

**EXAMEN DE POSTGRADO 2022
(MAESTRIA EN CIENCIA DE DATOS)**

APELLIDOS Y NOMBRES: SERDAN CARDENAS RAMON WILDER CEL: 73736998

CORREO: wserdan@gmail.com

1. (Regresión múltiple)

(Se pide dar solución en R) Un estudio quiere generar un modelo que permita predecir la esperanza de vida media de los habitantes de una ciudad en función de diferentes variables. Se dispone de información sobre: habitantes, analfabetismo, ingresos, esperanza de vida, asesinatos, universitarios, heladas, área y densidad poblacional. Con la siguiente información

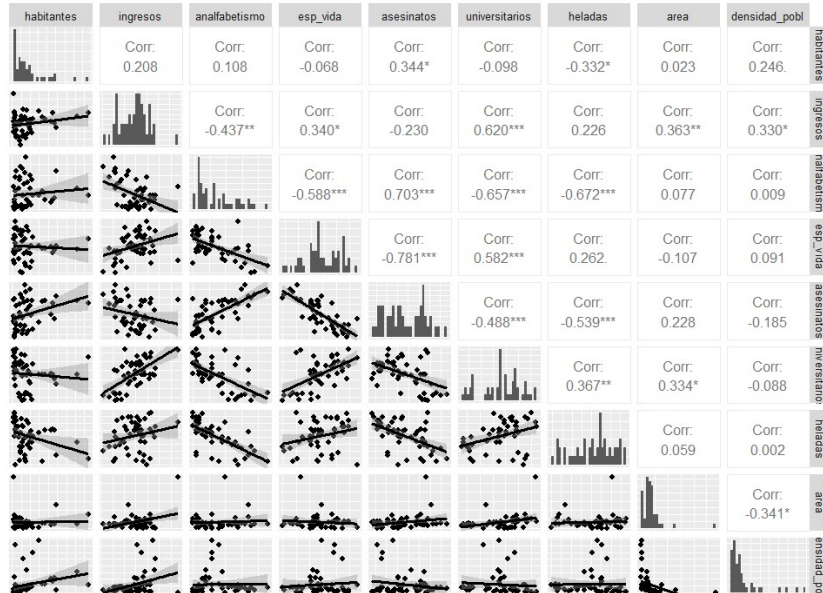
```
library(dplyr)
datos <- as.data.frame(state.x77)
datos <- rename(habitantes = Population,
               analfabetismo = Illiteracy,
               ingresos = Income,
               esp_vida = `Life Exp`,
               asesinatos = Murder,
               universitarios = `HS Grad`,
               heladas = Frost,
               area = Area,
               .data = datos)
datos <- mutate(.data = datos,
               densidad_pobl = habitantes * 1000/area)
datos
```

se pide: (todo con R)

a) Analizar la relación entre variables

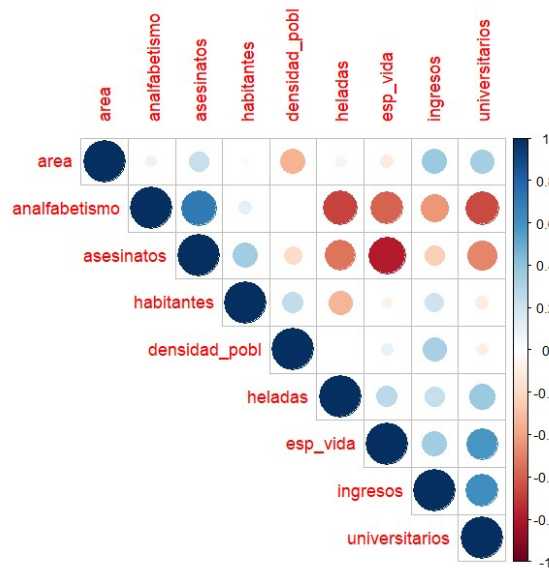
Las correlaciones de todas las variables cuantitativas son las siguientes:

```
library(ggally)
ggpairs(datos, lower = list(continuous = "smooth"),
        diag = list(continuous = "bar"), axisLabels = "none")
```



Utilizando la librería "corrplot" se determina que las correlaciones más importantes son :

```
install.packages('corrplot')
library(corrplot)
x <- cor(datos[,1:9])
corrplot(x, type="upper", order="hclust")
```



ANALISIS

Correlaciones positivas		Correlaciones negativas	
Porcentaje de Analfabetismo	Tasa de asesinatos	Número de días con helada	Porcentaje de Analfabetismo
Porcentaje de graduados (universitarios)	Esperanza de vida	Esperanza de vida	Tasa de asesinatos
Porcentaje de graduados (universitarios)	Ingresos	Porcentaje de graduados (universitarios)	Porcentaje de Analfabetismo

- b) Generar un modelo de regresión lineal con todas las variables (hacer un summary y analizar)

Aplicamos el modelo a las 9 variables:

```
modelo <- lm(esp_vida ~ habitantes + ingresos + analfabetismo +
             asesinatos + universitarios + heladas + area + densidad_pobl,
             data = datos )
summary(modelo)
```

Con el siguiente resultado:

```
Call:
lm(formula = esp_vida ~ habitantes + ingresos + analfabetismo +
    asesinatos + universitarios + heladas + area + densidad_pobl,
    data = datos)

Residuals:
    Min       1Q   Median       3Q      Max
-1.47514 -0.45887 -0.06352  0.59362  1.21823

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.995e+01  1.843e+00  37.956 < 2e-16 ***
habitantes    6.480e-05  3.001e-05   2.159  0.0367 *
ingresos      2.701e-04  3.087e-04   0.875  0.3867
analfabetismo  3.029e-01  4.024e-01   0.753  0.4559
asesinatos   -3.286e-01  4.941e-02  -6.652 5.12e-08 ***
universitarios 4.291e-02  2.332e-02   1.840  0.0730 .
heladas      -4.580e-03  3.189e-03  -1.436  0.1585
area         -1.558e-06  1.914e-06  -0.814  0.4205
densidad_pobl -1.105e-03  7.312e-04  -1.511  0.1385
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7337 on 41 degrees of freedom
Multiple R-squared:  0.7501,    Adjusted R-squared:  0.7013
F-statistic: 15.38 on 8 and 41 DF, p-value: 3.787e-10
```

ANALISIS

De acuerdo al Summary del modelo se tiene un R^2 importante de 0,75 que muestra una alta capacidad de explicar la variabilidad observada en la esperanza de vida.

El p-value del modelo es prácticamente cero ($3,787e-10$) por lo que se puede aceptar que el modelo no es por el azar. Una buena parte de los coeficientes no son significativos con un aporte casi nulo al modelo.

- c) Selección de los mejores predictores (hacer un summary y analizar)

Aplicando

```
step(object = modelo, direction = "both", trace = 1)
```

Se obtienen los siguientes resultados:

```
Start: AIC=-22.89
esp_vida ~ habitantes + ingresos + analfabetismo + asesinatos +
  universitarios + heladas + area + densidad_pobl

  Df Sum of Sq  RSS   AIC
- analfabetismo 1  0.3050 22.373 -24.208
- area          1  0.3564 22.425 -24.093
- ingresos      1  0.4120 22.480 -23.969
<none>         1  22.068  -22.894
- heladas       1  1.1102 23.178 -22.440
- densidad_pobl 1  1.2288 23.297 -22.185
- universitarios 1  1.8225 23.891 -20.926
- habitantes    1  2.5095 24.578 -19.509
- asesinatos    1 23.8173 45.886  11.707

Step: AIC=-24.21
esp_vida ~ habitantes + ingresos + asesinatos + universitarios +
  heladas + area + densidad_pobl

  Df Sum of Sq  RSS   AIC
- area          1  0.1427 22.516 -25.890
- ingresos      1  0.2316 22.605 -25.693
<none>         1  22.373  -24.208
- densidad_pobl 1  0.9286 23.302 -24.174
- universitarios 1  1.5218 23.895 -22.918
+ analfabetismo 1  0.3050 22.068 -22.894
- habitantes    1  2.2047 24.578 -21.509
- heladas       1  3.1324 25.506 -19.656
- asesinatos    1 26.7071 49.080  13.072

Step: AIC=-25.89
esp_vida ~ habitantes + ingresos + asesinatos + universitarios +
  heladas + densidad_pobl

  Df Sum of Sq  RSS   AIC
- ingresos      1  0.132 22.648 -27.598
- densidad_pobl 1  0.786 23.302 -26.174
<none>         1  22.516  -25.890
+ universitarios 1  1.424 23.940 -24.824
+ area          1  0.143 22.373 -24.208
+ analfabetismo 1  0.091 22.425 -24.093
- habitantes    1  2.332 24.848 -22.962
- heladas       1  3.304 25.820 -21.043
- asesinatos    1 32.779 55.295  17.033

Step: AIC=-27.6
esp_vida ~ habitantes + asesinatos + universitarios + heladas +
  densidad_pobl

  Df Sum of Sq  RSS   AIC
- densidad_pobl 1  0.660 23.308 -28.161
<none>         1  22.648  -27.598
+ ingresos      1  0.132 22.516 -25.890
+ analfabetismo 1  0.061 22.587 -25.732
+ area          1  0.043 22.605 -25.693
- habitantes    1  2.659 25.307 -24.046
- heladas       1  3.179 25.827 -23.030
- universitarios 1  3.966 26.614 -21.529
- asesinatos    1 33.626 56.274  15.910

Step: AIC=-28.16
esp_vida ~ habitantes + asesinatos + universitarios + heladas

  Df Sum of Sq  RSS   AIC
<none>         1  23.308  -28.161
+ densidad_pobl 1  0.660 22.648 -27.598
+ ingresos      1  0.006 23.302 -26.174
+ analfabetismo 1  0.004 23.304 -26.170
+ area          1  0.001 23.307 -26.163
- habitantes    1  2.064 25.372 -25.920
- heladas       1  3.122 26.430 -23.877
- universitarios 1  5.112 28.420 -20.246
- asesinatos    1 34.816 58.124  15.528

Call:
lm(formula = esp_vida ~ habitantes + asesinatos + universitarios +
  heladas, data = datos)

Coefficients:
(Intercept)  habitantes  asesinatos  universitarios  heladas
  7.103e+01   5.014e-05  -3.001e-01   4.658e-02  -5.943e-03
```

ANALISIS

Entonces el mejor modelo estaría dado por:

```
mejormodelo <- (lm(formula = esp_vida ~ habitantes + asesinatos + universitarios + heladas, data = datos))  
summary(mejormodelo)
```

Resultando:

```
Call:
lm(formula = esp_vida ~ habitantes + asesinatos + universitarios + heladas, data = datos)

Residuals:
    Min       1Q   Median       3Q      Max
-1.47095 -0.53464 -0.03701  0.57621  1.50683

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.103e+01  9.529e-01  74.542  < 2e-16 ***
habitantes    5.014e-05  2.512e-05   1.996  0.05201 .
asesinatos   -3.001e-01  3.661e-02  -8.199  1.77e-10 ***
universitarios 4.658e-02  1.483e-02   3.142  0.00297 **
heladas      -5.943e-03  2.421e-03  -2.455  0.01802 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7197 on 45 degrees of freedom
Multiple R-squared:  0.736,    Adjusted R-squared:  0.7126
F-statistic: 31.37 on 4 and 45 DF,  p-value: 1.696e-12
```

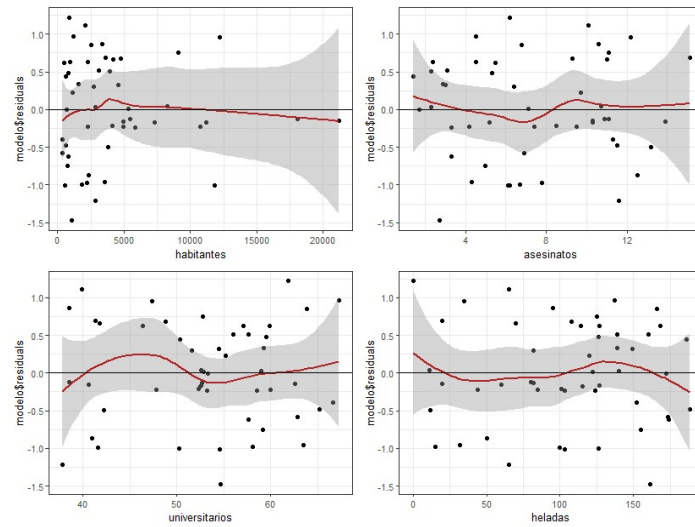
Se pueden interpretar cada uno de los coeficientes como elasticidades que explican la variable dependiente manteniendo las demás constantes. Según los resultados una reducción de los días con heladas en una unidad porcentual, la esperanza de vida aumenta en promedio 0,0059 puntos porcentuales manteniendo constantes los demás predictores.

- d) Realizar la validación de condiciones para la regresión múltiple lineal

Aplicando:

```
library(ggplot2)
install.packages('gridExtra')
library(gridExtra)
plotHabitantes <- ggplot(data = datos, aes(habitantes, modelo$residuals)) +
  geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
  theme_bw()
plotAsesinatos <- ggplot(data = datos, aes(asesinatos, modelo$residuals)) +
  geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
  theme_bw()
plotUniversitarios <- ggplot(data = datos, aes(universitarios, modelo$residuals)) +
  geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
  theme_bw()
plotHeladas <- ggplot(data = datos, aes(heladas, modelo$residuals)) +
  geom_point() + geom_smooth(color = "firebrick") + geom_hline(yintercept = 0) +
  theme_bw()
grid.arrange(plotHabitantes, plotAsesinatos, plotUniversitarios, plotHeladas)
```

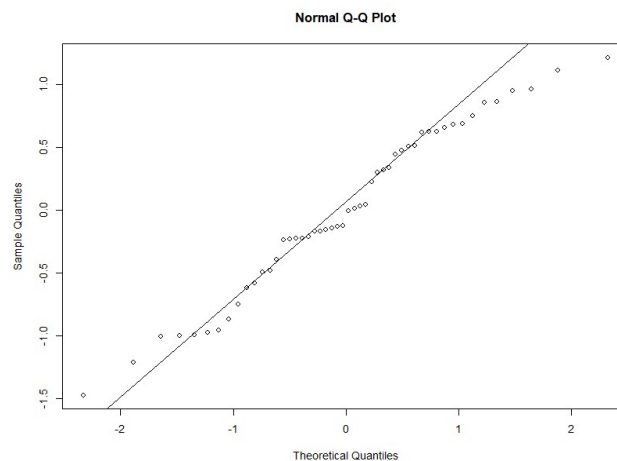
Gráficamente se verifica que la relación es lineal ya que los residuos se distribuyen aleatoriamente en torno al origen con una variabilidad constante en el x.



e) Determinar si cumple con la distribución normal de los residuos

Vemos gráficamente aplicando el siguiente comando.

```
qqnorm(modelo$residuals)
qqline(modelo$residuals)
```



f) Realizar el test de normalidad

Aplicamos:

```
shapiro.test(modelo$residuals)
```

Con los siguientes resultados:

```
Shapiro-Wilk normality test
data:  modelo$residuals
W = 0.97147, p-value = 0.2654
```

ANALISIS

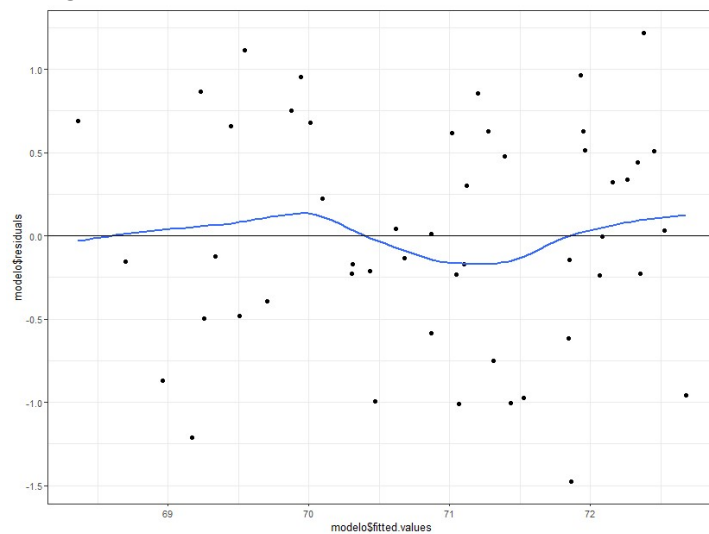
Debido a que el p-value es mayor a un nivel de significancia de 5%, no se rechaza la hipótesis nula de distribución normal de los residuos.

g) Realizar el test homocedasticidad

Realizamos un análisis gráfico con los siguientes comandos:

```
ggplot(data = datos, aes(modelo$fitted.values, modelo$residuals)) +  
  geom_point() +  
  geom_smooth(se = FALSE) +  
  geom_hline(yintercept = 0) +  
  theme_bw()
```

Obteniendo el siguiente resultado:



h) Realizar el Análisis de Inflación de Varianza (VIF)

Utilizando la librería:

```
install.packages('car')  
library(car)  
vif(mejormodelo)
```

Obtenemos el siguiente resultado:

```
vif(mejormodelo)  
habitantes      asesinatos universitarios      heladas  
1.189835      1.727844      1.356791      1.498077
```


2. Considerando el estudio de rendimiento antes y después cuya información es:

grupoAntes: 2, 4, 6, 1, 3
grupoDespues: 5, 2, 7, 1, 6

Se pide:

- a) Realizar la prueba de rangos de signos de Wilcoxon en sus tres tipos de pruebas)
(Se pide dar solución en R)

Detallamos las diferencias entre ambos grupos:

```
> grupoAntes <- c( 2, 4, 6, 1, 3 )
> grupoDespues <- c( 5, 2, 7, 1, 6 )
> diferencias <- c(grupoAntes - grupoDespues)
> rbind(grupoAntes, grupoDespues, diferencias)
      [,1] [,2] [,3] [,4] [,5]
grupoAntes    2    4    6    1    3
grupoDespues   5    2    7    1    6
diferencias   -3    2   -1    0   -3
```

En caso que la diferencia sea cero, completamos el vector con cero:

```
> rangosDeDiferencias <- rank(abs(diferencias[diferencias != 0]))
> rangosDeDiferencias <- c(rangosDeDiferencias[1:3], 0, rangosDeDiferencias[4])
> rangosDeDiferencias
[1] 3.5 2.0 1.0 0.0 3.5
```

Computamos los signos y las diferencias:

```
> DataFrame <- data.frame(grupoAntes = grupoAntes, grupoDespues = grupoDespues,
+                           signo = sign(diferencias),
+                           diferencia = abs(diferencias), rangos = rangosDeDiferencias)
> DataFrame
  grupoAntes grupoDespues signo diferencia rangos
1         2           5    -1          3     3.5
2         4           2     1          2     2.0
3         6           7    -1          1     1.0
4         1           1     0          0     0.0
5         3           6    -1          3     3.5
```

Realizamos el cálculo con el siguiente resultado:

```
> positivos <- sum(DataFrame[DataFrame$signo == 1,"rangos"])
> negativos <- sum(DataFrame[ DataFrame$signo == -1,"rangos"] )
> estadisticow <- min(c( positivos, negativos ))
> estadisticow
[1] 2
```


- b) Verifique el resultado de R con el cálculo a mano. (en ambos incisos realizar la interpretación de todo el proceso)

Comprobamos el resultado manual con el siguiente código:

```
> wilcox.test(x = grupoAntes, y = grupoDespues,  
+            alternative = "two.sided", mu = 0, paired = TRUE)  
  
    wilcoxon signed rank test with continuity correction  
  
data: grupoAntes and grupoDespues  
V = 2, p-value = 0.3573  
alternative hypothesis: true location shift is not equal to 0
```

```
> library(coin)  
> data <- data.frame(grupoAntes = grupoAntes, grupoDespues = grupoDespues)  
> wilcoxsign_test(grupoAntes ~ grupoDespues, data = data, distribution = "exact")  
  
    Exact wilcoxon-Pratt Signed-Rank Test  
  
data: y by x (pos, neg)  
      stratified by block  
Z = -1.0937, p-value = 0.375  
alternative hypothesis: true mu is not equal to 0
```

ANALISIS

Concluimos que el verdadero valor de μ es no es igual a 0. No existen elementos para rechazar la hipótesis nula

3. Un equipo de biólogos quiere generar un modelo estadístico que permita identificar a que especie (a o b) pertenece un determinado insecto. Para ello se han medido tres variables (longitud de las patas, diámetro del abdomen y diámetro del órgano sexual) en 10 individuos de cada una de las dos especies. Los datos son los siguientes (aplique análisis discriminante con R). Con sus correspondientes interpretaciones.

	especie	pata	abdomen	organo_sexual
1	a	191	131	53
2	a	185	134	50
3	a	200	137	52
4	a	173	127	50
5	a	171	128	49
6	a	160	118	47
7	a	188	134	54
8	a	186	129	51
9	a	174	131	52
10	a	163	115	47
11	b	186	107	49
12	b	211	122	49
13	b	201	144	47
14	b	242	131	54
15	b	184	108	43
16	b	211	118	51
17	b	217	122	49
18	b	223	127	51
19	b	208	125	50
20	b	199	124	46

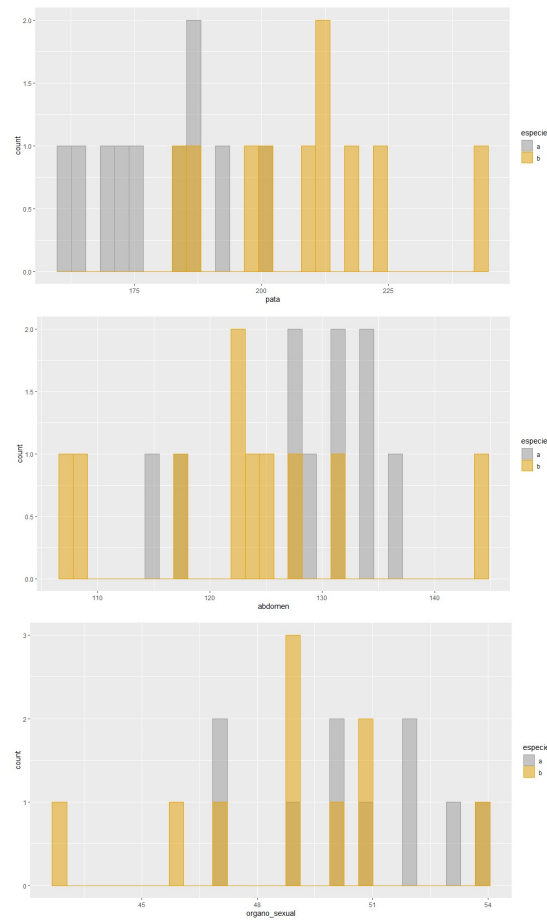
Primero, hacemos un análisis de las variables:

```
library(ggplot2)
install.packages('ggpubr')
library(ggpubr)

pata <- ggplot(data = insectos, aes(x = pata, color = especie, fill = especie)) +
  geom_histogram(position = "identity", alpha = 0.5)
pata + scale_color_manual(values = c("#999999", "#E69F00", "#56B4E9")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#56B4E9"))

abdomen <- ggplot(data = insectos, aes(x = abdomen, color = especie, fill = especie)) +
  geom_histogram(position = "identity", alpha = 0.5)
abdomen + scale_color_manual(values = c("#999999", "#E69F00", "#56B4E9")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#56B4E9"))

organo_sexual <- ggplot(data = insectos, aes(x = organo_sexual, color = especie,
  fill = especie)) +
  geom_histogram(position = "identity", alpha = 0.5)
organo_sexual + scale_color_manual(values = c("#999999", "#E69F00", "#56B4E9")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#56B4E9"))
```



ANALISIS

El largo de la pata de los insectos muestra una clara separación en las dos especies de insectos a y b.



Análisis Discriminante Lineal

```
install.packages('MASS')
library(MASS)
modeloDiscriminante <- lda(formula = especie ~ pata + abdomen + organo_sexual,
                             data = insectos)
modeloDiscriminante
```

Con el siguiente resultado:

```
> modeloDiscriminante
call:
lda(especie ~ pata + abdomen + organo_sexual, data = insectos)

Prior probabilities of groups:
  a  b
0.5 0.5

Group means:
      pata abdomen organo_sexual
a 179.1   128.4         50.5
b 208.2   122.8         48.9

Coefficients of linear discriminants:
              LD1
pata          0.13225339
abdomen       -0.07941509
organo_sexual -0.52655608
```

Podemos hacer una predicción en base al modelo:

```
> # predicción
> nuevoInput <- data.frame(pata = 180, abdomen = 130,
+                           organo_sexual = 40)
> predict(object = modeloDiscriminante, newdata = nuevoInput)
$class
[1] b
Levels: a b

$posterior
      a          b
1 2.592394e-07 0.9999997

$x
      LD1
1 2.952909
```

ANALISIS

Con base a la función discriminante, la probabilidad posterior de que la nueva información del insecto pertenezca a la especie b es del 99,9% en contraposición al 0,1% de que pertenezca a la especie a.

4. Considerando la siguiente información. La cuantificación del contenido en grasa de la carne puede hacerse mediante técnicas de analítica química, sin embargo, este proceso es costoso en tiempo y recursos. Una posible alternativa para reducir costes y optimizar tiempo es emplear un espectrofotómetro (instrumento capaz de detectar la absorbancia que tiene un material a diferentes tipos de luz en función de sus características). Para comprobar su efectividad se mide el espectro de absorbancia de 100 longitudes de onda en 215 muestras de carne, cuyo contenido en grasa se obtiene también por análisis químico para poder comparar los resultados. El set de datos `meatspec` del paquete `faraway` contiene toda la información.

```
library(faraway)
data(meatspec)
```

Se pide determinar:

- a) el modelo inicial

Instalamos el paquete “faraway” y aplicamos el modelo lineal a la totalidad de los datos.

```
install.packages('faraway')
library(faraway)
data(meatspec)
meatspec

modelo <- lm(fat ~ ., data = meatspec)
summary(modelo)
```

Resultando:

```

Call:
lm(formula = fat ~ ., data = meatspec)

Residuals:
    Min       1Q   Median       3Q      Max
-2.9833 -0.4982  0.0135  0.4864  3.1727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.302      1.876   3.892 0.000168 ***
V1             10898.047    3003.614   3.628 0.000428 ***
V2             -12174.864    5520.233  -2.205 0.029426 *
V3              -5953.285     8868.517  -0.671 0.503398
V4              23229.862    15426.530   1.506 0.134875
V5              -28386.219    19758.501  -1.437 0.153554
V6              12748.270    17381.421   0.733 0.464794
V7              -11422.335    11454.169  -0.997 0.320769
V8               7102.332     7123.935   0.997 0.320892
V9               783.655      5228.808   0.150 0.881130
V10             3512.239     6787.803   0.517 0.605856
V11            -10547.574    10580.407  -0.997 0.320926
V12             34638.288    18344.772   1.888 0.061543 .
V13            -38705.447    23098.395  -1.676 0.096542 .
V14             28895.947    19952.355   1.448 0.150293
V15            -13726.347    13312.307  -1.031 0.304676
V16             -7062.769     8172.878  -0.864 0.389308
V17             2571.597     6279.661   0.410 0.682932
V18             5263.427     6183.397   0.851 0.396432
V19             8860.827     8925.154   0.993 0.322914
V20            -12149.937    15184.189  -0.800 0.425276
V21            -19184.872    20536.132  -0.939 0.349680
V22             36626.953    22847.592   1.603 0.111680
V23            -11165.390    19302.712  -0.578 0.564111
V24            -15008.939    13616.072  -1.102 0.272655
V25             16698.992     8582.462   1.946 0.054151 .
V26             -4891.852     5901.456  -0.829 0.408880
V27             -6334.752     6072.685  -1.043 0.299084
V28             24043.786     8144.906   2.952 0.003834 **
V29            -39940.900    12335.575  -3.238 0.001578 **
V30             33309.092    17674.622   1.885 0.062034 .
V31            -23174.509    20974.708  -1.105 0.271539
V32             18764.305    18959.821   0.989 0.324423
V33            -37477.892    13458.994  -2.778 0.007158 **
V34            -6671.747     9353.448  -0.713 0.477122
V35             -5318.549     7534.861  -0.706 0.481716
V36             10488.898     5773.159   1.817 0.071869 .
V37            -8410.539     5892.265  -1.427 0.156202
V38            -408.228     7970.269  -0.051 0.959241
V39             19815.971    11338.219   1.748 0.083206 .
V40            -23690.179    15971.026  -1.483 0.140748
V41             29398.659    19340.032   1.520 0.131256
V42            -32055.252    20639.448  -1.553 0.123170
V43             11826.000    17491.895   0.676 0.500356
V44            -9994.257     11436.392  -0.874 0.383969
V45             23017.798     8927.175   2.578 0.011200 *
V46            -9041.633     6218.630  -1.454 0.148705
V47            -4846.799     3520.124  -1.377 0.171246
V48             1536.042     4401.789   0.349 0.727764
V49             2188.418     7363.225   0.297 0.766848
V50            -13170.870     9829.843  -1.340 0.182947
V51             26420.737    13371.372   1.976 0.050580 .
V52            -23565.834    16339.395  -1.442 0.151968
V53            -2005.210     16742.496  -0.120 0.904878
V54             30327.413    14023.378   2.163 0.032658 *
V55            -31802.344    10650.780  -2.986 0.003461 **
V56             12428.271     6395.916   1.943 0.054463 .
V57            -1021.107     4676.993  -0.219 0.826220
V58             210.251      4388.133   0.048 0.961869
V59            -7679.011     4511.526  -1.702 0.091465 .
V60             11590.949     3967.244   2.922 0.004199 **
V61            -6559.639     3756.703  -1.746 0.083485 .
V62             2533.819     3939.248   0.643 0.521370
V63             11950.924     5296.267   2.256 0.025947 *
V64            -18515.851     7070.171  -2.619 0.010021 *
V65             4051.697     8539.248   0.474 0.636066
V66             222.861      9691.472   0.023 0.981694
V67             10439.030    10111.231   1.032 0.304061
V68            -22570.742     9493.417  -2.378 0.019094 *
V69             17268.149     8163.742   2.115 0.036320 *
V70             -45.036      7357.838  -0.006 0.995127
V71            -8134.714     6796.093  -1.197 0.233802
V72            -1768.780     6344.295  -0.279 0.780905
V73             15744.948     5531.706   2.846 0.005246 **
V74            -11219.545     5666.910  -1.980 0.050132 .
V75             5289.427     5067.718   1.044 0.298810
V76            -2454.612     4760.274  -0.516 0.607101
V77             740.608     4922.688   0.150 0.880677
V78            -5730.806     5518.607  -1.038 0.301257
V79             12166.493     6026.835   2.019 0.045863 *
V80            -22688.979     7023.823  -3.230 0.001616 **
V81             14991.763     8595.338   1.744 0.083824 .
V82             3331.367     9984.910   0.334 0.739264
V83            -6651.082    11358.746  -0.586 0.559337
V84            -6752.949    12405.922  -0.544 0.587276
V85             16271.066    12434.546   1.309 0.193323
V86             5512.031    13689.180   0.403 0.687955
V87            -21092.220    15770.171  -1.337 0.183730
V88             9657.690    15143.593   0.638 0.524921
V89             273.586     13103.448   0.021 0.983379
V90            -5489.915    13927.199  -0.394 0.694180
V91             2891.941     15479.740   0.187 0.852133
V92             10160.850    14407.777   0.705 0.482103
V93            -3183.235    11882.686  -0.268 0.789269
V94            -7330.650    10859.287  -0.669 0.504913
V95             5551.521     9450.485   0.587 0.558075
V96            -3320.415     8349.562  -0.398 0.691613
V97            -2512.787     7974.922  -0.315 0.753272
V98            -5979.563     7355.289  -0.813 0.417935
V99             8283.253     7911.765   1.047 0.297336
V100           -101.926     3591.166  -0.028 0.977407

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.22 on 114 degrees of freedom
Multiple R-squared:  0.9951,    Adjusted R-squared:  0.9908
F-statistic: 232 on 100 and 114 DF,  p-value: < 2.2e-16

```

b) la suma de cuadrados del error

```
Residual standard error: 1.22 on 114 degrees of freedom  
Multiple R-squared: 0.9951, Adjusted R-squared: 0.9908  
F-statistic: 232 on 100 and 114 DF, p-value: < 2.2e-16
```

```
> SCE <- mean((modelo$fitted.values - meatspec$fat)^2)  
> SCE  
[1] 0.7898249
```

c) los predictores mediante stepwise

```
> modeloSegunStepwise <- step(object = modelo, trace = FALSE)  
> # Cantidad de predictores  
> length(modeloSegunStepwise$coefficients)  
[1] 64
```

d) el número óptimo de componentes principales identificado por cross validation

```
install.packages('pls')  
library(pls)  
set.seed(2222)  
modelo1 <- pcr(formula = fat ~ ., data = meatspec, scale. = TRUE,  
               validation = "cv")  
modeloCROSS <- MSEP(modelo1, estimate = "cv")  
which.min(modeloCROSS$val)
```

e) el test-MSE

```
> testMSE <- mean((modelo1$fitted.values - meatspec$fat)^2)  
> testMSE  
[1] 5.859152
```

f) los gráficos correspondientes.

