"/>

multiDA and genDA

Discriminant Analysis Methods for Large Scale and Complex Datasets

Sarah Romanes @sarah_romanes

12-Jul-2019

bit.ly/SR-useR-2019

Discriminant Analysis

What is Discriminant Analysis?

- Discriminant Analysis (Fisher, 1936) is a ML technique that seeks to find a linear combination of features that separates classes of objects.

- It *strictly* assumes the conditional distribution of the data, given class grouping, is multivariate normal.

- Available through `MASS` package in R with functions `lda` (common covariance) and `qda`.

Issues with DA

Does not work in high dimensions

DA does not work when p>n due to a required covariance inverse being singular.

Solution? **multiDA**

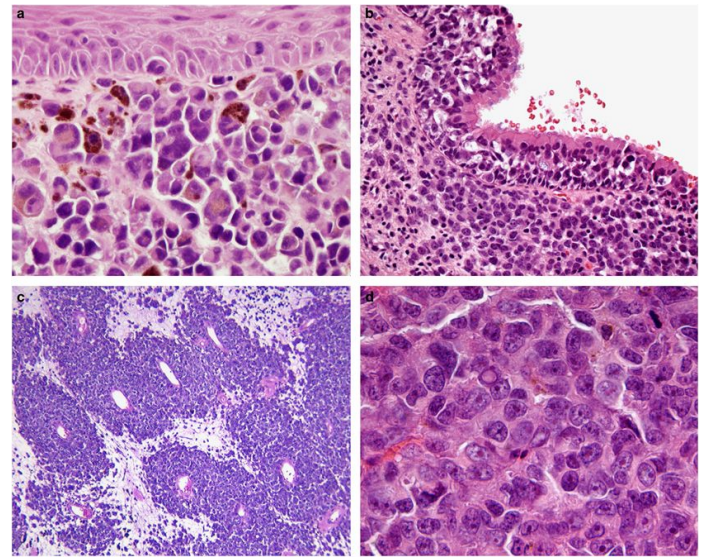
Does not work for non-Gaussian response

Cannot use for count/ skewed/ binary/ mixed response data, etc.

Solution? **genDA**

multiDA

SRBCT data



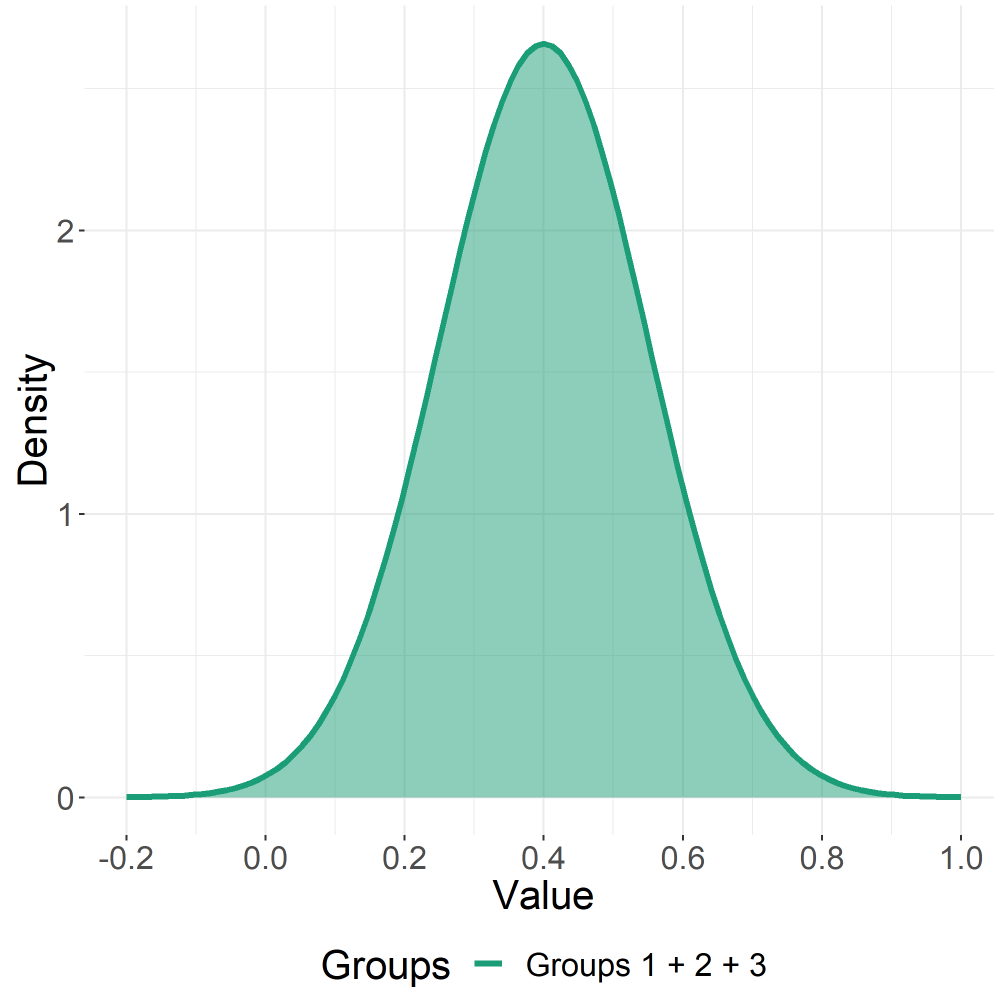
Source: Nature

Pipeline

So, what makes a discriminative feature?

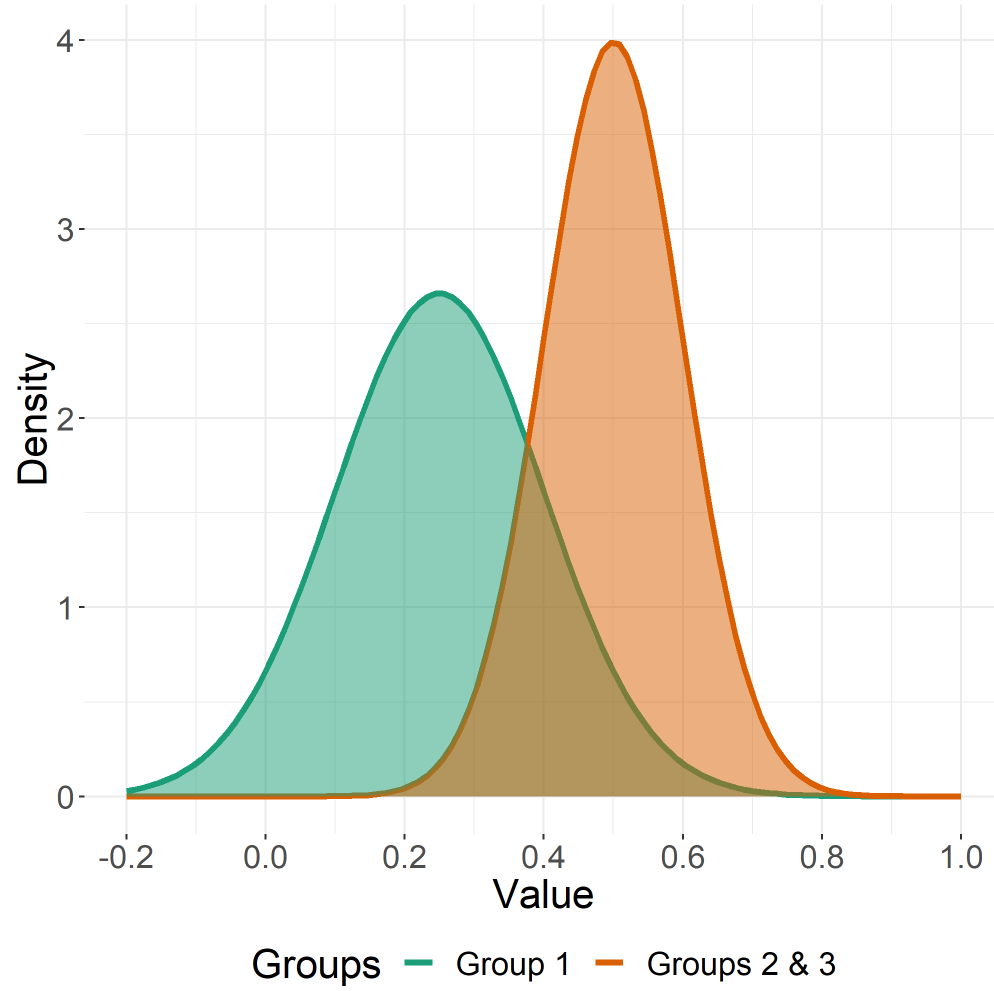
Suppose we have 3 classes to model. We could group them as

One group... (aka, NOT a discriminative feature)



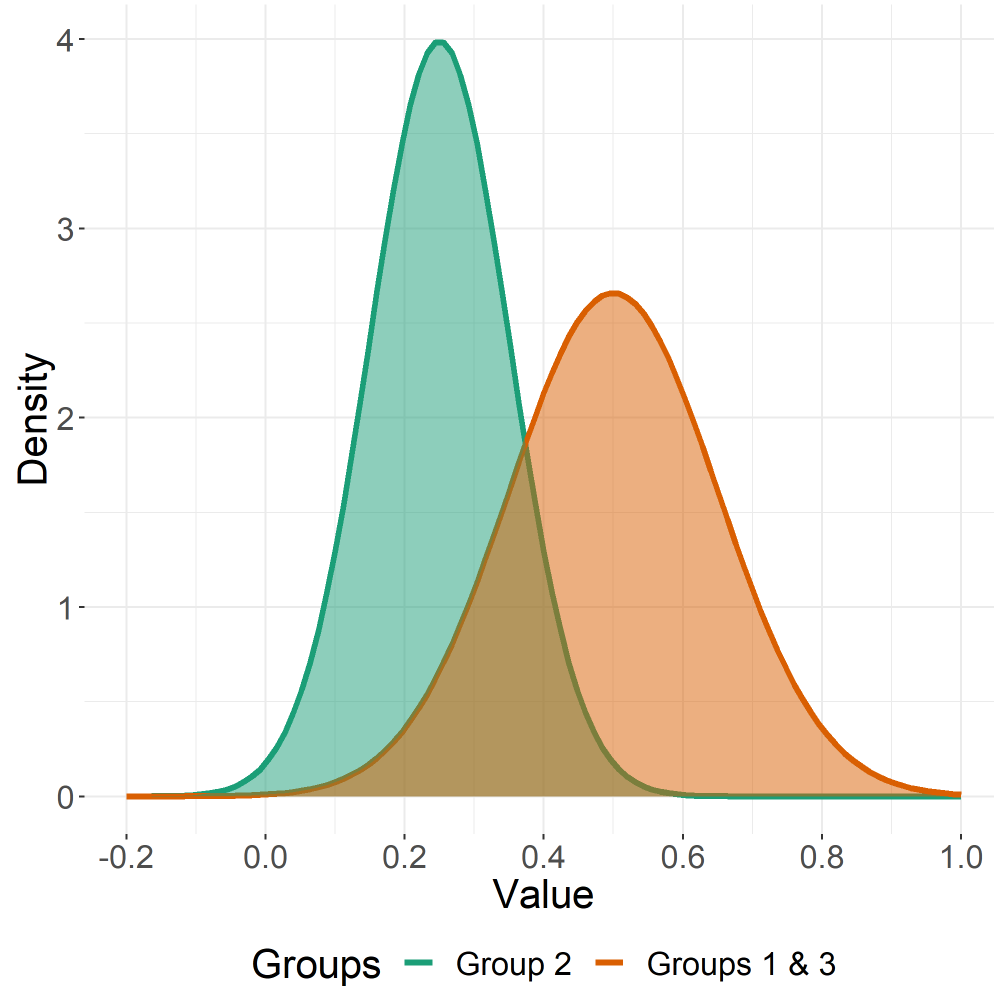
So, what makes a discriminative feature?

Two groups group 1 against 2 and 3



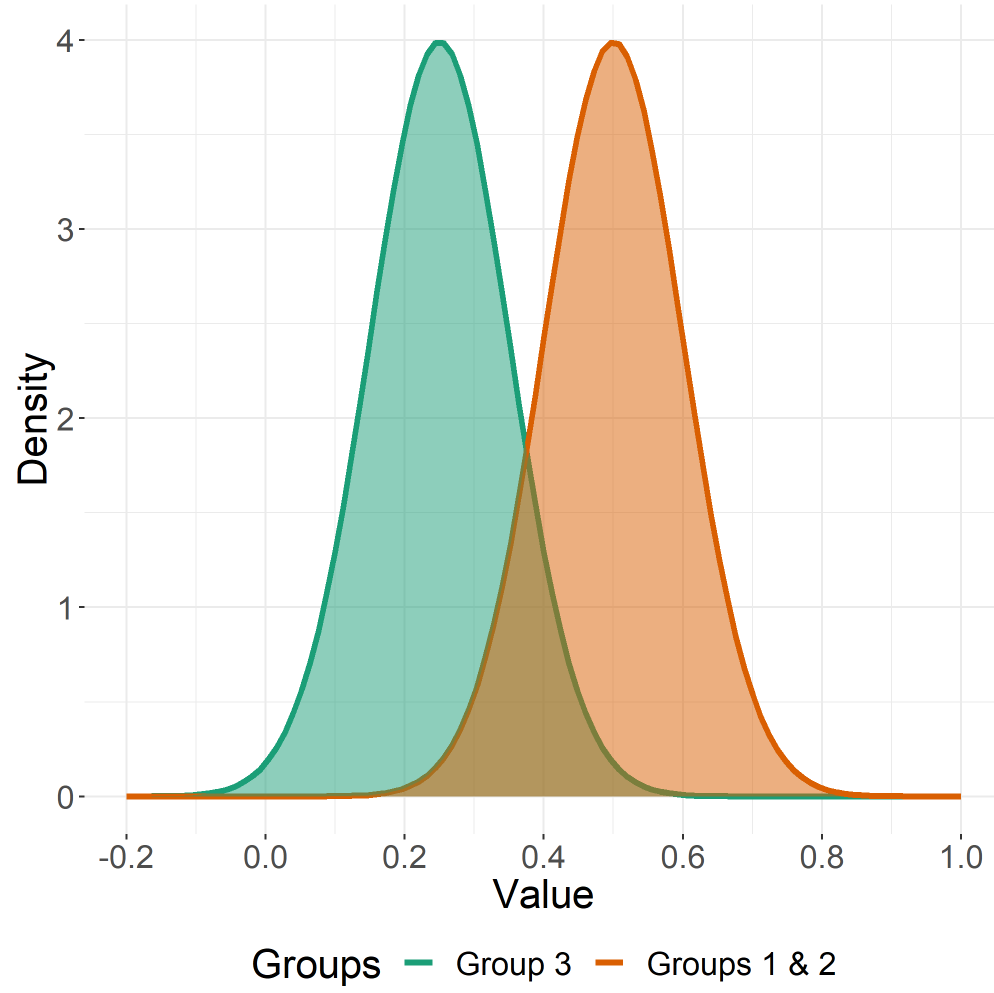
So, what makes a discriminative feature?

Two groups group 2 against 1 and 3



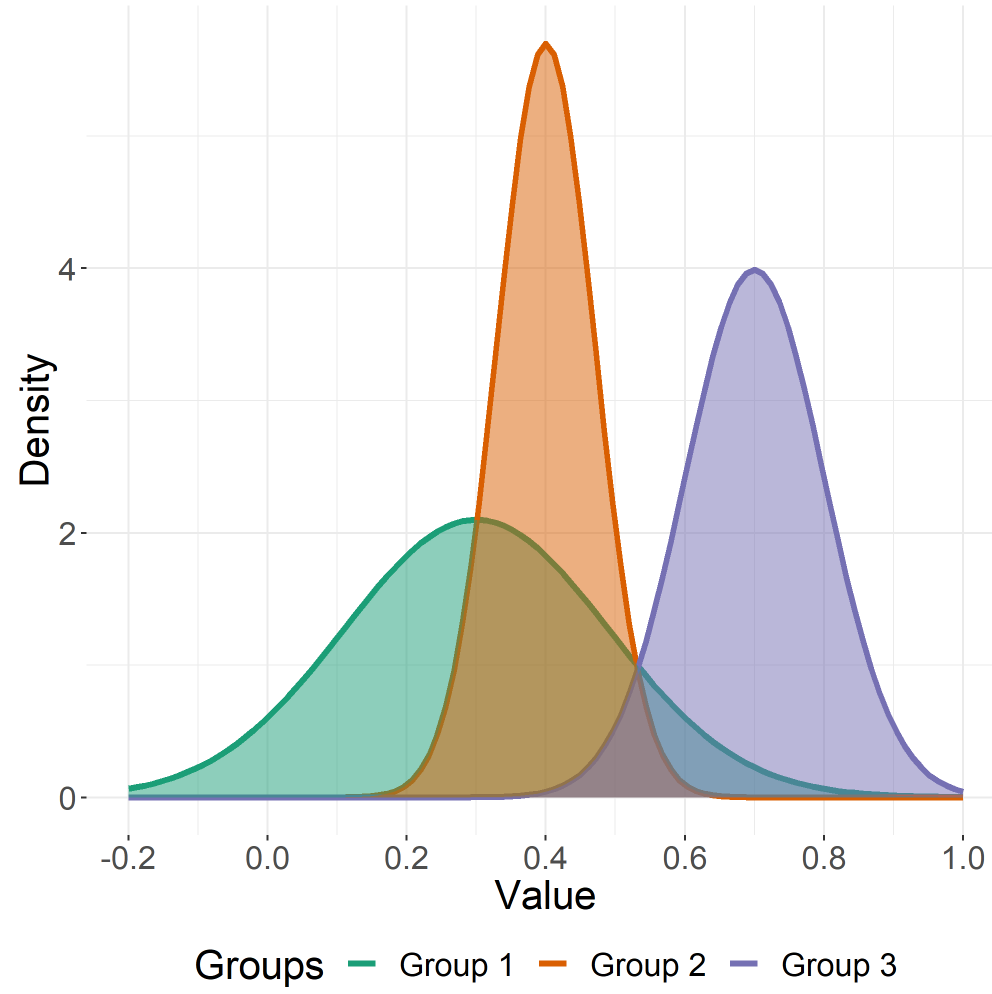
So, what makes a discriminative feature?

... and **Two groups** group 3 against 1 and 2

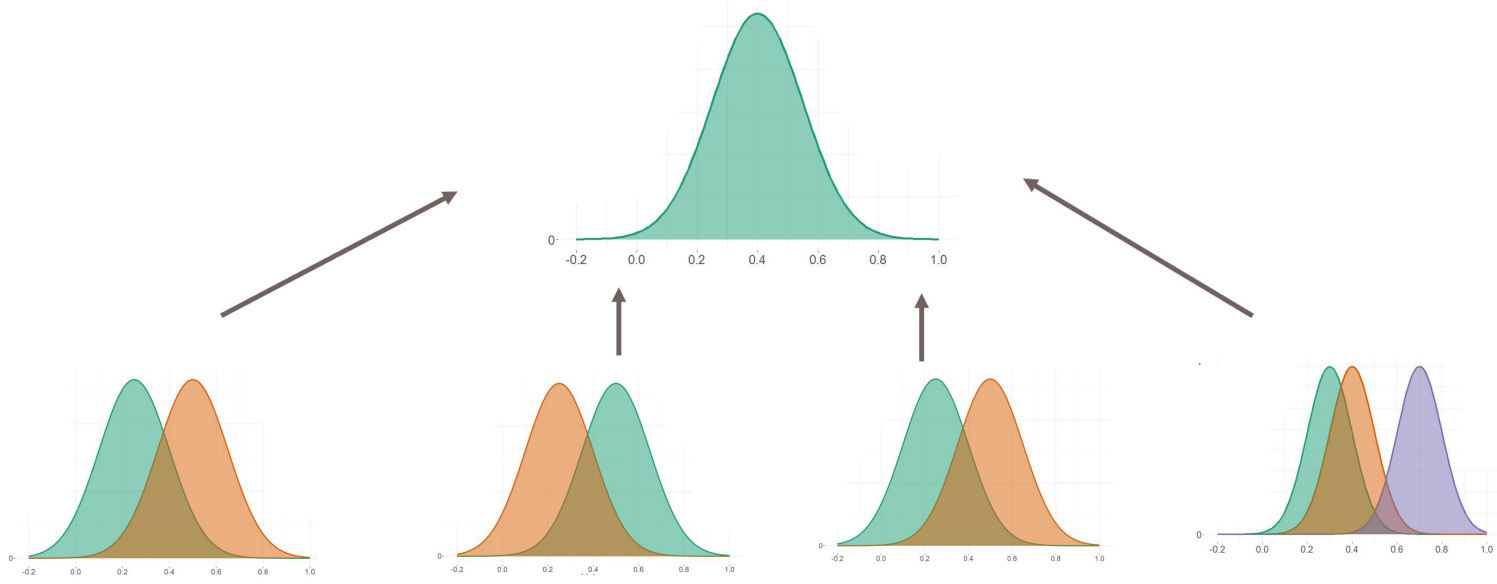


So, what makes a discriminative feature?

... and finally, **three groups** (all are different).



A Penalised LRT is used to determine best fit for each feature



multiDA - syntax

```
res <- multiDA(y = y,
               X = X,
               penalty = "EBIC",
               equal.var = TRUE,
               set.options = "exhaustive")
```

multiDA - syntax

```
res <- multiDA(y = y,
               X = X,
               penalty = "EBIC",
               equal.var = TRUE,
               set.options = "exhaustive")
```

multiDA - syntax

```
res <- multiDA(y = y,
               X = X,
               penalty = "EBIC",
               equal.var = TRUE,
               set.options = "exhaustive")
```

multiDA - syntax

```
res <- multiDA(y = y,
               X = X,
               penalty = "EBIC",
               equal.var = TRUE,
               set.options = "exhaustive")
```

multiDA - syntax

```
res <- multiDA(y = y,
               X = X,
               penalty = "EBIC",
               equal.var = TRUE,
               set.options = "exhaustive")
```

multiDA - syntax

```
res <- multiDA(y = y,
               X = X,
               penalty = "EBIC",
               equal.var = TRUE,
               set.options = "exhaustive")
```

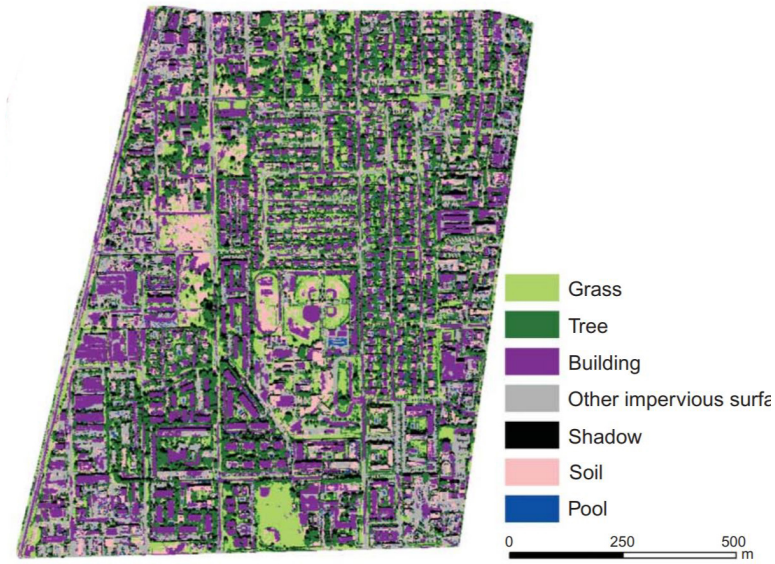
```
predict(res, newdata = newdata)
```

A generic S3 `predict` method is used for prediction as follows

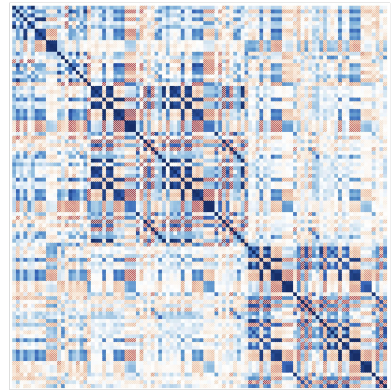
CV results

genDA

Urban Cover data

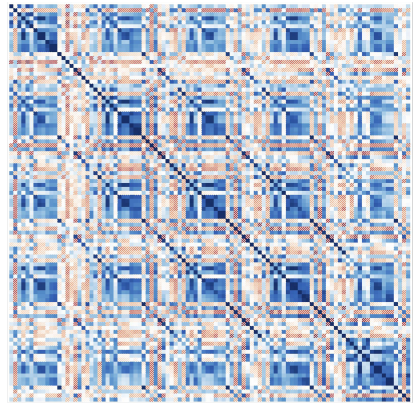


Urban Cover data



CLASS
'CAR'

CLASS
'SHADOW'

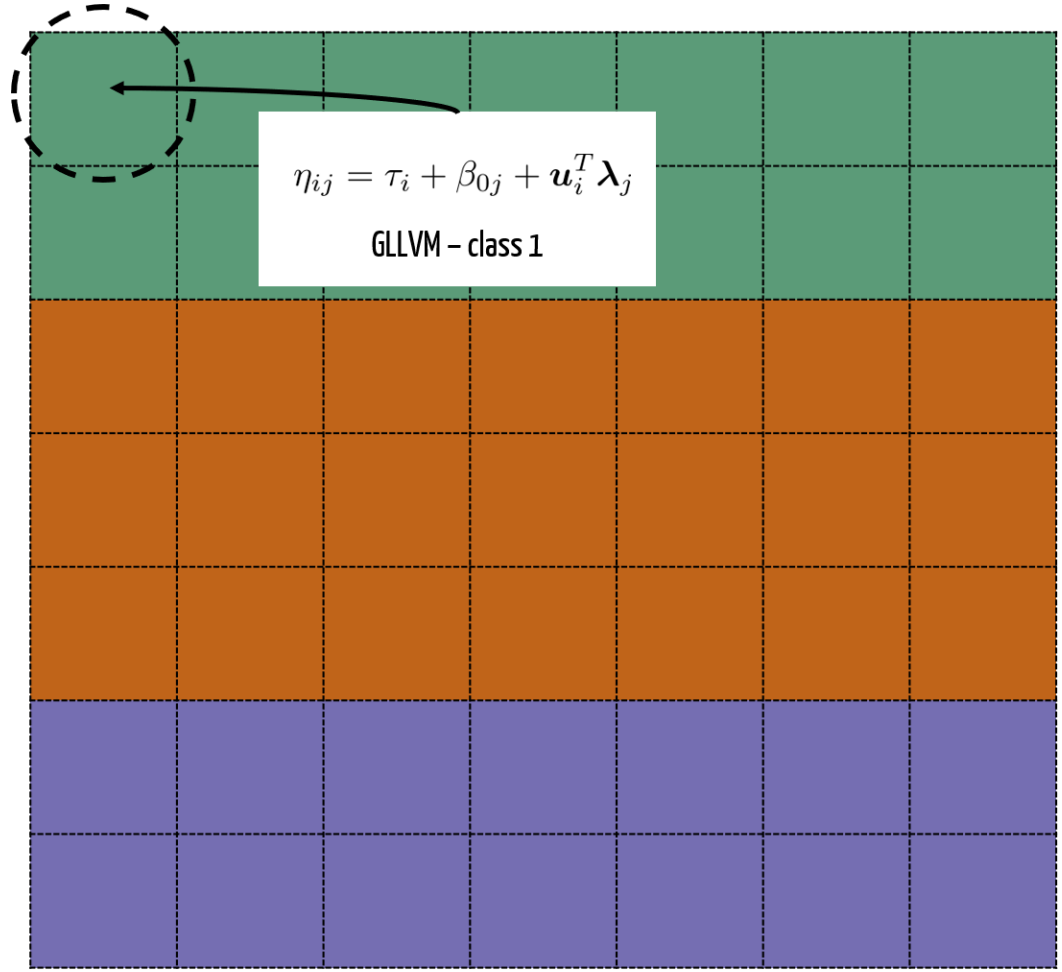


GLLVM

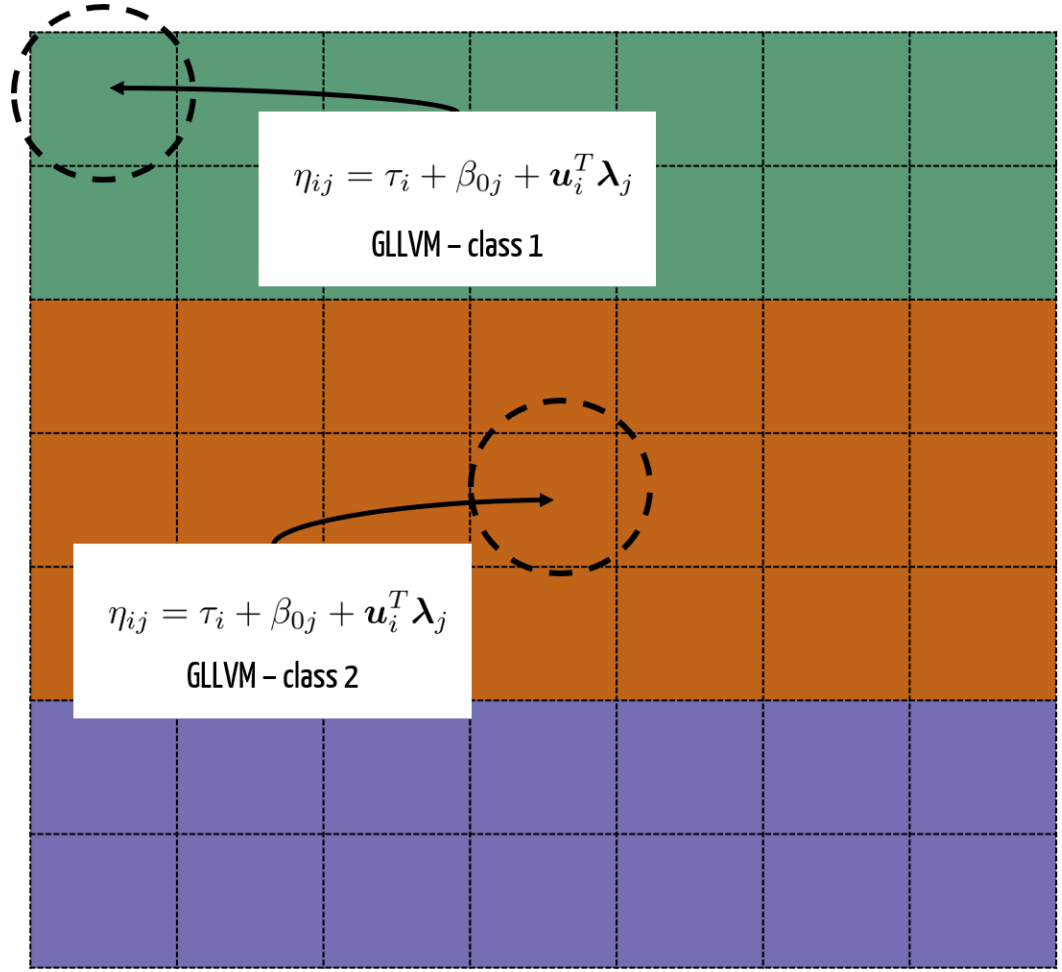
```
# A tibble: 168 x 148
   class     BrdIndx  Area Round Bright Compact ShpI
   <fct>       <dbl> <int> <dbl>  <dbl>   <dbl>   <d
 1 car          1.27    91  0.97   231.    1.39     1
 2 concrete     2.36   241  1.56   216.    2.46     2
 3 concrete     2.12   266  1.47   232.    2.07     2
 4 concrete     2.42   399  1.28   230.    2.49     2
 5 concrete     2.15   944  1.73   193.    2.28     4
 6 tree         3.11   169  1.47   172.    2.49     3
 7 car          1.2     44  0.79   209.    1.14     1
 8 car          1       88  0.22   235.    1.11     1
 9 building     1.59  1737  0.67   220.    1.3      1
10 tree         2.37   153  1.3    120.    2.85     2


#"log-normal"        "negative-binomial" "log-norma
```

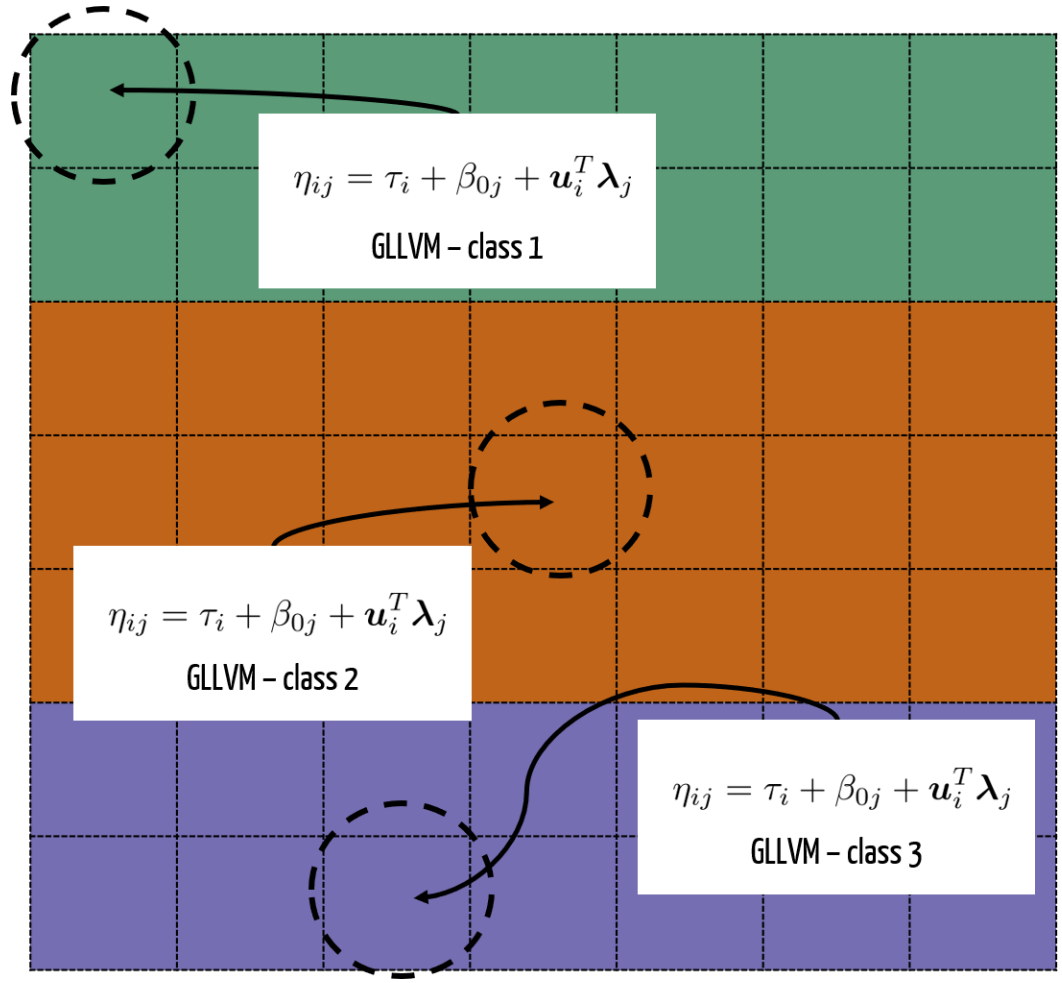
GLLVM matrix



GLLVM matrix



GLLVM matrix



TMB package

genDA - syntax

```
res <- genDA(Y = Y,
             class = class,
             num.lv = 2,
             family = family,
             common.covariance = TRUE,
             row.eff = FALSE,
             standard.errors = FALSE)
```

genDA - syntax

```
res <- genDA(Y = Y,
             class = class,
             num.lv = 2,
             family = family,
             common.covariance = TRUE,
             row.eff = FALSE,
             standard.errors = FALSE)
```

genDA - syntax

```
res <- genDA(Y = Y,
             class = class,
             num.lv = 2,
             family = family,
             common.covariance = TRUE,
             row.eff = FALSE,
             standard.errors = FALSE)
```

genDA - syntax

```
res <- genDA(Y = Y,
             class = class,
             num.lv = 2,
             family = family,
             common.covariance = TRUE,
             row.eff = FALSE,
             standard.errors = FALSE)
```

genDA - syntax

```
res <- genDA(Y = Y,
             class = class,
             num.lv = 2,
             family = family,
             common.covariance = TRUE,
             row.eff = FALSE,
             standard.errors = FALSE)
```

genDA - syntax

```
res <- genDA(Y = Y,
             class = class,
             num.lv = 2,
             family = family,
             common.covariance = TRUE,
             row.eff = FALSE,
             standard.errors = FALSE)
```

genDA - syntax

```
res <- genDA(Y = Y,
             class = class,
             num.lv = 2,
             family = family,
             common.covariance = TRUE,
             row.eff = FALSE,
             standard.errors = FALSE)
```

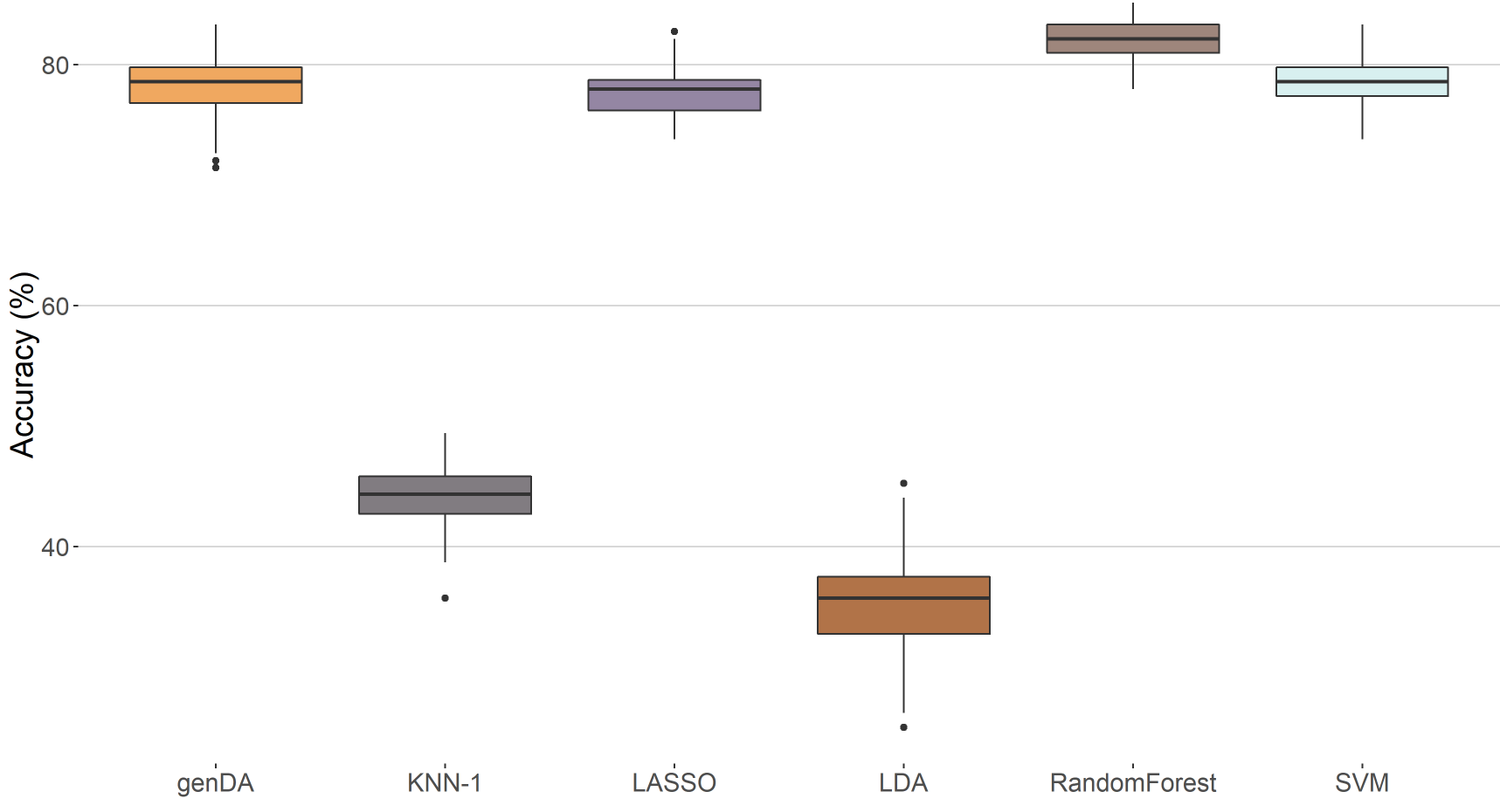
genDA - syntax

```
res <- genDA(Y = Y,
             class = class,
             num.lv = 2,
             family = family,
             common.covariance = TRUE,
             row.eff = FALSE,
             standard.errors = FALSE)
```

```
predict(res, newdata = newdata)
```

A generic S3 `predict` method is used for prediction as follows

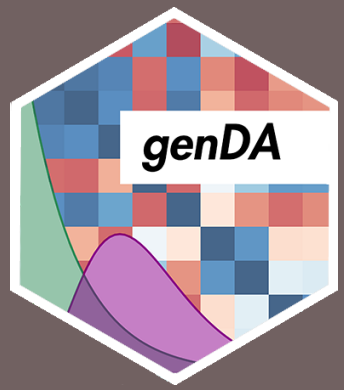
100 Trial, 5 Fold CV



: sarah_romanes

: sarahromanes.github.io

genDA



sarahromanes/genDA

multiDA

sarahromanes/multiDA