PAC 1 Regresión Lineal

Maria Lucas

2023-04-22

Índice

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

```
# 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 (= $\beta 4$) 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.

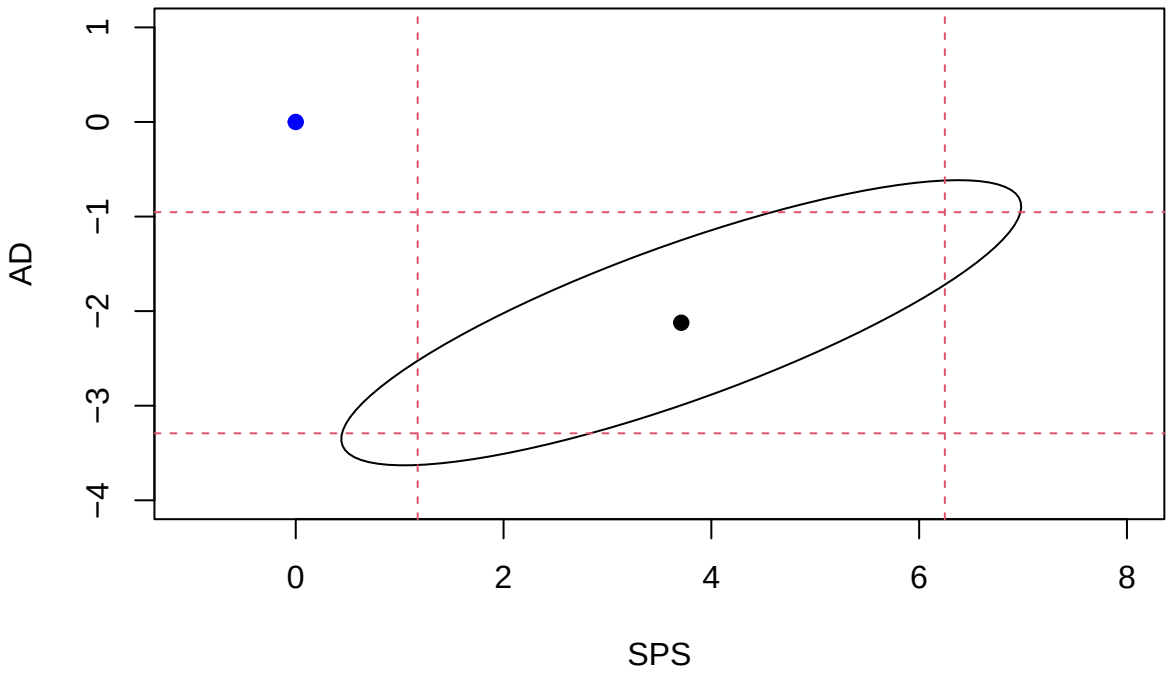
Por otro lado, en el modelo reducido todas las variables explican de manera significa la variable respuesta. Además, el valor de R ajustado es similar en ambos modelos (0.93), este valor indica el porcentaje de la variable respuesta que es explicado por el modelo.

(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
```

```r
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

Si la elipse de confianza no incluye el (0,0), esto sugiere que los coeficientes son distintos a 0 de forma estadísticamente significativa. Esto sugiere que las variables predictoras usadas para construir la elipse, tienen un efecto sobre la variable respuesta. Por otro lado, si no se incluye, indica que los coeficientes estimados no son distintos que 0 y que por lo tanto las variables predictoras no aportan al modelo. Esto no es necesariamente cierto, ya que existen múltiples motivos por los cuales la elipse incluiría el (0,0), como por ejemplo, que el modelo no sea lineal, y por lo tanto no veamos relación.

En este caso al no incluirlo, podemos deducir que las variables SPS y AD sí explican la variable BD.

(f) Predicción

```r
new_ob = data.frame(SPS = 5, AD = 11)
#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.
```

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

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

```
# Alternativamente
range_SPS = range(data1$SPS)
range_AD = range(data1$AD)
cat("Min AD:", range_AD[1], " Max AD:", range_AD[2], " Observed value:", new_ob$AD)
```

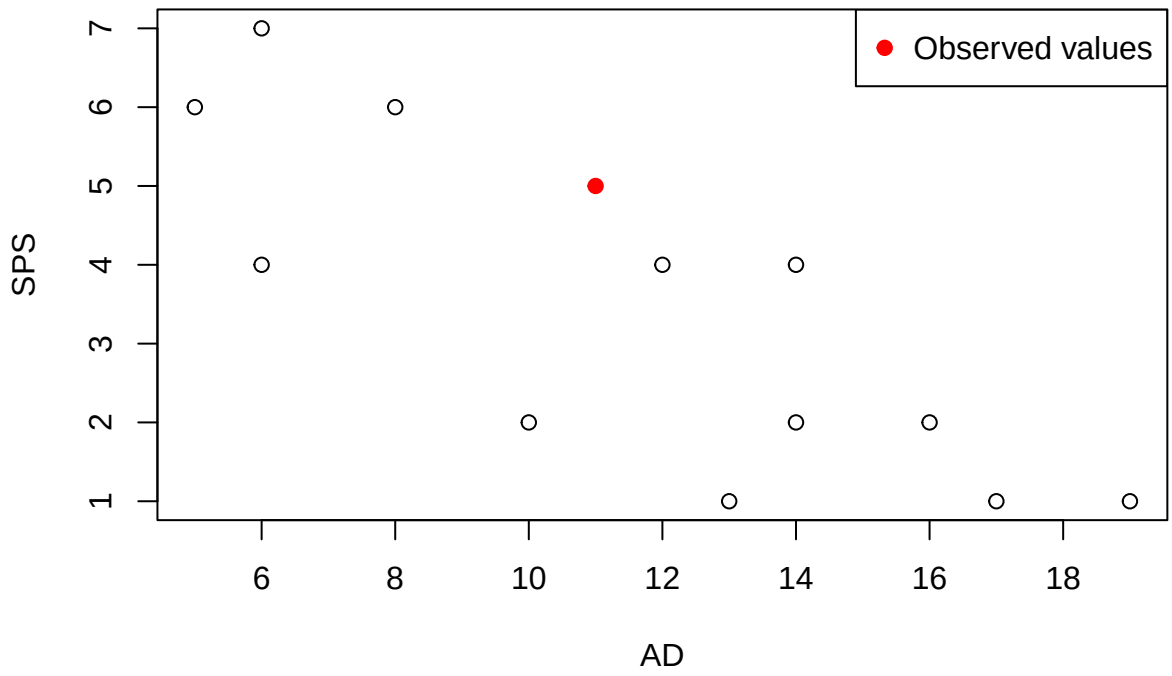
```
## Min AD: 5  Max AD: 19  Observed value: 11
```

```
cat("\n")
```

```
cat("Min SPS:", range_SPS[1], " Max SPS:", range_SPS[2], " Observed value:", new_ob$SPS)
```

```
## Min SPS: 1  Max SPS: 7  Observed value: 5
```

```
# 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. Usando la función range también podemos determinarlo, ya que nos indica el mínimo y el máximo de las variables señaladas. Si nuestra observación cae en ese rango no es una extrapolación.

```
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
```

Ejercicio 2

(a) 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
```

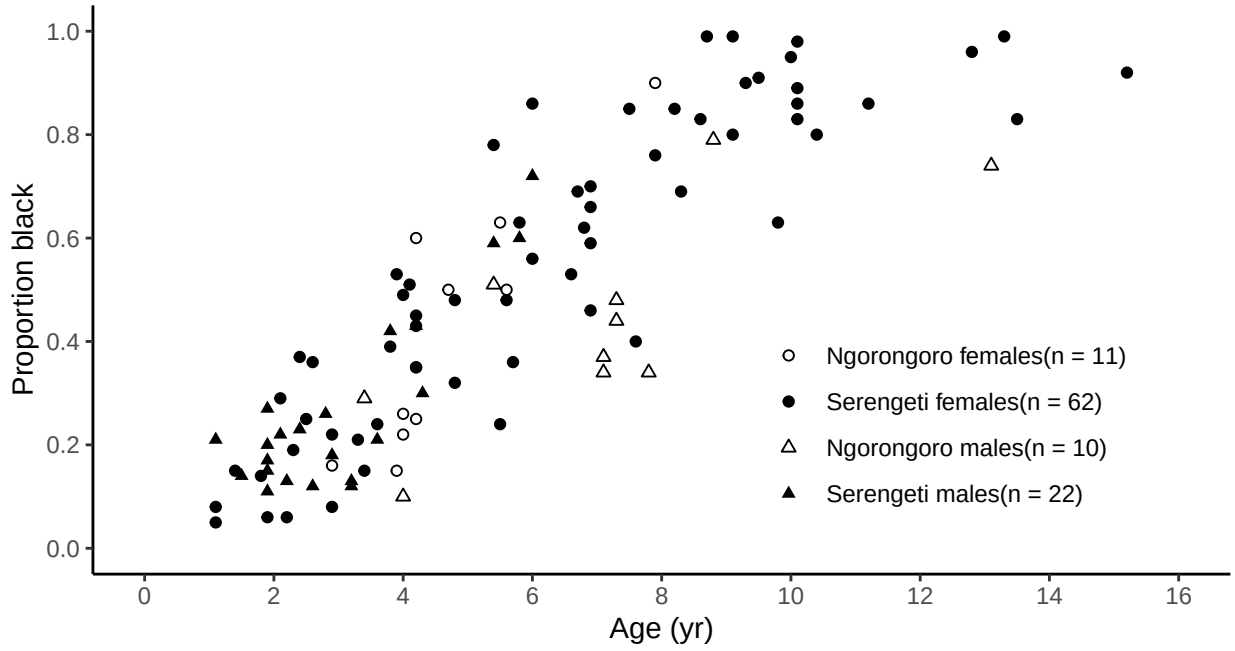
```
## The following object is masked from 'package:car':
##
##     recode
```

```
## 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.
```



(b) Modelos según área

```
lmod_all = lm(prop.black ~ age + sex + area, data = data2)
summary(lmod_all)
```

```
##
## Call:
## lm(formula = prop.black ~ age + sex + area, data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30265 -0.09116  0.00592  0.10049  0.32242
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.023324   0.044314   0.526   0.5998
## age          0.074464   0.004396  16.939   <2e-16 ***
## sexM        -0.068416   0.030662  -2.231   0.0279 *
## areaS        0.067473   0.034106   1.978   0.0506 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1367 on 101 degrees of freedom
## Multiple R-squared:  0.7713, Adjusted R-squared:  0.7645
## F-statistic: 113.5 on 3 and 101 DF,  p-value: < 2.2e-16
```

```r
data2_split_area = split(data2, f=data2$area)

lmod_N = lm(prop.black ~ age + sex, data = data2_split_area$N)
lmod_S = lm(prop.black ~ age + sex, data = data2_split_area$S)
summary(lmod_N)
```

```
##
## Call:
## lm(formula = prop.black ~ age + sex, data = data2_split_area$N)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20193 -0.11281 -0.02567  0.14511  0.23160
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04337    0.09071   0.478 0.638321
## age          0.07912    0.01681   4.707 0.000176 ***
## sexM        -0.16748    0.07885  -2.124 0.047776 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1531 on 18 degrees of freedom
## Multiple R-squared:  0.5538, Adjusted R-squared:  0.5042
## F-statistic: 11.17 on 2 and 18 DF,  p-value: 0.000701
```

```r
summary(lmod_S)
```

```
##
## Call:
## lm(formula = prop.black ~ age + sex, data = data2_split_area$S)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32208 -0.08310  0.00054  0.09561  0.33087
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.064161   0.034787   1.844   0.0688 .
## age          0.077495   0.004805  16.127   <2e-16 ***
## sexM        -0.030123   0.036358  -0.829   0.4098
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1316 on 81 degrees of freedom
## Multiple R-squared:  0.8065, Adjusted R-squared:  0.8017
## F-statistic: 168.8 on 2 and 81 DF,  p-value: < 2.2e-16
```

```r
lmod_all = lm(prop.black ~ age + sex + area, data = data2)
summary(lmod_all)
```

```
##
## Call:
## lm(formula = prop.black ~ age + sex + area, data = data2)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30265 -0.09116  0.00592  0.10049  0.32242
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.023324   0.044314   0.526   0.5998
## age          0.074464   0.004396  16.939   <2e-16 ***
## sexM        -0.068416   0.030662  -2.231   0.0279 *
## areaS        0.067473   0.034106   1.978   0.0506 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1367 on 101 degrees of freedom
## Multiple R-squared:  0.7713, Adjusted R-squared:  0.7645
## F-statistic: 113.5 on 3 and 101 DF,  p-value: < 2.2e-16
```

```
lmod_all2 = lm(prop.black ~ age * (sex + area), data = data2)
summary(lmod_all2)
```

```
##
## Call:
## lm(formula = prop.black ~ age * (sex + area), data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32206 -0.09746 -0.01365  0.10173  0.32558
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.003101   0.105559   0.029 0.976627
## age          0.081609   0.021302   3.831 0.000224 ***
## sexM        -0.005786   0.071180  -0.081 0.935374
## areaS        0.069820   0.106242   0.657 0.512593
## age:sexM    -0.015094   0.017246  -0.875 0.383571
## age:areaS   -0.004692   0.021184  -0.221 0.825161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1373 on 99 degrees of freedom
## Multiple R-squared:  0.7738, Adjusted R-squared:  0.7624
## F-statistic: 67.74 on 5 and 99 DF,  p-value: < 2.2e-16
```

En el modelo con todas las posibles variables y sin contrastar si existe interacción entre las variables, podemos observar que el sexo influye significativamente sobre la variable respuesta (proporción de negro en la nariz), con un pvalor = 0.0279.

Al separar por área, obtenemos los mismos resultados que en el artículo, en el que en la población Ngorongoroel sexo influye significativamente en la variable respuesta (pv = 0.04), mientras que en los Serengeti no influye (pv = 0.4). Podemos deducir que en el caso de los machos tienen una nariz más clara por el coeficiente estimado que es negativo (-0.16), es decir que el ser macho se relaciona significativamente con tener la nariz más clara.

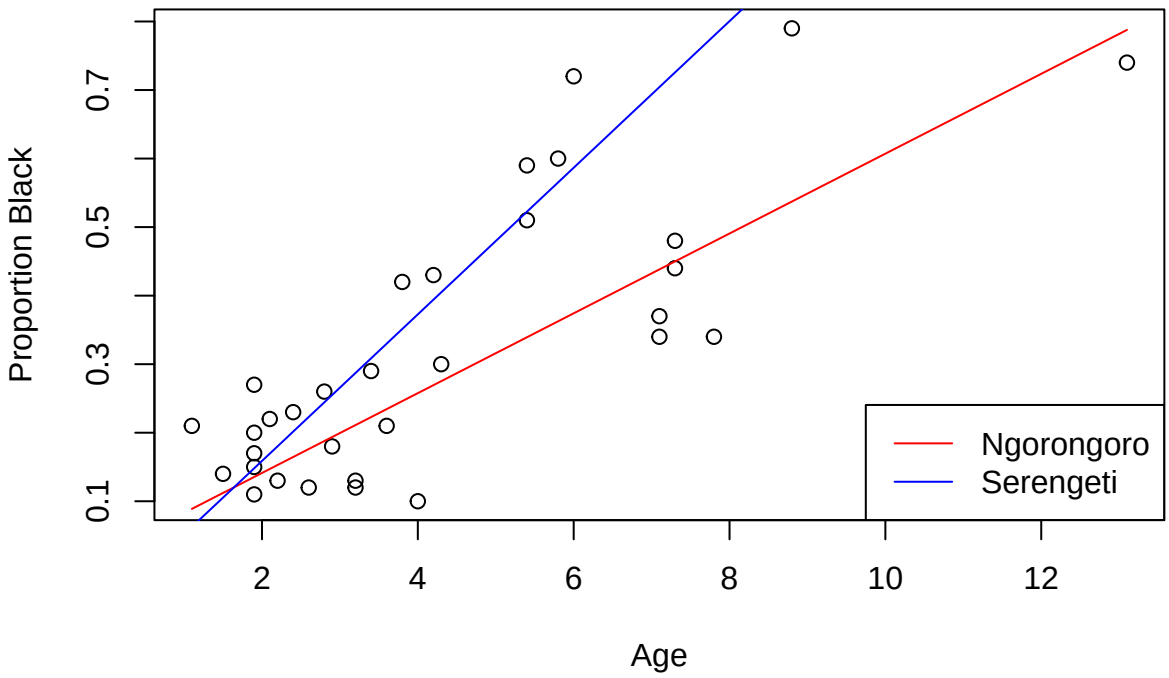
(c) Modelo leones macho

```
data2_split_sex = split(data2, f = data2$sex)
lmod_male = lm(prop.black ~ age + area, data = data2_split_sex$M)
summary(lmod_male)
```

```
##
## Call:
## lm(formula = prop.black ~ age + area, data = data2_split_sex$M)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16192 -0.08356 -0.01158  0.08842  0.22278
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.10826    0.08831  -1.226   0.2301
## age          0.07689    0.01126   6.827  1.7e-07 ***
## areaS        0.14411    0.06402   2.251   0.0321 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1162 on 29 degrees of freedom
## Multiple R-squared:  0.6798, Adjusted R-squared:  0.6577
## F-statistic: 30.78 on 2 and 29 DF,  p-value: 6.745e-08
```

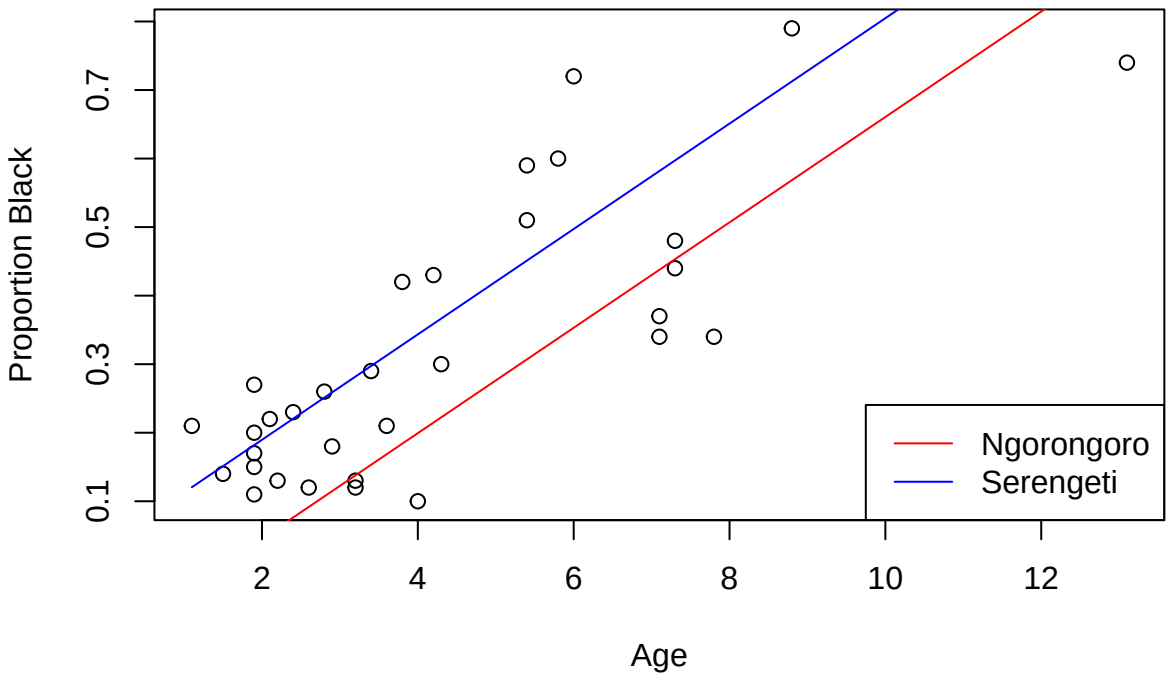
Confirmamos que para los machos existen diferencias significativas según el área (pv = 0.03), teniendo los Serengeti una nariz más oscura que los Ngorongoro (coef = 0.14).

```
# Fit a linear regression model with an interaction term
lmod_male_interaction <- lm(prop.black ~ age * area, data = data2_split_sex$M)

# Create a sequence of ages to use for plotting
age_seq <- seq(min(data2_split_sex$M$age), max(data2_split_sex$M$age), length.out = 100)

# Predict proportion black for each area at each age
pred_N <- predict(lmod_male_interaction, newdata = data.frame(age = age_seq, area = "N"))
pred_S <- predict(lmod_male_interaction, newdata = data.frame(age = age_seq, area = "S"))

# Plot regression lines for each area
plot(data2_split_sex$M$age, data2_split_sex$M$prop.black, xlab = "Age", ylab = "Proportion Black")
lines(age_seq, pred_N, col = "red")
lines(age_seq, pred_S, col = "blue")
legend("bottomright", legend = c("Ngorongoro", "Serengeti"), col = c("red", "blue"), lty = 1)
```



Cabe destacar que para dibujar la regresión por áreas, hemos ajustado un modelo dónde se permite la interacción entre la edad y el área para permitir que la pendiente de regresión cambie según el área. Si no lo hicieramos, las líneas aparecerían como paralelas tal que así:

```r
# Predict proportion black for each area at each age
pred_N_noi <- predict(lmod_male, newdata = data.frame(age = age_seq, area = "N"))
pred_S_noi <- predict(lmod_male, newdata = data.frame(age = age_seq, area = "S"))

# Plot regression lines for each area
plot(data2_split_sex$M$age, data2_split_sex$M$prop.black, xlab = "Age", ylab = "Proportion Black")
lines(age_seq, pred_N_noi, col = "red")
lines(age_seq, pred_S_noi, col = "blue")
legend("bottomright", legend = c("Ngorongoro", "Serengeti"), col = c("red", "blue"), lty = 1)
```



(d) Predicción de la edad de una leona

No, ninguno de los modelos ajustados hasta el momento serviría para predecir la edad de una leona según su proporción de pigmentación. Hasta ahora hemos usado la pigmentación de la nariz como variable respuesta, que era explicada por la edad, área y sexo del león. No es posible simplemente "revertir" el modelo, necesitaríamos estimar un nuevo modelo dónde la variable respuesta fuera la edad, y la variable predictora fuera el color de la nariz.

El modelo que proponen en el artículo, utilizan la función arcsin(sqrt) para transformar la proporción de negro en la nariz, y hacerla más simétrica y adecuada para el estudio estadístico.

```
library(stats)

# Aplicamos la transformación sólo a
data2_split_sex$F$prop.black.transformed = asin(sqrt(data2_split_sex$F$prop.black))

lmod_age = lm(age ~ prop.black.transformed, data = data2_split_sex$F)

# Compute the predicted age on the transformed scale

# Define the proportion black for which to make the prediction
prop_black = 0.5
prop_black_transformed <- asin(sqrt(prop_black))

# Compute the predicted age and confidence intervals
predicted_age <- predict(lmod_age, newdata = data.frame(prop.black.transformed = prop_black_transformed
ci_95 <- predict(lmod_age, newdata = data.frame(prop.black.transformed = prop_black_transformed), inter
```

```r
ci_75 <- predict(lmod_age, newdata = data.frame(prop.black.transformed = prop_black_transformed), interv
ci_50 <- predict(lmod_age, newdata = data.frame(prop.black.transformed = prop_black_transformed), interv

# Compute se
summary_lmod_age <- summary(lmod_age)
se_predicted_age <- summary_lmod_age$sigma * sqrt(1 + 1/nrow(data2_split_sex$F) + (prop_black_transforme

# Create a data frame with the predicted age and confidence intervals
result_table <- data.frame("Proportion black" = prop_black,
                           "Estimated age in years" =paste(round(predicted_age,2), "(", round(se_predict
                           "95% CI" = paste(round(ci_95[2],2), round(ci_95[3],2), sep = "-"),
                           "75% CI" = paste(round(ci_75[2],2), round(ci_75[3],2), sep = "-"),
                           "50% CI" = paste(round(ci_50[2],2), round(ci_50[3],2), sep = "-"))

library(knitr)

new_names = c("Proportion black", "Estimated age in years (s.e.)", "95% p.i.", "75% p.i.", "50% p.i.")

names(result_table) <- new_names

# Create a table using the kable function from the knitr package
kable(result_table, format = "markdown")
```

Proportion black	Estimated age in years (s.e.)	95% p.i.	75% p.i.	50% p.i.
0.5	5.71 (1.62)	2.5-8.91	3.84-7.57	4.61-6.8

Se o standard error es una medida de la variabilidad de los errores de predicción de la variable dependiente a partir de las variables independientes. Se calcula como la desviación estándar de los residuos de la regresión (diferencias entre los valores predichos y los valores observados) dividida por la raíz cuadrada del número de observaciones.En resumen, el error estándar indica la precisión de las predicciones de la variable dependiente y es una medida importante para evaluar la calidad de un modelo de regresión lineal. Un error estándar más bajo indica una mayor precisión en las predicciones del modelo.

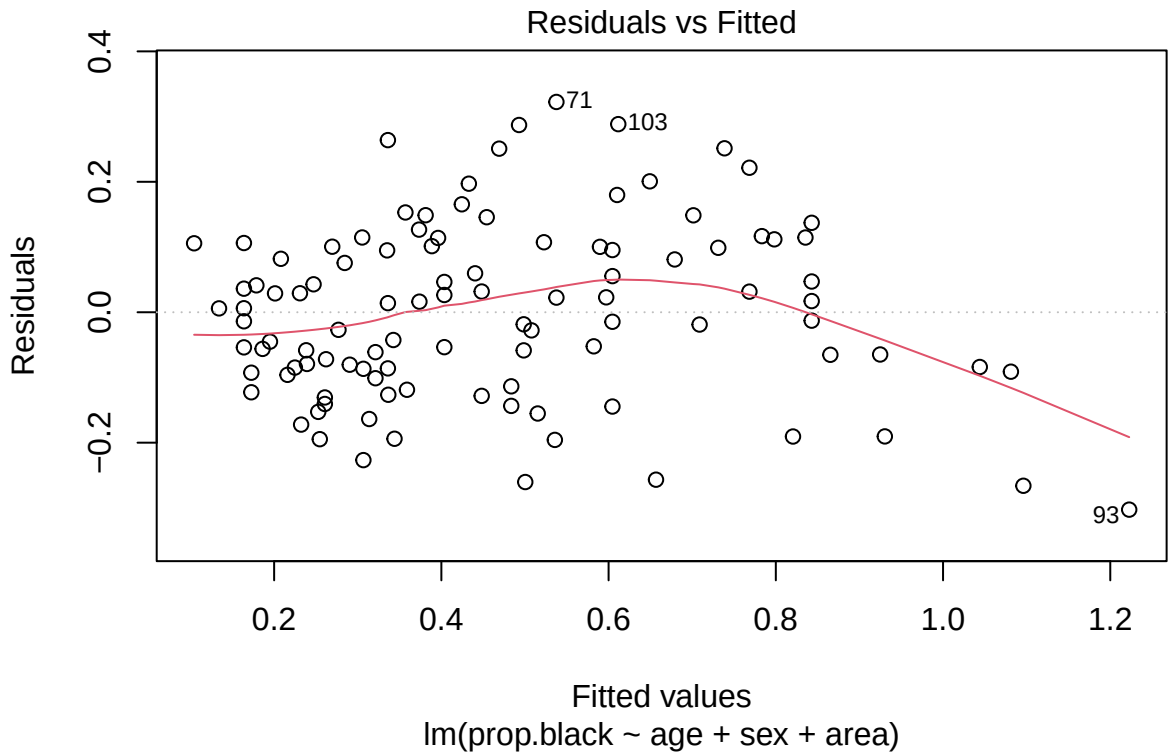
Ejercicio 3

(a) Gauss-Markov y condiciones del modelo re regresión

Las hipótesis de Gauss-Markov son:

- Linealidad: La relación entre la variable dependiente y las independientes debe ser lineal.
- Independencia de errores
- Homocedasticidad: La variancia de los errores o residuos debe ser constante entre todas las variables.
- Normalidad de errores
- No correlación de las variables independientes
- Media condicional de 0: El valor esperado de los errores debe ser 0 para todas las variables independientes

```r
# Load required packages
library(ggplot2)

# Residuals vs. Fitted plot
plot(lmod_all, which = 1)
```



Linealidad

```
# Create dummy variables for sex and area
dummy_sex <- model.matrix(~ sex, data = data2)
dummy_area <- model.matrix(~ area, data = data2)

data2 = cbind(data2, dummy_sex)
data2 = cbind(data2, dummy_area)

# Add quadratic terms for age, sex, and area to lmod_all
lmod_quad <- lm(prop.black ~ age + I(age^2) + dummy_sex[, 2] + I(dummy_sex[, 2]^2) + dummy_area[, 2] + I

# Perform an F-test to compare lmod_all and lmod_quad
anova(lmod_all, lmod_quad)
```

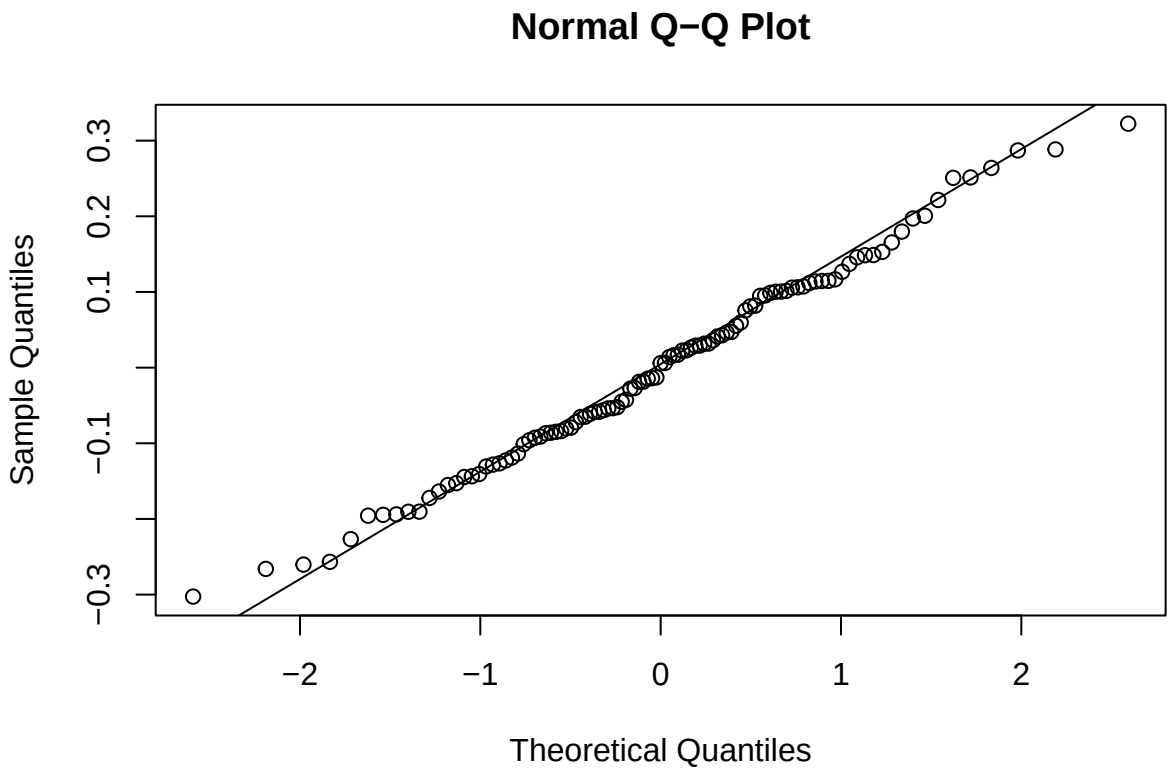
```
## Analysis of Variance Table
##
## Model 1: prop.black ~ age + sex + area
## Model 2: prop.black ~ age + I(age^2) + dummy_sex[, 2] + I(dummy_sex[,
##     2]^2) + dummy_area[, 2] + I(dummy_area[, 2]^2)
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    101 1.8872
## 2    100 1.5756  1   0.31152 19.771 2.265e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Observando el gráfico, vemos que no se acaba de cumplir linealidad. Es normal en modelos cuya variable dependiente es una proporción que sigan un patrón sigmoideo, tal y como observamos en el gráfico. Para acabar de testear linealidad, creamos un modelo añadiendo las variables cuadráticos y realizamos una comparación

entre modelos. Como el pvalor de la anova es significativo (pv<0.05), determinamos que la transformación cuadrática mejora el modelo, y por tanto hay evidencia de no-linealidad.

Nota: Al tener variables factoriales (área y sexo) hemos realizado un previo dummy coding a la generación del modelo cuadrático.

```
# Normality of residuals
qqnorm(resid(lmod_all))
qqline(resid(lmod_all))
```



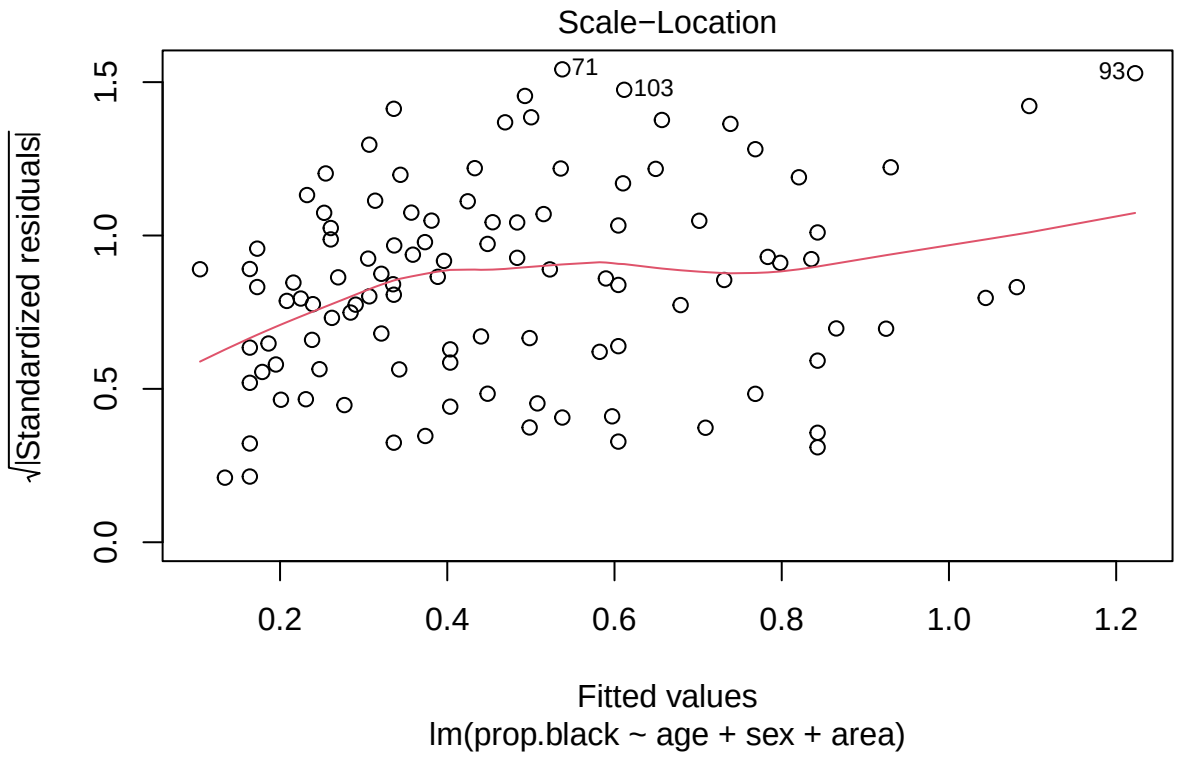
Normalidad

```
# H0: follows normality H1: does not follow normality
shapiro.test(resid(lmod_all))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lmod_all)
## W = 0.9909, p-value = 0.7072
```

Tanto el test de Shapiro como el qqplot nos indican que los residuos siguen una distribución normal. Podemos estar seguros porque aceptamos la hipótesis nula del test (pv = 0.7) y en el gráfico los valores siguen que manera casi perfecta la línea recta.

```
# Scale-Location plot
plot(lmod_all, which = 3)
```



Homocedasticidad

```
# Load required package
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'lmtest'
```

```
## The following object is masked from 'package:VGAM':
##
##     lrtest
```

```
# Perform Breusch-Pagan test
bptest(lmod_all)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmod_all
## BP = 10.171, df = 3, p-value = 0.01717
```

```
summary(lm(sqrt(abs(residuals(lmod_all))) ~ fitted(lmod_all)))
```

```
##
## Call:
## lm(formula = sqrt(abs(residuals(lmod_all))) ~ fitted(lmod_all))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.237971 -0.086909  0.004351  0.078468  0.249834
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.25904    0.02432   10.65   <2e-16 ***
## fitted(lmod_all)  0.10965    0.04587    2.39   0.0186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1157 on 103 degrees of freedom
## Multiple R-squared:  0.05256,    Adjusted R-squared:  0.04336
## F-statistic: 5.714 on 1 and 103 DF,  p-value: 0.01865
```

Parece ser que el modelo no cumple homocedasticidad. En el gráfico podemos ver como los valores no acaban de tener una forma rectangular, ya que aumenta su dispersión a medida que aumenta el valor ajustado, dando una ligera forma de cono.

Por otro lado el test de Breusch-Pagan también nos indica que hay heterocedasticidad, ya que con pv = 0.01 aceptamos la hipótesis alternativa: la varianza de los residuos no es constante.

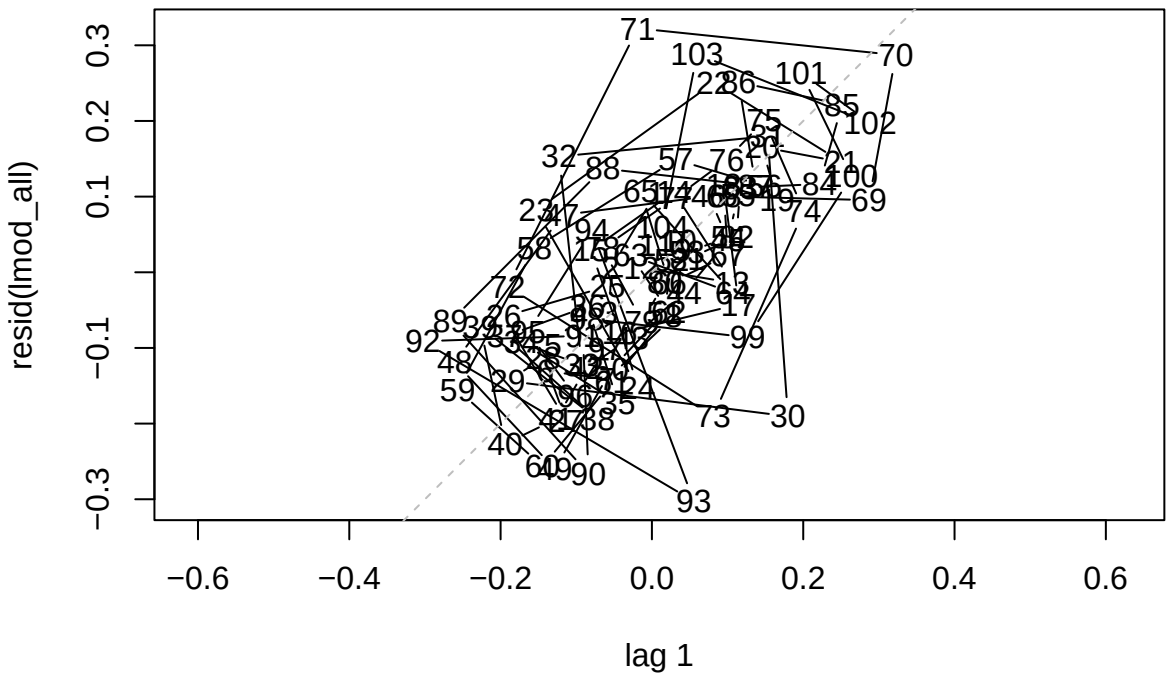
Adicionalmente, en el resumen del modelo ajustado a la raíz cuadrada de los residuos absolutos, observamos un pvalor significativo (pv = 0.01). Esto sugiere que su relación no es aleatoria, y por lo tanto, el modelo original viola homocedasticidad.

```
# Load the lmtest package
library(lmtest)

# Perform the Durbin-Watson test on lmod_all
#H0: No hay autocorrelación H1: Hay correlación
dwtest(lmod_all)
```

Independencia de errores

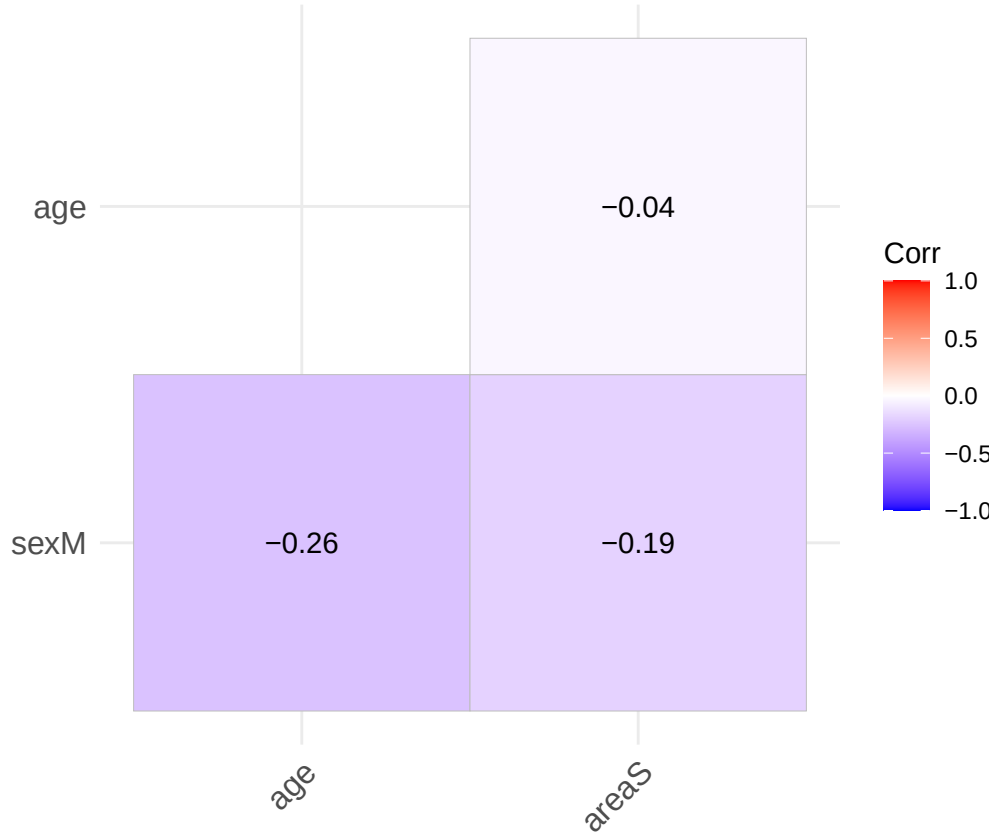
```
##
##  Durbin-Watson test
##
## data:  lmod_all
## DW = 0.82096, p-value = 8.419e-11
## alternative hypothesis: true autocorrelation is greater than 0
```

```
# Load the graphics package
library(graphics)

# Create a lag plot of the residuals from lmod_all
lag.plot(resid(lmod_all))
```



Según el test de Durbin-Watson, hay evidencia de correlación de residuos (pv < 0.05). Si miramos el gráfico, también determinados que existe correlación de residuos, pues los puntos no se reparten equitativamente sobre la línea horizontal y=0.

```
library(ggcorrplot)

# Compute correlation matrix of independent variables
cor_mat <- cor(data2[, c("age", "sexM", "areaS")])

# Create correlation plot
ggcorrplot(cor_mat, hc.order = TRUE, type = "lower", lab = TRUE)
```



Correlación de variables

```
# Compute correlation and p-value between age and dummy_sex
cor.test(data2$age, data2$sexM)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data2$age and data2$sexM
## t = -2.7308, df = 103, p-value = 0.007434
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.43007820 -0.07173848
## sample estimates:
##        cor
## -0.2598311
```

```
# Compute correlation and p-value between age and dummy_area
cor.test(data2$age, data2$areaS)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data2$age and data2$areaS
## t = -0.45041, df = 103, p-value = 0.6534
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2340136  0.1485910
## sample estimates:
```

```
##         cor
## -0.04433691
```

```
# Compute correlation and p-value between dummy_sex and dummy_area
cor.test(data2$areaS, data2$sexM)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data2$areaS and data2$sexM
## t = -1.9235, df = 103, p-value = 0.05718
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.364854816  0.005655778
## sample estimates:
##        cor
## -0.1862113
```

Existe correlación entre las variables sexo y edad. Parece ser que los machos tienen menor edad que las hembras (-0.26), y esta relación es significativa (pv = 0.007). El resto de variables no parecen estar correlacionadas.

```
# H0: Intercept pv = 0 H1: Distinto de 0
summary(lmod_all)
```

Media condicional de 0

```
##
## Call:
## lm(formula = prop.black ~ age + sex + area, data = data2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30265 -0.09116  0.00592  0.10049  0.32242
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.023324   0.044314   0.526   0.5998
## age          0.074464   0.004396  16.939   <2e-16 ***
## sexM        -0.068416   0.030662  -2.231   0.0279 *
## areaS        0.067473   0.034106   1.978   0.0506 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1367 on 101 degrees of freedom
## Multiple R-squared:  0.7713, Adjusted R-squared:  0.7645
## F-statistic: 113.5 on 3 and 101 DF,  p-value: < 2.2e-16
```

Podemos ver que el pvalor del intercepto no es significativo, y por tanto aceptamos la hipótesis nula de que la media condicional de los errores es 0. Es importante que la media condicional sea 0, ya que si no el modelo sobreestimaría o subestimaría los valores reales de la población, llevando así a predicciones sesgados.

```
# Leverage

# Calculate the threshold based on the rule of thumb
```

```r
p <- ncol(model.matrix(lmod_all)) - 1
n <- nrow(data2)
threshold <- 2*p/n

# Calculate the number of observations with hat values above the threshold
hatv <- hatvalues(lmod_all)
num_outliers <- sum(hatv > threshold)

# Print the number of outliers
cat("Number of outliers according to hatvalues:", num_outliers)
```

Observaciones inusuales

```
## Number of outliers according to hatvalues: 25
```

```r
# Half normal plot with hatvalues
library(faraway)
```

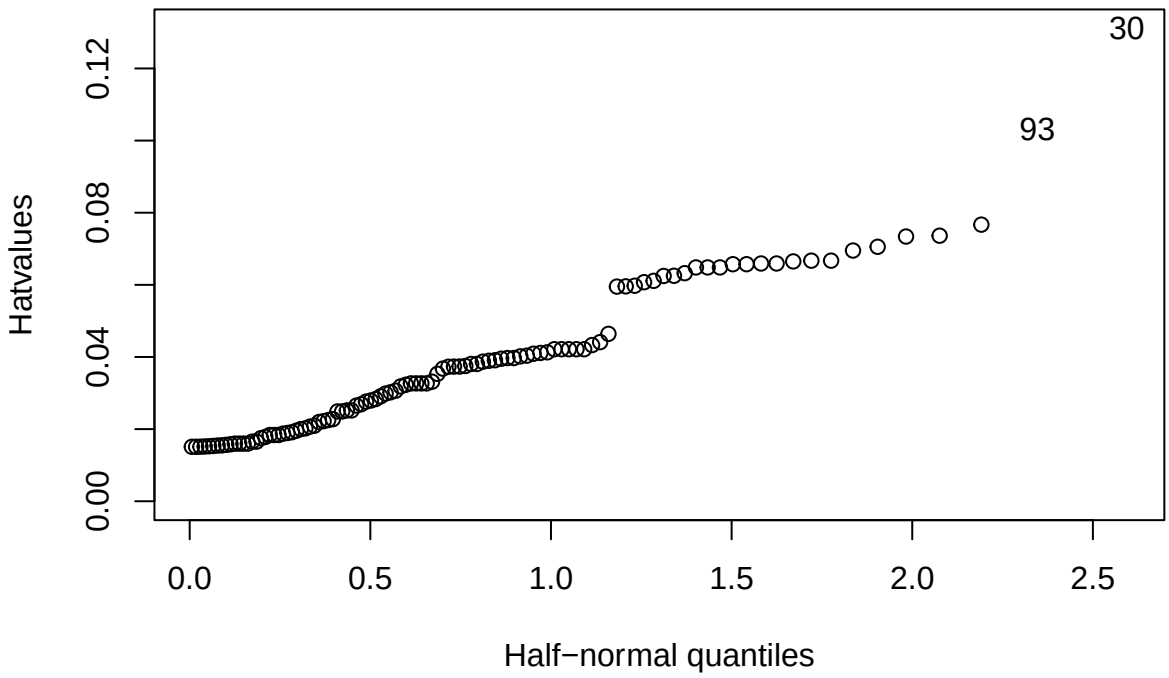
```
##
## Attaching package: 'faraway'
```

```
## The following object is masked from 'package:rpart':
##
##     solder
```

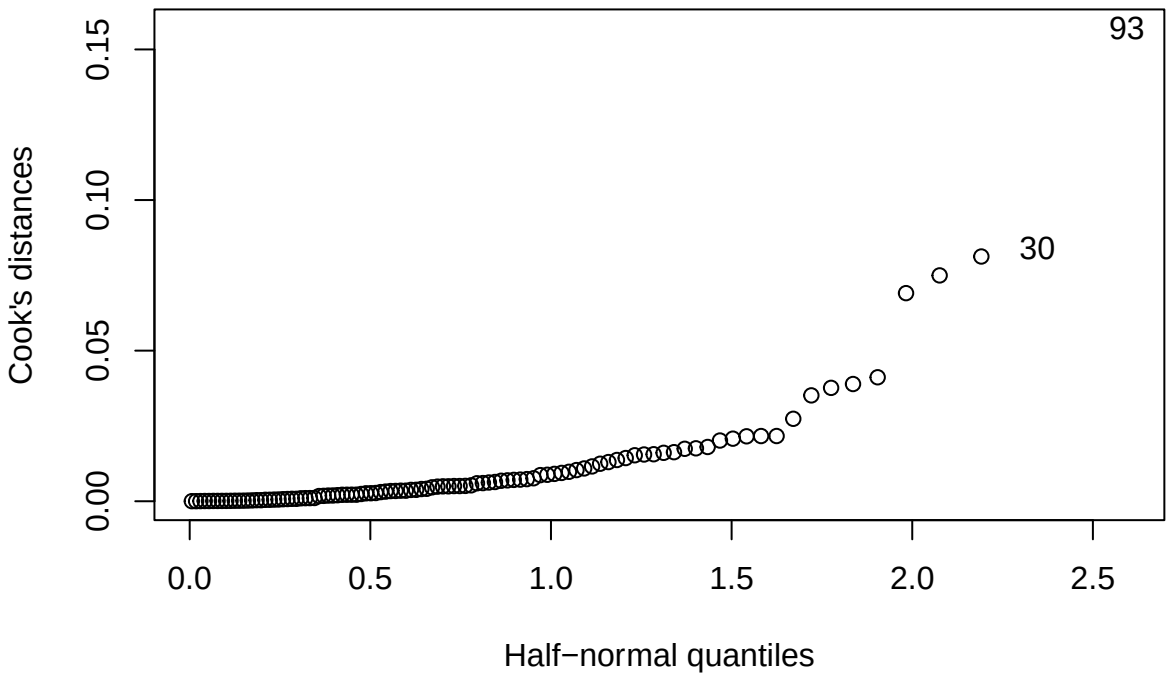
```
## The following objects are masked from 'package:VGAM':
##
##     hormone, logit, pneumo, prplot
```

```
## The following objects are masked from 'package:car':
##
##     logit, vif
```

```r
halfnorm(hatv, ylab="Hatvalues")
```




```
# Half normal plot with cook's distance
cook = cooks.distance(lmod_all)
halfnorm(cook, ylab="Cook's distances")
```



En estadística, un punto influyente es un punto que tiene un gran efecto en la estimación de los coeficientes de regresión de un modelo. Vemos que en este estudio existen múltiples puntos influyentes con valores atípicos de las variables dependientes (90, 93, 30...). Si usamos la regla general de 2p/n (siendo p el número de variables independientes del modelo y n el tamaño muestral), obtenemos 25 puntos Mirando el gráfico Half-normal de hatvalues, podemos ver que efectivamente son 25 los puntos que se desvían de la recta principal.Un número tan elevado de puntos influyentes puede significar que el modelo no se ajusta bien a los datos.

Por otro lado, calculamos la distancia de cook para cada uno de los puntos. Mientras que el hatvalue mide cuán desviado está un punto del centro de los datos en cuanto a variables dependientes, la distancia de cook mide el efecto que tendría eliminar el punto en el modelo. En este nuevo gráfico, no parece que los 25 puntos influyentes influyan por igua al modelo, son aproximadamente 10 los que más influyen.

```
# Outliers

stud = rstudent(lmod_all)
maxo = stud[which.max(abs(stud))]
# Calculate Bonferroni critical value
bonf_crit <- qt(0.05/(2*n), df = lmod_all$df.residual)

cat("\n")
```

```
if (maxo > abs(bonf_crit)) {
  cat("The point is an outlier")
} else {
  cat("The point is not an outlier")
}
```

```
## The point is not an outlier
```

En estadística, un outlier es un punto significativamente distinto al resto. En este estudio parece que no tenemos outliers. Un punto influyente no tiene porque ser un outlier y un outlier no tiene porque ser un punto influyente. En el artículo original contaban con algún outlier que fue eliminado, como nuestros datos están extrapolados del artículo no contienen outliers.

Conclusión

(b) Variable respuesta proporción

El mayor problema que conlleva el hecho de que la variable dependiente sea una proporción, es que nuestro modelo podría predecir valores que no son posibles (por debajo de 0 o encima de 1). Adicionalmente, las relaciones de estas proporciones no siguen una linea recta, si no una sigmoidal (con forma de "S"). Es común también que este tipo de modelos con proporciones no presenten heterocedasticidad y normalidad de errores.

Para mejorar el ajuste de los datos existen múltiples opciones. Se puede realizar una regresión beta o regresión de respuesta fraccional. En la regresión beta los valores predichos se encuentran entre 0 y 1 (no incluidos).

https://raymondltremblay.github.io/ANALITICA/TF7_Regresion_beta.html

Otra aproximación que se puede realizar, es una transformación de los datos. Se pueden realizar ciertas transformaciones (como hacer la raíz cuadrada), para que los datos se alejen de los extremos (0 y 1). Con valores entre 0.2 y 0.8 nuestro modelo no llegará a predecir valores fuera del rango entre 0 y 1.

En este caso en particular, se puede aplicar esta transformación aplicando la raíz cuadrada. Por otro lado, se podría medir la cantidad de negro en la nariz de los leones por milímetro cuadrado, y así esta variable dejaría de ser una proporción y no presentaría estos problemas.

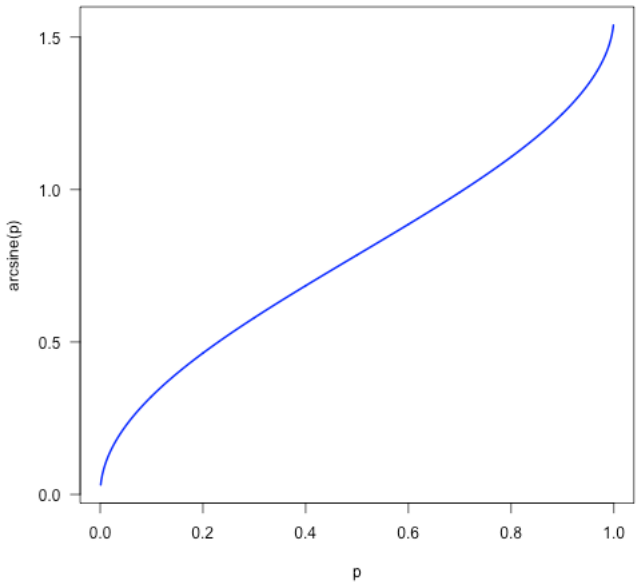
(c) Transformación de la variable

(d) Ajuste modelo transformado

(e) Discusión uso arcsin

Tal y como comentamos en el apartado 2d, el modelo que proponen en el artículo utiliza la función arcsin(sqrt) para transformar la proporción de negro en la nariz, y hacerla más simétrica y adecuada para el estudio estadístico. Con esta transformación, los valores medios de proporción (0.3-0.7) siguen una distribución normal, gracias a esto se puede realizar una regresión lineal.

Se define de la siguiente manera: arcsin(x) = sinˆ(-1)(x). La transformación sqrt hace que los valores más extremos (cercanos a 0 y 1), se desplacen hacia el centro, ayudando así a que caigan en la zona de máxima linealidad.



ANEXO