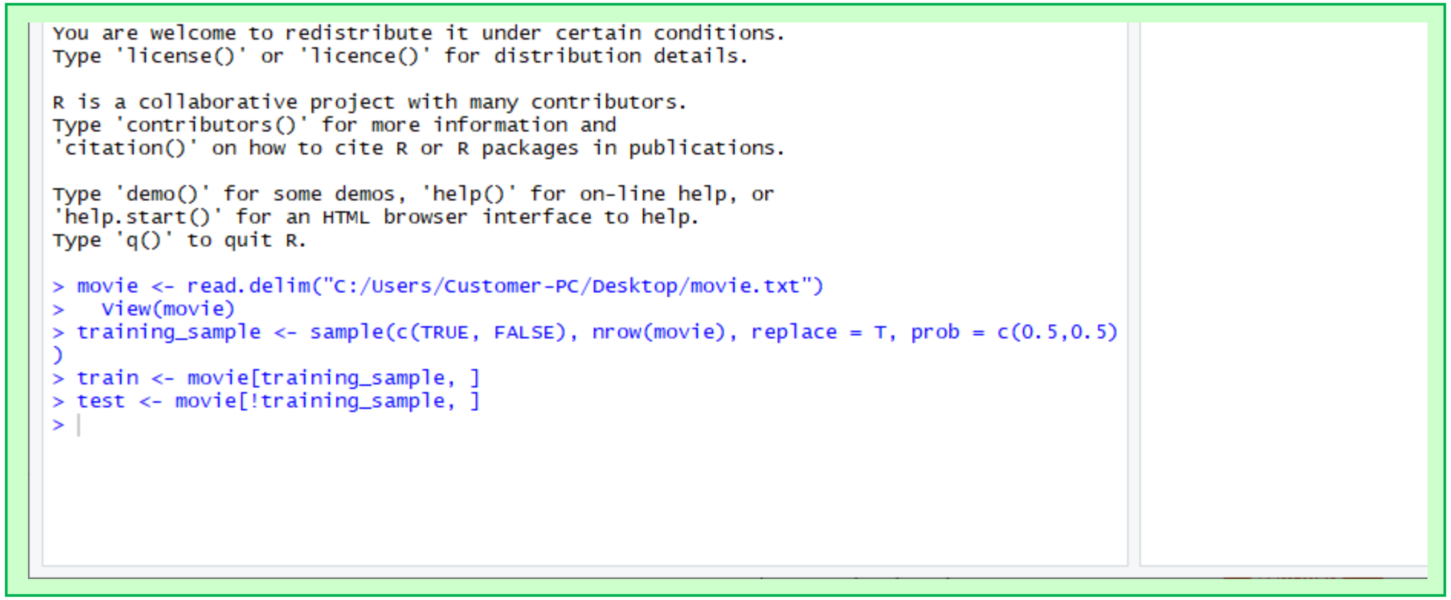
Showing 1 to 12 of 1,753 entries

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> movie <- read.delim("C:/Users/Customer-PC/Desktop/movie.txt")
>   View(movie)
> |
```

DISCRIMINANT ANALYSIS IN R

We can separate the data into two subsets: a training set (building the model) and a testing set (evaluate the accuracy of the model). For convenience sake we will use a 50/50 split, using 50% of the data as the training set and the remaining 50% for the testing set.

```
# Split the data 50/50
training_sample <- sample(c(TRUE, FALSE), nrow(movie), replace = T, prob = c(0.5,0.5))

# Define the training data
train <- SATB[training_sample, ]

# Define the testing data
test <- SATB[!training_sample, ]
```



```
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> movie <- read.delim("C:/Users/Customer-PC/Desktop/movie.txt")
>   View(movie)
> training_sample <- sample(c(TRUE, FALSE), nrow(movie), replace = T, prob = c(0.5,0.5)
)
> train <- movie[training_sample, ]
> test <- movie[!training_sample, ]
> 
```

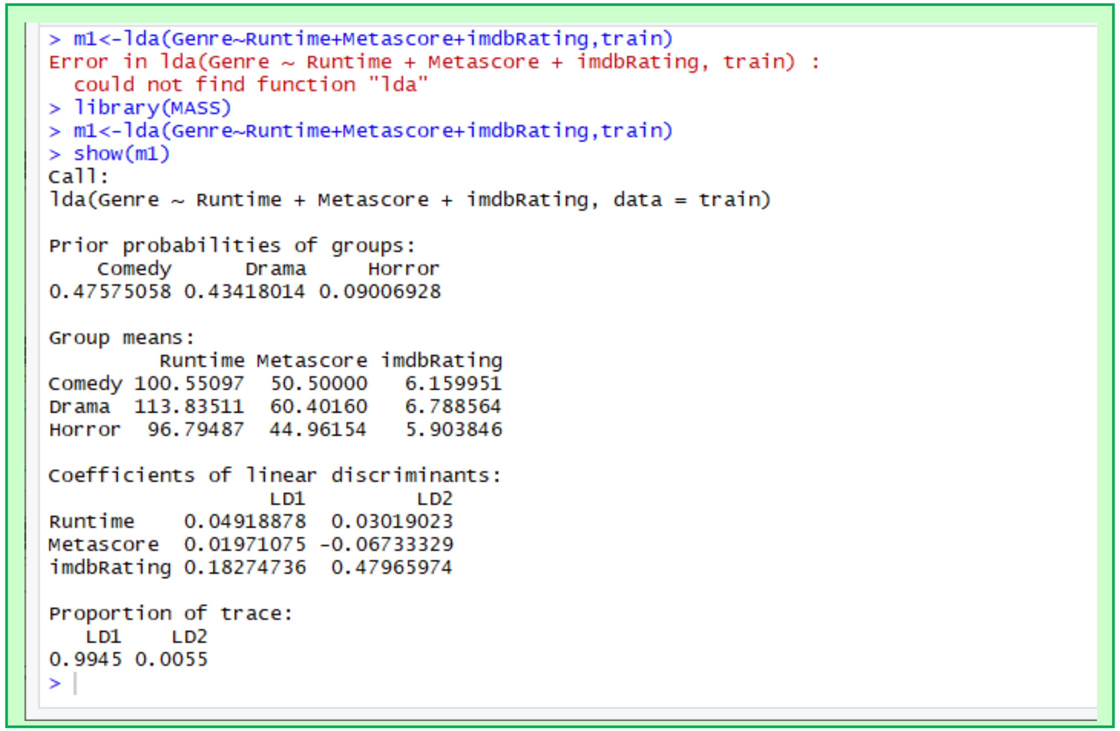
DISCRIMINANT ANALYSIS IN R

# Create an initial LDA model (m1) based upon the data

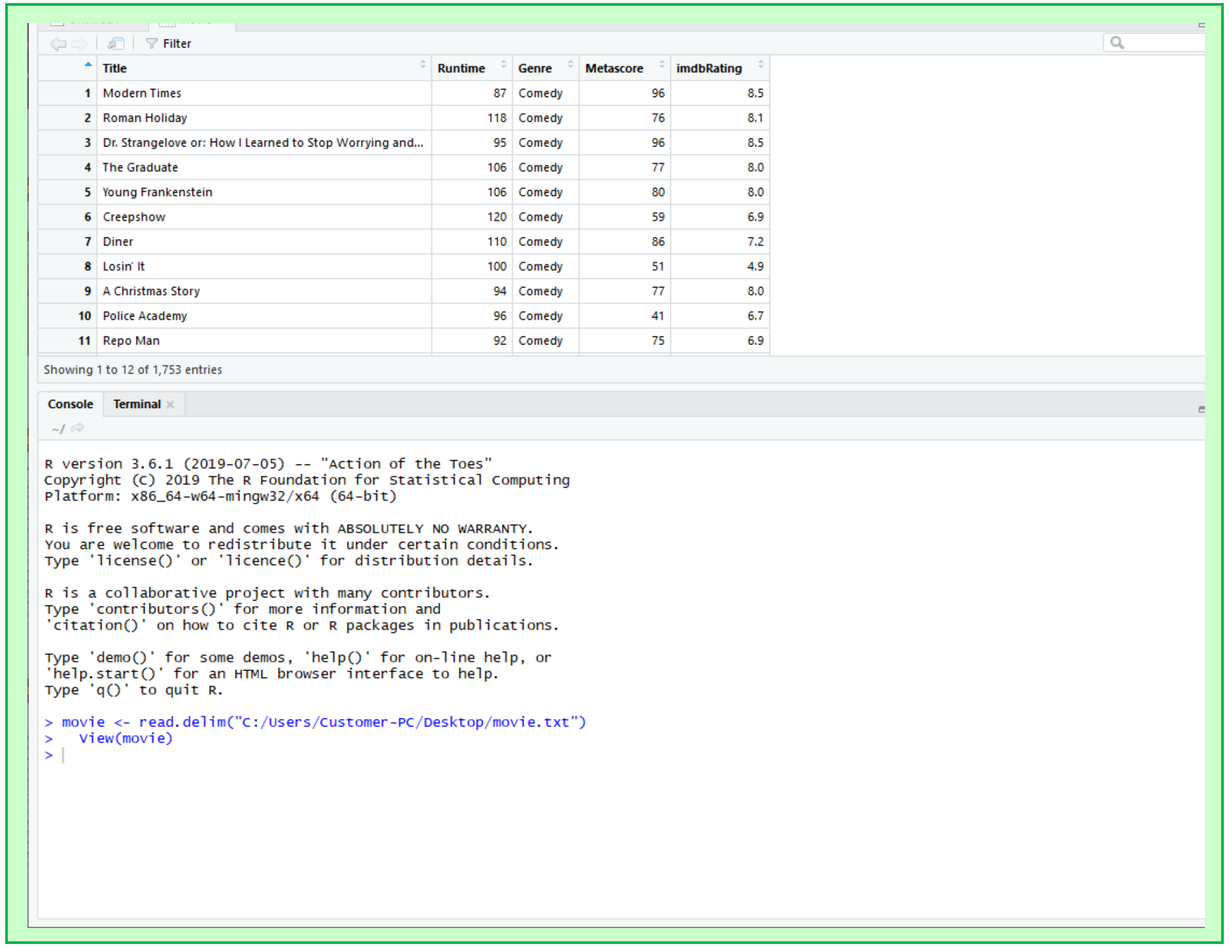
```
m1<-lda(Genre ~ Runtime + Metascore + imdbRating, train)
```

# Show the results

```
show(m1)
```



```
> m1<-lda(Genre~Runtime+Metascore+imdbRating,train)
Error in lda(Genre ~ Runtime + Metascore + imdbRating, train) :
  could not find function "lda"
> library(MASS)
> m1<-lda(Genre~Runtime+Metascore+imdbRating,train)
> show(m1)
Call:
lda(Genre ~ Runtime + Metascore + imdbRating, data = train)

Prior probabilities of groups:
    Comedy      Drama     Horror
0.47575058 0.43418014 0.09006928

Group means:
         Runtime Metascore imdbRating
Comedy 100.55097  50.50000   6.159951
Drama  113.83511  60.40160   6.788564
Horror  96.79487  44.96154   5.903846

Coefficients of linear discriminants:
                  LD1         LD2
Runtime    0.04918878  0.03019023
Metascore  0.01971075 -0.06733329
imdbRating 0.18274736  0.47965974

Proportion of trace:
   LD1    LD2
0.9945 0.0055
> |
```

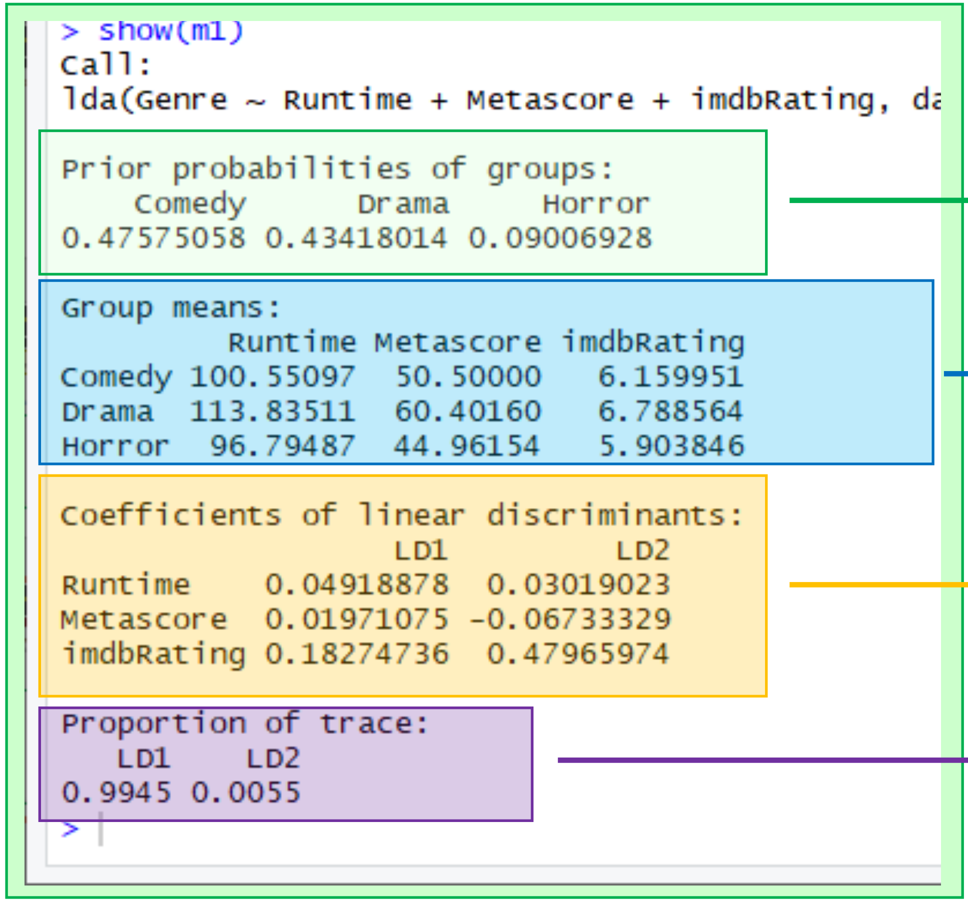


	Title	Runtime	Genre	Metascore	imdbRating
1	Modern Times	87	Comedy	96	8.5
2	Roman Holiday	118	Comedy	76	8.1
3	Dr. Strangelove or: How I Learned to Stop Worrying and...	95	Comedy	96	8.5
4	The Graduate	106	Comedy	77	8.0
5	Young Frankenstein	106	Comedy	80	8.0
6	Creepshow	120	Comedy	59	6.9
7	Diner	110	Comedy	86	7.2
8	Losin' It	100	Comedy	51	4.9
9	A Christmas Story	94	Comedy	77	8.0
10	Police Academy	96	Comedy	41	6.7
11	Repo Man	92	Comedy	75	6.9

Showing 1 to 12 of 1,753 entries

```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> movie <- read.delim("C:/Users/Customer-PC/Desktop/movie.txt")
>   View(movie)
> |
```

DISCRIMINANT ANALYSIS IN R



```
> show(m1)
Call:
lda(Genre ~ Runtime + Metascore + imdbRating, da

Prior probabilities of groups:
    Comedy      Drama     Horror 
0.47575058 0.43418014 0.09006928 

Group means:
         Runtime Metascore imdbRating
Comedy 100.55097  50.50000   6.159951
Drama  113.83511  60.40160   6.788564
Horror  96.79487  44.96154   5.903846

Coefficients of linear discriminants:
                  LD1         LD2
Runtime    0.04918878  0.03019023
Metascore  0.01971075 -0.06733329
imdbRating 0.18274736  0.47965974

Proportion of trace:
   LD1    LD2 
0.9945 0.0055 
> 
```

The Prior probabilities of groups show the probability of randomly selecting an observation from class the total training set (e.g. 47.6% chance to be comedy, 43.4% drama and 9.0% to be horror)

Shows the mean values for the different variables split by the classification factor (in this example: Genre)

LD1 = 0.05(Runtime) + 0.02(Metascore) + 0.18(imdbRating)

LD1 = 0.03(Runtime) – 0.07(Metascore) + 0.48(imdbRating)

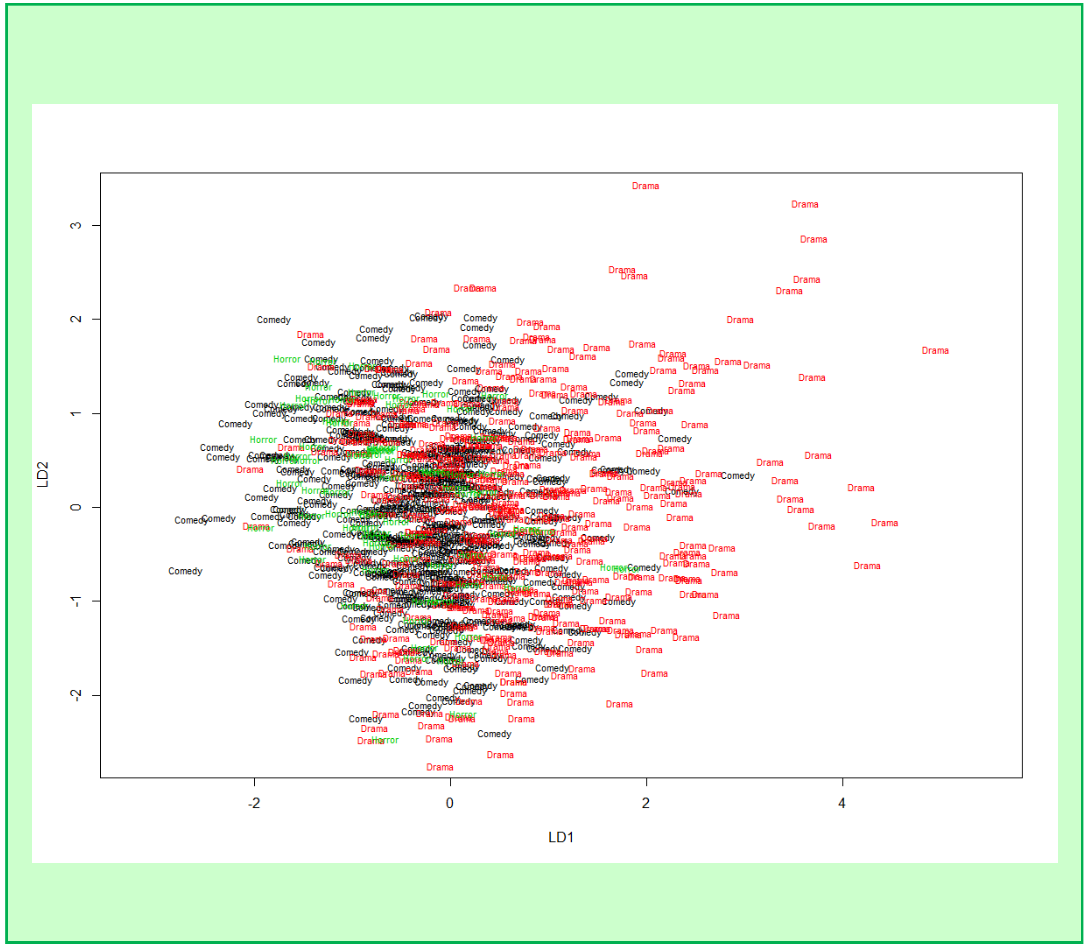
The Proportions of trace describes the proportion of between-class variance that is explained by successive discriminant functions. As you can see LD1 explains 99.45% of the variance.

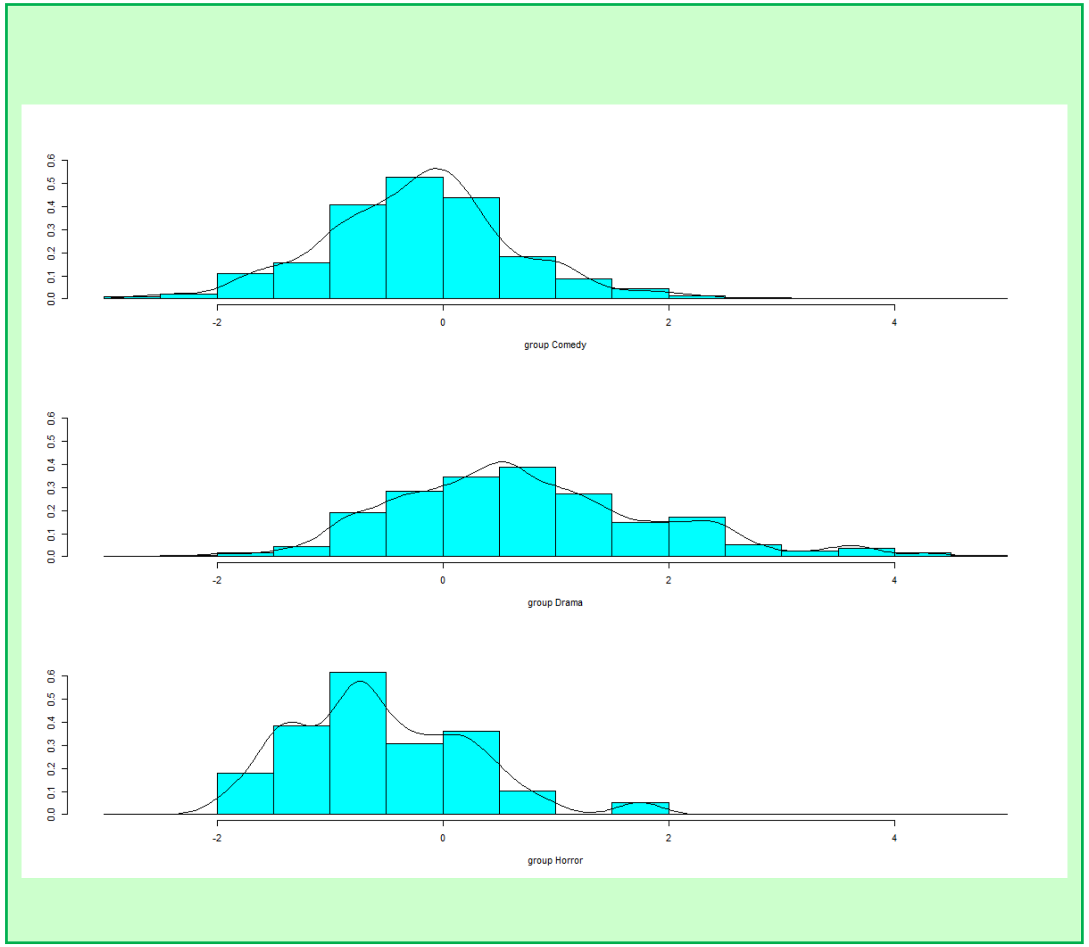
DISCRIMINANT ANALYSIS IN R

Data Preparation | Splitting data (train / test) | Applying LDA | Visualisation | Predictions

```
# Plot LD1 and LD2
plot(m1, col=as.integer(train$Genre))
```

```
# Plot LD1 only
plot(m1, dimen = 1, type = "b")
```



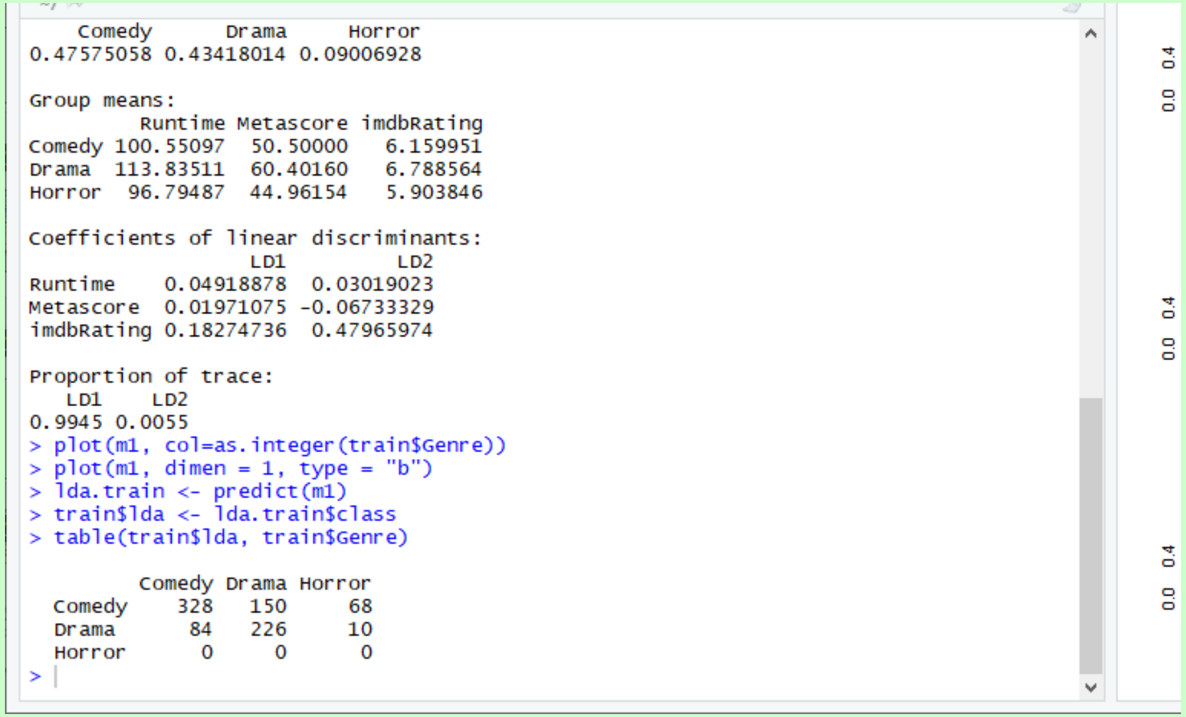


DISCRIMINANT ANALYSIS IN R

Next let’s evaluate the prediction accuracy of our model. First we’ll run the model against the training set used to verify the model fits the data properly by using the command predict. The table output is a confusion matrix with the actual as the rows and the predicted as columns

```
# Compare model against test set
lda.train <- predict(m1)
train$lda <- lda.train$class
table(train$lda, train$Genre)
```

- The total number of correctly predicted observations is the sum of the diagonal (328 + 226 + 0 = 554). So this model fit the training data correctly for 63.97% of cases.
- Verifying the training set doesn’t prove accuracy, but a poor fit to the training data could be a sign that the model isn’t a good one.



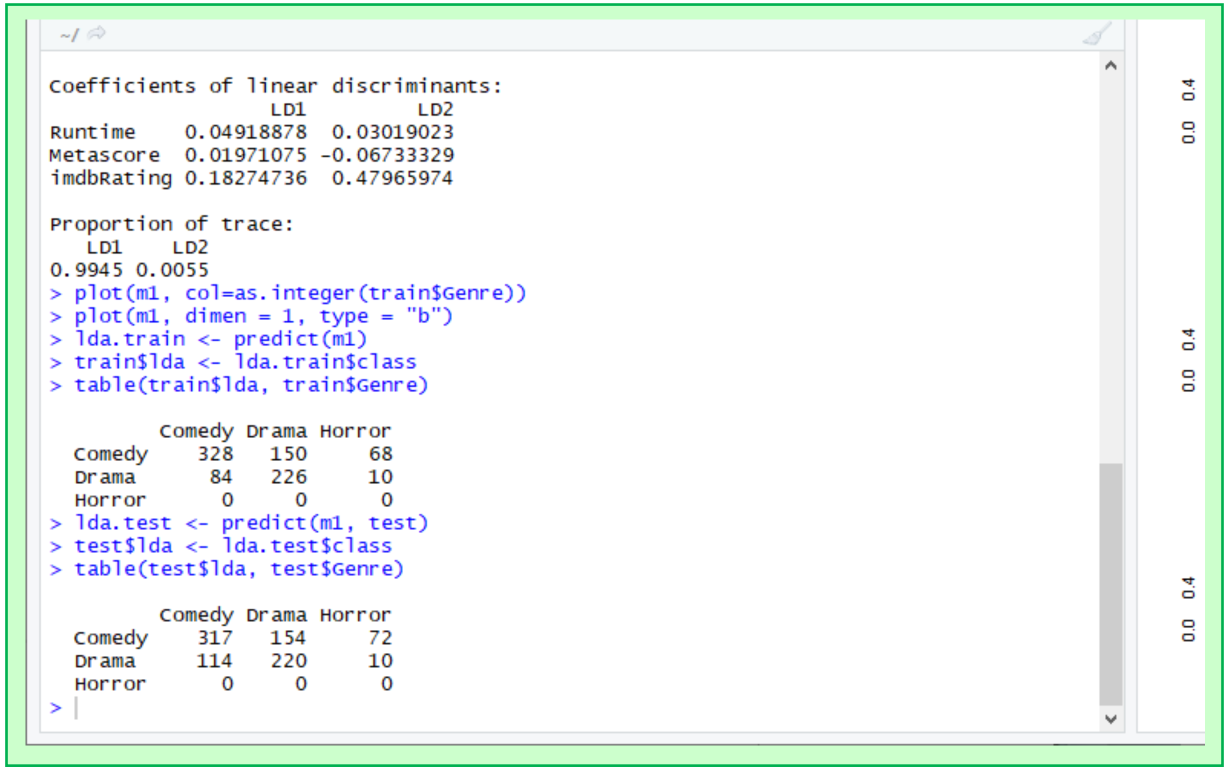
```
    Comedy      Drama      Horror
0.47575058 0.43418014 0.09006928

Group means:
         Runtime Metascore imdbRating
Comedy 100.55097  50.50000   6.159951
Drama  113.83511  60.40160   6.788564
Horror  96.79487  44.96154   5.903846

Coefficients of linear discriminants:
                  LD1         LD2
Runtime    0.04918878  0.03019023
Metascore  0.01971075 -0.06733329
imdbRating 0.18274736  0.47965974

Proportion of trace:
   LD1    LD2
0.9945 0.0055
> plot(m1, col=as.integer(train$Genre))
> plot(m1, dimen = 1, type = "b")
> lda.train <- predict(m1)
> train$lda <- lda.train$class
> table(train$lda, train$Genre)

         Comedy Drama Horror
  Comedy    328   150     68
  Drama      84   226     10
  Horror      0     0      0
>
```

DISCRIMINANT ANALYSIS IN R

Now let’s run our test set against this model to determine its accuracy.

```
# Compare model against test set
lda.test <- predict(m1, test)
test$lda <- lda.test$class
table(test$lda, test$Genre)
```

- The total number of correctly predicted observations is the sum of the diagonal (317 + 154 + 0 = 471).
- The overall accuracy is only 53.1%
- Therefore these three variables are not good at discriminating between movie genres (horror in particular was really bad)
- Can you think of better variables?



```
Coefficients of linear discriminants:
                 LD1         LD2
Runtime    0.04918878  0.03019023
Metascore  0.01971075 -0.06733329
imdbRating 0.18274736  0.47965974

Proportion of trace:
   LD1    LD2 
0.9945 0.0055 
> plot(m1, col=as.integer(train$Genre))
> plot(m1, dimen = 1, type = "b")
> lda.train <- predict(m1)
> train$lda <- lda.train$class
> table(train$lda, train$Genre)
         
          Comedy Drama Horror
  Comedy     328   150     68
  Drama       84   226     10
  Horror       0     0      0
> lda.test <- predict(m1, test)
> test$lda <- lda.test$class
> table(test$lda, test$Genre)
         
          Comedy Drama Horror
  Comedy     317   154     72
  Drama      114   220     10
  Horror       0     0      0
> 
```

DISCRIMINANT ANALYSIS IN R

Now repeat the task with the IRIS data frame (base R data)

[this one discriminates quite well]