Practica Final

Giuliana Scoppettone

2023-01-28

Practica Final

Dataset de semillas (disponible en la URL):

https://archive.ics.uci.edu/ml/datasets/seeds

Librerías:

Se cargan las librerías necesarias:

```
library(tidyverse)
```

Se carga el dataset:

```
df_seeds <- read.table('https://archive.ics.uci.edu/ml/machine-learning-databases/00236/seeds_dataset.t
                   'longitud','anchura','coeficient.asimetria',
                   'longitud.ranura','tipo'))
```

PREGUNTA 1 ¿Cuántas filas y cuántas columnas tiene el dataframe df_seeds?

Respuesta:

```
dim(df_seeds)
```

```
## [1] 210   8
```

```
str(df_seeds)
```

```
## 'data.frame':    210 obs. of  8 variables:
##  $ area               : num  15.3 14.9 14.3 13.8 16.1 ...
##  $ perimetro          : num  14.8 14.6 14.1 13.9 15 ...
##  $ compacto           : num  0.871 0.881 0.905 0.895 0.903 ...
##  $ longitud           : num  5.76 5.55 5.29 5.32 5.66 ...
##  $ anchura            : num  3.31 3.33 3.34 3.38 3.56 ...
##  $ coeficient.asimetria: num  2.22 1.02 2.7 2.26 1.35 ...
##  $ longitud.ranura    : num  5.22 4.96 4.83 4.8 5.17 ...
##  $ tipo               : int  1 1 1 1 1 1 1 1 1 1 ...
```

El dataset contiene 210 filas y 8 columnas; 8 varibles y 210 observaciones.

PREGUNTA 2 Vamos a convertir en factor la columna tipo. Vamos a reemplazar los números por su correspondiente etiqueta (label). La correspondencia entre el código y el tipo es:

```
names(df_seeds)
```

```
## [1] "area"                "perimetro"           "compacto"
## [4] "longitud"            "anchura"             "coeficient.asimetria"
## [7] "longitud.ranura"     "tipo"
```

- 1 - Kama
- 2 - Rosa
- 3 - Canadian

Convierte en factor la columna tipo, respetando las etiquetas:

Respuesta:

```
df_seeds$tipo <- as.factor(df_seeds$tipo)
```

```
df_seeds <- df_seeds %>%
  mutate (tipo = case_when(
  tipo == "1" ~ "Kama",
  tipo == "2" ~ "Rosa",
  tipo == "3" ~ "Canadian"))

head(df_seeds, 5)
```

```
##    area perimetro compacto longitud anchura coeficient.asimetria
## 1 15.26     14.84   0.8710    5.763   3.312                2.221
## 2 14.88     14.57   0.8811    5.554   3.333                1.018
## 3 14.29     14.09   0.9050    5.291   3.337                2.699
## 4 13.84     13.94   0.8955    5.324   3.379                2.259
## 5 16.14     14.99   0.9034    5.658   3.562                1.355
##   longitud.ranura tipo
## 1           5.220 Kama
## 2           4.956 Kama
## 3           4.825 Kama
## 4           4.805 Kama
## 5           5.175 Kama
```

PREGUNTA 3 ¿Cual es la media del área de cada uno de los tipos?

Respuesta

La media del área de cada uno de los tipos en la siguiente tabla:

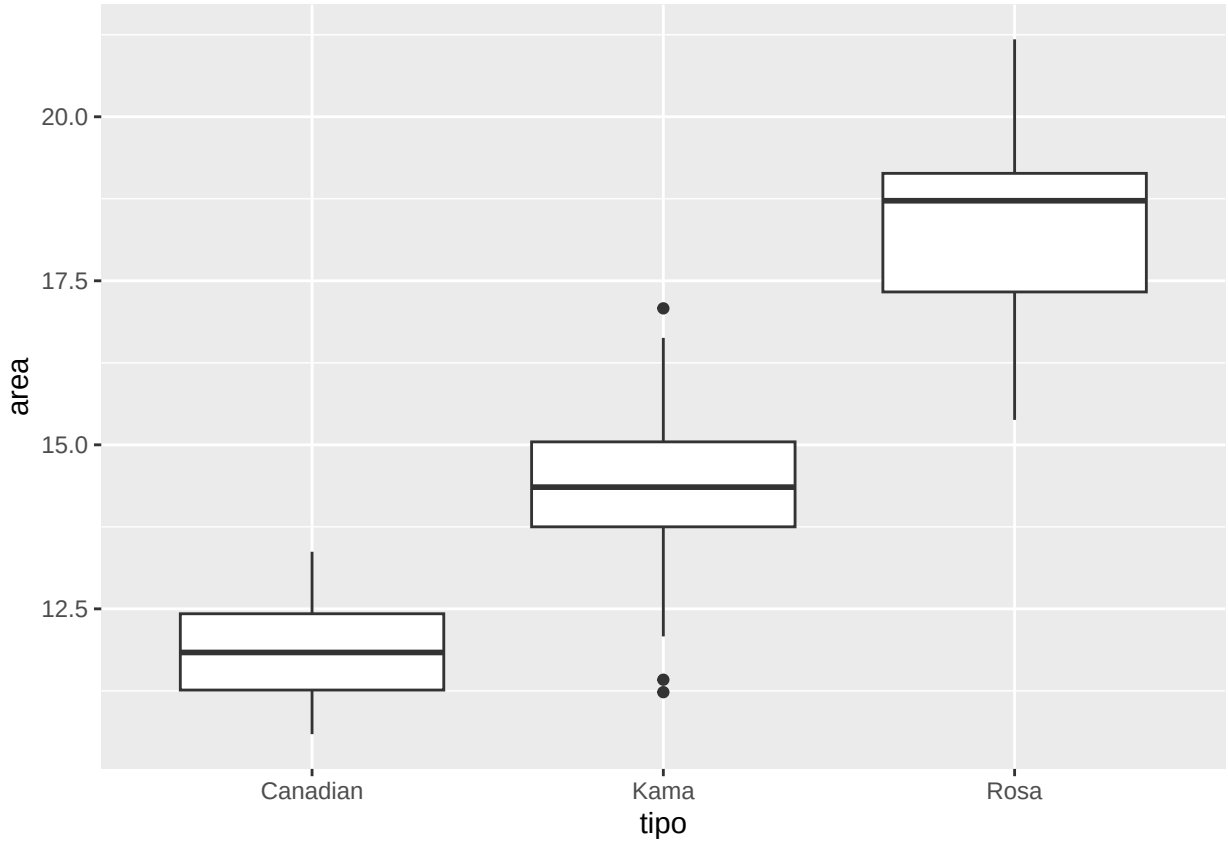
```
media_Tipos <- df_seeds %>%
    group_by(tipo) %>%
    dplyr::summarize(Mean = mean(area, na.rm=TRUE))

media_Tipos
```

```
## # A tibble: 3 x 2
##   tipo      Mean
##   <chr>    <dbl>
## 1 Canadian  11.9
## 2 Kama      14.3
## 3 Rosa      18.3
```

PREGUNTA 4 ¿Cómo se llama el siguiente tipo de gráfico?. ¿Qué representa la línea del centro de la caja?

```
ggplot(df_seeds, aes(x=tipo, y=area)) +
  geom_boxplot()
```

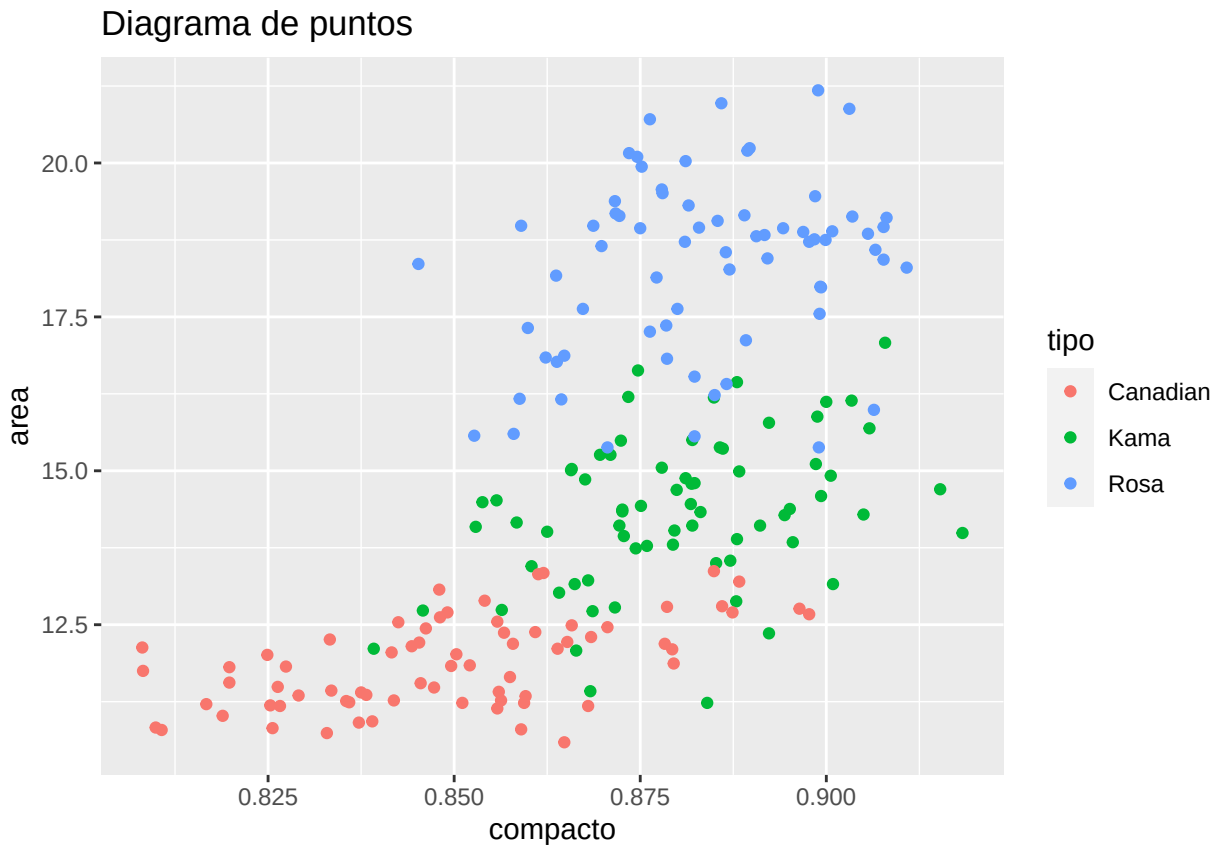


Respuesta: 'Es un gráfico de cajas. La línea central de cada caja representa el cuartil Q2 o mediana (el 50% de los datos)'

PREGUNTA 5 ¿Cómo pintarías un diagrama de puntos (o scatterplot) con ggplot con las siguientes características? - En el eje X la variable compacto - En el eje Y la variable area - Cada tipo de semilla debería tener un color diferente

Respuesta:

```
ggplot(df_seeds, aes(x=compacto, y=area,
      color=tipo)) +
  ggtitle("Diagrama de puntos") +
  geom_point()
```



PREGUNTA 6 ¿Qué hace la siguiente línea?:

```
df_seeds |> mutate(is_kama = tipo=='Kama') -> df_seeds
head(df_seeds,5)
```

```
##    area perimetro compacto longitud anchura coeficient.asimetria
## 1 15.26     14.84   0.8710    5.763   3.312                2.221
## 2 14.88     14.57   0.8811    5.554   3.333                1.018
## 3 14.29     14.09   0.9050    5.291   3.337                2.699
## 4 13.84     13.94   0.8955    5.324   3.379                2.259
## 5 16.14     14.99   0.9034    5.658   3.562                1.355
##   longitud.ranura tipo is_kama
## 1           5.220 Kama    TRUE
## 2           4.956 Kama    TRUE
## 3           4.825 Kama    TRUE
## 4           4.805 Kama    TRUE
## 5           5.175 Kama    TRUE
```

Respuesta: 'La línea crea la variable is_kama y en ella le asigna verdadero al tipo de semilla Kama, y falso a los demás tipos de semilla.'

PREGUNTA 7 Vamos a dividir el conjunto de datos en test y training porque vamos a entrenar un modelo que me permita diferenciar si una semilla es de tipo Kama o no. ¿Por qué es aconsejable dividir el dataset en los grupos de train y test?

```
set.seed(123) # Este set.seed hace que a todos nos generen los mismos número aleatorios
```

```
idx <- sample(1:nrow(df_seeds), 0.7 * nrow(df_seeds))

df_seeds_train <- df_seeds[idx, ]

df_seeds_test <- df_seeds[-idx, ]
```

Respuesta:

'Estos modelos son aconsejables porque permiten comparar los datos de entrenamiento y prueba para comprobar que el modelo final funciona correctamente o para identificar si apunta hacia alguna dirección en específico.

Los conjuntos de entrenamiento se usan comúnmente para estimar diferentes parámetros o para comparar el rendimiento de diferentes modelos.

El conjunto de datos de prueba se usa después de que se realiza el entrenamiento.'

PREGUNTA 8 Vamos a crear un modelo para realizar una clasificación binaria, donde le pasaremos como entrada las columnas: área, perimetro, compacto, longitud, coeficient.asimetria y longitud.ranura

¿Qué tipo de algoritmo o modelo debería usar?

Respuesta: 'Modelo de regresion logística:

la regresión logística es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (en este caso is_kama; TRUE y FALSE) en función de las variables independientes o predictoras.'

PREGUNTA 9 Crea un modelo que me permita clasificar si una semilla es de tipo Kama o no con las siguientes columnas: área, perimetro, compacto, longitud, coeficient.asimetria, longitud.ranura

Respuesta:

Se cargan las librerías:

```
library(caTools)
library(ROCR)
```

Análisis de observaciones:

Antes de contruir cualquier modelo se intenta explorar un poco la información disponible.

En este caso se tiene 140 observaciones con FALSE (No Kama) y 70 con TRUE (Kama).

```
table(df_seeds$is_kama)
```

```
##
## FALSE  TRUE
##   140    70
```

En la siguiente tabla se pueden observar los estadísticos básicos:

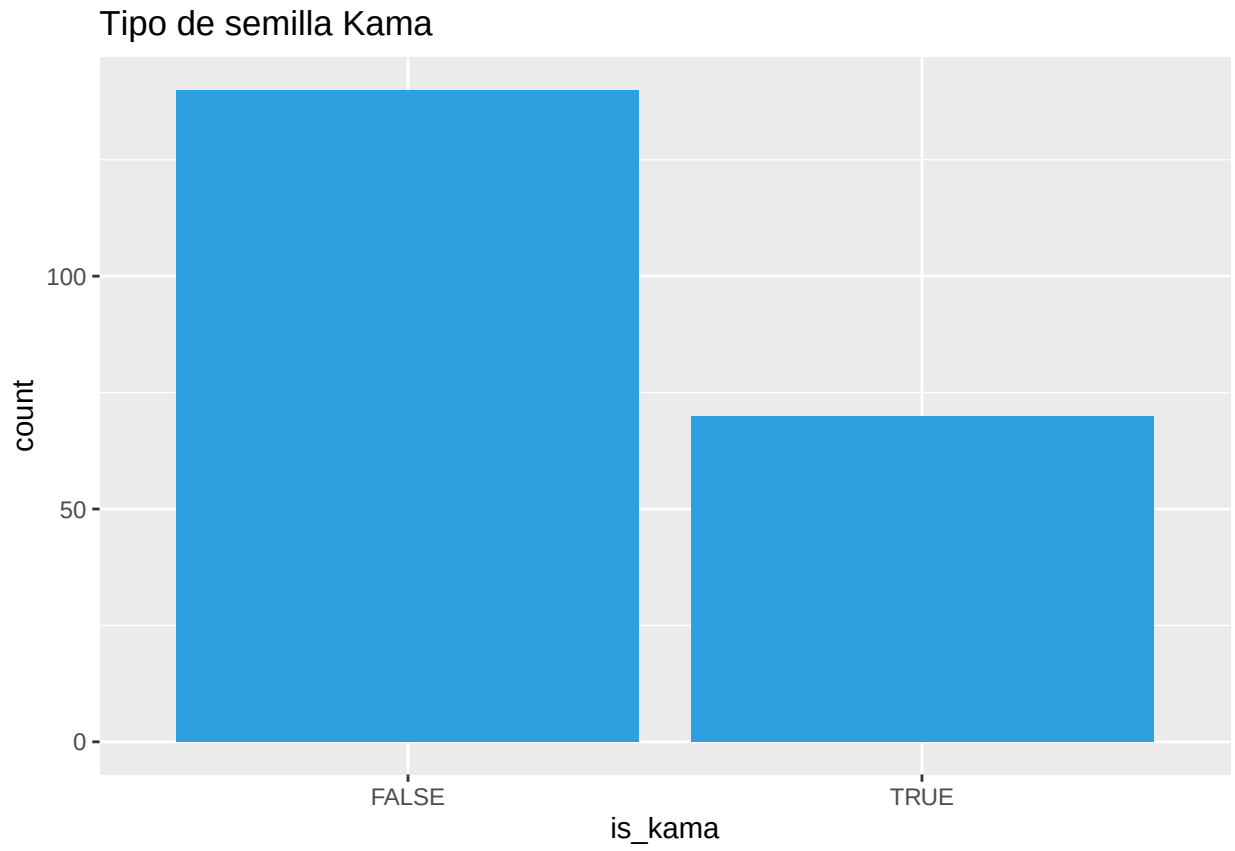
```
summary(df_seeds)
```

```
##       area         perimetro        compacto         longitud
##  Min.   :10.59   Min.   :12.41   Min.   :0.8081   Min.   :4.899
##  1st Qu.:12.27   1st Qu.:13.45   1st Qu.:0.8569   1st Qu.:5.262
##  Median :14.36   Median :14.32   Median :0.8734   Median :5.524
##  Mean   :14.85   Mean   :14.56   Mean   :0.8710   Mean   :5.629
##  3rd Qu.:17.30   3rd Qu.:15.71   3rd Qu.:0.8878   3rd Qu.:5.980
##  Max.   :21.18   Max.   :17.25   Max.   :0.9183   Max.   :6.675
##     anchura      coeficient.asimetria longitud.ranura     tipo
```

```
##  Min.   :2.630   Min.   :0.7651       Min.   :4.519   Length:210
##  1st Qu.:2.944   1st Qu.:2.5615       1st Qu.:5.045   Class :character
##  Median :3.237   Median :3.5990       Median :5.223   Mode  :character
##  Mean   :3.259   Mean   :3.7002       Mean   :5.408
##  3rd Qu.:3.562   3rd Qu.:4.7687       3rd Qu.:5.877
##  Max.   :4.033   Max.   :8.4560       Max.   :6.550
##   is_kama
##  Mode :logical
##  FALSE:140
##  TRUE :70
##
##
##
```

Ahora se observa graficamente las observaciones FALSE (tipo no Kama) y TRUE (tipo Kama).

```
ggplot(df_seeds, aes(is_kama)) +
  geom_bar(fill="#2E9FDF") +
  ggtitle("Tipo de semilla Kama")
```




```
summary(df_seeds_train)
```

Modelo de entrenamiento:

```
##       area          perimetro        compacto          longitud
##  Min.   :10.59   Min.   :12.41   Min.   :0.8082   Min.   :4.899
##  1st Qu.:12.28   1st Qu.:13.43   1st Qu.:0.8576   1st Qu.:5.236
```

```
##  Median :14.33   Median :14.28   Median :0.8746   Median :5.563
##  Mean   :14.92   Mean   :14.59   Mean   :0.8717   Mean   :5.640
##  3rd Qu.:18.06   3rd Qu.:15.93   3rd Qu.:0.8881   3rd Qu.:6.037
##  Max.   :21.18   Max.   :17.23   Max.   :0.9183   Max.   :6.675
##     anchura      coeficient.asimetria longitud.ranura     tipo
##  Min.   :2.642   Min.   :0.7651       Min.   :4.519   Length:147
##  1st Qu.:2.947   1st Qu.:2.4615       1st Qu.:5.047   Class :character
##  Median :3.212   Median :3.5980       Median :5.231   Mode  :character
##  Mean   :3.266   Mean   :3.7427       Mean   :5.427
##  3rd Qu.:3.574   3rd Qu.:4.7645       3rd Qu.:5.887
##  Max.   :4.033   Max.   :8.4560       Max.   :6.550
##   is_kama
##  Mode :logical
##  FALSE:100
##  TRUE :47
##
##
##
```

```
summary(df_seeds_test)
```

```
##       area         perimetro        compacto         longitud
##  Min.   :10.79   Min.   :12.83   Min.   :0.8081   Min.   :5.008
##  1st Qu.:12.32   1st Qu.:13.48   1st Qu.:0.8569   1st Qu.:5.272
##  Median :14.70   Median :14.41   Median :0.8722   Median :5.482
##  Mean   :14.67   Mean   :14.50   Mean   :0.8695   Mean   :5.602
##  3rd Qu.:16.32   3rd Qu.:15.25   3rd Qu.:0.8863   3rd Qu.:5.861
##  Max.   :20.97   Max.   :17.25   Max.   :0.9153   Max.   :6.666
##     anchura      coeficient.asimetria longitud.ranura     tipo
##  Min.   :2.630   Min.   :0.8551       Min.   :4.607   Length:63
##  1st Qu.:2.933   1st Qu.:2.6995       1st Qu.:5.041   Class :character
##  Median :3.286   Median :3.6000       Median :5.219   Mode  :character
##  Mean   :3.240   Mean   :3.6011       Mean   :5.363
##  3rd Qu.:3.495   3rd Qu.:4.7680       3rd Qu.:5.749
##  Max.   :3.991   Max.   :7.0350       Max.   :6.498
##   is_kama
##  Mode :logical
##  FALSE:40
##  TRUE :23
##
##
##
```

Modelo glm:

```
logistic_model <- glm(is_kama ~ area + perimetro +
                        compacto + longitud + coeficient.asimetria +
                        longitud.ranura,
                      data = df_seeds_train,
                      family = "binomial")
summary(logistic_model)
```

```
##
## Call:
## glm(formula = is_kama ~ area + perimetro + compacto + longitud +
##     coeficient.asimetria + longitud.ranura, family = "binomial",
```

```
##     data = df_seeds_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.02351  -0.01523  -0.00234   0.00214   2.26723
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -671.4230   240.4966  -2.792  0.00524 **
## area                  -20.5841     7.4615  -2.759  0.00580 **
## perimetro              32.0042    14.2003   2.254  0.02421 *
## compacto              431.4114   157.0099   2.748  0.00600 **
## longitud               59.7991    25.4630   2.348  0.01885 *
## coeficient.asimetria   -1.8366     0.6386  -2.876  0.00403 **
## longitud.ranura       -36.6699    14.0928  -2.602  0.00927 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 184.239  on 146  degrees of freedom
## Residual deviance:  23.053  on 140  degrees of freedom
## AIC: 37.053
##
## Number of Fisher Scoring iterations: 9
```

```
logistic_model$coefficients
```

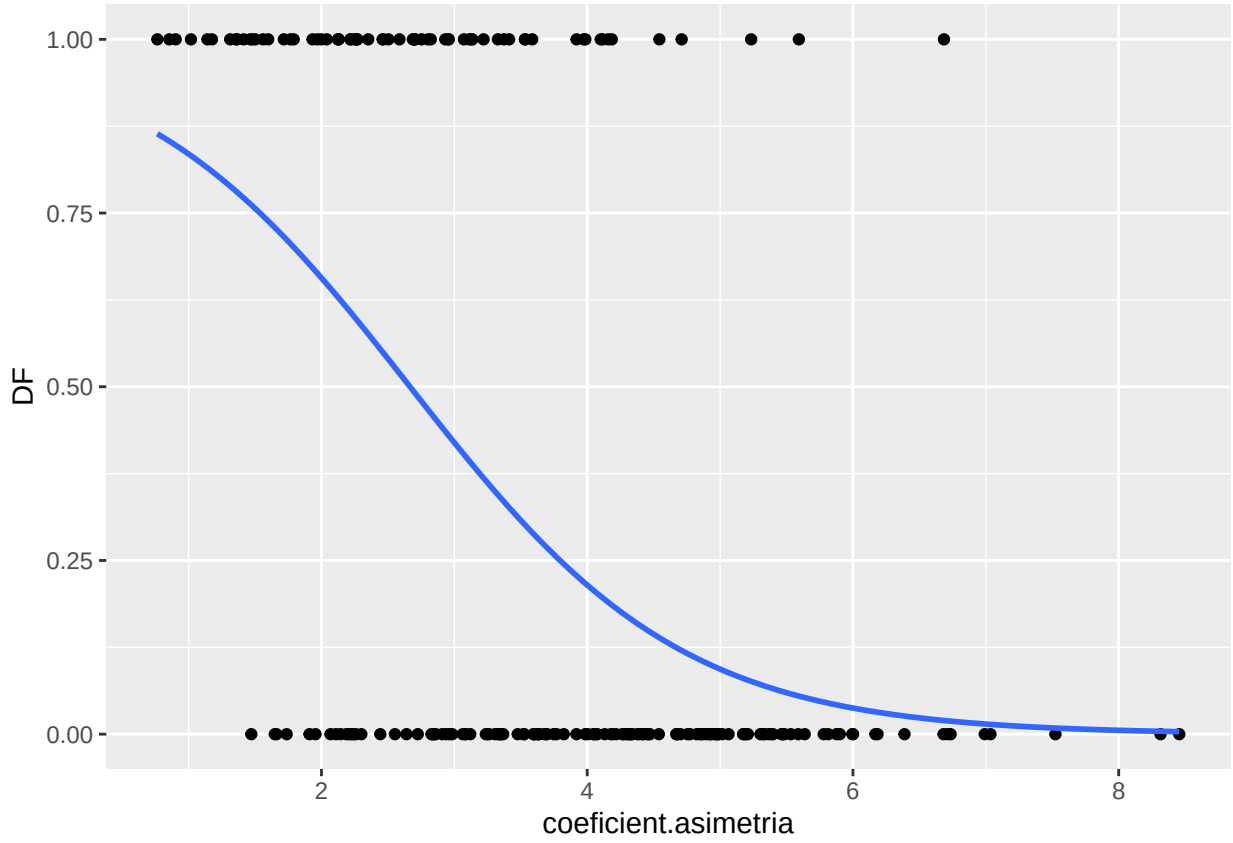
```
##          (Intercept)                 area            perimetro
##           -671.42304            -20.58410             32.00424
##              compacto              longitud coeficient.asimetria
##            431.41137             59.79908              -1.83660
##      longitud.ranura
##            -36.66987
```

Se establece como notación: 1 -> TRUE y 0 -> FALSE

```
DF <- ifelse(df_seeds$is_kama == "TRUE", 1, 0)
```

Gráfica con curva logística para la variable “coeficient.asimetria”.

```
ggplot(df_seeds, aes(x=coeficient.asimetria , y=DF, na.rm = TRUE)) +
  geom_point() +
  geom_smooth(method = "glm",
    method.args = list(family = "binomial"),
    se = FALSE)
```



Predicciones:

```
probabilities <- logistic_model %>%
  predict(df_seeds_test, type = "response")

predicted.classes <- ifelse(probabilities > 0.5, "TRUE", "FALSE")
```

Precisión de predicción

```
observed.classes <- df_seeds_test$is_kama
mean(predicted.classes == observed.classes, na.rm = TRUE)
```

```
## [1] 0.984127
```

Precisión de predicción

```
observed.classes <- df_seeds_test$is_kama
mean(predicted.classes == observed.classes, na.rm = TRUE)
```

```
## [1] 0.984127
```

```
performance_data <- data.frame(observed = df_seeds_test$is_kama,
           predicted = predicted.classes)
```

```
verdadero <- sum(performance_data$observed =="TRUE")
falso <- sum(performance_data$observed =="FALSE")
predicted_positive <- sum(performance_data$predicted=="TRUE")
predicted_negative <- sum(performance_data$predicted=="FALSE")
total <- nrow(performance_data)
```

```
data.frame(verdadero, falso,predicted_positive,predicted_negative)
```

```
##   verdadero falso predicted_positive predicted_negative
## 1        23    40                24                39
```

```
tp<-sum(performance_data$observed=="TRUE" & performance_data$predicted=="TRUE")
tn<-sum(performance_data$observed=="FALSE" & performance_data$predicted=="FALSE")
fp<-sum(performance_data$observed=="FALSE" & performance_data$predicted=="TRUE")
fn<-sum(performance_data$observed=="TRUE" & performance_data$predicted=="FALSE")
data.frame(tp,tn,fp,fn)
```

```
##   tp tn fp fn
## 1 23 39  1  0
```

```
accuracy <- (tp+tn)/total
error_rate <- (fp+fn)/total
sensitivity <- tp/verdadero
especificity <- tn/falso
precision <- tp/predicted_positive
npv <- tn / predicted_negative
data.frame(accuracy,error_rate,sensitivity,especificity,precision,npv)
```

```
##   accuracy error_rate sensitivity especificity precision npv
## 1 0.984127 0.01587302           1        0.975 0.9583333   1
```

PREGUNTA 10 Si usamos un umbral de 0 en la salida del modelo (lo que equivale a probabilidad de 0.5 cuando usamos el predict con type='response') ¿Cuáles son los valores de precisión y exhausitividad?

Respuesta.

```
probabilities <- logistic_model %>%
  predict(df_seeds_test, type = "response")

predicted.classes <- ifelse(probabilities > 0.5, "TRUE", "FALSE")
```

Precisión de predicción

```
observed.classes <- df_seeds_test$is_kama
mean(predicted.classes == observed.classes, na.rm = TRUE)
```

```
## [1] 0.984127
```

```
performance_data <- data.frame(observed = df_seeds_test$is_kama,
           predicted = predicted.classes)
```

```
verdadero <- sum(performance_data$observed =="TRUE")
falso <- sum(performance_data$observed =="FALSE")
predicted_positive <- sum(performance_data$predicted=="TRUE")
predicted_negative <- sum(performance_data$predicted=="FALSE")
total <- nrow(performance_data)
data.frame(verdadero, falso,predicted_positive,predicted_negative)
```

```
##   verdadero falso predicted_positive predicted_negative
## 1        23    40                24                39
```

```
tp<-sum(performance_data$observed=="TRUE" & performance_data$predicted=="TRUE")
tn<-sum(performance_data$observed=="FALSE" & performance_data$predicted=="FALSE")
fp<-sum(performance_data$observed=="FALSE" & performance_data$predicted=="TRUE")
```

```r
fn<-sum(performance_data$observed=="TRUE" & performance_data$predicted=="FALSE")
data.frame(tp,tn,fp,fn)
```

```
##   tp tn fp fn
## 1 23 39  1  0
```

```r
precision <- (tp+tn)/total
error_rate <- (fp+fn)/total
sensibilidad <- tp/verdadero
especificidad <- tn/falso
precision <- tp/predicted_positive
npv <- tn / predicted_negative
data.frame(accuracy,error_rate,sensitivity,especificity,precision,npv)
```

```
##   accuracy error_rate sensitivity especificity precision npv
## 1 0.984127 0.01587302           1        0.975 0.9583333   1
```

PREGUNTA 11 ¿Qué están haciendo las siguientes líneas?

```r
set.seed(123)

cl <- df_seeds |>
  select(area,perimetro,compacto,longitud,
                      anchura,coeficient.asimetria,longitud.ranura) |>
  kmeans(3)

table(real=df_seeds$tipo, cluster=cl$cluster)
```

```
##           cluster
## real        1  2  3
##   Canadian  0  2 68
##   Kama      1 60  9
##   Rosa     60 10  0
```

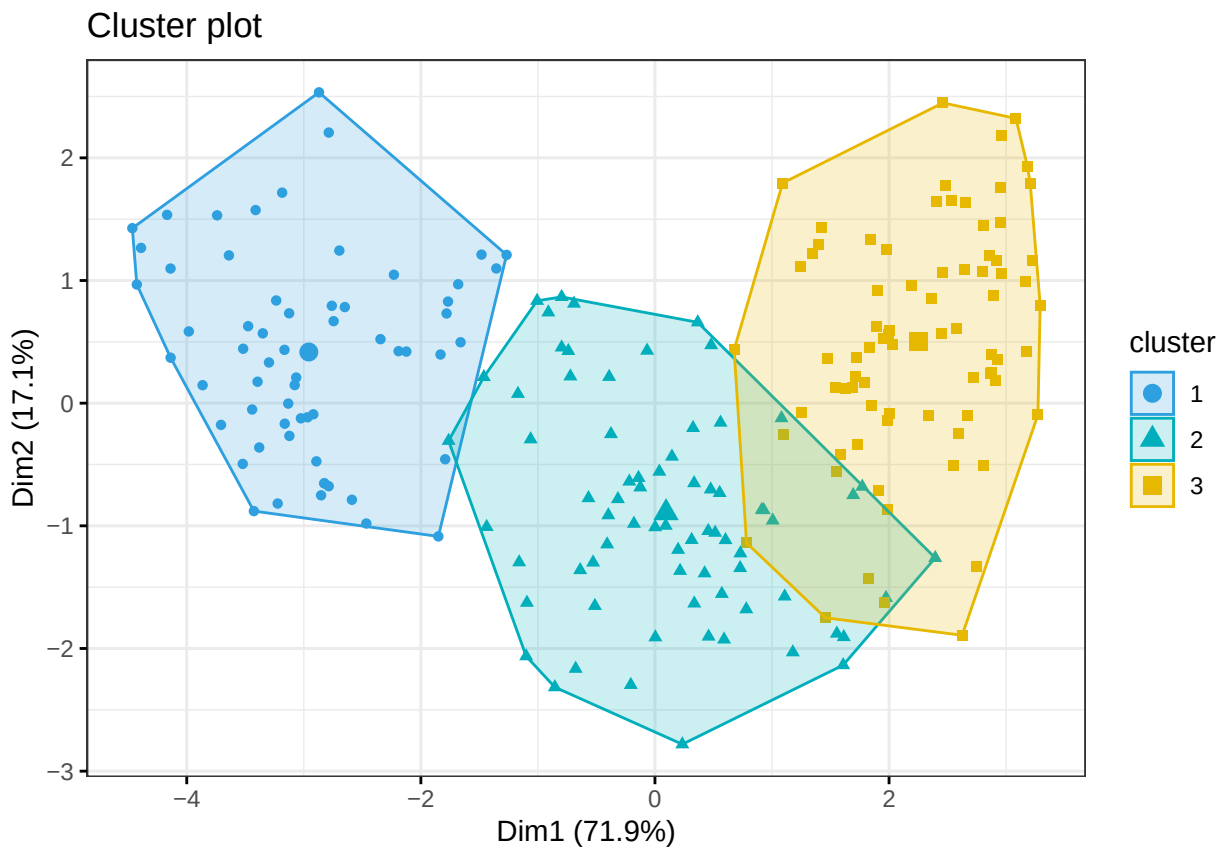
Respuesta: 'En este caso se genera un agrupamiento k-medias, entre tres categorías: Canadian, Kama, Rosa.'

```r
library(factoextra)

cluster=cl$cluster

df_seeds <- df_seeds[, -8]
df_seeds <- df_seeds[, -8]

fviz_cluster(cl, data = df_seeds,
             palette = c("#2E9FDF", "#00AFBB", "#E7B800"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
             )
```



Gráficamente se pueden notar los centriodes y los agrupamientos para cada una de las categorías.