DATA 622 Assignment 2

CUNY: Spring 2021

Philip Tanofsky

15 March 2021

Introduction

Purpose, prompt of the exercise

```r
# Import required R libraries
library(palmerpenguins)
library(tidyverse)
library(caret)
library(MASS)
library(ggplot2)
library(mvtnorm)
theme_set(theme_classic())
```

Initial Data Inspection

```r
ds <- penguins

head(ds)
```

```
## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g sex
##   <fct>   <fct>           <dbl>         <dbl>            <int>       <int> <fct>
## 1 Adelie  Torge~           39.1          18.7              181        3750 male
## 2 Adelie  Torge~           39.5          17.4              186        3800 fema~
## 3 Adelie  Torge~           40.3          18                195        3250 fema~
## 4 Adelie  Torge~           NA            NA                 NA          NA <NA>
## 5 Adelie  Torge~           36.7          19.3              193        3450 fema~
## 6 Adelie  Torge~           39.3          20.6              190        3650 male
## # ... with 1 more variable: year <int>
```

```r
summary(ds)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
##  Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
##                                  Mean   :43.92   Mean   :17.15
##                                  3rd Qu.:48.50   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##                                  NA's   :2       NA's   :2
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172.0     Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197.0     Median :4050   NA's  : 11   Median :2008
##  Mean   :200.9     Mean   :4202                Mean   :2008
##  3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
##  NA's   :2         NA's   :2
```

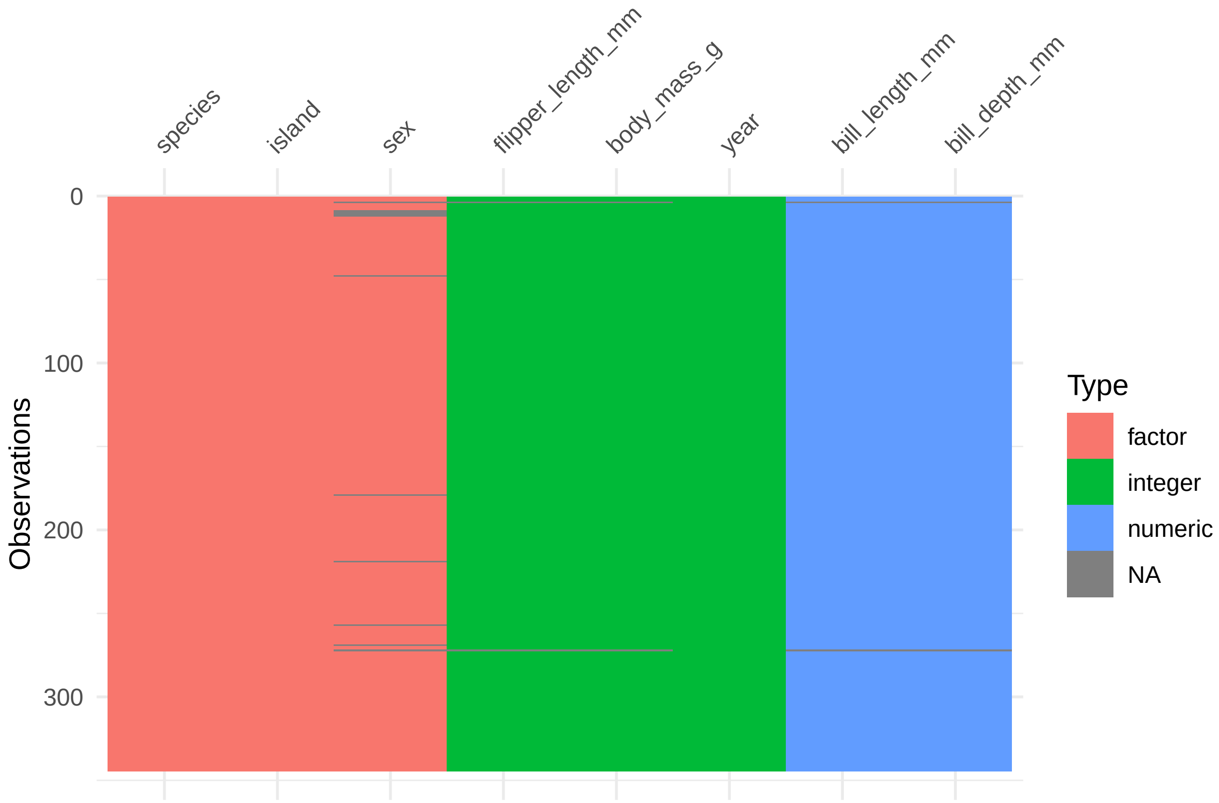
```
dim(ds)
```

```
## [1] 344   8
```

```
glimpse(ds)
```

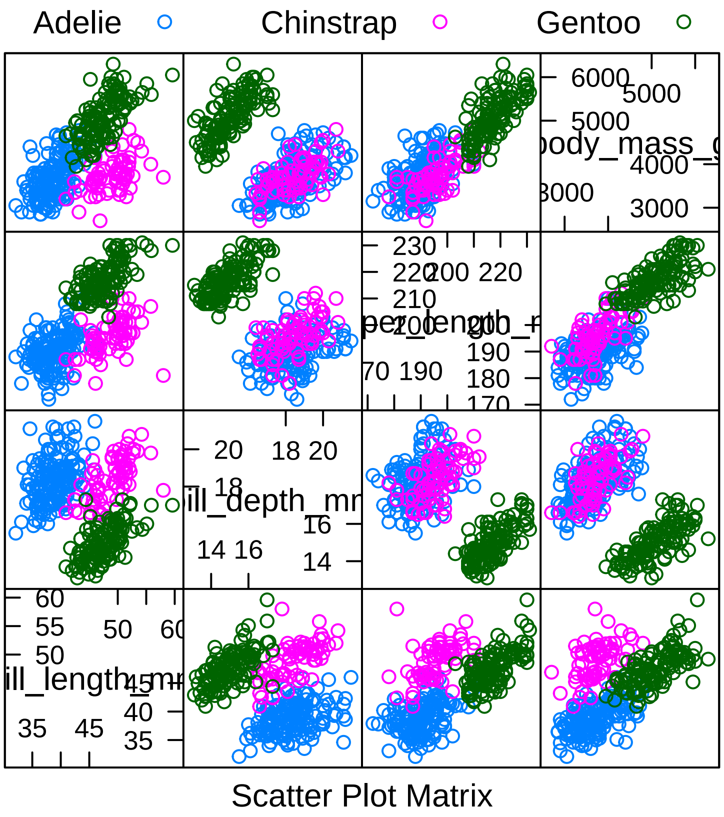
```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, A...
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torge...
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34....
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18....
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, ...
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 347...
## $ sex               <fct> male, female, female, NA, female, male, female, m...
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2...
```

```
visdat::vis_dat(ds)
```

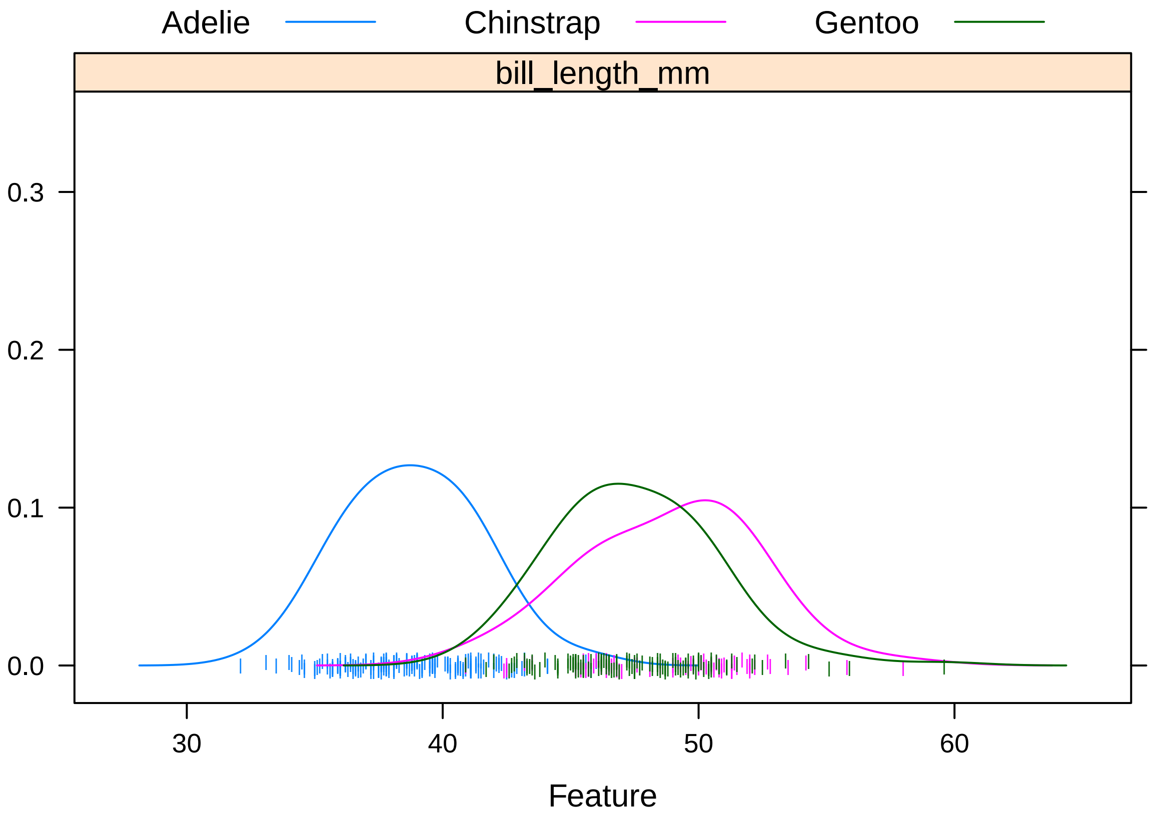


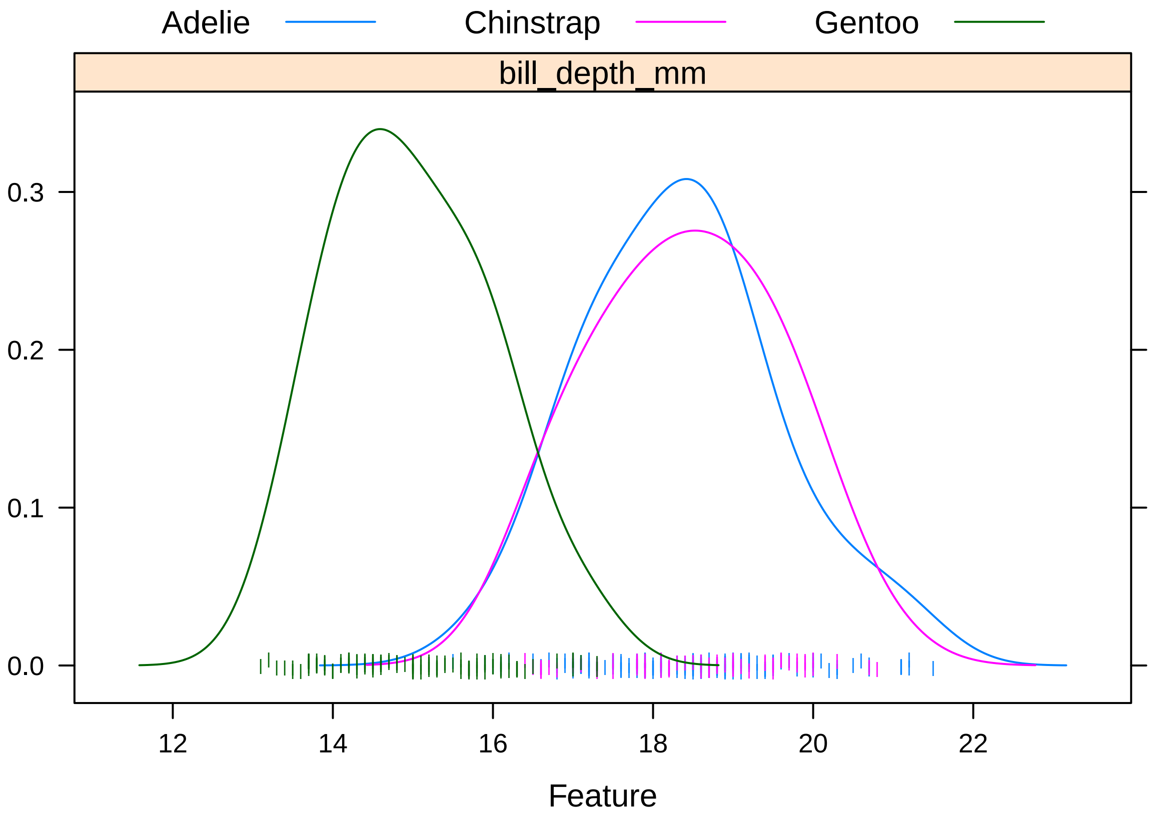
Exploratory Data Analysis

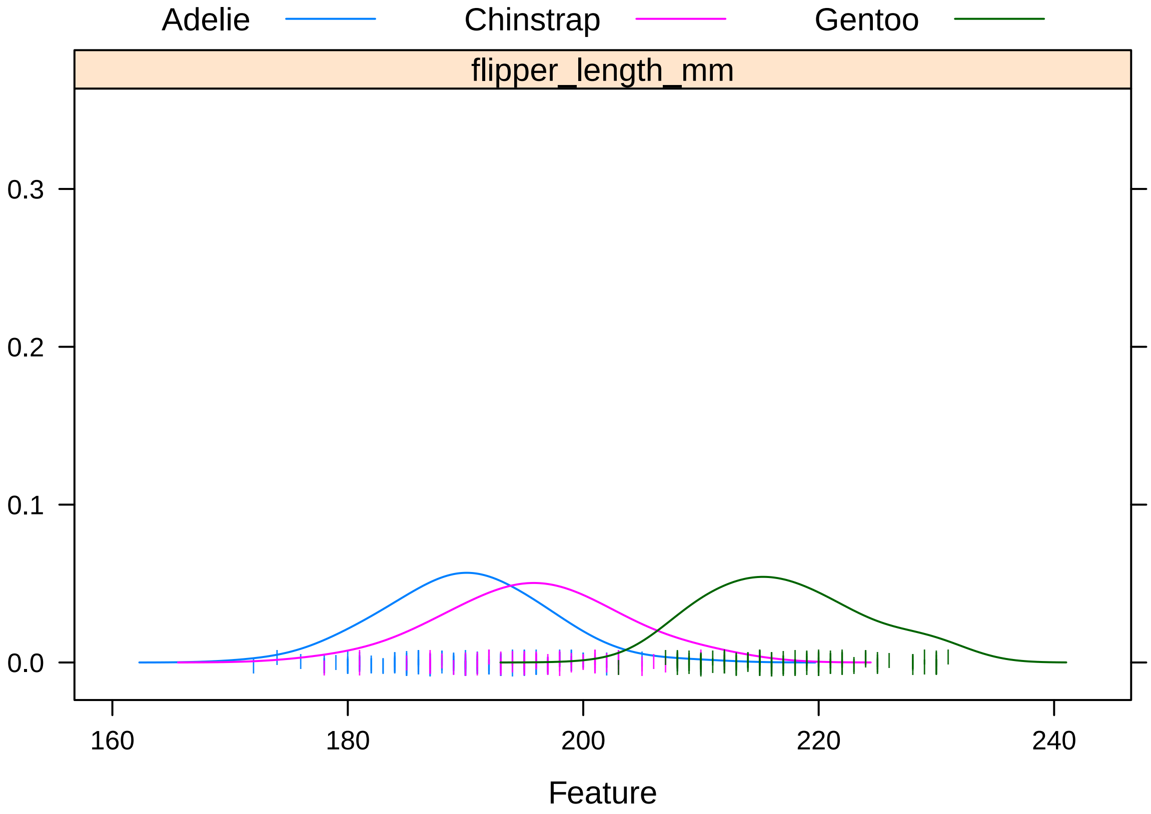
```
# Use featurePlot
# https://topepo.github.io/caret/visualizations.html

# Scatterplot
featurePlot(x = penguins[, 3:6],
            y = penguins$species,
            plot = "pairs",
            # Add a key at the top
            auto.key = list(columns = 3))
```

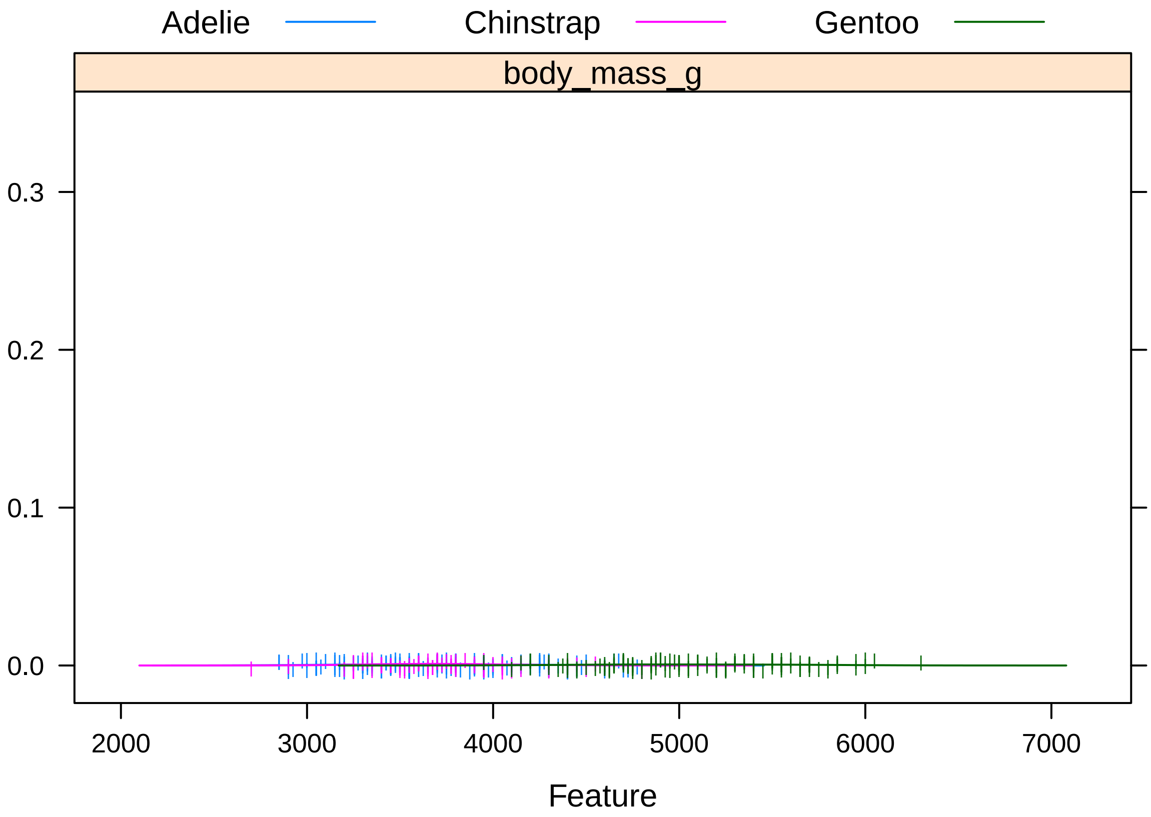


```r
# Overlayed density plots
featurePlot(x = penguins[, 3:6],
            y = penguins$species,
            plot = "density",
            # Pass in options to xyplot() to
            # make it prettier
            scales = list(x = list(relation="free"),
                          x = list(relation="free")),
            adjust = 1.5,
            pch = "|",
            layout = c(1, 1),
            auto.key = list(columns = 3))
```


Adelie
Chinstrap
Gentoo
bill_length_mm
0.3
0.2
0.1
0.0
30
40
50
60
Feature


Adelie
Chinstrap
Gentoo
bill_depth_mm
0.3
0.2
0.1
0.0
12
14
16
18
20
22
Feature

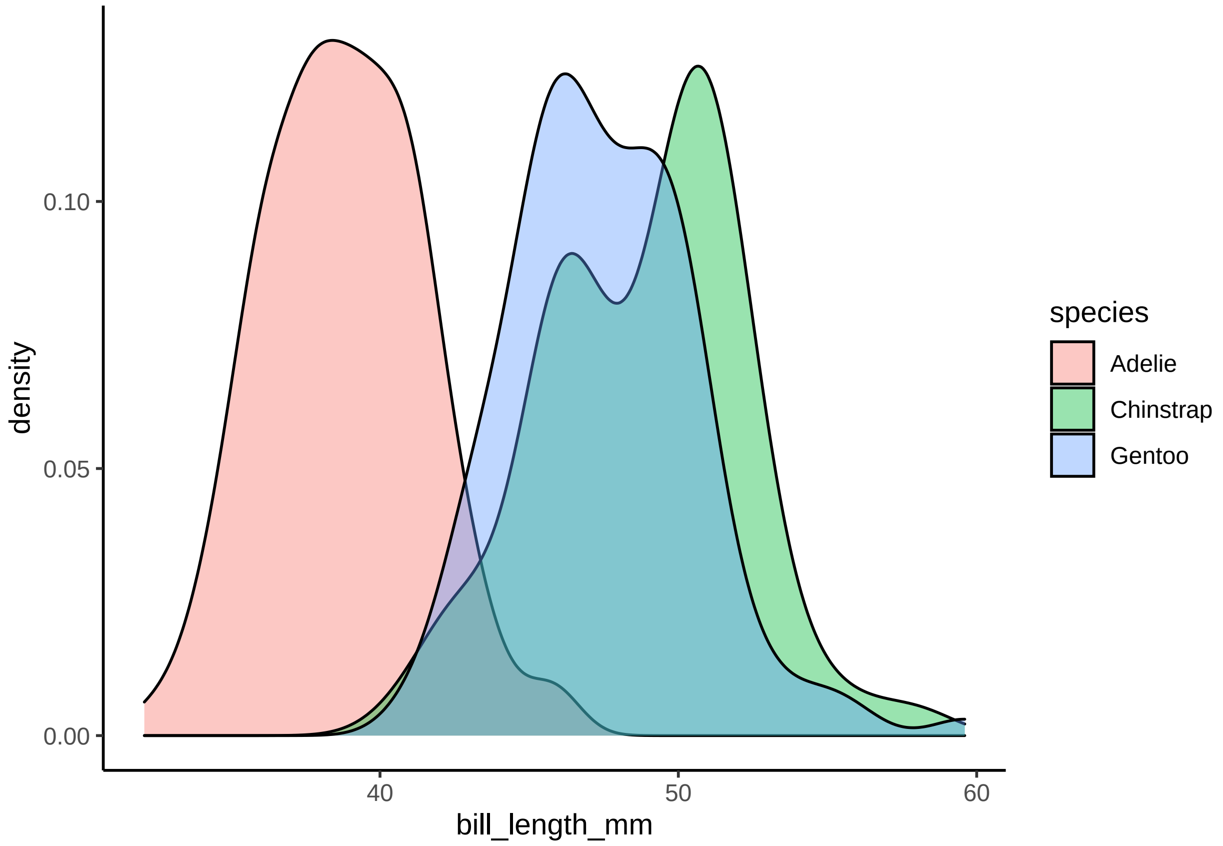

Adelie
Chinstrap
Gentoo
flipper_length_mm
0.3
0.2
0.1
0.0
160
180
200
220
240
Feature



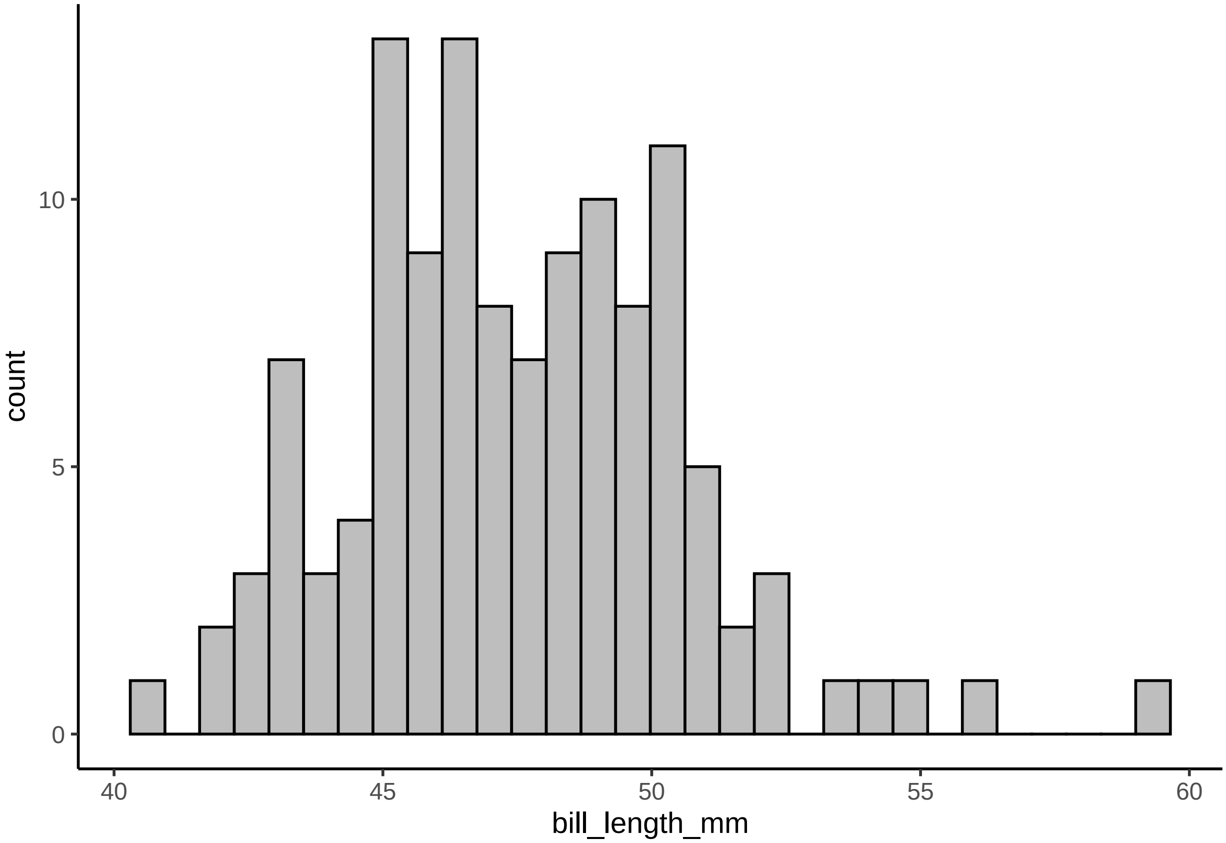
From scatterplot above featurePlot function, bill_length_mm appears to differentiate bill_depth_mm doesn't differentiate well,

but bill_length_mm and flipper_length_mm appear highly correlated

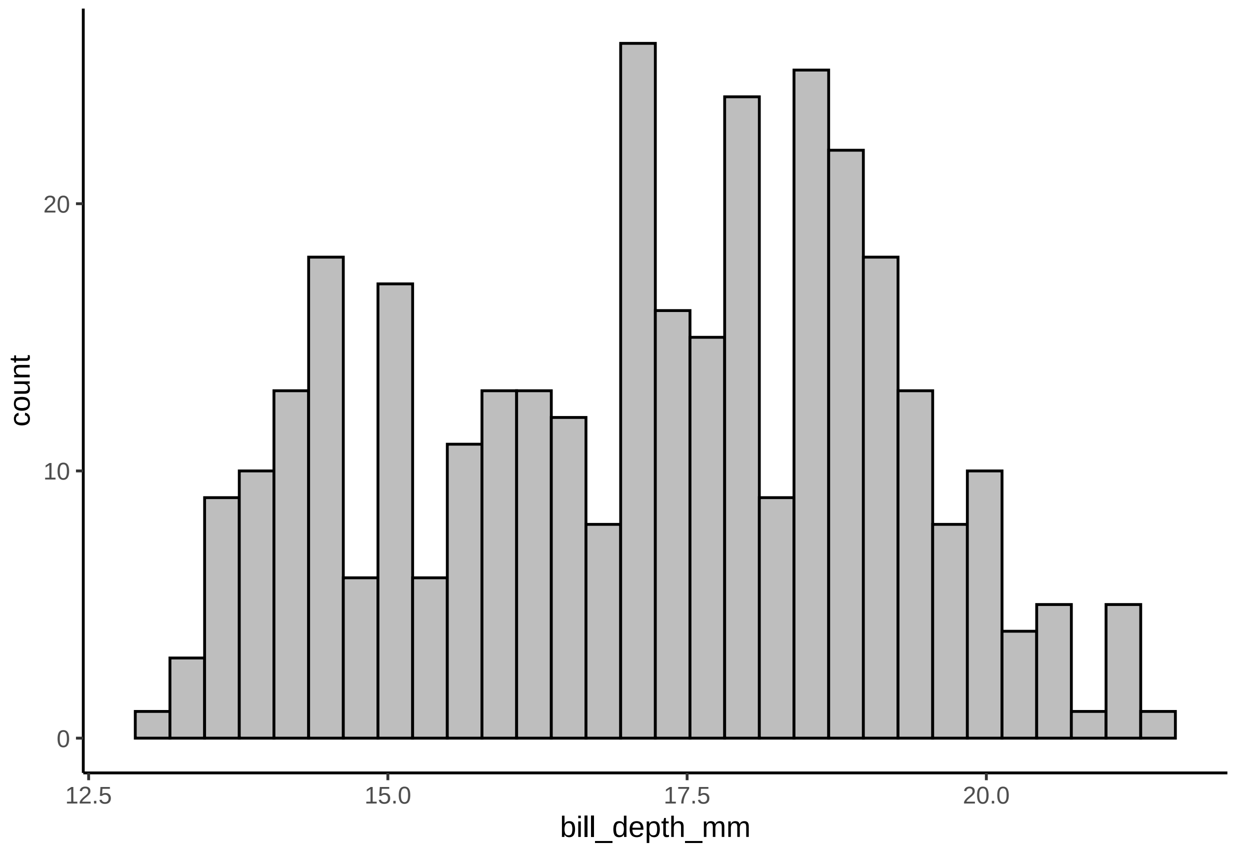
```
# Plots of individual variables
#http://www.sthda.com/english/articles/32-r-graphics-essentials/133-plot-one-variable-frequency-graph-d


# Change density plot fill colors by groups
p <- ggplot(penguins, aes(x=bill_length_mm, fill=species)) +
  geom_density(alpha=0.4)

p
```



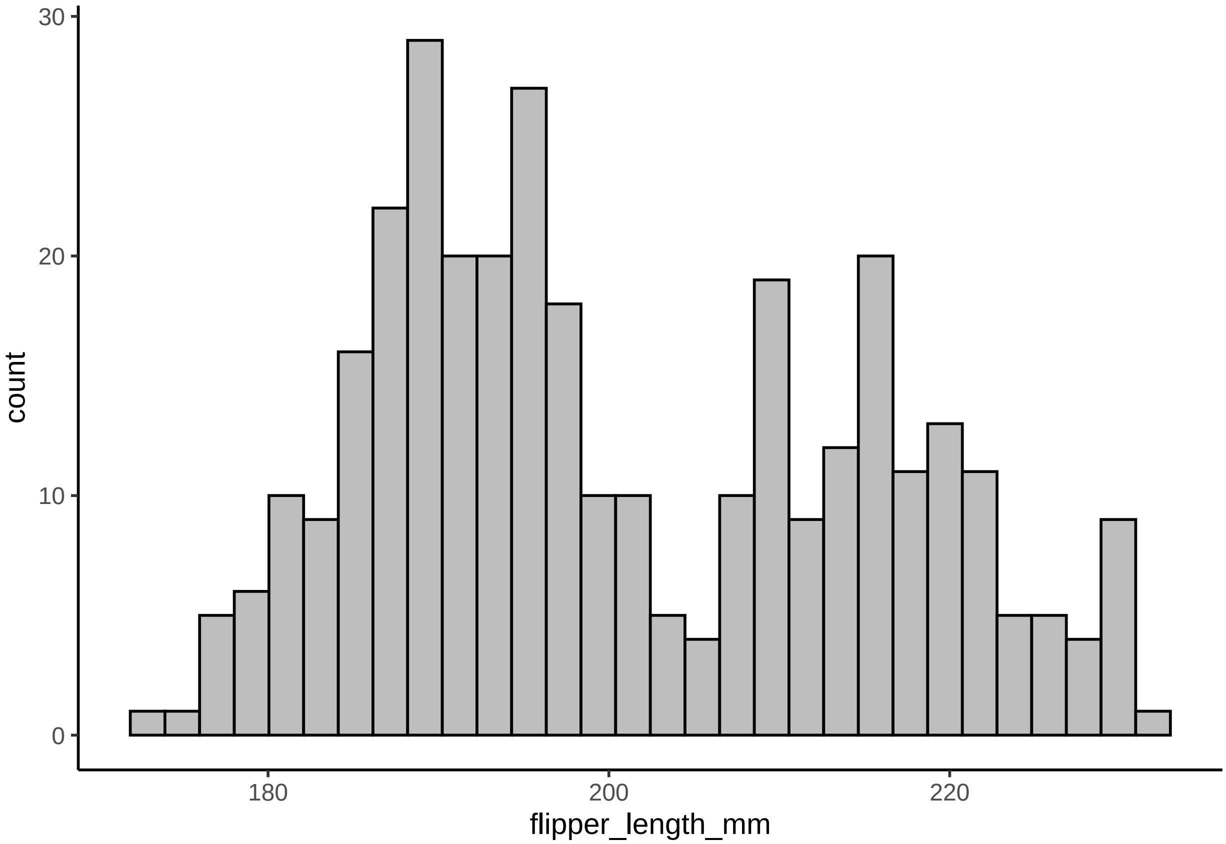
```
a <- penguins %>%
      filter(species == 'Gentoo') %>%
      ggplot(aes(x = bill_length_mm))

#a <- ggplot(penguins, aes(x = bill_length_mm))

a + geom_histogram(bins = 30, color = "black", fill = "gray") +
  geom_vline(aes(xintercept = mean(bill_length_mm)),
             linetype = "dashed", size = 0.6)
```



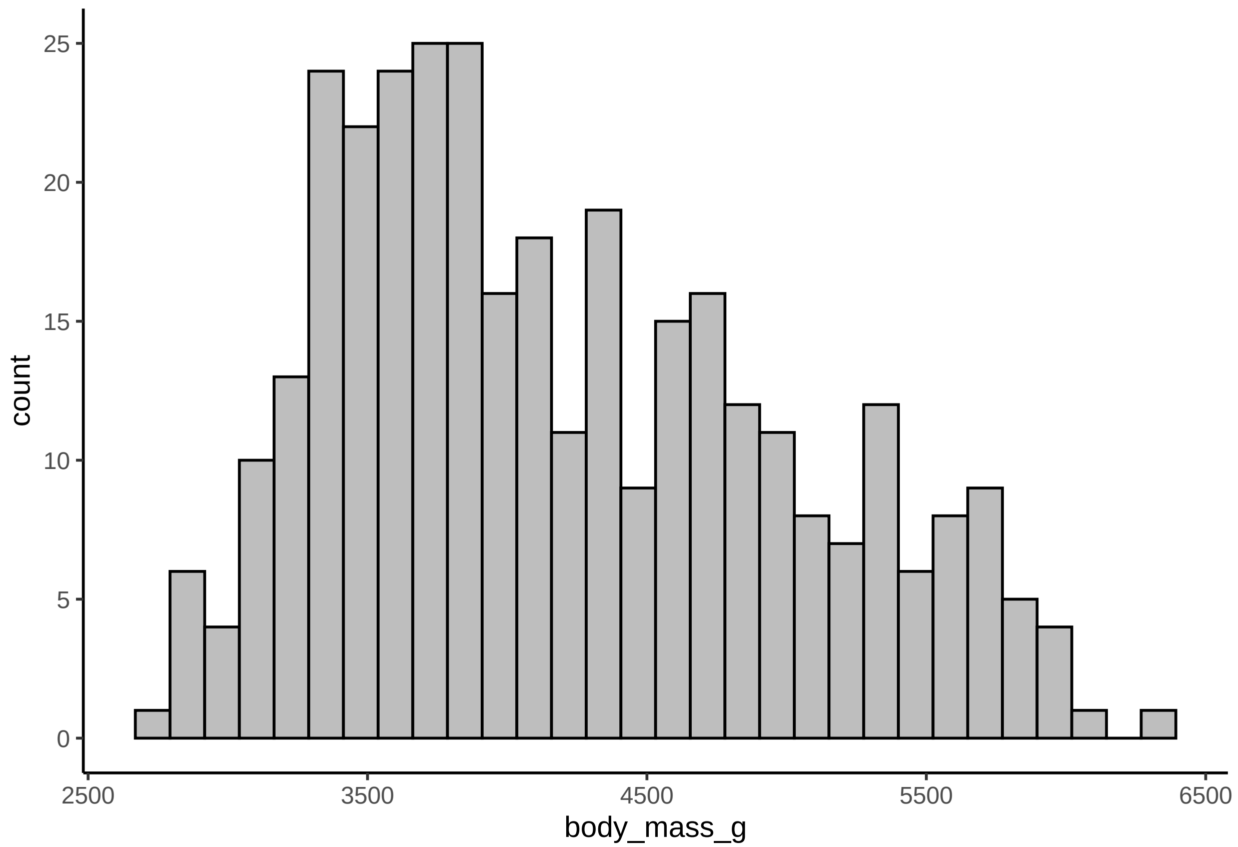
```r
b <- ggplot(penguins, aes(x = bill_depth_mm))

b + geom_histogram(bins = 30, color = "black", fill = "gray") +
  geom_vline(aes(xintercept = mean(bill_depth_mm)),
             linetype = "dashed", size = 0.6)
```



```
c <- ggplot(penguins, aes(x = flipper_length_mm))

c + geom_histogram(bins = 30, color = "black", fill = "gray") +
  geom_vline(aes(xintercept = mean(flipper_length_mm)),
             linetype = "dashed", size = 0.6)
```



```r
d <- ggplot(penguins, aes(x = body_mass_g))

d + geom_histogram(bins = 30, color = "black", fill = "gray") +
  geom_vline(aes(xintercept = mean(body_mass_g)),
             linetype = "dashed", size = 0.6)
```

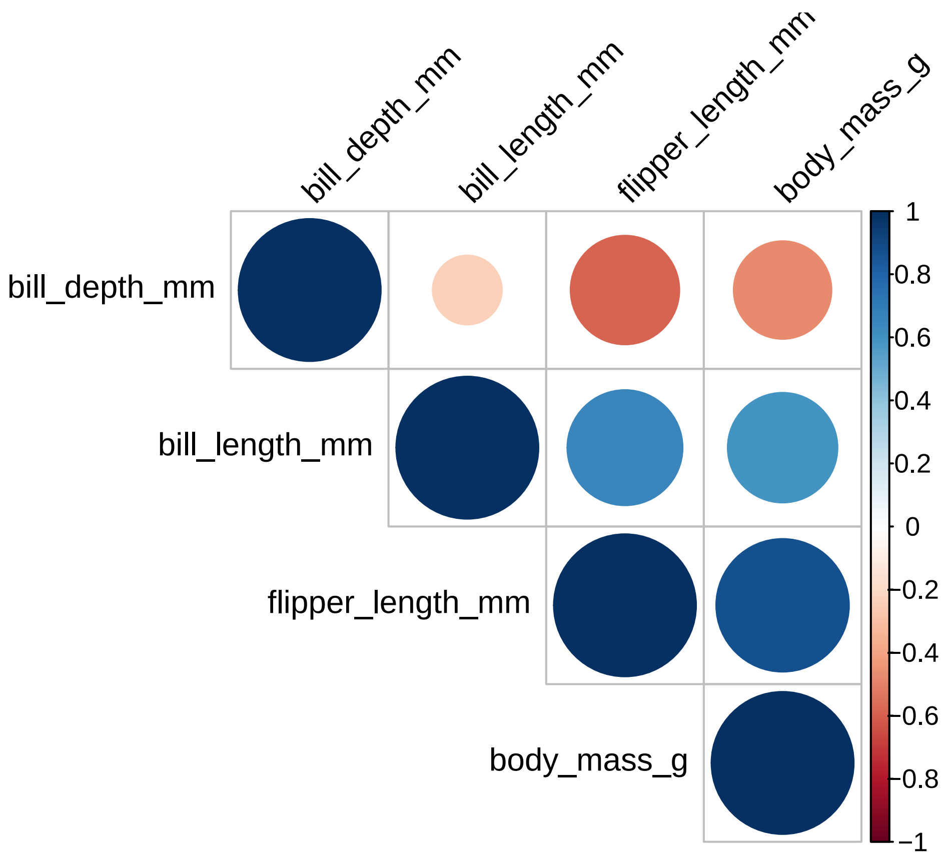


Covariance checks

```
# Compute correlation matrix
cor_mat <- cor(penguins[,3:6],  use = "complete.obs")
round(cor_mat, 2)
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm              1.00         -0.24              0.66        0.60
## bill_depth_mm              -0.24          1.00             -0.58       -0.47
## flipper_length_mm           0.66         -0.58              1.00        0.87
## body_mass_g                 0.60         -0.47              0.87        1.00
```

```
library(corrplot)
corrplot(cor_mat, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```



```r
# Compute covariance matrix
cov_mat <- cov(penguins[,3:6],  use = "complete.obs")
round(cov_mat, 2)
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm             29.81         -2.53             50.38     2605.59
## bill_depth_mm              -2.53          3.90            -16.21     -747.37
## flipper_length_mm          50.38        -16.21            197.73     9824.42
## body_mass_g              2605.59       -747.37           9824.42   643131.08
```

```r
p_g <- penguins %>% filter(species == 'Gentoo')
cov_mat <- cov(p_g[,3:6],  use = "complete.obs")
round(cov_mat, 2)
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm              9.50          1.95             13.21     1039.63
## bill_depth_mm               1.95          0.96              4.50      355.69
## flipper_length_mm          13.21          4.50             42.05     2297.14
## body_mass_g              1039.63        355.69           2297.14   254133.18
```

```r
p_a <- penguins %>% filter(species == 'Adelie')
cov_mat <- cov(p_a[,3:6],  use = "complete.obs")
round(cov_mat, 2)
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm              7.09          1.27              5.67      670.36
## bill_depth_mm               1.27          1.48              2.45      321.44
## flipper_length_mm           5.67          2.45             42.76     1404.03
## body_mass_g               670.36        321.44           1404.03   210282.89
```

```r
p_c <- penguins %>% filter(species == 'Chinstrap')

cov_mat <- cov(p_c[,3:6],  use = "complete.obs")
round(cov_mat, 2)
```

```
##                   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm             11.15          2.48             11.23      659.20
## bill_depth_mm               2.48          1.29              4.70      263.79
## flipper_length_mm          11.23          4.70             50.86     1758.54
## body_mass_g               659.20        263.79           1758.54   147713.45
```

LDA: Linear Discriminant Analysis

LDA does not handle categorial data well.

LDA assumes the feature variables come from multivariate normal distribution, all of them continuous

http://www.sthda.com/english/articles/36-classification-methods-essentials/146-discriminant-analysis-essentials-in-r/

LDA assumes the predictors are normally distributed (Gaussian distribution) and that the different classes have class-specific means and equal variance/covariance

Make sure each variable is normally distributed

https://web.stanford.edu/class/stats202/notes/Classification/LDA.html That is, within each class the features have multivariate normal distribution with center depending on the class and common covariance

bill_depth, bill_length, body_mass

```r
# Load the data
data("penguins")

# Only complete entries
penguins <- na.omit(penguins)

# Remove 'year' and 'sex feature
# Apparently leaving 'island' in for LDA improves the model
drops <- c("year", "sex")
penguins <- penguins[ , !(names(penguins) %in% drops)]

#Split the data into training (80%) and test set (20%)
set.seed(123)
training.samples <- penguins$species %>%
  createDataPartition(p = 0.8, list=FALSE)
train.data <- penguins[training.samples, ]
test.data <- penguins[-training.samples, ]

#2. Normalize the data. Categorial variables are automatically ignored from normalizing
```

```r
# Estimate preprocessing parameters
preproc.param <- train.data %>%
  preProcess(method = c("center", "scale"))

# Transform the data using the estimated parameters
train.transformed <- preproc.param %>% predict(train.data)
test.transformed <- preproc.param %>% predict(test.data)
```

Focus on normal distributions

If n is small and the distribution of the predictors X is approximately normal in each of the classes, the LD model is more stable than logistic regression

Correlation will cause a line in the gaussian density plot

When there are K classes, linear discriminant analysis can be viewed exactly in a K-1 dimensional plot. Measuring which centroid is the closest. Distance in the subspace

Co-variance matrix wold be 4 x 4 for 4 features

Distribution and common covariance for each class

```r
# Fit the model
model <- lda(species~., data = train.transformed)
# Make predictions
predictions <- model %>% predict(test.transformed)

# Confusion matrix
table(predictions$class, test.transformed$species)
```

```
## 
##             Adelie Chinstrap Gentoo
##   Adelie        29         0      0
##   Chinstrap      0        13      0
##   Gentoo         0         0     23
```

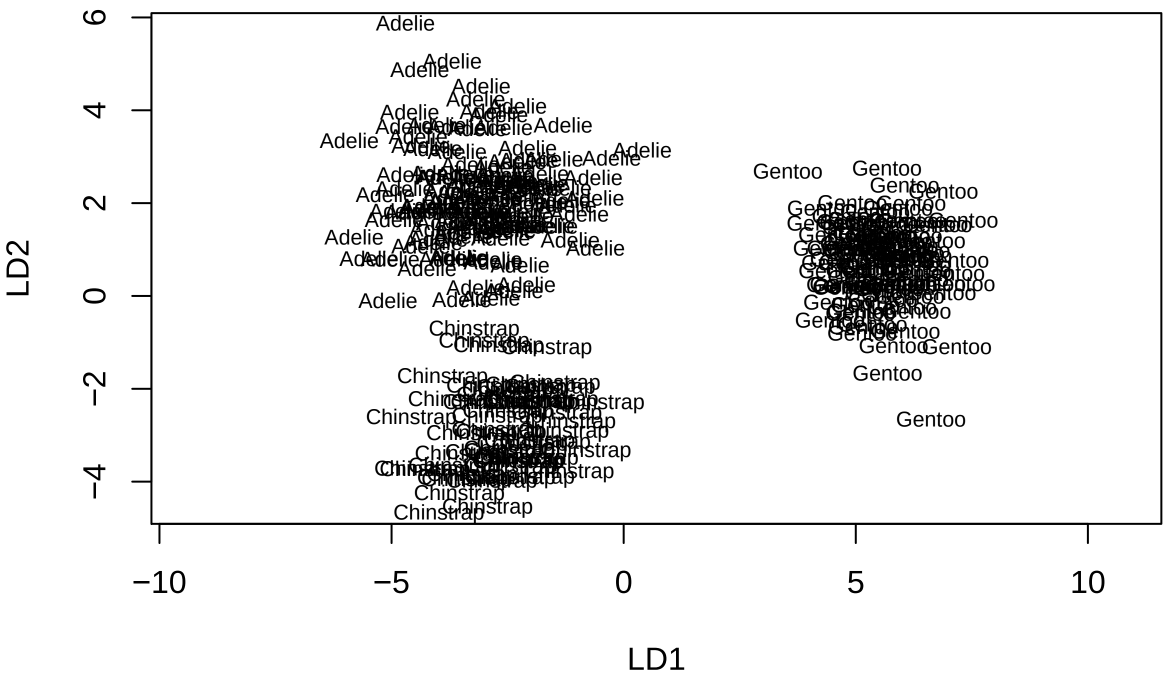
```r
# Model accuracy
mean(predictions$class == test.transformed$species)
```

```
## [1] 1
```

```r
# Output Model
model
```

```
## Call:
## lda(species ~ ., data = train.transformed)
## 
## Prior probabilities of groups:
##    Adelie Chinstrap    Gentoo 
## 0.4365672 0.2052239 0.3582090 
## 
## Group means:
##           islandDream islandTorgersen bill_length_mm bill_depth_mm
## Adelie      0.4017094       0.3247863     -0.9555507     0.6009140
## Chinstrap   1.0000000       0.0000000      0.9088301     0.6564651
## Gentoo      0.0000000       0.0000000      0.6438935    -1.1084638
##            flipper_length_mm body_mass_g
## Adelie            -0.7707877  -0.6092842
## Chinstrap         -0.3716006  -0.5901947
## Gentoo             1.1522937   1.0806975
##
## Coefficients of linear discriminants:
##                          LD1        LD2
## islandDream       -1.3494973 -1.6233797
## islandTorgersen   -1.1326926 -0.2217997
## bill_length_mm     0.3074563 -2.2954800
## bill_depth_mm     -1.9480393  0.3330398
## flipper_length_mm  1.1902763  0.2251057
## body_mass_g        0.9771679  0.9796534
##
## Proportion of trace:
##    LD1    LD2
## 0.8236 0.1764
```

```r
# Display model
plot(model)
```



```r
names(predictions)
```

```
## [1] "class"     "posterior" "x"
```

```r
# Predicted classes
head(predictions$class, 6)
```

```
## [1] Adelie Adelie Adelie Adelie Adelie Adelie
## Levels: Adelie Chinstrap Gentoo
```

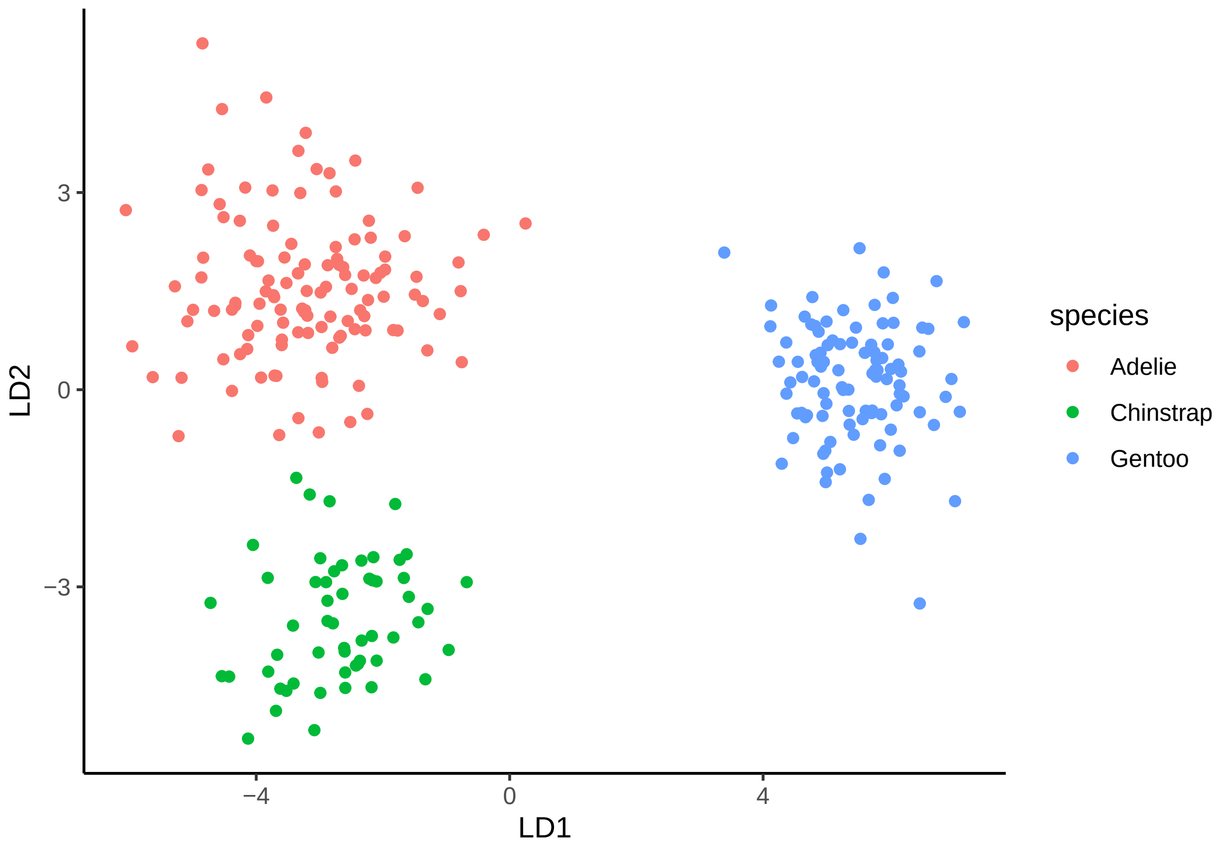
```r
# Predicted probabilities of class membership
head(predictions$posterior, 6)
```

```
##      Adelie    Chinstrap       Gentoo
## 1 0.9999999 1.122150e-07 1.063675e-22
## 2 0.9999990 9.659679e-07 7.497505e-16
## 3 1.0000000 1.680975e-08 2.464359e-27
## 4 1.0000000 1.864923e-09 1.913351e-19
## 5 1.0000000 1.785146e-11 2.750650e-19
## 6 1.0000000 5.858899e-09 1.029719e-14
```

```r
# Linear discriminants
head(predictions$x, 3)
```

```
##         LD1      LD2
## 1 -4.516164 1.880100
## 2 -2.737947 1.636759
## 3 -5.705103 2.133684
```

```r
# Plot
lda.data <- cbind(train.transformed, predict(model)$x)
ggplot(lda.data, aes(LD1, LD2)) +
  geom_point(aes(color = species))
```



```
# Model accuracy
mean(predictions$class==test.transformed$species)
```

```
## [1] 1
```

```
sum(predictions$posterior[ ,1] >= .5)
```

```
## [1] 29
```

```
# QDA

# Remove 'island' feature as it was causing rank deficiency in group Chinstrap
#drops <- c("flipper_length_mm", "body_mass_g", "island")
drops <- c("island")
train.transformed <- train.transformed[ , !(names(train.transformed) %in% drops)]

# Fit the model
model <- qda(species~., data = train.transformed)

# Output model results
model
```

```
## Call:
## qda(species ~ ., data = train.transformed)
```

```
##
## Prior probabilities of groups:
##    Adelie Chinstrap    Gentoo
## 0.4365672 0.2052239 0.3582090
##
## Group means:
##           bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## Adelie        -0.9555507     0.6009140        -0.7707877  -0.6092842
## Chinstrap      0.9088301     0.6564651        -0.3716006  -0.5901947
## Gentoo         0.6438935    -1.1084638         1.1522937   1.0806975
```

```
# Make predictions
predictions <- model %>% predict(test.transformed)

# Model accuracy
mean(predictions$class == test.transformed$species)
```

```
## [1] 0.9538462
```

https://www.geeksforgeeks.org/linear-discriminant-analysis-in-r-programming/

QDA: Quadratic Discrimant Analysis

Same link as above

QDA: works well with fewer features, that's when NB works well, works with higher number of features

mixed features can be used for NB

NB: Naive Bayes

https://www.r-bloggers.com/2018/01/understanding-naive-bayes-classifier-using-r/

```
library(e1071)

# Next load the Titantic dataset
data("Titanic")

# Save into a data frame and view it
t_df <- as.data.frame(Titanic)

# Creating data from table
repeating_sequence <- rep.int(seq_len(nrow(t_df)), t_df$Freq)

# Create the dataset by row repetition created
t_ds <- t_df[repeating_sequence, ]

# We no longer need the frequency, drop the feature
t_ds$Freq = NULL

# Fitting the Naive Bayes model
```

```
nbm <- naiveBayes(Survived~., data=t_ds)
# Output the model
nbm

# Prediction on the dataset
nb_predictions <- predict(nbm, t_ds)
# Confusion matrix to check accuracy
table(nb_predictions, t_ds$Survived)

# Getting started with Naive Bayes in mlr
library(mlr)

# Create a classification task for learning on Titantic Dataset and specify the target feature
task <- makeClassifTask(data = t_ds, target="Survived")

# Initialize the Naive Bayes classifier
selected_model <- makeLearner("classif.naiveBayes")

# Train the model
nb_mlr <- train(selected_model, task)

# Read the model learned
nb_mlr$learner.model

# Predict on the dataset without passing the target feature
predictions_mlr <- as.data.frame(predict(nb_mlr, newdata = t_ds[,1:3]))

# Confusion matrix to check accuracy
table(predictions_mlr[,1], t_ds$Survived)
```

https://www.geeksforgeeks.org/naive-bayes-classifier-in-r-programming/

==== Prompt =====

```
Homework # 2 (Generative Models) (100 points) Due on March 12, 11:59pm EST
We will be working with the Penguin dataset again as we did for Homework #1. Please use "Species" as yo
Using the target variable, Species, please conduct:
a. LinearDiscriminantAnalysis(30points):
a. Youwanttoevaluateallthe'features'ordependentvariablesandsee what should be in your model. Please comm
b. Justasuggestion:YoumightwanttoconsiderexploringfeaturePlot on the caret package. Basically, you look
c. Fit your LDA model using whatever predictor variables you deem appropriate. Feel free to split the da
d. Lookatthefitstatistics/accuracyrates.
b. QuadraticDiscriminantAnalysis(30points)
a. Samestepsasabovetoconsider
c. Naive Bayes (30 points)
a. Samestepsasabovetoconsider
d. Commentonthemodelsfits/strength/weakness/accuracyforallthesethree models that you worked with. (10 po
```