Ejercicios de libro - Sección 2.4

Diego Velasco

03/04/2023

Ejercicio 1

For each of parts (a) through (d), indicate whether we would generally expect the performance of a flexible statistical learning method to be better or worse than an inflexible method. Justify your answer.

(a) The sample size n is extremely large, and the number of predictors p is small.

(b) The number of predictors p is extremely large, and the number of observations n is small.

(c) The relationship between the predictors and response is highly non-linear.

(d) The variance of the error terms, i.e. $\sigma^2 = Var(\epsilon)$, is extremely high.

Respuesta

(a) Cuando el tamaño de la muestra es muy grande comparado con la dimensionalidad de la misma, es esperable que un modelo flexible sea más apropiado.

Los modelos más flexibles tienden a tener menor error de sesgo (debido a que no se limita la clase de funciones a utilizar a ningún tipo en particular) y mayor poder predictivo, pero pueden tener desventajas desde el punto de vista de interpretabilidad, sobreajuste y varianza. Sin embargo, estas desventajas se verán disminuidas en un dataset de este tipo.

En primer lugar, un dataset con una gran cantidad de observaciones implica que es posible generar un dataset de entrenamiento de gran tamaño. Esto disminuye el riesgo de sobreajuste tanto en modelos paramétricos como no paramétricos, ya que es más difícil que el modelo se ajuste a la gran cantidad de observaciones en dicha partición. Asimismo, si bien los métodos más flexibles tienen mayor varianza asociada, entrenar un modelo de este tipo con mayor cantidad de datos tenderá a reducir este problema. En esta línea, la varianza del método podría reducirse utilizando métodos de agregación o de validación cruzada. Dado que se tienen pocas variables, el costo computacional adicional por entrenar al modelo más de una vez puede resultar manejable.

Por el otro lado, un dataset de pocas variables predictoras resulta naturalmente más fácil de interpretar, lo que resulta más favorable al usar modelos que sufran de poca interpretabilidad. Asimismo, un dataset de gran tamaño y baja dimensionalidad es un escenario ideal para utilizar métodos basados en distancias (KNN, por ejemplo), ya que a medida que aumenta la dimensionalidad, los puntos tienden a estar más lejos entre sí.

(b) Por el contrario, cuando el tamaño de la muestra es muy pequeño comparado con la dimensionalidad de la misma, es esperable que un modelo menos flexible sea más apropiado.

Los modelos más flexibles requieren mayor cantidad de datos para ajustar, por lo que un dataset con pocos datos de entrenamiento implica un mayor riesgo de sobreajuste. En esta línea, los modelos menos flexibles tienden a tener menor varianza, reduciendo el riesgo de sobreajuste. Asimismo, una mayor dimensionalidad implica mayor distancias entre puntos, lo que es desfavorable para el uso de métodos basados en distancias.

Por otro lado, en el caso de que se tengan más variables que datos, es posible entrenar un modelo paramétrico utilizando un proceso de selección de variables que reduzca la cantidad de las mismas a un número razonable (Subset Selection o Lasso, por ejemplo).

(c) Cuando la relación entre las variables predictoras y la variable objetivo es altamente no lineal, es esperable que un modelo flexible sea más apropiado.

La principal razón de esto es que los modelos flexibles tienden a tener un error de sesgo menor, por lo que permiten capturar una variedad más amplia de relaciones entre los datos, lo que es útil en un dataset como el planteado. Por el otro lado, los métodos menos flexibles introducen un error de sesgo mayor. Si la relación entre variables predictoras y objetivo es altamente no lineal, la utilización de un método con poca flexibilidad implicaría la elección previa de una clase de funciones en base a datos de difícil interpretación.

(d) Si la varianza del error irreducible es extremadamente alta, es esperable que un modelo menos flexible sea más apropiado.

Dada una observación $X = (x_1, ..., x_d)$, el valor de la variable objetivo para esa observación será $Y = f(X) + \epsilon$. El objetivo es estimar la función $f(X)$ lo más precisamente posible. Aún en el caso de que esta función sea razonablemente sencilla, una alta varianza de ϵ agregará ruido a los datos y "esconderá" la verdadera función $f(X)$. Un modelo flexible será más sensible a este ruido, sobreajustando las predicciones a los valores afectados por este ruido y así perdiendo la relación buscada. Por el otro lado, un modelo menos flexible tendrá un menor nivel de ajuste a los datos, teniendo menor sensibilidad al ruido, pudiendo aproximar la función $f(X)$ con mayor precisión.

Ejercicio 4

4. You will now think of some real-life applications for statistical learning.

(a) Describe three real-life applications in which classification might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

(b) Describe three real-life applications in which regression might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

(c) Describe three real-life applications in which cluster analysis might be useful.

Respuesta

a.
- **Aplicación 1**: Predecir si un paciente tendrá una enfermedad o no, en base a resultados de análisis médicos, como pueden ser: resultados de análisis de sangre (glucosa, glóbulos rojos y blancos, plaquetas, etc.), resultados de análisis de orina, placas o radiografías, etc. El objetivo, en este caso, sería de predicción.
- **Aplicación 2:** Predecir si un equipo fallará o no, en base a mediciones de sensores relacionados con el equipo. Las variables que podrían resultar de interés pueden ser: temperatura del equipo, temperatura ambiente, presión del líquido, voltaje, amperaje, hora de medición, etc. y la variable objetivo sería "falla" (variable binaria). Los sensores medirían estas cantidades con cierta periodicidad en el tiempo, y se buscaría determinar si un equipo falla o no en función de los valores de estas variables cierto tiempo antes de la medición en la que se registra la falla, de manera de predecir la falla con un margen de acción suficiente para prevenirla. En este caso, el objetivo principal sería predictivo. Sin embargo, podría también realizarse con fines de inferencia de manera de averiguar qué parámetros anticipan con mayor poder la presencia de una falla en el corto plazo (valores medios de presión o temperatura, valores picos, valores mínimos, etc.).
- **Aplicación 3:** Análisis crediticio. Un banco podría determinar si es conveniente autorizar una línea de crédito a una persona, en base a datos personales y antecedentes financieros. Las variables que podrían resultar de interés pueden ser: edad, género, estado civil, tenencia de empleo (variable

binaria), ingreso mensual, cantidad de presencias en clearing, patrimonio de la persona, etc. En este caso, el objetivo principal sería predictivo.

b.
- **Aplicación 1**: Predicción del valor de venta de una casa en función de parámetros asociados a la misma. Las variables predictoras razonablemente relevantes podrían ser: m2 construidos, m2 totales, cantidad de ambientes, cantidad de pisos, ubicación, antigüedad, si la misma es un reciclaje o no, etc. El objetivo, en este caso, podría ser tanto predictivo o de inferencia. Por ejemplo, un potencial inversor inmobiliario podría querer determinar el precio de venta de la casa previo a su construcción para determinar si invertir o no, mientras que una persona que está buscando una casa para comprar podría querer determinar las variables que más juegan en el precio final para buscar casas más baratas.
- **Aplicación 2:** Identificar las variables que más influyen en la cantidad de rapiñas (o algún tipo de crimen) en una ciudad. Las variables predictoras razonablemente relevantes podrían ser: población total, cantidad de gente por encima y debajo de la franja de pobreza, cantidad de gente por nivel educativo, país, cantidad de armas por habitante, cantidad de otro crimen relacionado, ubicación de la ciudad (país, continente, etc.), mes del año. Las observaciones deberían incluir datos de varias ciudades, así como de varios momentos en el tiempo para dicha ciudad. El objetivo más inmediato sería el de inferencia, logrando identificar qué variables tienen mayor predictivo sobre la variable objetivo y así enfocarse en dichas variables para reducir el número de crímenes.
- **Aplicación 3:** Predecir la cantidad de horas que llevará la construcción de una obra en función de datos de obras pasadas. Las variables predictoras razonablemente relevantes podrían ser: m3 a construir, tipo de obra, cantidad de rubros de construcción, cantidades de los rubros más significativos (hormigón, acero, movimiento de suelos), cantidad de días de lluvia en el período construido, ubicación de la obra, cantidad de km al centro poblado más cercano, cantidad de gente en la obra. Con las variables indicadas, el problema debería ser de inferencia, ya que no se contará con valores como la cantidad de días de lluvia de una obra previo a su contrucción. Sin embargo, podría encararse como un problema de inferencia y luego de predicción, omitiendo ese tipo de variables y evaluando la precisión del método.

c.
- **Aplicación 1:** Segmentar los clientes de alguna empresa en base a criterios no establecidos de antemano, de manera de generar estrategias de marketing personalizadas. Las variables predictoras razonablemente relevantes podrían ser personales del cliente (edad o rango etario, género, nacionalidad) y de hábitos comerciales (cantidad de compras por mes, categoría más comprada, categoría más buscada, años de antigüedad, periodicidad de entrada a la plataforma de la empresa, etc.)
- **Aplicación 2:** Segmentación de productos que se compran juntos. Una empresa que vende productos de alguna categoría (limpieza, comestibles, productos del hogar, etc) podría querer agrupar productos que frecuentemente se compran juntos de manera de ofrecerlos en promoción, colocarlos juntos en una tienda física, o potenciar la venta de esos grupos de productos de alguna manera. Las variables de interés podrían ser: producto, identificador de compra (para identificar que los productos fueron comprados juntos), precio de los productos, cantidad comprada, lugar donde se realizó la compra, si la compra se realizó en tienda digital o física (variable binaria), monto total de la compra, etc.
- **Aplicación 3:** Planificando una campaña política, el equipo de planificación podría buscar segmentar una población objetivo en base a datos de orientación política y antecedentes de las personas, de manera de construir una estrategia de campaña personalizada para cada grupo generado. Las variables de interés podrían ser personales: edad, género, nivel educativo, ingreso anual, estado civil, religión, cantidad de hijos, etc. así como relacionadas a la orientación política: cantidad de veces que votó a un partido político, opinión sobre temas sociales y políticos particulares, partido político de preferencia, si ejerce el voto o no, etc. Los datos podrían ser obtenidos en base a una encuesta en la cual las preguntas de carácter político sean de múltiple opción, de manera de que las respuestas posibles sean las mismas para todos los encuestados.

Ejercicio 9

```
auto = read.table('Auto.data', header = TRUE, na.strings = '?', stringsAsFactors = TRUE)
head(auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
View (auto)
```

a. **Which of the predictors are quantitative, and which are qualitative?**

```
for (var in names(auto)){
  cat(paste0('number of unique values of ', var, ': ',(length(unique((auto[,var])))), '\n'))
}
```

```
## number of unique values of mpg: 129
## number of unique values of cylinders: 5
## number of unique values of displacement: 82
## number of unique values of horsepower: 94
## number of unique values of weight: 350
## number of unique values of acceleration: 95
## number of unique values of year: 13
## number of unique values of origin: 3
## number of unique values of name: 304
```

```
unique(auto$origin)
```

```
## [1] 1 3 2
```

```
unique(auto$cylinders)
```

```
## [1] 8 4 6 3 5
```

Las variables **cuantitativas** son: mpg, displacement, horsepower, weight, acceleration, year.

Las variables **cualitativas** son: cylinders, origin, name.

Cabe destacar que, si bien las variable "origin" presenta valores numéricos, la misma presenta únicamente tres valores distintos, los cuales parecen ser una codificación de una variable categórica, por lo que se considera que la misma es categórica. Asimismo, la variable "cylinders" indica la cantidad de cilindros del auto, por lo que la naturaleza de la misma es numérica. Sin embargo, dada que la misma presenta únicamente cinco valores distintos, la misma también se considera como cualitativa.

b. **What is the range of each quantitative predictor? You can answer this using the range() function.**

```
for (var in names(auto)){
  if (var != 'name'& var != 'origin' & var != 'cylinders' & var != 'horsepower'){
    cat('Range of', var, ':', range(auto[,var]),'\n')
  }
  else if (var == 'horsepower'){
    cat ('Range of', var, ':', range(na.omit(auto[,var])), '\n')
  }
}
```

```
## Range of mpg : 9 46.6
## Range of displacement : 68 455
## Range of horsepower : 46 230
## Range of weight : 1613 5140
## Range of acceleration : 8 24.8
## Range of year : 70 82
```

c. **What is the mean and standard deviation of each quantitative predictor?**

```
for (var in names(auto)){
  if (var != 'name'& var != 'origin' & var != 'cylinders' & var != 'horsepower'){
    cat(var, '\n')
    cat('Mean:', mean(auto[,var]),'\n')
    cat('Standard deviation:', sd(auto[,var]),'\n')
    cat('------------------------------------------------------', '\n')
  }
  else if (var == 'horsepower'){
    cat (var, '\n')
    cat ('Mean:', mean(na.omit(auto[,var])), '\n')
    cat ('Standard deviation: ', sd(na.omit(auto[,var])), '\n')
    cat('------------------------------------------------------', '\n')
  }
}
```

```
## mpg
## Mean: 23.51587
## Standard deviation: 7.825804
## ------------------------------------------------------
## displacement
## Mean: 193.5327
## Standard deviation: 104.3796
## ------------------------------------------------------
## horsepower
## Mean: 104.4694
## Standard deviation:  38.49116
## ------------------------------------------------------
## weight
## Mean: 2970.262
## Standard deviation: 847.9041
## ------------------------------------------------------
## acceleration
## Mean: 15.55567
## Standard deviation: 2.749995
## ------------------------------------------------------
```

```
## year
## Mean: 75.99496
## Standard deviation: 3.690005
## ------------------------------------------------------
```

d. **Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?**

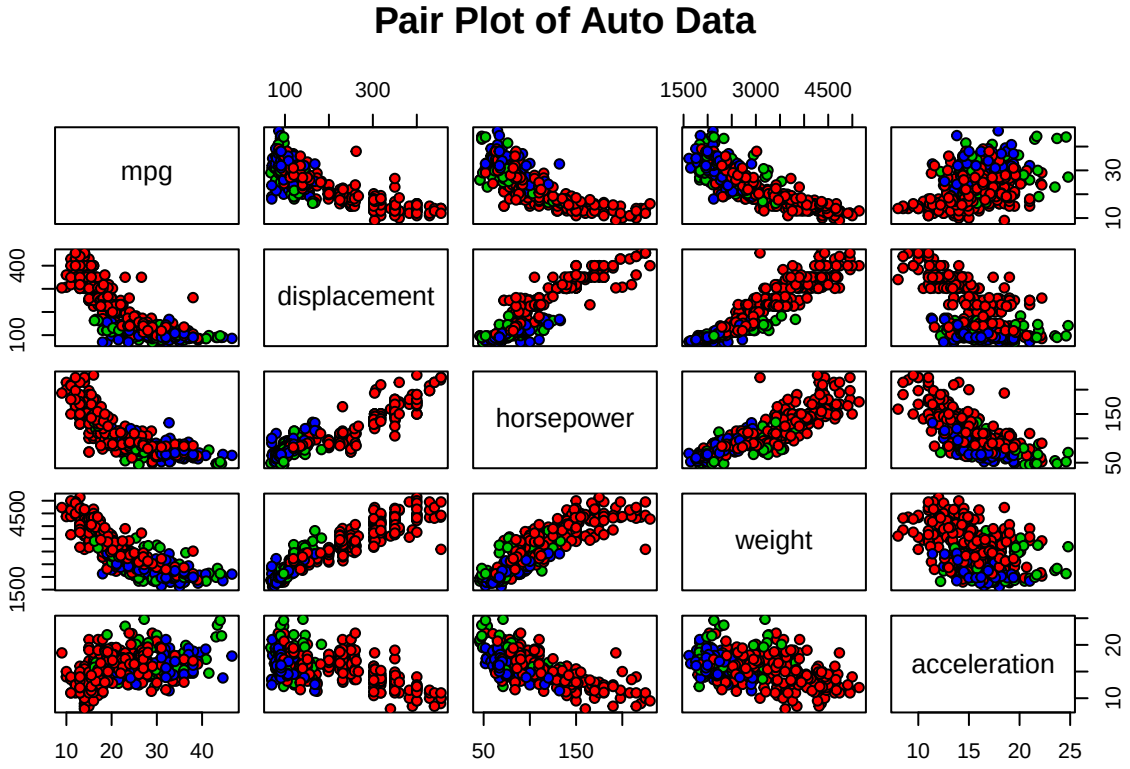
```
auto_filtered = auto[-c(10:85), ]

for (var in names(auto_filtered)){
  if (var != 'name'& var != 'origin' & var != 'cylinders' & var != 'horsepower'){
    cat(var, '\n')
    cat('Range:', range(auto_filtered[,var]),'\n')
    cat('Mean:', mean(auto_filtered[,var]),'\n')
    cat('Standard deviation:', sd(auto_filtered[,var]),'\n')
    cat('------------------------------------------------------', '\n')
  }
  else if (var == 'horsepower'){
    cat(var, '\n')
    cat ('Range:', range(na.omit(auto_filtered[,var])), '\n')
    cat ('Mean:', mean(na.omit(auto_filtered[,var])), '\n')
    cat ('Standard deviation: ', sd(na.omit(auto_filtered[,var])), '\n')
    cat('------------------------------------------------------', '\n')
  }
}
```

```
## mpg
## Range: 11 46.6
## Mean: 24.43863
## Standard deviation: 7.908184
## ------------------------------------------------------
## displacement
## Range: 68 455
## Mean: 187.0498
## Standard deviation: 99.63539
## ------------------------------------------------------
## horsepower
## Range: 46 230
## Mean: 100.9558
## Standard deviation:  35.89557
## ------------------------------------------------------
## weight
## Range: 1649 4997
## Mean: 2933.963
## Standard deviation: 810.6429
## ------------------------------------------------------
## acceleration
## Range: 8.5 24.8
## Mean: 15.72305
## Standard deviation: 2.680514
## ------------------------------------------------------
## year
## Range: 70 82
## Mean: 77.15265
```

```
## Standard deviation: 3.11123
## ----------------------------------------------------------
```

e. **Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.**

```
# Filtrado de dataset para mostrar variables de interés
auto_to_plot = auto[,-c(2,7:9)]

# Pairplots, diferenciando por color según valor de origin
pairs(auto_to_plot, pch = 21, bg = c("red", "green3", "blue")[unclass(auto$origin)],
      col = "black", main = "Pair Plot of Auto Data")
```

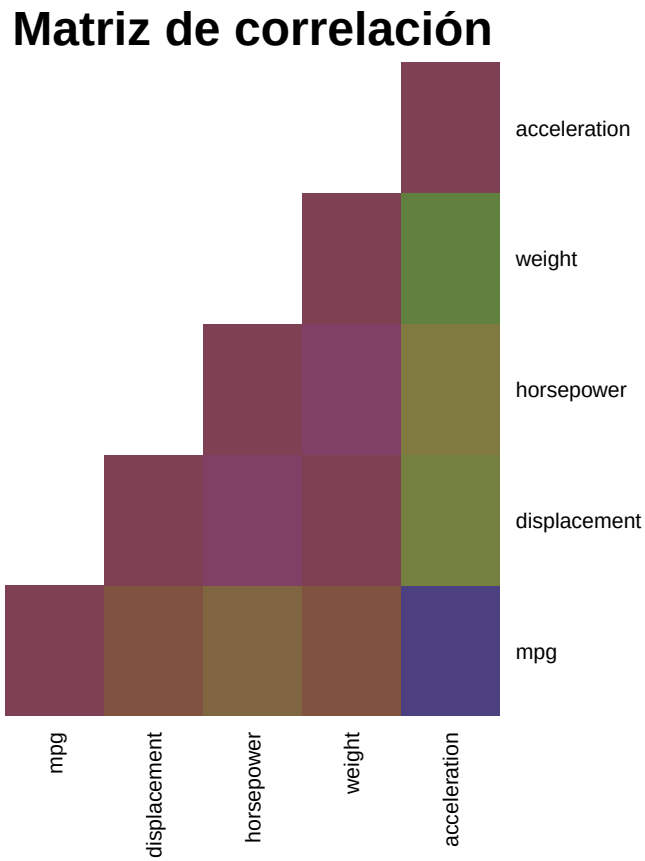



```
# Matriz de correlación
cor_mat = cor(na.omit(auto_to_plot))
cor_mat[upper.tri(cor_mat)] = NA
print (cor_mat, na.print = '')
```

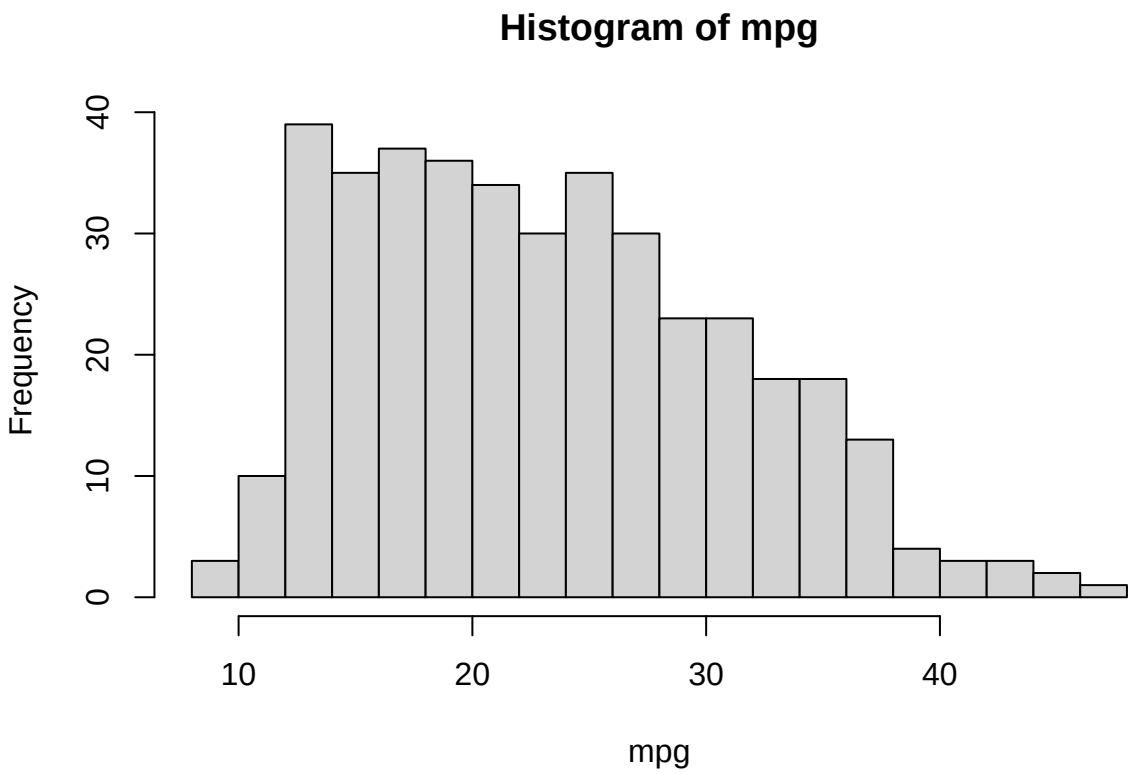
```
##                     mpg displacement horsepower     weight acceleration
## mpg           1.0000000
## displacement -0.8051269    1.0000000
## horsepower   -0.7784268    0.8972570  1.0000000
## weight       -0.8322442    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285   -0.5438005 -0.6891955 -0.4168392            1
```

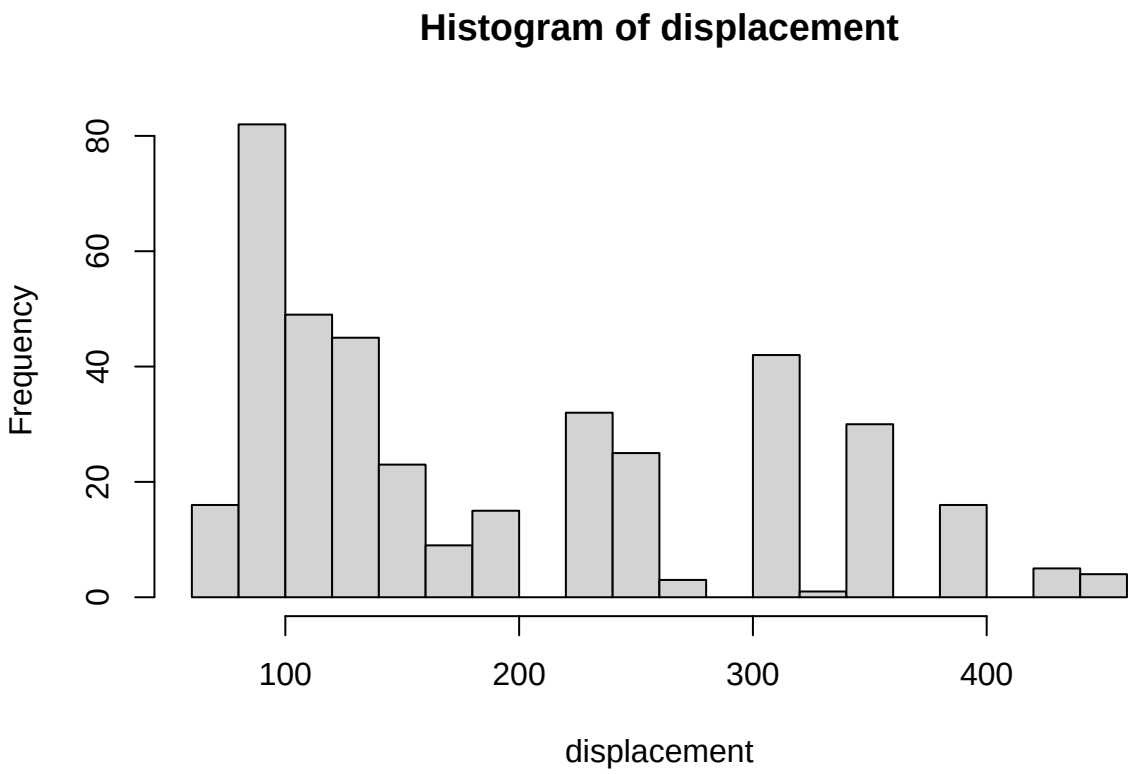
```
heatmap(t(cor_mat),
        col = rainbow(20, s = 0.5, v = 0.5),
```

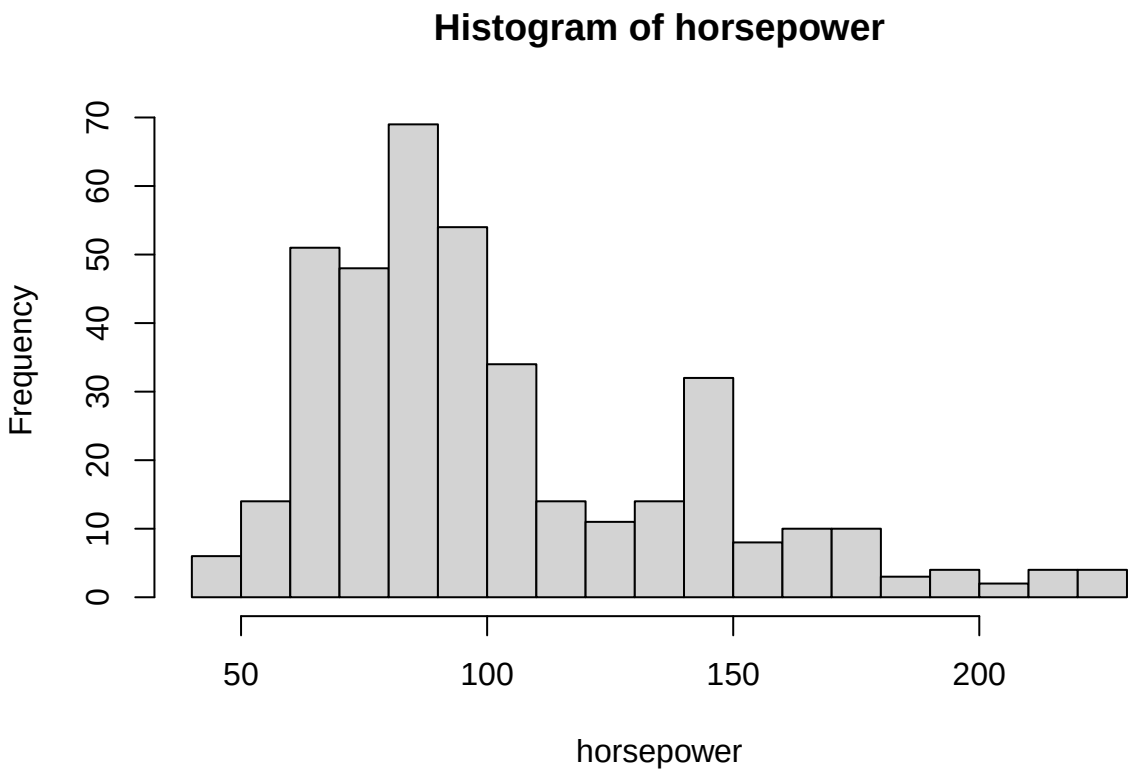
```
        Rowv = NA, Colv = NA, cexRow = 0.8, cexCol = 0.8,
        main = "Matriz de correlación",
        scale = "none",
        margins = c(5, 10),
        breaks = seq(-1, 1, by = 0.1),
        )
```

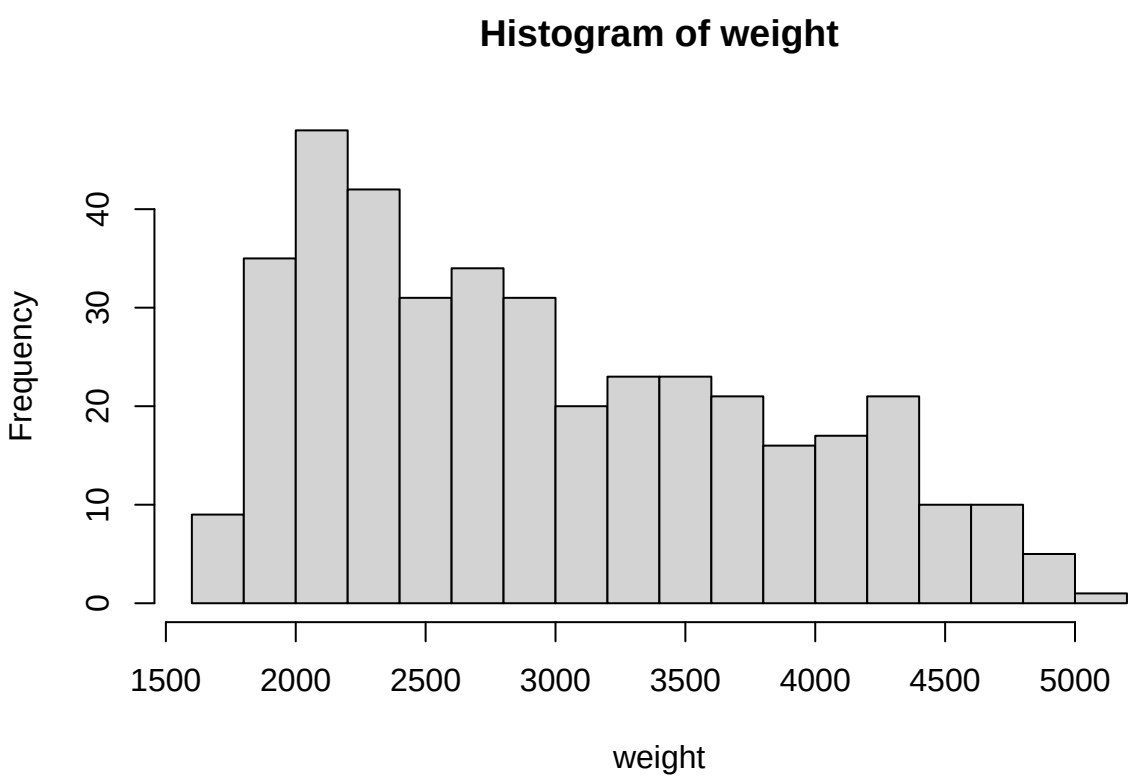


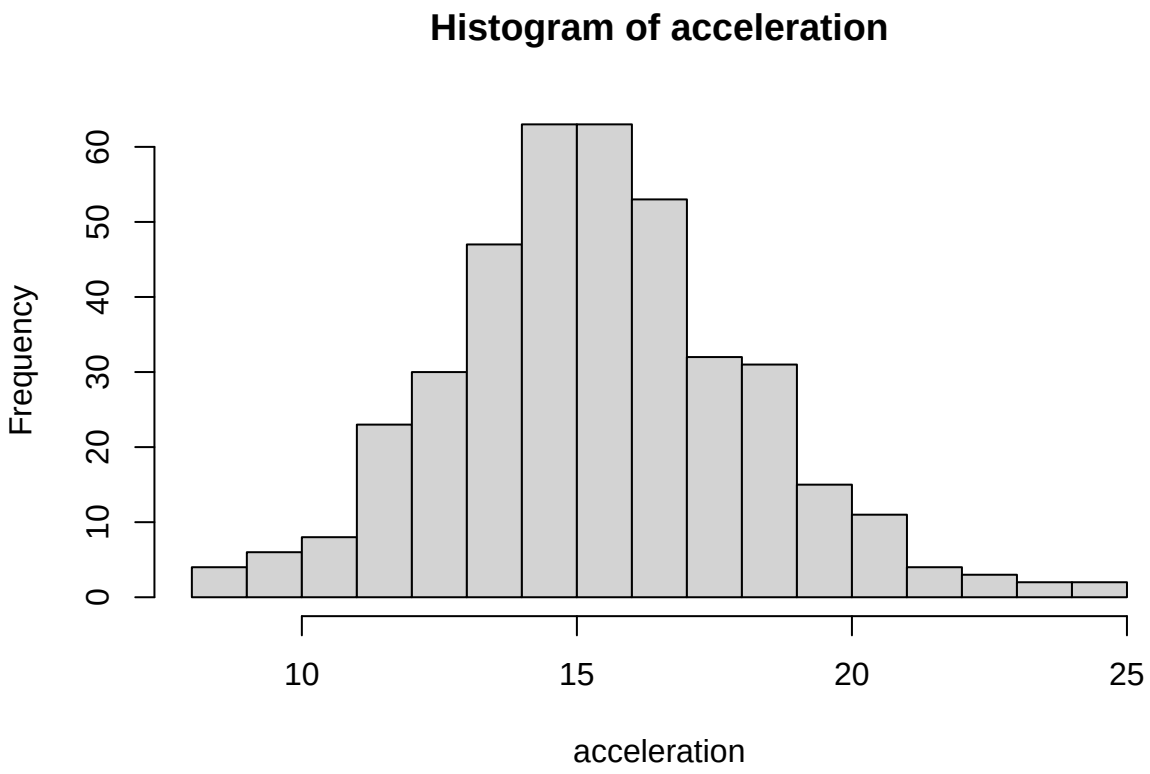

```
for (var in names(auto_to_plot)){
  hist(auto_to_plot[,var], xlab = var, main = paste0('Histogram of ', var), breaks = 15)
}
```


Histogram of mpg
Frequency
0
10
20
30
40
10
20
30
40
mpg

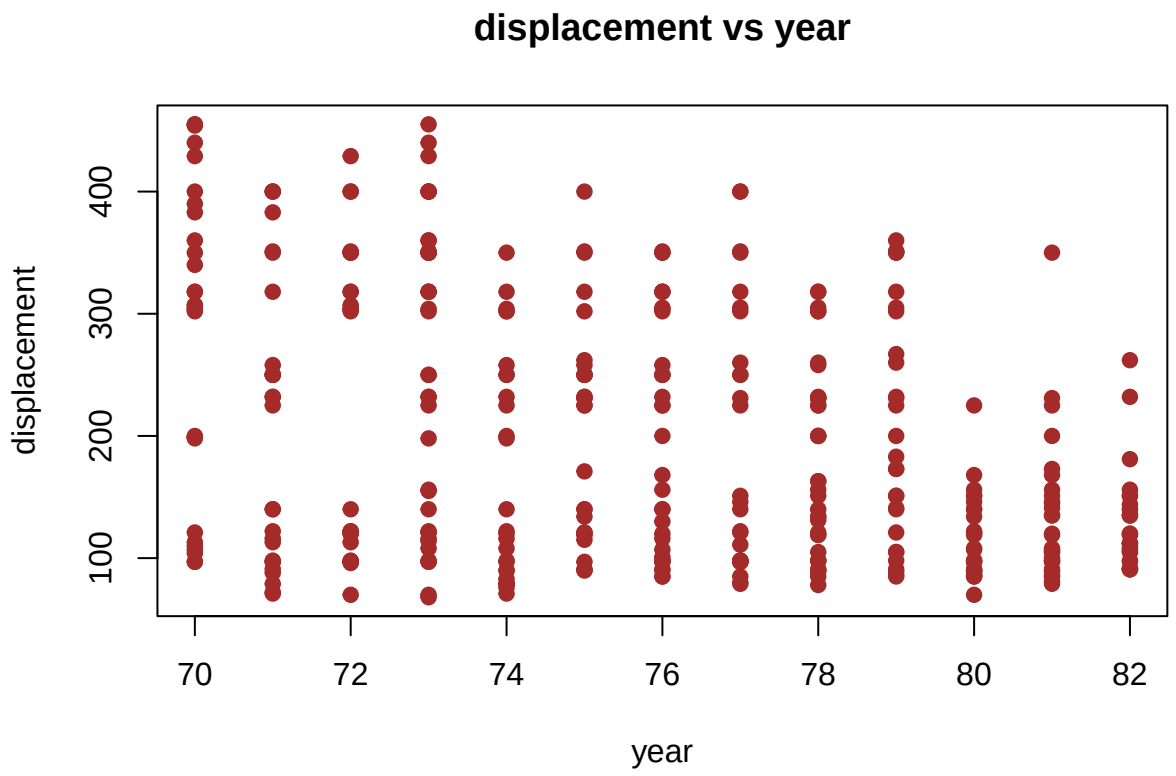

Histogram of displacement
Frequency
0
20
40
60
80
100
200
300
400
displacement


Histogram of horsepower
Frequency
0
10
20
30
40
50
60
70
50
100
150
200
horsepower

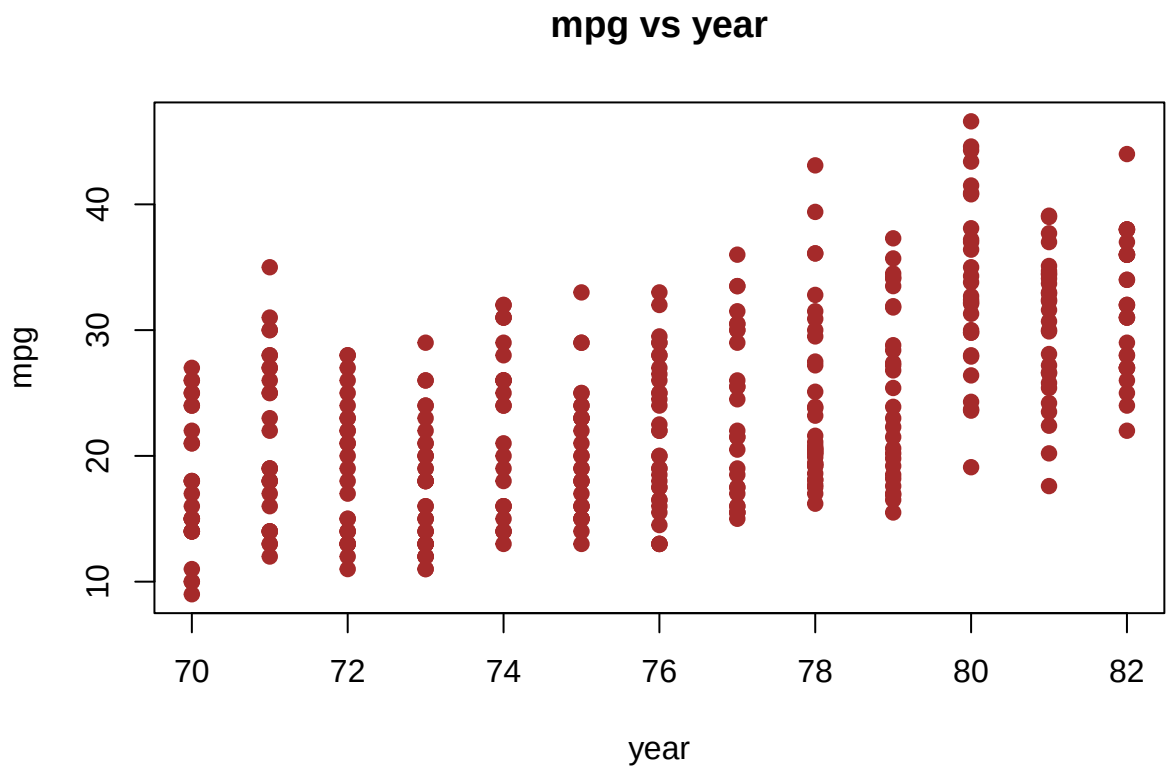

Histogram of weight
Frequency
0
10
20
30
40
1500
2000
2500
3000
3500
4000
4500
5000
weight



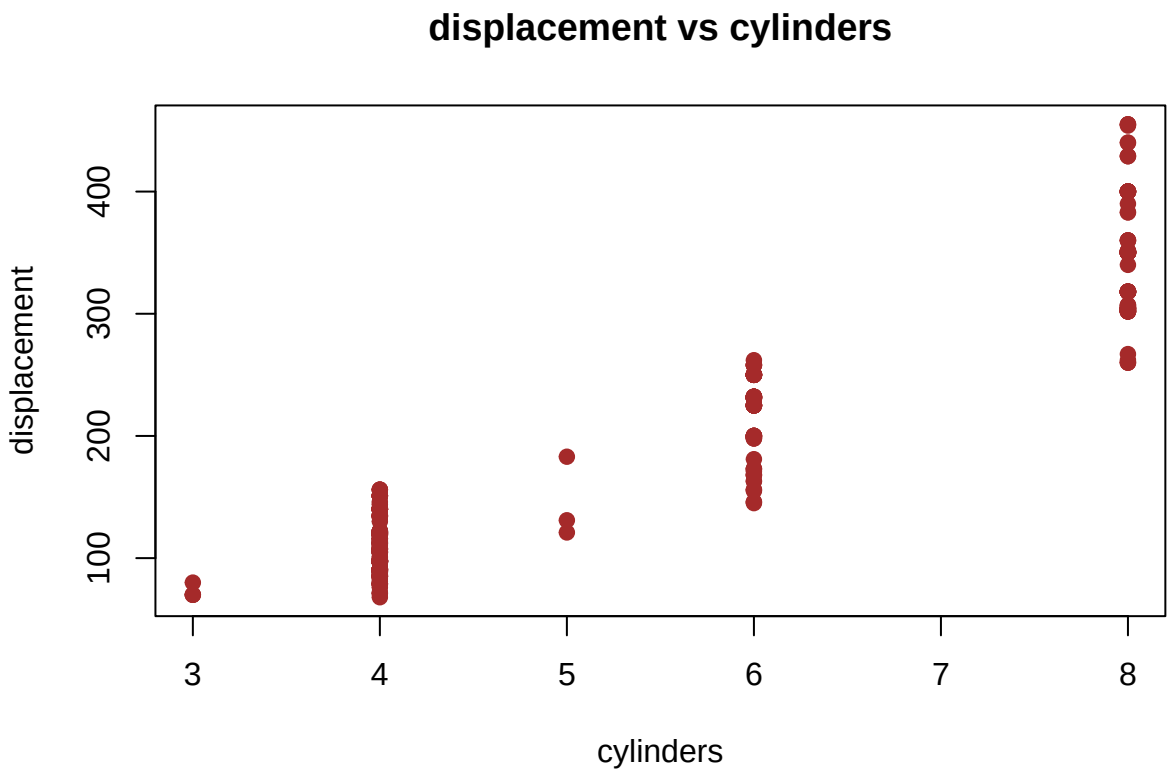

```
plot (x = auto$year, y = auto$displacement, col = 'brown', xlab = 'year', ylab = 'displacement',
      main = 'displacement vs year', pch = 19)
```



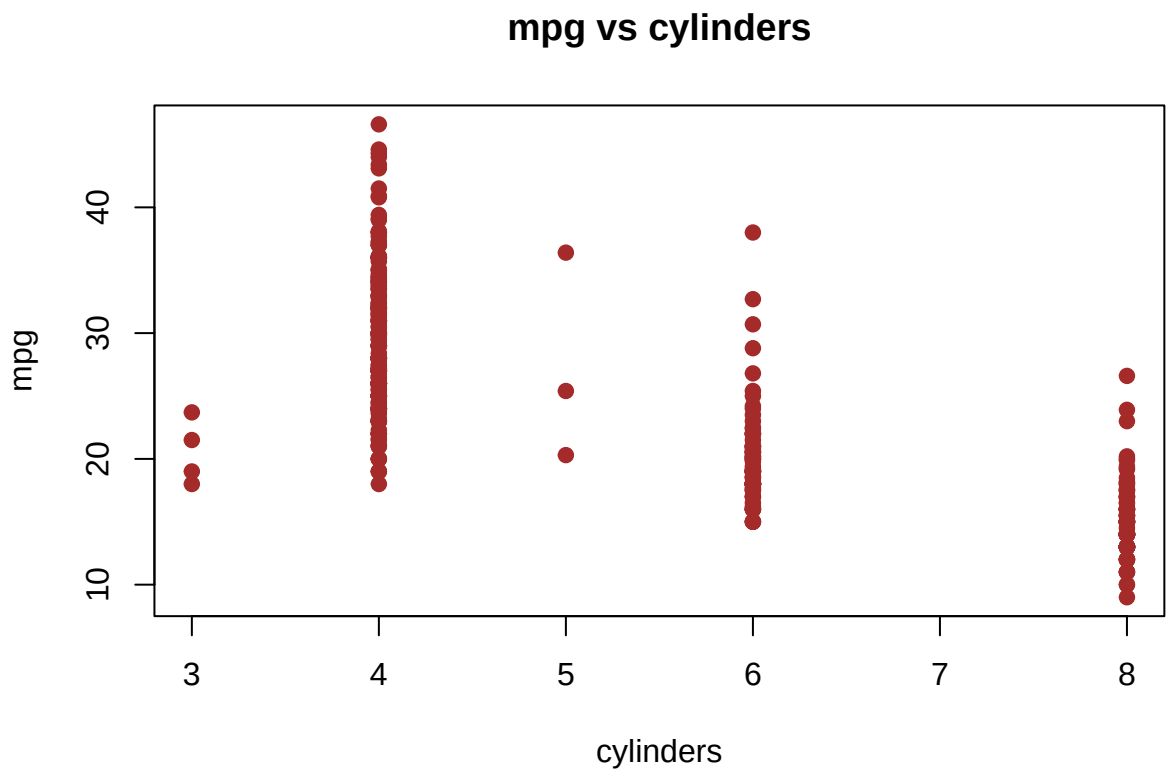

```
plot (x = auto$year, y = auto$mpg, col = 'brown', xlab = 'year', ylab = 'mpg',
      main = 'mpg vs year', pch = 19)
```




```
plot (x = auto$cylinders, y = auto$displacement, col = 'brown', xlab = 'cylinders',
      ylab = 'displacement', main = 'displacement vs cylinders', pch = 19)
```




```
plot (x = auto$cylinders, y = auto$mpg, col = 'brown', xlab = 'cylinders', ylab = 'mpg',
      main = 'mpg vs cylinders', pch = 19)
```



Se realizan pares de scatterplots de las variables de mayor cardinalidad, clasificándolas con un color distinto según la variable categórica "origin". Es posible ver como varias de las variables presentan una clara relación entre ellas, entre las que se destacan:

- "mpg" en función de "displacement", "horsepower" y "weight": en estos tres casos, la variable "mpg" presenta una relación inversamente proporcional a las mencionadas. Esto tiene sentido, debido a que un vehículo con mayor tamaño de motor, potencia y peso tendrá razonablemente un menor rendimiento, es decir, recorrerá menos millas por galón de combustible. A su vez, la relación entre la aceleración del auto y su rendimiento parece tener cierto grado de proporcionalidad, aunque la relación es menos marcada. Esto también se aprecia en la matriz de correlación, teniendo valores aproximados a -0.80 para las primeras variables y +0.42 para el caso de aceleración
- Esta relación entre variables también es apreciable en los gráficos de "displacement", "horsepower" y weight. Observando el centro de la imagen, estos gráficos presentan una clara relación lineal positiva, lo que se evidencia aún más en la matriz de correlación, la cual presenta valores en el entorno de +0.90 para estas variables. Nuevamente, esto tiene sentido, ya que autos más pesados generalmente cuentan con motores más grandes (displacement) y mayor potencia. Asimismo, estas tres variables presentan una relación de proporcionalidad inversa con la variable de aceleración, pero nuevamente la misma es menos marcada que con las restantes. Esto también tiene sentido, ya que tiende a ser más difícil que autos más grandes y pesados alcancen altas velocidades en poco tiempo. Esto es coherente con la relación entre aceleración y mpg, ya que, como se mencionó antes, un alto mpg está relacionado con autos más livianos y de menor potencia, lo cuál lleva a que tengan mayores valores de aceleración. Además, en ambos casos la relación entre aceleración y el resto de las variables es débil.
- Dado que las variables "displacement", "horsepower" y "weight" presentan el mismo comportamiento entre ellas, se grafica únicamente una de ellas (displacement), así como la variable mpg, en función de las variables no utilizadas en los gráficos de pares (year y cylinders). Es posible ver como el rendimiento medio de los vehículos (mpg) aumenta con el año de fabricación, y disminuye con la cantidad de cilindros,

lo que es razonable ya que se espera que autos más modernos puedan tener una mayor eficiencia, así como se espera que autos con motores más grandes y de mayor cilindrada consuman más combustible. Asimismo, también es posible ver como el tamaño del motor (displacement) aumenta con la cantidad de cilindros del mismo, lo que también resulta intuitivo.

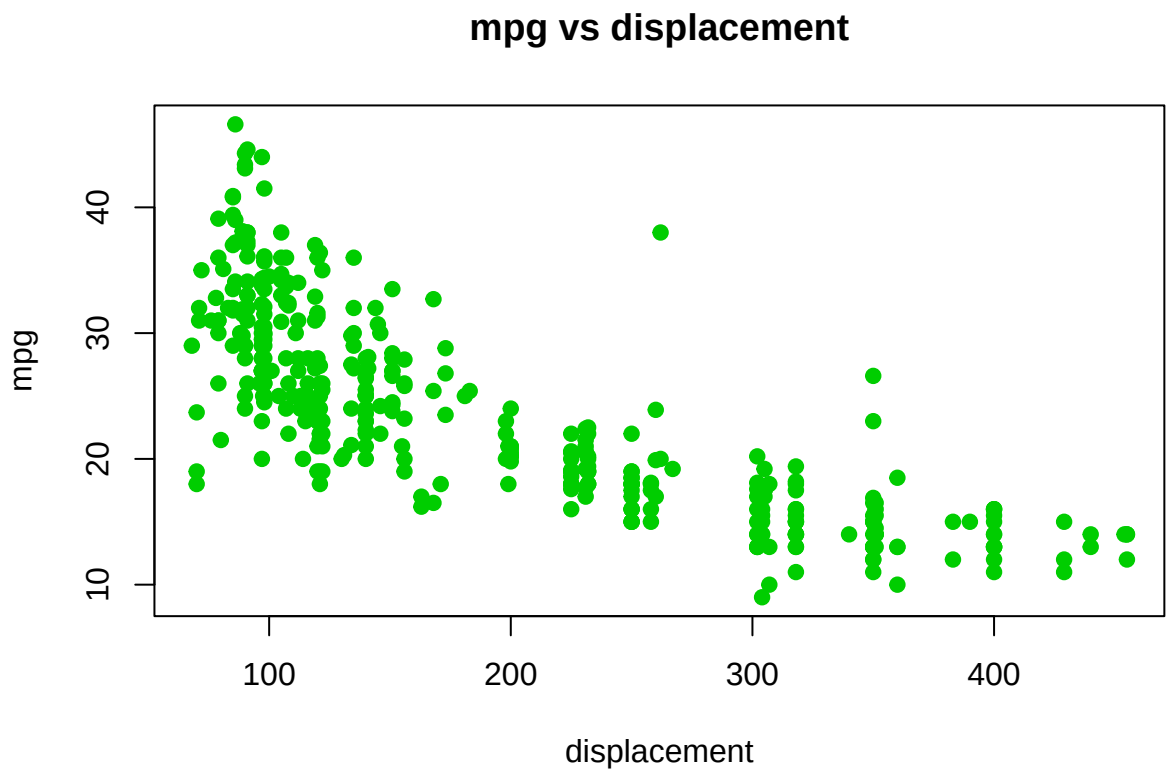
- Por último, se realizan histogramas de las variables numéricas. Es posible ver como las variables mpg, weight y horsepower presentan un gran nivel de dispersión, con observaciones en todo el rango de cada variable pero con mayor cantidad de observaciones en torno al extremo inferior de la misma. La variable displacement también presenta un comportamiento similar, pero con valores dentro del rango para los cuales no existen observaciones. Luego, respecto a la variable de aceleración, el histograma marca una distribución muy aproximada a una distribución normal.

f. **Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.**

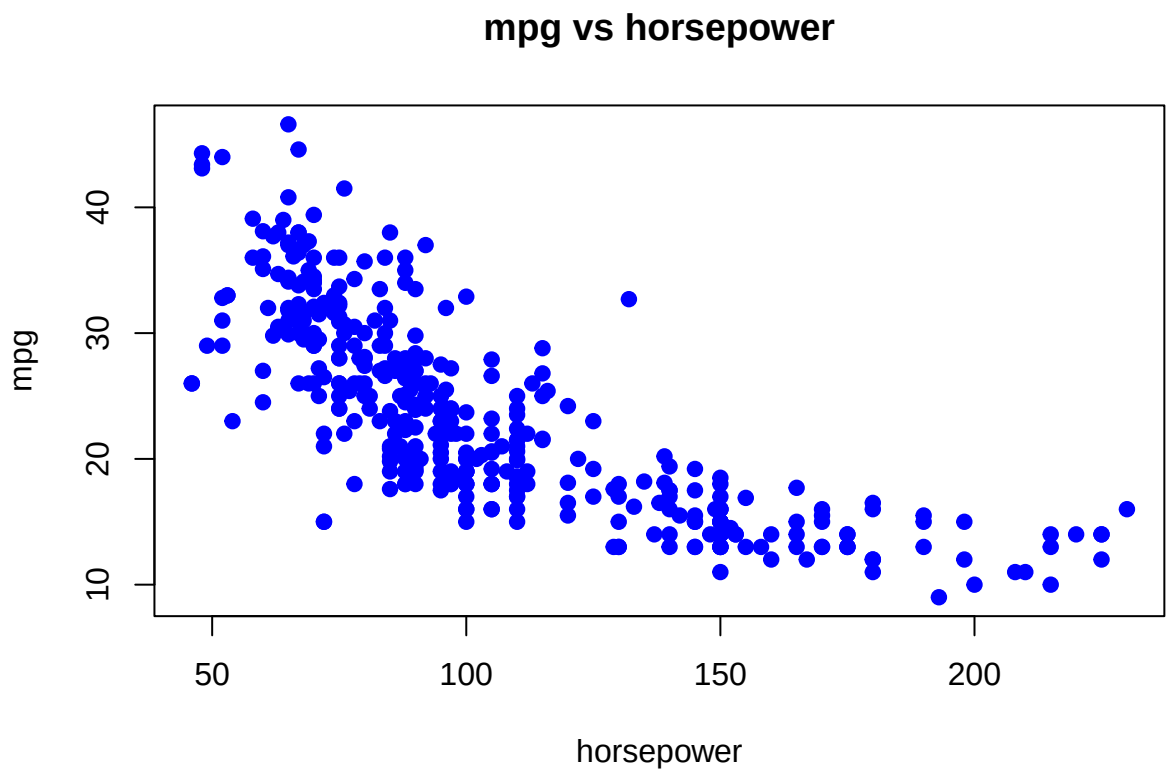
```
# Matriz de correlación
cor_mat = cor(na.omit(auto_to_plot))
cor_mat[upper.tri(cor_mat)] = NA
print (cor_mat, na.print = '')
```

```
##                     mpg displacement horsepower     weight acceleration
## mpg           1.0000000
## displacement -0.8051269    1.0000000
## horsepower   -0.7784268    0.8972570  1.0000000
## weight       -0.8322442    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285   -0.5438005 -0.6891955 -0.4168392            1
```

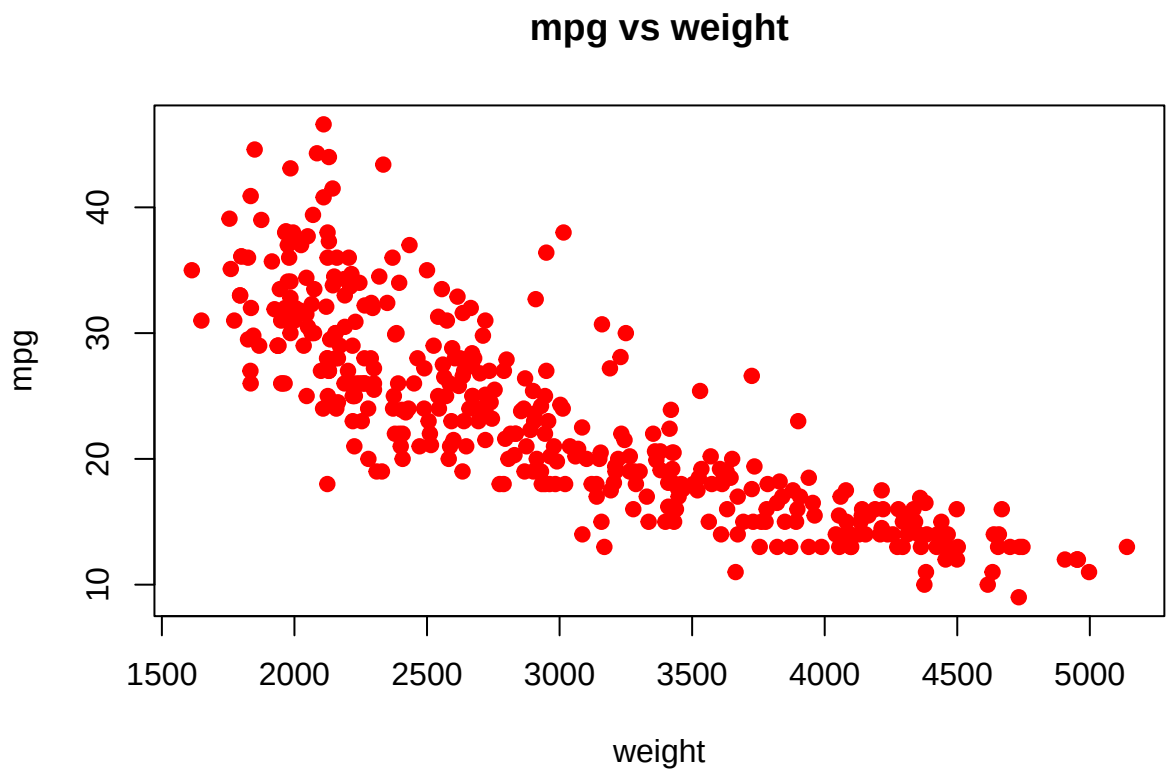
```
# Scatterplots de mpg en función de variables restantes
plot (x = auto$displacement, y = auto$mpg, col = 'green3', xlab = 'displacement', ylab = 'mpg',
      main = 'mpg vs displacement', pch = 19)
```



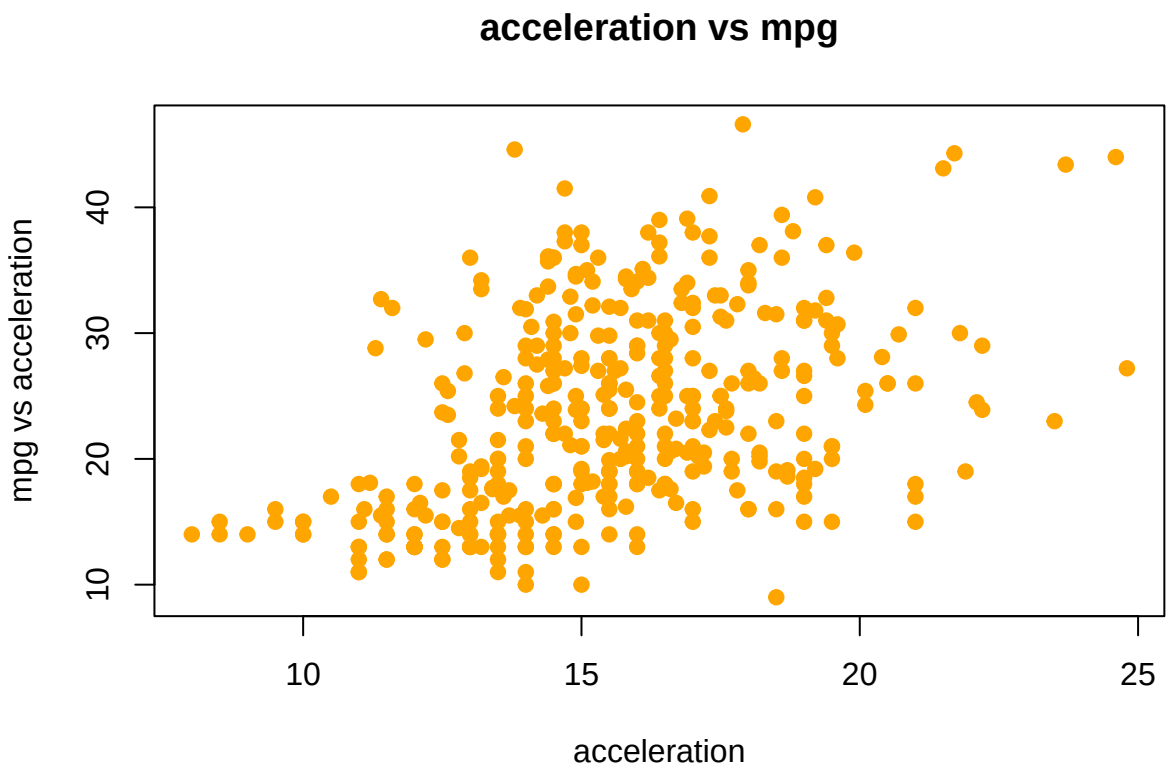

```
plot (x = auto$horsepower, y = auto$mpg, col = 'blue', xlab = 'horsepower', ylab = 'mpg',
      main = 'mpg vs horsepower', pch = 19)
```



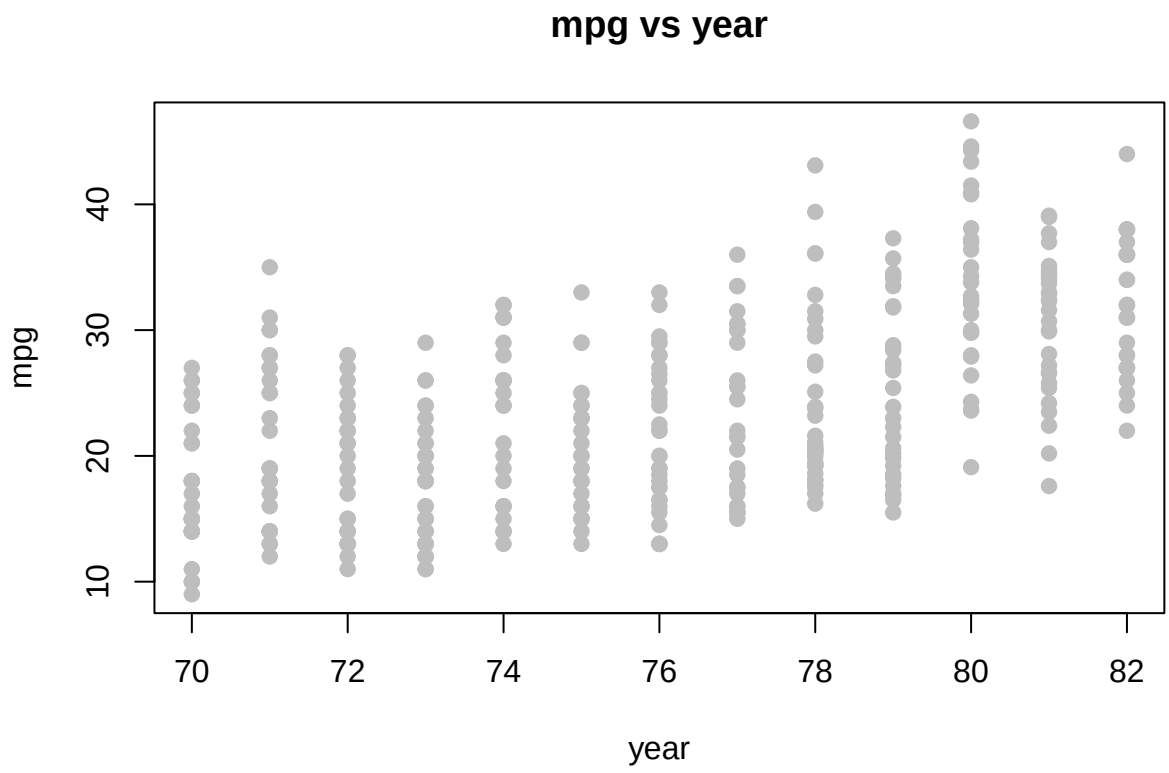

```
plot (x = auto$weight, y = auto$mpg, col = 'red', xlab = 'weight', ylab = 'mpg',
      main = 'mpg vs weight', pch = 19)
```



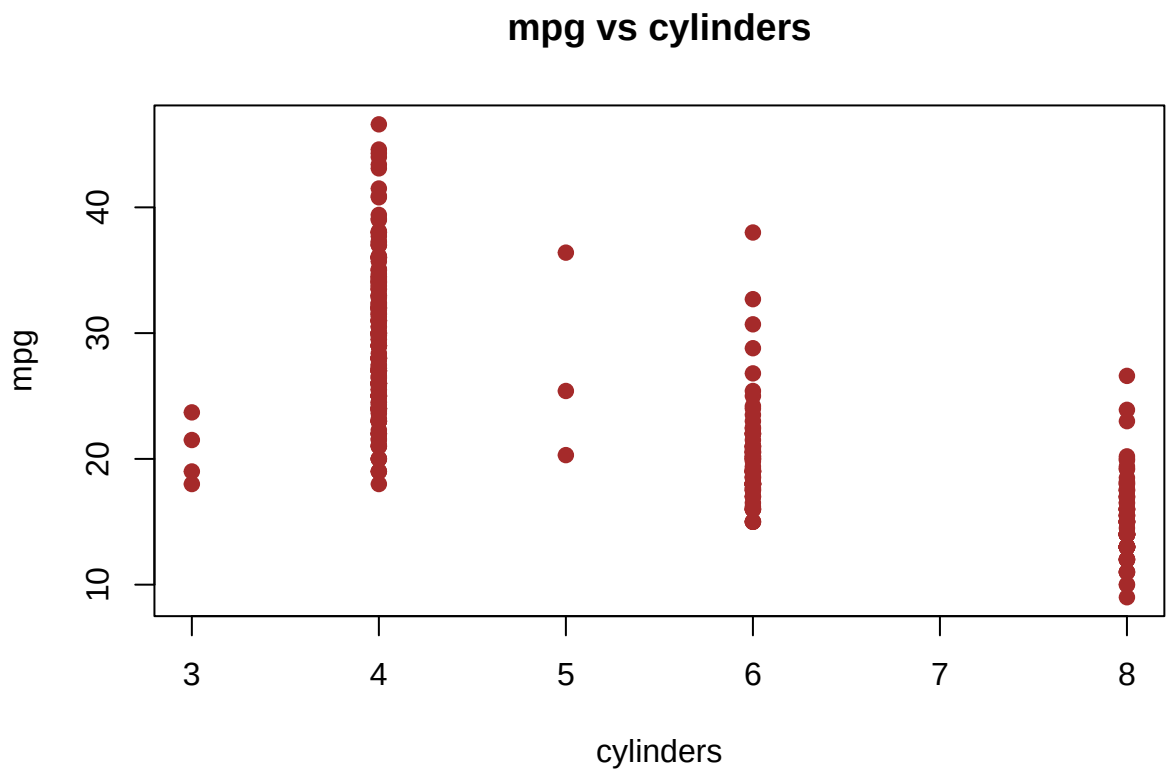

```
plot (x = auto$acceleration, y = auto$mpg, col = 'orange', xlab = 'acceleration',
      ylab = 'mpg vs acceleration', main = 'acceleration vs mpg', pch = 19)
```




```
plot (x = auto$year, y = auto$mpg, col = 'gray', xlab = 'year', ylab = 'mpg', main = 'mpg vs year',
      pch = 19)
```




```
plot (x = auto$cylinders, y = auto$mpg, col = 'brown', xlab = 'cylinders', ylab = 'mpg',
      main = 'mpg vs cylinders', pch = 19)
```



Se realizan scatterplots de todas las variables predictoras en función de la variable objetivo “mpg”. Como se mencionó anteriormente, las variables “displacement”, “horsepower” y “weight” son las que tienen una mayor correlación con la variable objetivo, presentando coeficientes de correlación en el entorno de -0.80, por lo que se deduce que son las que tienen mayor valor predictivo. Cabe destacar que la relación parece ser más débil para valores más altos de la variable objetivo, por lo que es esperable que las predicciones tengan mayor nivel de error en el entorno de estos valores. Asimismo, los gráficos de las variables “cylinders” e “year” indican una tendencia decreciente y creciente respectivamente con la variable objetivo, aunque los rangos para cada valor de dicha variable son bastante amplios. Se espera que estas variables tengan un poder predictivo significativamente menor que las ya mencionadas. Por último, se espera que la variable “acceleration” tenga un poder predictivo despreciable en comparación con las ya mencionadas.

Ejercicios de Práctico - Clasificación Binaria

Diego Velasco

03/04/2023

Ejercicio 1

From the Bayes Classifier, predict the class for each test data and compute the error.

Training sample

x_1	a	a	b	a	a	b	b	b
x_2	b	a	a	a	a	b	b	b
Y	1	1	1	1	-1	-1	-1	-1

Test sample

x_1	a	a	b	b
x_2	a	b	a	b
Y_{pred}	?	?	?	?
Y_{real}	1	-1	1	1

Respuesta

Versión manual: El clasificador de Bayes es aquel que maximiza las probabilidades a posteriori. En este caso, dado que la variable binaria objetivo puede tomar valores de 1 y -1, una observación será clasificada como 1 si se cumple:

$$P(Y = 1|X = X) > P(Y = -1|X = X)$$

Utilizando la fórmula de Bayes, cada término de esta inecuación puede escribirse de la siguiente manera:

$$P(Y = 1|X = X) = \frac{P(Y = 1)P(X = X|Y = 1)}{P(X = X)}$$

$$P(Y = -1|X = X) = \frac{P(Y = -1)P(X = X|Y = -1)}{P(X = X)}$$

De manera que, simplificando el denominador, la inecuación planteada se expresa de la siguiente manera:

$$P(X = X|Y = 1) \times P(Y = 1) > P(X = X|Y = -1) \times P(Y = -1)$$

Dado que se tienen igual cantidad de datos de cada categoría en el dataset de entrenamiento, $P(Y = 1) = P(Y = -1) = 0.5$, de manera que una observación será clasificada como 1 si se cumple la siguiente desigualdad:

$$P(X = X|Y = 1) > P(X = X|Y = -1)$$

Para hallar estas probabilidades, se utiliza el estimador de Naive Bayes, el cual asume distribución normal e independencia condicional de las variables univariantes x_1, x_2, de manera que:

$$P(X = (x_1, x_2)|Y = 1) = P(X_1 = x_1|Y = 1) \times P(X_2 = x_2|Y = 1)$$

$$P(X = (x_1, x_2)|Y = -1) = P(X_1 = x_1|Y = -1) \times P(X_2 = x_2|Y = -1)$$

Las probabilidades condicionales de cada variable univariante x_1 y x_2 son calculadas como la cantidad de observaciones de cada una de ellas que son iguales al valor correspondiente a ó b para cada subconjunto del dataset de entrenamiento $Y = 1$ e $Y = -1$. Estos cálculos se resumen en la siguiente tabla.

x_1	x_2	$P(X = (x_1, x_2)\|Y = 1)$	$P(X = (x_1, x_2)\|Y = -1)$	Y_{pred}
a	a	$0.75 \times 0.75 = 0.5625$	$0.25 \times 0.25 = 0.0625$	1
a	b	$0.75 \times 0.25 = 0.1875$	$0.25 \times 0.75 = 0.1875$	$1^{(*)}$
b	a	$0.25 \times 0.75 = 0.1875$	$0.75 \times 0.25 = 0.1875$	$-1^{(*)}$
b	b	$0.25 \times 0.25 = 0.0625$	$0.75 \times 0.75 = 0.5625$	-1

(*) Dado que las probabilidades de las observaciones 2 y 3 son iguales, se clasifica de manera arbitraria.

```
library(e1071)

# Crear dataset de entrenamiento
x1_train = c('a','a','b','a','a','b','b','b')
x2_train = c('b','a','a','a','a','b','b','b')
y_train = c(1,1,1,1,-1,-1,-1,-1)

data_train = data.frame (x1 = x1_train,  x2 = x2_train, y = y_train)
data_train
```

Versión en R con paquete e1071:

```
##   x1 x2  y
## 1  a  b  1
## 2  a  a  1
## 3  b  a  1
## 4  a  a  1
## 5  a  a -1
## 6  b  b -1
## 7  b  b -1
## 8  b  b -1
```

```
# Ajuste de modelo
nb = naiveBayes(y~x1+x2, data = data_train)

# Crear dataset de testeo
x1_test = c('a','a','b','b')
x2_test = c('a','b','a','b')

data_test = data.frame(x1 = x1_test, x2 = x2_test)
```

```
data_test
```

```
##   x1 x2
## 1  a  a
## 2  a  b
## 3  b  a
## 4  b  b
```

```
# Predicciones
y_test_prob = predict(nb, newdata = data_test, type = 'raw')
y_test = predict(nb, newdata = data_test)

y_test
```

```
## [1] 1  -1 -1 -1
## Levels: -1 1
```

```
y_test_prob
```

```
##       -1   1
## [1,] 0.1 0.9
## [2,] 0.5 0.5
## [3,] 0.5 0.5
## [4,] 0.9 0.1
```

Es posible apreciar que las predicciones obtenidas son las mismas para la primera y última observación. Asimismo, también es posible ver que las probabilidades a posteriori son 0.5 para cada categoría en el caso de las observaciones 2 y 3, lo que coincide con la resolución manual. Por último, si bien no es posible comparar directamente las probabiliades a posteriori para los casos 1 y 4 con las calculadas manualmente debido a que en el desarrollo manual se realizó una simplificación por $P(X = X)$, sí es posible comparar el ratio entre $P(X = X|Y = 1)$ y $P(X = X|Y = -1)$. Para el caso manual, este coeficiente vale $\frac{0.5625}{0.0625} = 9$ para la observación 1 y $\frac{0.0625}{0.5625} = \frac{1}{9}$ para la observación 4. Para el caso en R, este coeficiente vale $\frac{0.9}{0.1} = 9$ para la observación 1 y $\frac{0.1}{0.9} = \frac{1}{9}$ para la observación, por lo que se concluye que se obtiene los mismos resultados.

Ejercicio 2

Suppose that $\pi_1 = \pi_0 = 0.5$ and the densities are $g_1 = N(0, 1)$ and $g_0 = 0.7N(-3, 1) + 0.3N(1, 2)$

a. Assuming equal cost find:

   1. Plot the densities and write the Bayes rule for this classification task.
   2. Write the Bayes decision boundary and find its solutions.

b. Assume that C(1; 0) = 2 and C(0; 1) = 6. Repeat questions above.

   Nota: Se condidera $g_0(x) = 0.7N(-3, 1) + 0.3N(1, 2)$ según nota en sitio eva.

Respuesta Siendo:

- $C(1, 0)$ el costo de clasificar erróneamente una observación $Y = 1$, como $Y = 0$
- $C(0, 1)$ el costo de clasificar erróneamente una observación $Y = 0$, como $Y = 1$

Se define el riesgo de clasificar erróneamente una observación $Y \neq j$ como $Y = j$ como:

$R(Y = j|X = x) = \sum_i P(Y = i|X = x).C(i, j)$ con $i \neq j$ ya que se considera que $C(i, i) = 0 \rightarrow$ en un problema de clasificación binaria con categorías 0 y 1, se tiene que:

$$R(Y=1|X=x) = P(Y=0|X=x) \times C(0,1)$$

$$R(Y=0|X=x) = P(Y=1|X=x) \times C(1,0)$$

Luego, una observación se clasifica con $Y=1$ si se cumple que el riesgo de clasificarla erróneamente como $Y=1$ es menor que el de clasificarla como $Y=0$, es decir:

$$R(Y=1|X=x) < R(Y=0|X=x) \leftrightarrow P(Y=0|X=x) \times C(0,1) < P(Y=1|X=x)] \times C(1,0)$$

$$\leftrightarrow \frac{P(Y=1|X=x)}{P(Y=0|X=x)} > \frac{C(0,1)}{C(1,0)}$$

Utilizando la fórmula de Bayes para cada término del lado izquierdo de la inecuación planteada, teniendo en cuenta que $P(Y=1) = \pi_1 = P(Y=0) = \pi_0 = 0.5$, simplificando por $P(X=x)$ en el numerador y denominador, y aplicando el teorema del valor medio para el cálculo de la probabilidad de una variable continua, se tiene que:

$$\frac{P(Y=1|X=x)}{P(Y=0|X=x)} = \frac{P(X=x|Y=1)}{P(X=x|Y=0)} = \frac{g_1(x)\Delta x}{g_0(x)\Delta x} = \frac{g_1(x)}{g_0(x)}$$

De manera que una observación se clasificará como $Y=1$ si se cumple la siguiente relación:

$$\frac{g_1(x)}{g_0(x)} > \frac{C(0,1)}{C(1,0)}$$

Asimismo, se utiliza la fórmula de la densidad de la distribución normal univariada, presentada a continuación:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

De manera que la expresión para $g_1(x)$ queda de la siguiente manera:

$$g_1(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Análogamente, la expresión para $g_0(x)$ es:

$$g_0(x) = \frac{0.7}{\sqrt{2\pi}} e^{-\frac{1}{2}(x+3)^2} + \frac{0.3}{\sqrt{2\pi}\sqrt{2}} e^{-\frac{1}{2}\frac{(x-1)^2}{(\sqrt{2})^2}} g_0(x) = \frac{1}{\sqrt{2\pi}} \left[0.7e^{-\frac{1}{2}(x+3)^2} + \frac{0.3}{\sqrt{2}} e^{-\frac{1}{4}(x-1)^2} \right]$$

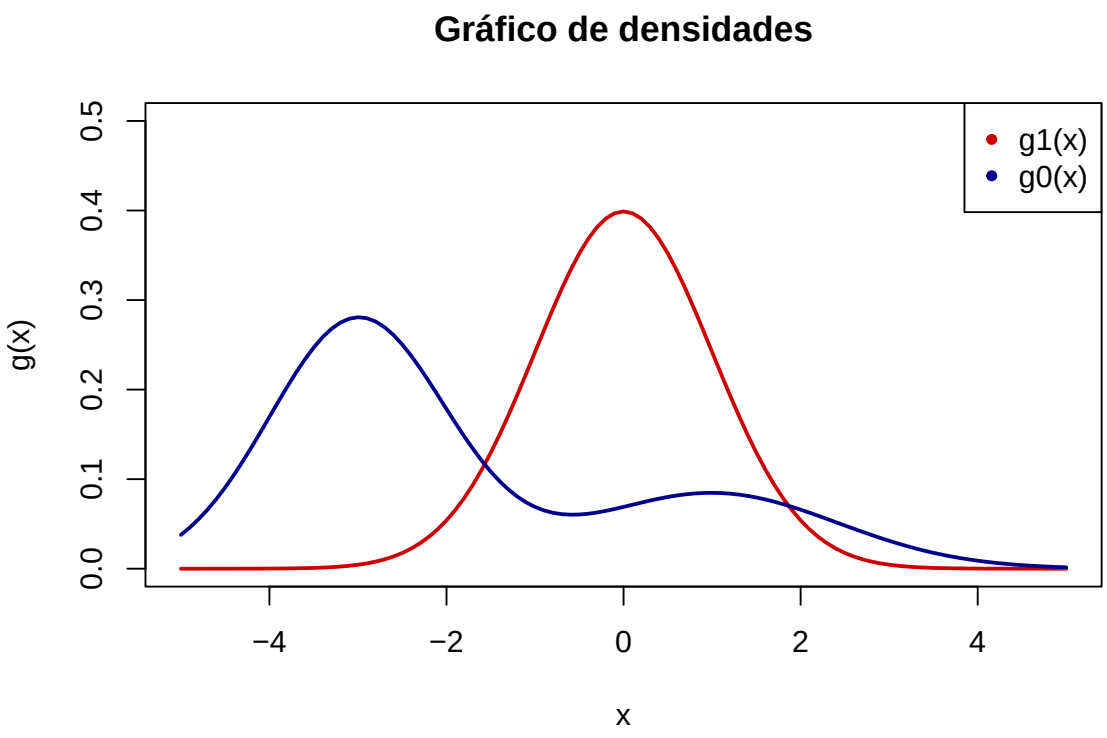
De manera que la expresión de $\frac{g_1(x)}{g_0(x)}$ será:

$$\frac{g_1(x)}{g_0(x)} = \frac{e^{-\frac{x^2}{2}}}{0.7e^{-\frac{1}{2}(x+3)^2} + \frac{0.3}{\sqrt{2}} e^{-\frac{1}{4}(x-1)^2}}$$

a. **Asumiendo costos iguales:**

   1. **Plot the densities and write the Bayes rule for this classification task.**

```r
# graficar g1(x)
mu1 = 0 # media de g1(x)
sd1 = 1 # desviación estándar (raiz de la varianza) de g1(x)

# graficar g0(x)
mu01 = -3 # media de distribución N(-3,1)
mu02 = 1 # media de distribución N(1,2)

sd01 = 1 # desviación estándar (raiz de la varianza) de distribución N(-3,1)
sd02 = sqrt(2) # desviación estándar (raiz de la varianza) de distribución N(1,2)

N01 = function(x) dnorm(x, mean = mu01, sd = sd01)
N02 = function(x) dnorm(x, mean = mu02, sd = sd02)
g0 = function(x) 0.7*N01(x) + 0.3*N02(x)

curve(dnorm(x, mean = mu1, sd = sd1), xlim = c(-5,5), xlab = 'x', ylab = 'g(x)',
      col = 'red3', lwd = 2, ylim = c(0,0.5), main = 'Gráfico de densidades')

curve(g0, xlim = c(-5,5), xlab = 'x', ylab = 'g(x)', col = 'blue4', lwd = 2,
      add = TRUE)

legend('topright', legend = c('g1(x)', 'g0(x)'), col = c('red3', 'blue4'), pch = 20)
```



Dado que los costos de clasificar erróneamente son iguales, es decir, $C(0,1) = C(1,0)$, se tiene que la regla de Bayes para el problema de clasificación en cuestión es:

$$Y_{pred} = 1 si F(x) = \frac{g_1(x)}{g_0(x)} - 1 = \frac{e^{-\frac{x^2}{2}}}{0.7e^{-\frac{1}{2}(x+3)^2} + \frac{0.3}{\sqrt{2}}e^{-\frac{1}{4}(x-1)^2}} - 1 > 0$$

$$Y_{pred} = 0 si F(x) = \frac{g_1(x)}{g_0(x)} - 1 = \frac{e^{-\frac{x^2}{2}}}{0.7e^{-\frac{1}{2}(x+3)^2} + \frac{0.3}{\sqrt{2}}e^{-\frac{1}{4}(x-1)^2}} - 1 < 0$$

2. **Write the Bayes decision boundary and find its solutions**.

   La frontera de Bayes para el problema en cuestión se da cuando la función $F(x)$ se anula, es decir:

$$F(x) = \frac{e^{-\frac{x^2}{2}}}{0.7e^{-\frac{1}{2}(x+3)^2} + \frac{0.3}{\sqrt{2}}e^{-\frac{1}{4}(x-1)^2}} - 1 = 0$$

   En la celda de código siguiente se halla la frontera de Bayes calculando la función $F(x)$ directamente como la división $\frac{g_1(x)}{g_0(x)} - 1$, ya que se definen $g_1(x)$ y $g_0(x)$ previamente. Sin embargo, la función obtenida es idéntica a la función recién planteada

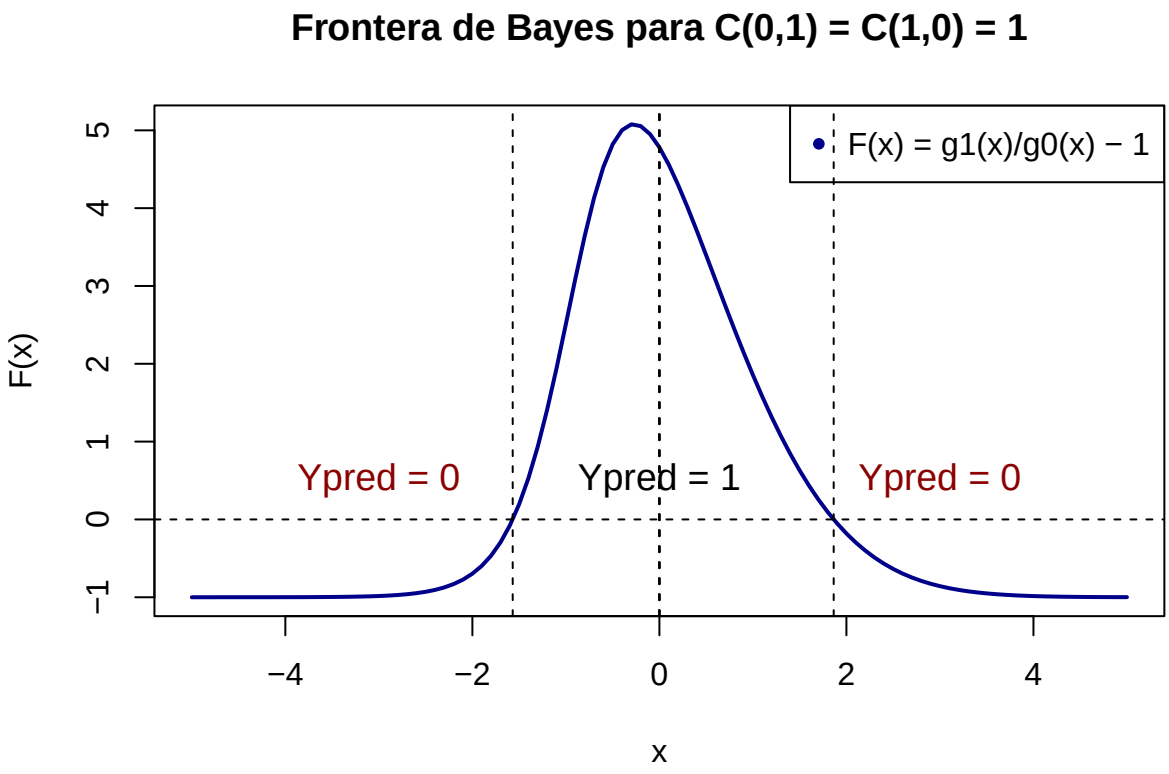
```
# Definición de funciones
g1 = function(x) dnorm(x, mean = mu1, sd = sd1)

f = function(x) g1(x)/g0(x) - 1

# Cálculo de raíces
root1 = uniroot(f,c(-5,0))
root2 = uniroot(f,c(0,5))
cat ('Primera raiz de f(x):',as.numeric(root1[1]),'\n')
```

```
## Primera raiz de f(x): -1.56881
```

```
cat ('Segunda raiz de f(x):',as.numeric(root2[1]))
```

```
## Segunda raiz de f(x): 1.863904
```

```
# Gráficos
curve(f, xlim = c(-5,5), xlab = 'x', ylab = 'F(x)', col = 'blue4', lwd = 2,
      main = 'Frontera de Bayes para C(0,1) = C(1,0) = 1')
abline(h = 0, col = 'black', lwd = 1, lty = 2)
abline(v = root1, col = 'black', lwd = 1, lty = 2)
abline(v = root2, col = 'black', lwd = 1, lty = 2)
text(x = 3, y = 0.5, labels = "Ypred = 0", col = 'red4', cex = 1.2)
text(x = 0, y = 0.5, labels = "Ypred = 1", col = 'black', cex = 1.2)
text(x = -3, y = 0.5, labels = "Ypred = 0", col = 'red4', cex = 1.2)

legend('topright', legend = 'F(x) = g1(x)/g0(x) - 1', col = 'blue4', pch = 20)
```



b. **Assume that C(1; 0) = 2 and C(0; 1) = 6. Repeat questions above**

Los gráficos de las densidades son idénticos a los generados en la parte anterior. Sin embargo, la regla de Bayes cambia para este caso. Como se mostró previamente, una clasificación será clasificada como $Y = 1$ si:

$$\frac{g_1(x)}{g_0(x)} > \frac{C(0,1)}{C(1,0)}$$

Dado que $C(0,1) = 6$ y $C(1,0) = 2$, se tiene que $\frac{C(0,1)}{C(1,0)} = 3$, de manera que la regla de Bayes para este caso sera:

$$Y_{pred} = 1 si F(x) = \frac{g_1(x)}{g_0(x)} - 3 = \frac{e^{-\frac{x^2}{2}}}{0.7e^{-\frac{1}{2}(x+3)^2} + \frac{0.3}{\sqrt{2}}e^{-\frac{1}{4}(x-1)^2}} - 3 > 0$$

$$Y_{pred} = 0 si F(x) = \frac{g_1(x)}{g_0(x)} - 3 = \frac{e^{-\frac{x^2}{2}}}{0.7e^{-\frac{1}{2}(x+3)^2} + \frac{0.3}{\sqrt{2}}e^{-\frac{1}{4}(x-1)^2}} - 3 < 0$$

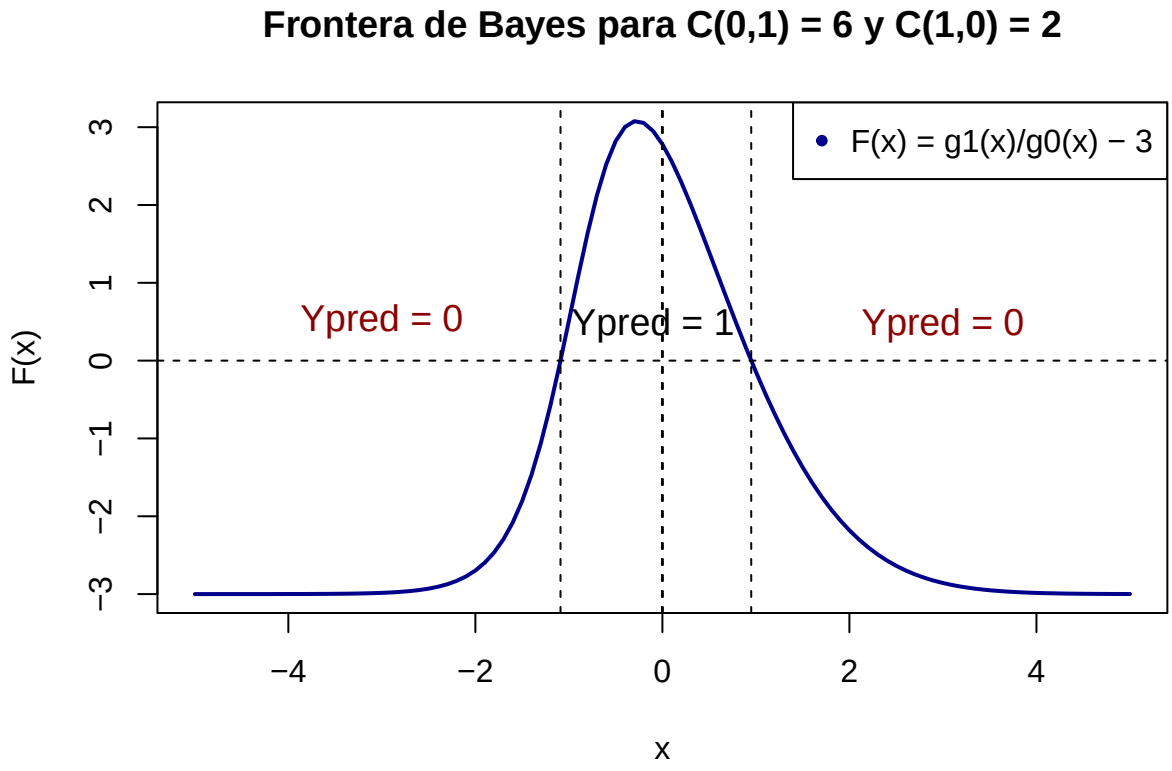
```
# Definición de funciones
f3 = function(x) g1(x)/g0(x) - 3

# Cálculo de raíces
root1 = uniroot(f3,c(-5,0))
root2 = uniroot(f3,c(0,5))
cat ('Primera raiz de f(x):',as.numeric(root1[1]),'\n')
```

```
## Primera raiz de f(x): -1.089666
```

```r
cat ('Segunda raiz de f(x):',as.numeric(root2[1]))
```

```
## Segunda raiz de f(x): 0.9499575
```

```r
# Gráficos
curve(f3, xlim = c(-5,5), xlab = 'x', ylab = 'F(x)', col = 'blue4', lwd = 2,
      main = 'Frontera de Bayes para C(0,1) = 6 y C(1,0) = 2')
abline(h = 0, col = 'black', lwd = 1, lty = 2)
abline(v = root1, col = 'black', lwd = 1, lty = 2)
abline(v = root2, col = 'black', lwd = 1, lty = 2)
text(x = 3, y = 0.45, labels = "Ypred = 0", col = 'red4', cex = 1.2)
text(x = -0.1, y = 0.45, labels = "Ypred = 1", col = 'black', cex = 1.2)
text(x = -3, y = 0.5, labels = "Ypred = 0", col = 'red4', cex = 1.2)

legend('topright', legend = 'F(x) = g1(x)/g0(x) - 3', col = 'blue4', pch = 20)
```



Es posible apreciar como, al aumentar el costo de clasificar erróneamente una observación $Y = 0$ respecto al costo de clasificar erróneamente una observación $Y = 1$, la zonas en la cuales la predicción será $Y = 0$ aumentan en tamaño, mientras que la zona en la que la predicción será $Y = 1$ disminuye en tamaño. Esto tiene sentido, ya que al aumentar el tamaño de la zona en la que la predicción será $Y = 0$, más observaciones serán clasificadas en esta clase, lo que reduce la probabilidad de clasificar erróneamente una observación de esta clase, mitigando así el costo mayor.

Ejercicio 3

Generate 100 observations from a bivariate Gaussian distribution $N(\mu_1, \Sigma_1)$ with $\mu_1 = (3, 1)'$ and $\Sigma_1 = I$ (identity matrix) and label them as 1. Generate another 100 observations from a bivariate Gaussian distribution $N(\mu_2, \Sigma_2)$ $\mu_2 = (1, 3)'$ and $\Sigma_2 = I$ and label them as 0. Together, these 200 observations constitute the training set.

a. Write an R code to generate this data set.

b. Plot this data using different colors for the two classes.

c. Assuming that priors are equals, find the Bayes Classifier.

d. Compute the training error.

e. Train a linear regression model, using the function $lm(y \sim x)$, with the training set.

f. Plot the boundary decision of Bayes Classifier and the line obtained by the linear regression model.

g. Generate a test set of 50 observations and compute the test error of Bayes Classifier and the linear model.

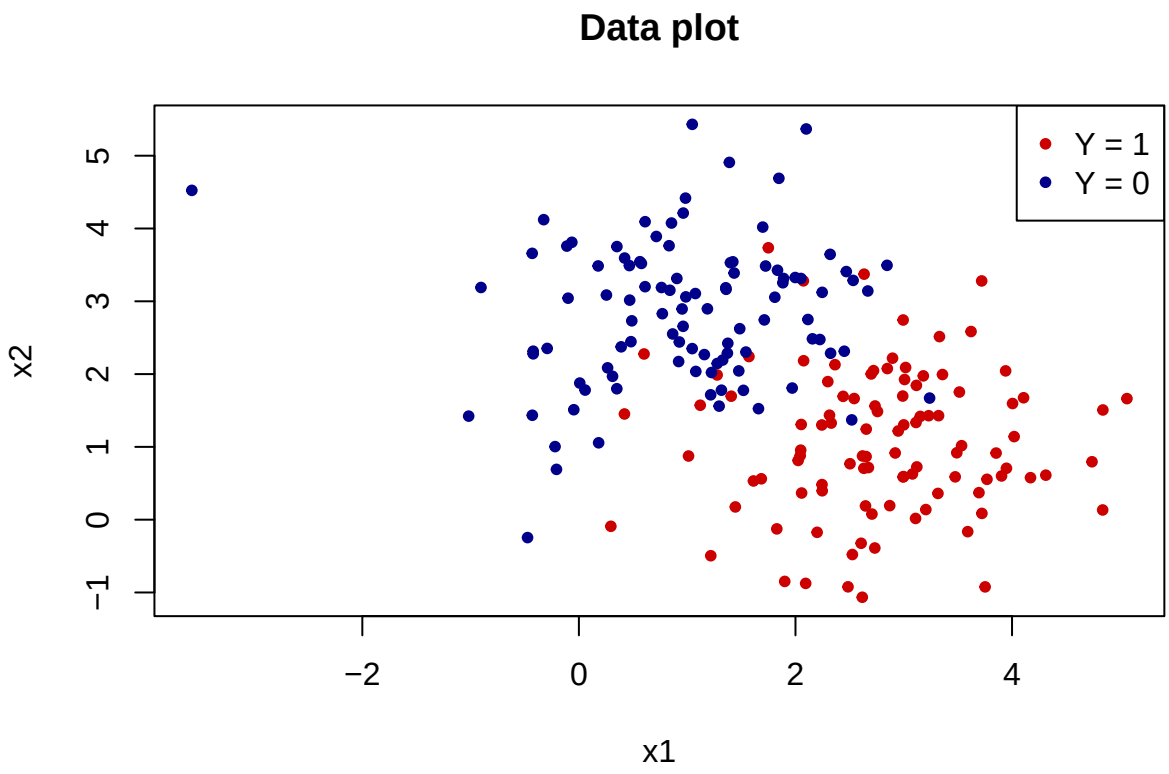
Respuesta

a. **Write an R code to generate this data set**

```
library(MASS)
set.seed(2023)
# Datos de categoría 1
mu1 = c(3, 1)
sigma1 = diag(2)
n1 = 100

obs1 = mvrnorm(n1, mu1, sigma1)
cat1 = rep(1,n1)

# Dataset de categoría 0
mu2 = c(1, 3)
sigma2 = diag(2)
n2 = 100

obs2 = mvrnorm(n2, mu2, sigma2)
cat2 = rep(0,n2)

# Dataset de entrenamiento
mat1 = matrix(cbind(obs1,cat1), ncol = 3)
mat2 = matrix(cbind(obs2,cat2), ncol = 3)

mat_train = rbind(mat1, mat2)
```

b. **Plot this data using different colors for the two classes**

```
plot (x = mat_train[,1], y = mat_train[,2], col = ifelse(mat_train[,3] == 1,'red3','blue4')
      , xlab = 'x1', ylab = 'x2', main = 'Data plot', pch = 20)

legend('topright', legend = c('Y = 1', 'Y = 0'), col = c('red3', 'blue4'), pch = 20)
```



c. **Assuming that priors are equals, find the Bayes Classifier.**

Del desarrollo del ejercicio 1, se tiene que, utilizando el clasificador de Bayes, una observación es clasificada como $Y = 1$ si se cumple:

$$P(X = X|Y = 1) \times P(Y = 1) > P(X = X|Y = 0) \times P(Y = 0)$$

Dado que se tienen 100 observaciones de cada categoría, se cumple que $P(Y = 1) = P(Y = 0) = 0.5$, por lo que una clasificación es clasificada como $Y = 1$ si:

$$\frac{P(X = X|Y = 1)}{P(X = X|Y = 0)} > 1$$

Asumiendo que las densidades de $X|Y = 1$ y $X|Y = 0$ son $f_1(X)$ y $f_0(X)$ respectivamente, y utilizando el teorema del valor medio, se tiene que una observación es clasificada como $Y = 1$ si se cumple:

$$f_1(X) > f_0(X)$$

Dado que los datos fueron generados en base a distribuciones normales bivariadas para cada categoría, se tiene que:

$$X|Y = 1 \sim N(\mu_1, \Sigma_1); con \mu_1 = (3, 1) y \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$X|Y = 0 \sim N(\mu_0, \Sigma_0); con \mu_0 = (1, 3) y \Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Asimismo, se usa la fórmula de la densidad de la distribución normal bivariada, presentada a continuación:

$$f_x(X) = \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)}$$

Entonces, para $X|Y=1$ se tiene que:

$$|\Sigma_1| = 1$$

$$(X-\mu_1)^T = (x_1-3, x_2-1)$$

$$\Sigma_1^{-1}(X-\mu_1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} x_1-3 \\ x_2-1 \end{pmatrix} = \begin{pmatrix} x_1-3 \\ x_2-1 \end{pmatrix}$$

Luego:

$$(X-\mu_1)^T\Sigma_1^{-1}(X-\mu 1) = (x_1-3, x_2-1) \times \begin{pmatrix} x_1-3 \\ x_2-1 \end{pmatrix} = (x_1-3)^2 + (x_2-1)^2$$

De manera que la función de densidad para $X|Y=1$ se puede escribir como:

$$f_1(X) = \frac{1}{2\pi} e^{-\frac{1}{2}\left[(x_1-3)^2+(x_2-1)^2\right]}$$

Análogamente, se puede escribir la función de densidad para $X|Y=0$ de la siguiente manera:

$$f_0(X) = \frac{1}{2\pi} e^{-\frac{1}{2}\left[(x_1-1)^2+(x_2-3)^2\right]}$$

Sustituyendo estas dos expresiones en la inecuación presentada anteriormente, se tiene que una observación será clasificada como $Y=1$ si se cumple que:

$$\frac{1}{2\pi} e^{-\frac{1}{2}\left[(x_1-3)^2+(x_2-1)^2\right]} > \frac{1}{2\pi} e^{-\frac{1}{2}\left[(x_1-1)^2+(x_2-3)^2\right]}$$

Simplificando el factor $\frac{1}{2\pi}$, aplicando la función logaritmo de ambos lados de la ecuación y operando con ambos términos, se tiene que:

$$-\frac{1}{2}(x_1-3)^2 + (x_2-1)^2 > -\frac{1}{2}(x_1-1)^2(x_2-3)^2$$

$$(x_1-1)^2 + (x_2-3)^2 > (x_1-3)^2 + (x_2-1)^2$$

$$x_1^2 - 2x_1 + 1 + x_2^2 - 6x_2 + 9 > x_1^2 - 6x_1 + 9 + x_2^2 - 2x_2 + 1$$

$$-2x_1 - 6x_2 > -6x_1 - 2x_2$$

$$4x_1 > 4x_2$$

$$x_1 > x_2$$

De manera que el clasificador de Bayes se puede escribir mediante la siguiente expresión:

$$Y_{pred} = 1 si x_1 > x_2$$

$$Y_{pred} = 0 si x_1 < x_2$$

Y la frontera de Bayes será la recta $x_1 = x_2$

d. **Compute the training error.**

Dado que la frontera de Bayes es la recta $x_1 = x_2$, y que la regla de Bayes indica que una observación debería ser clasificada con $Y = 1$ si $x_1 > x_2$ y con $Y = 0$ si $x_1 < x_2$, el error de entrenamiento será dado por la cantidad de observaciones que hayan sido clasificadas con $Y = 1$ cuando $x_1 < x_2$ sumadas a la cantidad de observaciones clasificadas como $Y = 0$ cuando $x_1 > x_2$

```
df_train = data.frame(mat_train)
false_positive = subset(df_train, df_train[,1]<df_train[,2] & df_train[,3] == 1)
false_negative = subset(df_train, df_train[,1]>df_train[,2] & df_train[,3] == 0)

fp = nrow(false_positive)
fn = nrow(false_negative)
tot = nrow(df_train)

bayes_train_error = (fp + fn)/tot

cat ('Falsos positivos:',fp,'\n')
```

```
## Falsos positivos: 10
```

```
cat ('Falsos negativos:',fn,'\n')
```

```
## Falsos negativos: 6
```

```
cat ('Total de observaciones:',tot,'\n')
```

```
## Total de observaciones: 200
```

```
cat ('Error de Bayes:',bayes_train_error)
```

```
## Error de Bayes: 0.08
```

e. **Train a linear regression model, using the function $lm(y \sim x)$, with the training set.**

```
colnames(df_train) = c('x1', 'x2', 'y')

regressor = lm(y~x1+x2, data = df_train)

coef(regressor)
```

```
## (Intercept)          x1          x2
##   0.4927216   0.1832768  -0.1736781
```

f. **Plot the boundary decision of Bayes Classifier and the line obtained by the linear regression model.**

Dado que el problema en cuestión es de clasificación, se plantea el siguiente clasificador en base al modelo de regresión planteado:

$$Y_{pred} = 1 si \beta_0^{est} + \beta_1^{est} x_1 + \beta_2^{est} x_2 > 0.5$$

$$Y_{pred} = 0 si \beta_0^{est} + \beta_1^{est} x_1 + \beta_2^{est} x_2 < 0.5$$

Siendo β_i^{est} el estimador del parámetro β_i. De esta manera, la frontera para el clasificador planteado será dada por la ecuación:

$$f(x_1, x_2) = \beta_0^{est} + \beta_1^{est} x_1 + \beta_2^{est} x_2 - 0.5 = 0$$

Operando con esta ecuación, se obtiene la ecuación de la recta de la frontera para el clasificador planteado:

$$\beta_0^{est} + \beta_1^{est} x_1 + \beta_2^{est} x_2 - 0.5 = 0$$

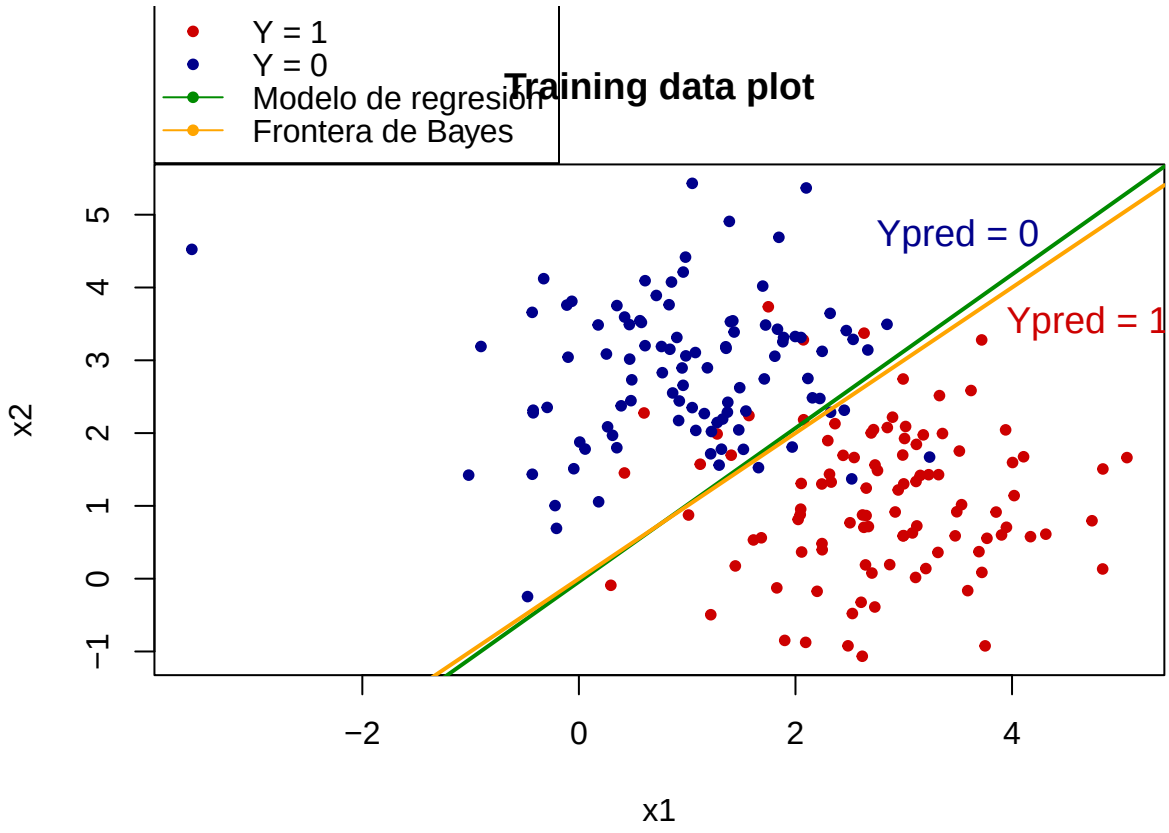
$$\beta_2^{pred} x_2 = -\beta_1^{pred} x_1 + 0.5 - \beta_0^{pred}$$

$$x_2 = -\frac{\beta_1^{pred}}{\beta_2^{pred}} x_1 + \frac{0.5 - \beta_0^{pred}}{\beta_2^{pred}}$$

$$x_2 = Ax_1 + B; con A = -\frac{\beta_1^{pred}}{\beta_2^{pred}} y B = \frac{0.5 - \beta_0^{pred}}{\beta_2^{pred}}$$

Cabe destacar que, dado que β_2 es negativo, una observación será clasificada como $Y = 1$ si $x_2 < Ax_1 + B$.

```
# Ploteo de puntos
plot (x = mat_train[,1], y = mat_train[,2], col = ifelse(mat_train[,3] == 1,'red3','blue4')
      , xlab = 'x1', ylab = 'x2', main = 'Training data plot', pch = 20)

# Recta obtenida por modelo de regresión lineal
beta0 = coef(regressor)[1]
beta1 = coef(regressor)[2]
beta2 = coef(regressor)[3]
umbral = 0.5

abline(a = (umbral - beta0)/beta2, b = -beta1/beta2, col = 'green4', lwd = 2)

# Frontera de Bayes
abline(a = 0, b = 1, col = 'orange', lwd = 2)

# Delimitación de zonas
text(x = 3.5, y = 4.7, labels = "Ypred = 0", col = 'blue4', cex = 1.2)
text(x = 4.7, y = 3.5, labels = "Ypred = 1", col = 'red3', cex = 1.2)

# Referencia
par(mar = c(5,5,2,2), xpd = TRUE)

legend('topleft', legend = c('Y = 1', 'Y = 0', 'Modelo de regresión', 'Frontera de Bayes'),
       col = c('red3', 'blue4', 'green4', 'orange'), lty = c(0,0,1,1), pch = 20,
       inset = c(0,-0.325))
```



g. **Generate a test set of 50 observations and compute the test error of Bayes Classifier and the linear model.**

```
# Dataset de testeo.
# Se generan con distribución normal en torno al (1,1) de manera que la media esté dentro de
# la frontera de Bayes

library(MASS)
mu = c(1, 1)
sigma = diag(2)
n = 50

obs = mvrnorm(n, mu, sigma)
cat = sample(1:0, size = n, replace = TRUE, prob = c(0.5, 0.5))

mat_test = matrix(cbind(obs,cat), ncol = 3)

df_test = data.frame(mat_test)
colnames(df_test) = c('x1', 'x2', 'y')

df_test
```

```
##             x1          x2 y
## 1   0.77418923  0.472781831 1
## 2   0.04736431  1.428993465 1
## 3   0.24185622  0.396797534 1
## 4   1.64004746  2.879212794 0
```

```
## 5   1.47530754  1.835359794 0
## 6   2.65078275  1.597903139 1
## 7   1.16296900 -0.051966477 1
## 8   0.43237782 -0.001889761 0
## 9  -0.22016721  1.079143456 1
## 10 -0.90883036 -0.270567337 0
## 11  1.98980833  2.460366845 1
## 12  0.41935901 -0.152867816 1
## 13 -0.43952029  0.359707774 1
## 14  0.96300155  1.276454814 1
## 15  0.35232485 -0.338594938 0
## 16  0.87125269  2.232124787 1
## 17  1.66365157  0.490534381 1
## 18  0.23111500  1.275890904 0
## 19  0.53654433 -0.012323967 1
## 20 -1.41893682  0.335474335 1
## 21  2.32164015  1.726952495 0
## 22  1.41480293  0.679003840 0
## 23 -0.19792665  1.402901903 1
## 24  1.66534843 -0.617393201 1
## 25  1.17322924  1.208616830 0
## 26  1.64572857  2.626670038 0
## 27  0.69889348  0.572598895 0
## 28  1.86779287  1.369476730 1
## 29  0.87031783  0.385767661 1
## 30 -0.15287912  1.434750414 1
## 31  1.79726653  0.272534969 1
## 32  2.37567574  0.670106364 0
## 33  1.09113450  0.475914435 1
## 34  0.78701249  1.829732553 1
## 35  1.09256861  1.805963866 1
## 36  0.36158190  0.302520391 0
## 37  2.84431310  0.805856191 0
## 38  2.02144763  1.099990030 1
## 39  1.72783296  0.124347410 1
## 40 -0.10477682  1.419235837 1
## 41  1.98073494 -0.504982580 1
## 42  0.13203486  1.362800602 0
## 43  1.21963326  0.791175627 0
## 44  1.94981872  0.558666774 1
## 45  0.31938720  1.246101840 0
## 46  3.78423896 -0.153364348 0
## 47  1.79190990  0.839401444 0
## 48  2.03007143 -0.546923398 0
## 49  1.58448808 -0.323331625 0
## 50  0.11359310  2.395835253 1
```

```r
# Test error para clasificador de Bayes
set.seed(2024)
false_positive_test = subset(df_test, df_test[,1]<df_test[,2] & df_test[,3] == 1)
false_negative_test = subset(df_test, df_test[,1]>df_test[,2] & df_test[,3] == 0)

fp_test = nrow(false_positive_test)
fn_test = nrow(false_negative_test)
```

```
tot_test = nrow(df_test)

bayes_test_error = (fp_test + fn_test)/tot_test

cat ('Falsos positivos:',fp_test,'\n')
```

```
## Falsos positivos: 14
```

```
cat ('Falsos negativos:',fn_test,'\n')
```

```
## Falsos negativos: 13
```

```
cat ('Total de observaciones:',tot_test,'\n')
```

```
## Total de observaciones: 50
```

```
cat ('Error de testeo de Bayes:',bayes_test_error)
```

```
## Error de testeo de Bayes: 0.54
```

Como fue explicado en la parte f, de acuerdo al modelo lineal, una observación será clasificada como $Y = 1$ si $x_2 < Ax_1 + B$, siendo $A = -\frac{\beta_1}{\beta_2}$ y $B = \frac{0.5-\beta_0}{\beta_2}$. Entonces, el error de testeo será dado por la cantidad de observaciones que hayan sido clasificadas con $Y = 1$ cuando $x_2 > Ax_1 + B$ sumadas a la cantidad de observaciones clasificadas como $Y = 0$ cuando $x_2 < Ax_1 + B$

```
# Test error para modelo de regresión lineal
A = -beta1/beta2
B = (umbral-beta0)/beta2

false_positive_test_lin = subset(df_test, df_test[,2]>A*df_test[,1]+B & df_test[,3] == 1)
false_negative_test_lin = subset(df_test, df_test[,2]<A*df_test[,1]+B & df_test[,3] == 0)

fp_test_lin = nrow(false_positive_test_lin)
fn_test_lin = nrow(false_negative_test_lin)
tot_test = nrow(df_test)

lin_test_error = (fp_test_lin + fn_test_lin)/tot_test

cat ('Falsos positivos:',fp_test_lin,'\n')
```

```
## Falsos positivos: 14
```

```
cat ('Falsos negativos:',fn_test_lin,'\n')
```

```
## Falsos negativos: 13
```

```
cat ('Total de observaciones:',tot_test,'\n')
```

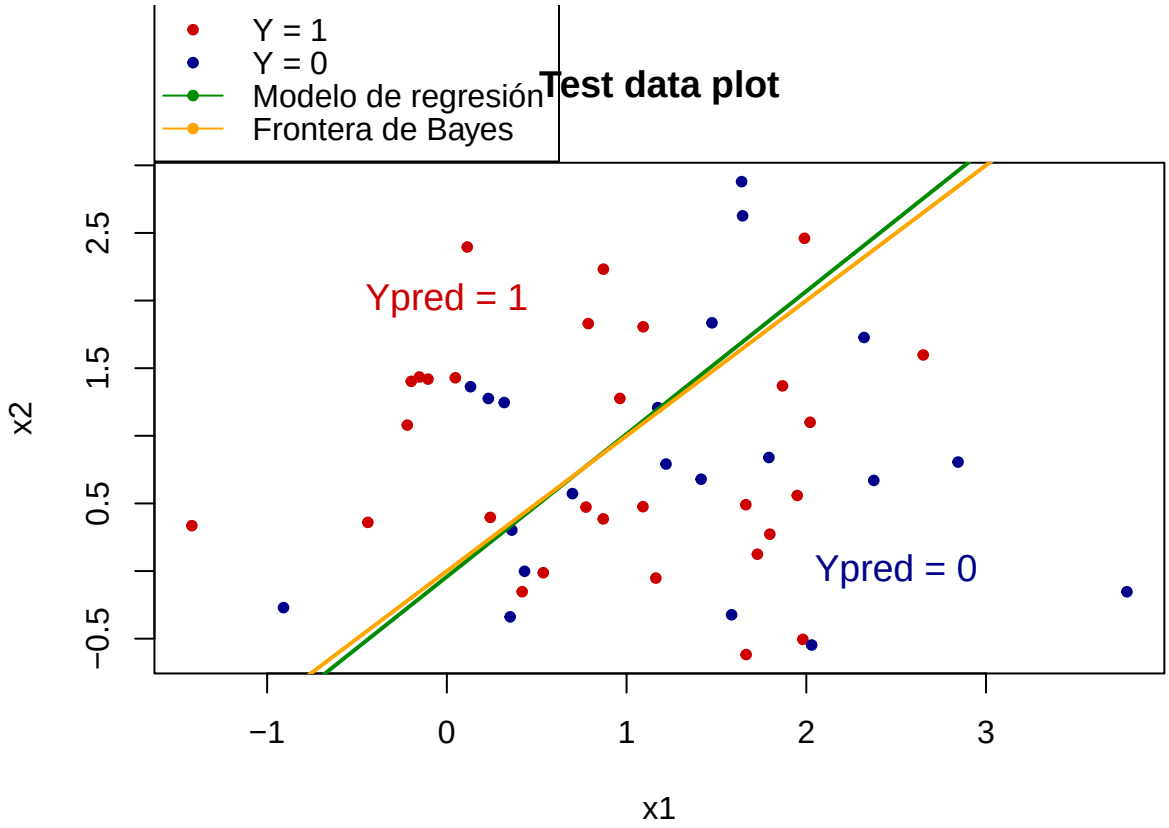
```
## Total de observaciones: 50
```

```
cat ('Error de testeo de modelo lineal:',lin_test_error)
```

```
## Error de testeo de modelo lineal: 0.54
```

```
# Ploteo de puntos
plot (x = mat_test[,1], y = mat_test[,2], col = ifelse(mat_test[,3] == 1,'red3','blue4'),
      xlab = 'x1', ylab = 'x2', main = 'Test data plot', pch = 20)

# Recta obtenida por modelo de regresión lineal
abline(a = (umbral - beta0)/beta2, b = -beta1/beta2, col = 'green4', lwd = 2)
```

```r
# Frontera de Bayes
abline(a = 0, b = 1, col = 'orange', lwd = 2)

# Delimitación de zonas
text(x = 2.5, y = 0, labels = "Ypred = 0", col = 'blue4', cex = 1.2)
text(x = 0, y = 2, labels = "Ypred = 1", col = 'red3', cex = 1.2)

# Referencia
par(mar = c(5,5,2,2), xpd = TRUE)

legend('topleft', legend = c('Y = 1', 'Y = 0', 'Modelo de regresión', 'Frontera de Bayes'),
       col = c('red3', 'blue4', 'green4', 'orange'), lty = c(0,0,1,1), pch = 20,
       inset = c(0,-0.325))
```



Es posible apreciar que el error obtenido para el clasificador de Bayes y para el modelo lineal es el mismo. Esto tiene sentido debido a que las fronteras de ambos modelos son casi iguales. Asimismo, si se observa el gráfico, es posible apreciar que no hay puntos que queden en una región para un modelo y en la otra región para el otro modelo, de manera que los errores obtenidos deben ser los mismos.

Ejercicio 4

Consider the following table.

x_1	x_2	x_3	y

0	3	0	Red
2	0	0	Red
0	1	3	Red
0	1	2	Green
-1	0	1	Green
1	1	1	Red

a. With the euclidean distance, what is the prediction with k = 1 and with k = 3 for the test observation (0; 0; 0)?

b. If the Bayes decision boundary in this problem is highly non-linear, then would we expect the best value for k to be large or small? Why?

Respuesta

a. **With the euclidean distance, what is the prediction with k = 1 and with k = 3 for the test observation (0; 0; 0)?**

Versión manual: En primer lugar se calculan las distancias euclídeas entre el punto para el cual se desea realizar la predicción y los puntos del dataset de entrenamiento. La distancia euclídea, para el caso de tres dimensiones x_1, x_2, x_2, entre un punto $x_0 = (x_{1,0}, x_{2,0}, x_{3,0})$ y un punto cualquiera $x = (x_1, x_2, x_3)$ se obtiene mediante la siguiente expresion:

$$d_{x,x_0} = \sqrt{(x_1 - x_{1,0})^2 + (x_2 - x_{2,0})^2 + (x_3 - x_{3,0})^2}$$

Dado que el punto en cuestión tiene coordenadas $x_0 = (0, 0, 0)$, la expresión anterior se simplifica de la siguiente manera:

$$d_{x,x_0} = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

Donde x_1, x_2, x_3 son las coordenadas de los puntos del dataset de entrenamiento. A continuación se calculan las distancias entre el punto para el cual se desea realizar la predicción y los puntos del dataset de entrenamiento.

$$d_1 = \sqrt{0^2 + 3^2 + 0^2} = \sqrt{9} = 3$$

$$d_2 = \sqrt{2^2 + 0^2 + 0^2} = \sqrt{4} = 2$$

$$d_3 = \sqrt{0^2 + 1^2 + 3^2} = \sqrt{10} \sim 3.16$$

$$d_4 = \sqrt{0^2 + 1^2 + 2^2} = \sqrt{5} \sim 2.24$$

$$d_5 = \sqrt{(-1)^2 + 0^2 + 1^2} = \sqrt{2} \sim 1.41$$

$$d_6 = \sqrt{1^2 + 1^2 + 1^2} = \sqrt{3} \sim 1.73$$

Luego, para $k = 1$, se clasifica la observación con la categoría del punto del dataset de entrenamiento más cercano al punto en cuestión. De las distancias calculadas arriba, se desprende que el punto más cercano es el

punto 5, con una distancia de $d_5 = \sqrt{2} \sim 1.41$, de manera que se clasifica a la observación con la categoría de dicho punto. Dado que dicha categoría es "Green", la predicción para el punto en cuestión también es "Green".

Para el caso de $k = 3$ se consideran los 3 puntos del dataset de entrenamiento más cercanos al punto en cuestión. Estos puntos son el punto 5, 6 y 2 con distancias de $d_5 = \sqrt{2} \sim 1.41$, $d_6 = \sqrt{3} \sim 1.73$ y $d_2 = 2$ respectivamente. La predicción para el punto en cuestión será la clase mayoritaria para estos tres puntos. Dado que las clases para los puntos 5, 6 y 2 son "Red", "Green" y "Red" respectivamente, la clase mayoritaria es "Red" ya que aparece dos veces, de manera que la predicción para el punto en cuestión es "Red".

```
# Generación de dataset
x1 = c(0,2,0,0,-1,1)
x2 = c(3,0,1,1,0,1)
x3 = c(0,0,3,2,1,1)
y = c('Red','Red','Red','Green', 'Green', 'Red')

mat_dat = matrix(cbind(x1,x2,x3,y), ncol = 4)
df = data.frame(mat_dat)
colnames(df) = c('x1', 'x2', 'x3', 'y')

df$x1 = as.integer(df$x1)
df$x2 = as.integer(df$x2)
df$x3 = as.integer(df$x3)
df
```

Versión en R:

```
##   x1 x2 x3     y
## 1  0  3  0   Red
## 2  2  0  0   Red
## 3  0  1  3   Red
## 4  0  1  2 Green
## 5 -1  0  1 Green
## 6  1  1  1   Red
```

```
# Entrenamiento de modelo y predicciones
library(class)

test_obs = c(0,0,0)

knn1 = knn(train = df[,1:3], test = test_obs, cl = df$y, k=1)
knn3 = knn(train = df[,1:3], test = test_obs, cl = df$y, k=3)

cat ('Prediccion con K = 1 para (0,0,0):',as.character(knn1[1]),'\n')
```

```
## Prediccion con K = 1 para (0,0,0): Green
```

```
cat ('Prediccion con K = 3 para (0,0,0):',as.character(knn3[1]))
```

```
## Prediccion con K = 3 para (0,0,0): Red
```

De manera que las predicciones realizadas en el desarrollo manual y computacional coinciden para ambos valores de k.

b. **If the Bayes decision boundary in this problem is highly non-linear, then would we expect the best value for k to be large or small? Why?**

Si la frontera de Bayes fuese extremadamente no lineal, sería esperable que el valor óptimo de k fuera bajo.

Un valor bajo de k produce un clasificador con bajo error de sesgo pero alta varianza, debido a que se utilizan pocos vecinos para realizar una predicción. En el caso extremo de k=1, en el cual se utiliza únicamente el punto más cercano del dataset de entrenamiento para realizar la predicción, la frontera obtenida mediante el modelo cambiará significativamente a medida que se aleja de un punto de determinada clase y se acerca a un punto de clase distinta, ya que la predicción será dada por este nuevo punto, produciendo así una frontera muy sinuosa. A medida que se aumenta el valor de k, cada punto tendrá menor peso en la predicción, lo que producirá curvas menos sinuosas.

Un valor alto de k produce un clasificador con alto error de sesgo pero varianza muy baja, ya que se utilizan muchos vecinos para realizar una predicción. Esto contribuye a alisar la frontera, ya que acercarse a un punto de determinada clase y alejarse de otro de clase distinta no generará cambios en muchas de las predicciones debido a que se considera una gran cantidad de puntos para realizar la predicción. Esto implica que sea esperable que un valor alto de k pueda producir fronteras más cercanas a fronteras lineales.

En virtud de esto, si la frontera de Bayes es extremadamente no lineal, es esperable que una frontera más sinuosa pueda generar mejores predicciones, lo que se logra con un valor de k bajo. Cabe destacar, igualmente, que un valor de k muy bajo puede llevar al sobreajuste del modelo, de manera que, si bien es esperable que el valor óptimo de k sea bajo, también es esperable que sea mayor a los valores contra el extremo mínimo del rango posible.