PAC 1 Regresión Lineal

Maria Lucas

2023-04-22

Ejercicio 1

Primero, cargamos los datos del documento excel.

```
#install.packages("readxl")
library("readxl")
data1 = read_excel("cicindela.xlsx")
names(data1)[1] <- "BD"
names(data1)[2] <- "WE"
names(data1)[3] <- "SPS"
names(data1)[4] <- "BS"
names(data1)[5] <- "AD"
```

(a) Ajuste del modelo

LA IA ME DICE QUE CHECKEE LAS ASUNCIONES (lineal, independencia, homocedasticidad, N de residuos)

```
# Creación del modelo
lmod = lm(BD ~ WE + SPS + BS + AD, data = data1)
sum = summary(lmod)
sum
```

```
##
## Call:
## lm(formula = BD ~ WE + SPS + BS + AD, data = data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3004 -2.7038  0.0795  2.6017  5.3924
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.9531    17.2661   0.866   0.4152
## WE            0.9123     1.0935   0.834   0.4317
## SPS           3.8970     1.1690   3.334   0.0125 *
## BS            0.6511     0.4530   1.437   0.1938
## AD           -1.5624     0.6610  -2.364   0.0501 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.513 on 7 degrees of freedom
## Multiple R-squared:  0.9578, Adjusted R-squared:  0.9337
## F-statistic: 39.71 on 4 and 7 DF,  p-value: 6.727e-05
```

Como podemos observar mediante la estimación de los coeficientes de regresión, la ecuación quedaría como: BD = 14.95 + 0.91WE + 3.89SPS + 0.65BS - 1.56AD.

El modelo obtenido es significativo, con un pvalor global = 6.727e-05. El test estadístico empleado es un F-test, éste testa como H0 que todos los coeficientes de regresión son 0, y como H1 que al menos uno es distinto de 0.

- H0: $\beta 1 = \beta 2 = \ldots = \beta p = 0$ (donde $\beta 1$, $\beta 2$, ..., βp son los coeficientes de regresión de las variables predictoras del modelo)
- H1: al menos un βi es diferente a 0, donde i = 1, 2, ..., p

En este caso al menos una de las variables tiene dependencia lineal con la variable respuesta (Beetle Density), ya que el pvalor es menor a 0.05 y por lo tanto, rechazamos la H0.

Tal y como se ve en la tabla de coeficientes, tanto SPS (Sand Particle Size, pvalor = 0.01) como AD (Amphipod Density, pvalor = 0.05) tienen un impacto significativo sobre la variable respuesta.

(b) Intervalos de confianza para AmphipodDensity

```
# CI a 95%
confint(lmod, "AD", level = 0.95)
```

```
##        2.5 %       97.5 %
## AD -3.125407 0.0007019125
```

```
# CI a 90%
confint(lmod, "AD", level = 0.9)
```

```
##          5 %       95 %
## AD -2.814699 -0.3100058
```

En ninguno de los dos intervalos se incluye el 0, es por ello que podemos deducir que el pvalor sea significativo a un nivel de confianza de 0.1 y 0.05, ya que como hemos explicado medimos si el parámetro es distinto a 0.

El coeficiente de regresión representa el cambio de la variable respuesta (BD o Beetle Density) por cada unidad que aumenta la variable predictora AD. Si este valor es 0 significa que la variable respuesta no varia conforme cambia el valor de la variable predictora. Si el valor es positivo un incremento de AD supone un incremento de BD, y si el valor es negativo un incremento de AD supone una reducción de BD.

(c) Multicolinealidad

```
library(car)
```

```
## Loading required package: carData
```

```
vif(lmod)
```

```
##       WE      SPS       BS       AD
## 3.771652 3.398998 1.158425 5.119632
```

El factor de inflación de la varianza o VIF mide cuánto se incrementa la varianza de los coeficientes de regresión estimados a causa de la colinealidad entre las variables predictoras. Valores de 1 indican que no hay correlación, valores de 1 a 5 que hay una ligera o moderada correlación, y valores mayores a 5 que las variables están altamente correlacionadas.

En este caso, podemos ver que sobretodo para AD hay una alta correlación y que por lo tanto no nos podemos fiar de la estimación de parámetros y pvalor.

El umbral del nivel de correlación aceptable entre variables dependerá de cada caso de estudio concreto.

(d) Modelo reducido

```
lmod_red= lm(BD ~ SPS + AD, data = data1)
summary(lmod_red)
```

```
##
## Call:
## lm(formula = BD ~ SPS + AD, data = data1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.933 -2.226 -0.512  3.315  5.787
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.5651     9.4259   3.773  0.00440 **
## SPS           3.7103     1.1215   3.308  0.00911 **
## AD           -2.1228     0.5167  -4.108  0.00264 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.621 on 9 degrees of freedom
## Multiple R-squared:  0.9431, Adjusted R-squared:  0.9305
## F-statistic: 74.58 on 2 and 9 DF,  p-value: 2.501e-06
```

```
anova(lmod_red, lmod)
```

```
## Analysis of Variance Table
##
## Model 1: BD ~ SPS + AD
## Model 2: BD ~ WE + SPS + BS + AD
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1      9 192.19
## 2      7 142.59  2     49.61 1.2178 0.3517
```

- H0: El modelo reducido es igual de bueno que el modelo con más variables.
- H1: El modelo con más variables explica mejor los datos.

O si lo escribimos de forma paramétrica:

- H0: RSS_reducido = RSS_completo
- H1: RSS_reducido > RSS_completo

Cabe destacar que el RSS (Residual Sum of Squares) mide la diferencia entre los valores reales de la variable respuesta y los valores predichos por el modelo. En otras palabras, es una medida de lo bien que se ajusta el modelo a los datos. Mediante la comparación de éste parámetro el F test nos ayuda a determinar si la adición de variables y con ello el aumento de grados de libertad mejoran el ajuste del modelo.

En nuestro caso como el pvalor = 0.35 aceptamos la hipótesis nula, el modelo BD ~ SPS + AD explica igual de bien los datos que el modelo con más variables, al ser más sencillo pero con iguales resultados, lo escogeríamos antes que el modelo más complejo.

Por otro lado, en el modelo reducido todas las variables explican de manera significa la variable respuesta. Además, el valor de R ajustado es similar en ambos modelos (0.93), este valor indica el porcentaje de la variable respuesta que es explicado por el modelo.

(e) Gráfico región de confianza

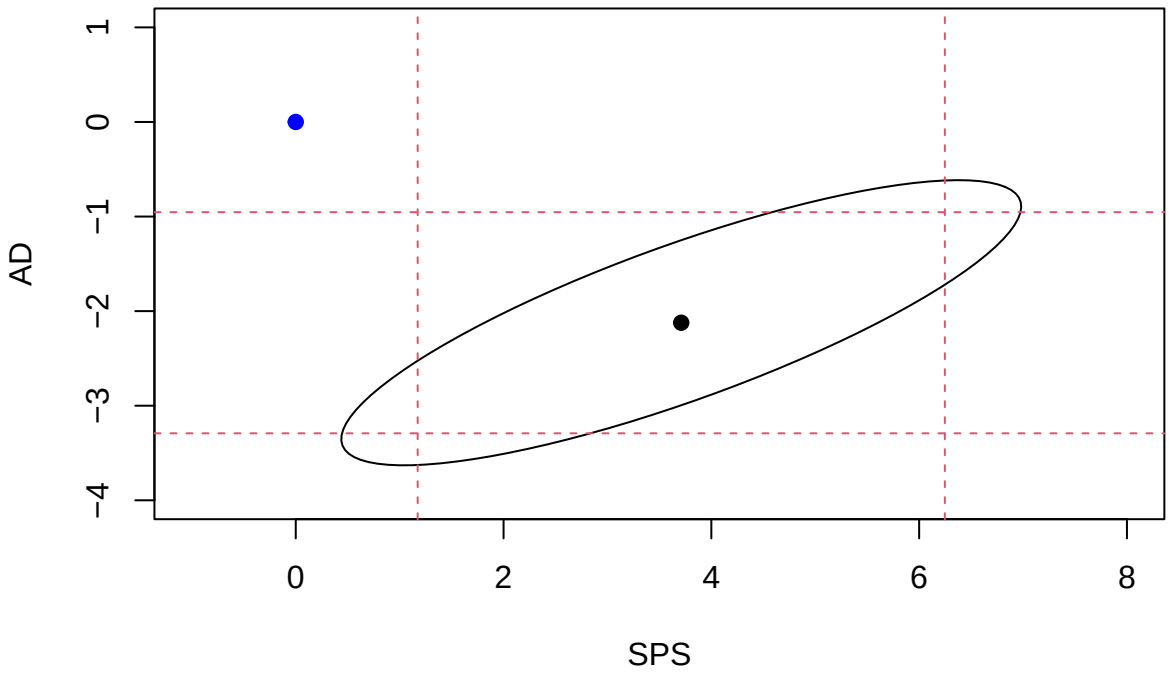
```
# install.packages('ellipse')
library(ellipse)
```

```
##
## Attaching package: 'ellipse'

## The following object is masked from 'package:car':
##
##     ellipse

## The following object is masked from 'package:graphics':
##
##     pairs
```

```
plot(ellipse(lmod_red, 2:3),type="l", xlim = c(-1, 8), ylim = c(-4, 1))
points(coef(lmod_red)[2], coef(lmod_red)[3], pch=19)
points(x=0, y=0, pch=19, col="blue")
abline(v=confint(lmod_red)[2,],lty=2,col=2)
abline(h=confint(lmod_red)[3,],lty=2,col=2)
```



El origen de coordenadas nos indica el resultado del test de Wald bajo las siguientes hipótesis:

- H0: $\beta 1 = \beta 2 = 0$. Los coeficientes de ambas variables son 0
- H1: $\beta 1 \neq 0$ y/o $\beta 2 \neq 0$. Caso contrario, al menos uno de los coeficientes es 0

Cuando la elipse de confianza incluye el (0,0) indica que los coeficientes estimados no son distintos que 0 y que por lo tanto las variables predictoras no aportan al modelo. Por otro lado, si no lo incluye significa que los coeficientes son distintos a 0 y las variables sí explican la variable respuesta. En este caso al no incluirlo, podemos deducir que las variables SPS y AD sí explican la variable BD.

(f) Predicción

```
new_ob = data.frame(SPS = 5, AD = 11)
pred <- predict(lmod_red, new_ob, interval = "confidence", level = 0.95)
cat("Predicted value:", pred[1], "\n")
```

```
## Predicted value: 30.76569
```

```
cat("95% confidence interval:", pred[2], "-", pred[3])
```

```
## 95% confidence interval: 26.05199 - 35.47939
```

```r
#install.packages('regclass')
library(regclass)
```

```
## Loading required package: bestglm
## Loading required package: leaps
## Loading required package: VGAM
## Loading required package: stats4
## Loading required package: splines
##
## Attaching package: 'VGAM'
## The following object is masked from 'package:car':
##
##     logit
## Loading required package: rpart
## Loading required package: randomForest
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

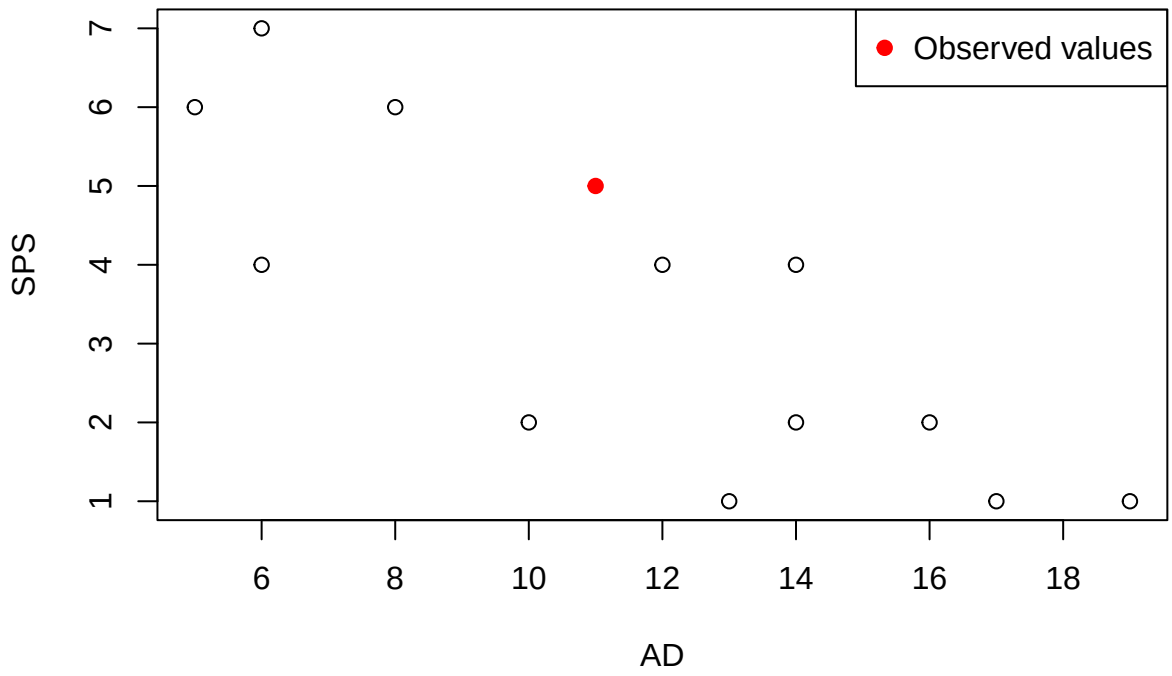
```r
extrapolation_check(lmod_red,new_ob)
```

```
##   Observation Percentile
## 1           1         25
```

```r
# create a scatter plot of SPS and AD
plot(SPS ~ AD, data = data1)

# add the observed values as points on the plot
points(x = 11, y = 5, col = "red", pch = 19)

# add a legend to the plot
legend("topright", legend = c("Observed values"), col = c("red"), pch = 19)
```



En este paquete (regclass) percentiles de aproximadamente 99 pueden implicar extrapolación, en nuestro caso obtenemos un percentil de 25 indicando que seguramente no la haya. Si revisamos el scatterplot podemos ver que estos valores de SPS y AD entran dentro del scope del modelo.

Ejercicio 2

Gráfico de dispersión

```
#install.packages("readxl")
library("readxl")
data2 = read.csv("lions.csv")
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library(dplyr)
```

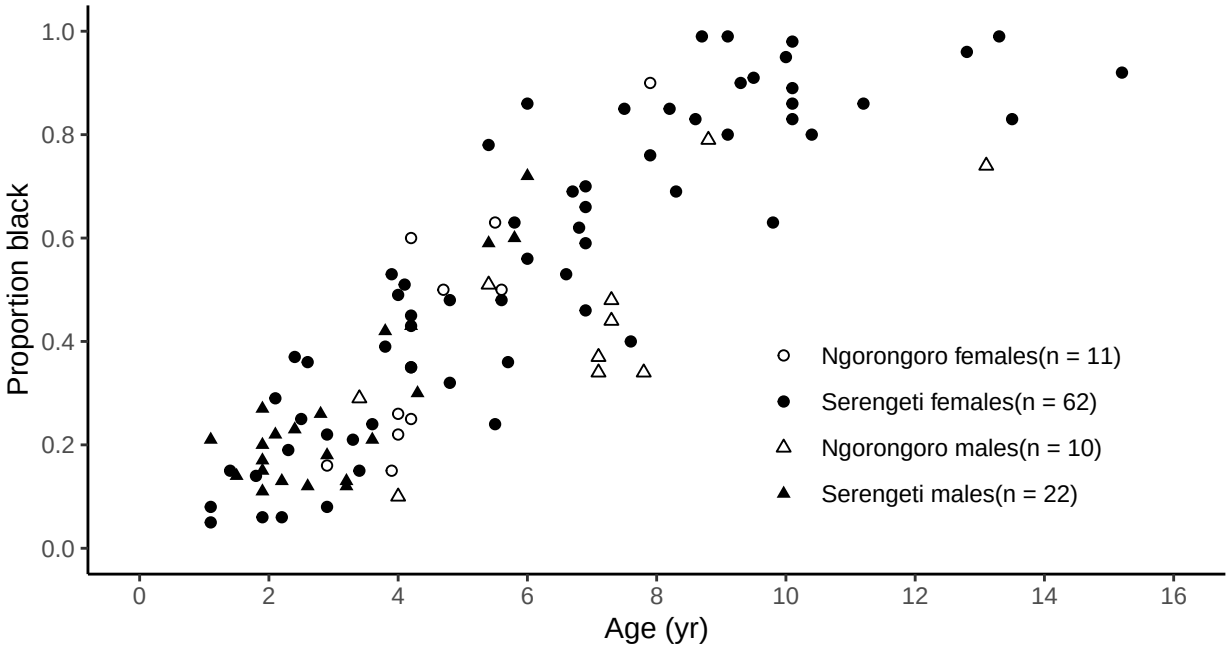
```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:randomForest':
##
```

```
##     combine
## The following object is masked from 'package:car':
##
##     recode
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
p = ggplot(data2, aes(age, prop.black))
p + geom_point(aes(shape = paste(ifelse(sex == "M", "males", "females"), ifelse(area == "N", "Ngorongor
  scale_shape_manual(name = "",
                     values = c(1, 19, 2, 17),
                     labels = data2 %>%
                       group_by(sex, area) %>%
                       summarize(n = n()) %>%
                       mutate(label = paste0(ifelse(area == "N", "Ngorongoro", "Serengeti"), " ", ifels
                       pull(label)) +
  labs(x = "Age (yr)", y = "Proportion black", shape = "") +
  scale_x_continuous(breaks = seq(0, 16, 2), limits = c(0, 16)) +
  scale_y_continuous(breaks = seq(0, 1, 0.2), limits = c(0, 1)) +
  scale_fill_discrete(breaks=c('F', 'M')) +
  theme_classic() +
  theme(aspect.ratio = 0.5, legend.position = c(0.75, 0.3))
```

```
## `summarise()` has grouped output by 'sex'. You can override using the `.groups`
## argument.
```


Proportion black
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
16
Age (yr)
Ngorongoro females(n = 11)
Serengeti females(n = 62)
Ngorongoro males(n = 10)
Serengeti males(n = 22)