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 el intervalo al 90% de confianza no se incluye el 0, es por ello que podemos deducir que el pvalor sea significativo a un nivel de confianza de 0.1, ya que como hemos explicado medimos si el parámetro es distinto a 0. En el caso del intervalo de confianza al 95% sí lo incluye por un margen muy pequeño.

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.

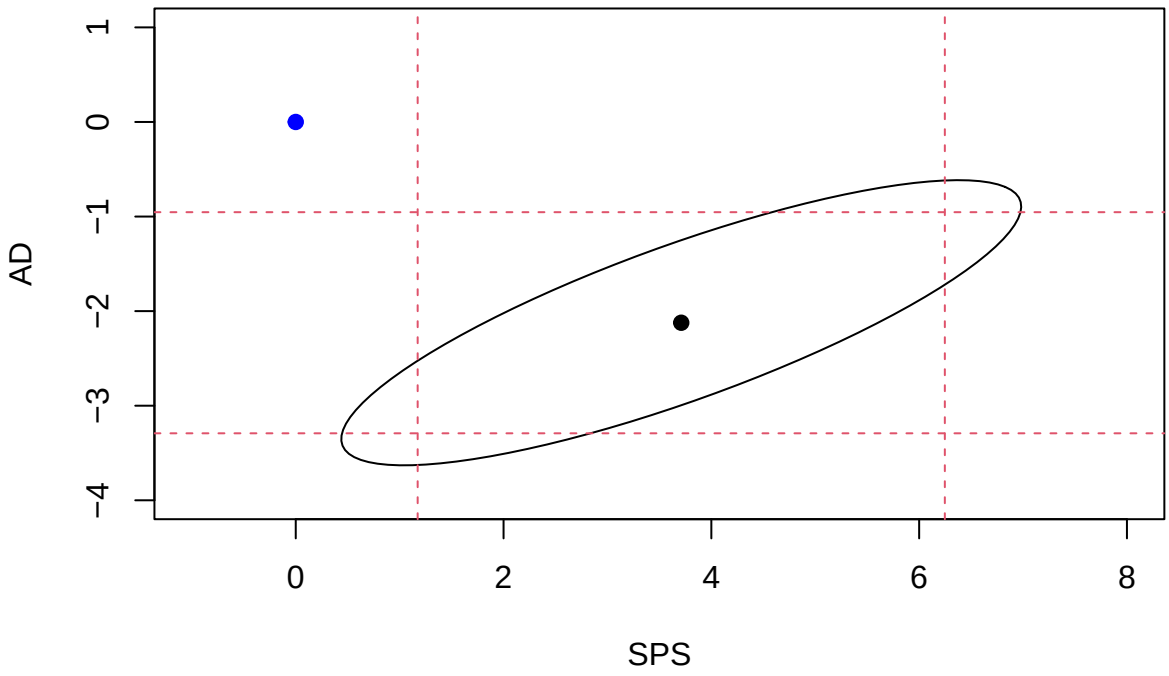
Por otro lado, en el modelo reducido todas las variables explican de manera significa la variable respuesta. Además, el valor de R ajustado es similar en ambos modelos (0.93), este valor indica el porcentaje de la variable respuesta que es explicado por el modelo.

(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
```

```
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 no 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

```
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)
```

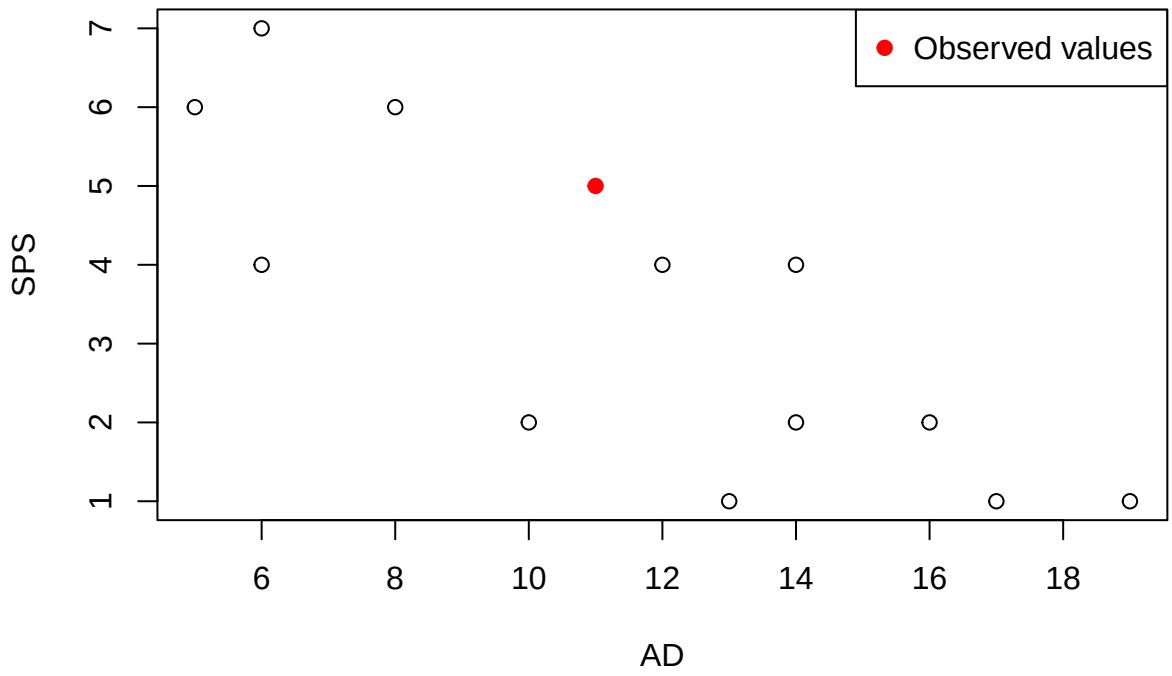
```
## Min AD: 5  Max AD: 19  Observed value: 11
```

```
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
```

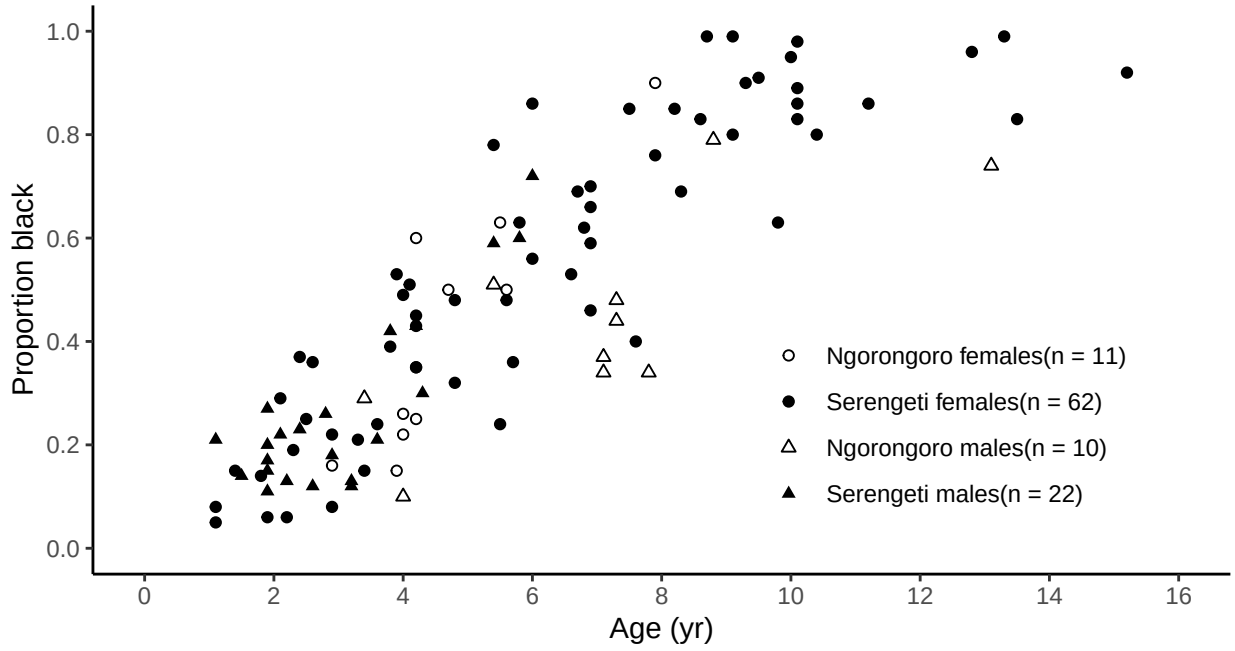
```
## The following object is masked from 'package:car':
##
##     recode
```

```
## 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
## -0.32206 -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
```

```
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
## -0.15771 -0.08862 -0.02669  0.06724  0.25969
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.32531    0.14975  -2.172   0.0443 *
## age          0.15848    0.03112   5.092 9.04e-05 ***
## sexM         0.35005    0.19249   1.819   0.0866 .
## age:sexM    -0.10024    0.03498  -2.866   0.0107 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1293 on 17 degrees of freedom
## Multiple R-squared:  0.6991, Adjusted R-squared:  0.6461
## F-statistic: 13.17 on 3 and 17 DF,  p-value: 0.000108
```

```
summary(lmod_S)
```

```
##
## Call:
## lm(formula = prop.black ~ age * sex, data = data2_split_area$S)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30712 -0.08717  0.02071  0.09153  0.33028
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.074893   0.035288   2.122   0.0369 *
## age          0.075804   0.004905  15.454   <2e-16 ***
## sexM        -0.130055   0.076583  -1.698   0.0934 .
## age:sexM     0.031156   0.021058   1.480   0.1429
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1307 on 80 degrees of freedom
## Multiple R-squared:  0.8116, Adjusted R-squared:  0.8046
## F-statistic: 114.9 on 3 and 80 DF,  p-value: < 2.2e-16
```

En el modelo con todas las variable, podemos observar que el sexo no influye significativamente sobre la variable respuesta (proporción de negro en la nariz), con un pvalor = 0.93.

Al separar por área, seguimos obteniendo que el sexo no influye de forma significativa para ninguna de las dos

áreas, al menos para un nivel de significación del 0.05. Si tomamos un nivel de 0.1, entonces en ambos el sexo pasa a ser significativo. En la población Ngorongoro los machos tienen la nariz más oscura que las hembras (coef sexM = 0.35), mientras que en los Serengeti los machos tienen la nariz más clara (coef sexM = -0.13).

Es importante estudiar la interacción entre la edad y el sexo. En el caso de los Serengeti ésta no es significativa. Y por lo tanto, los resultados se alinean con los del artículo (no hay efecto del sexo en el color de la nariz de los leones Serengeti). En cambio, en el caso de los Ngorongoro sí lo es (pv = 0.01), esto quiere decir que según la edad de los leones, sí observamos diferencias en cuanto al sexo y el color de la nariz. Nuevamente los hallazgos coindicen con los del artículo, pues los machos Ngorongoro tienen narices más claras que las hembras a ciertas edades (coef age:sexM = -0.1)

(c) Modelo leones macho

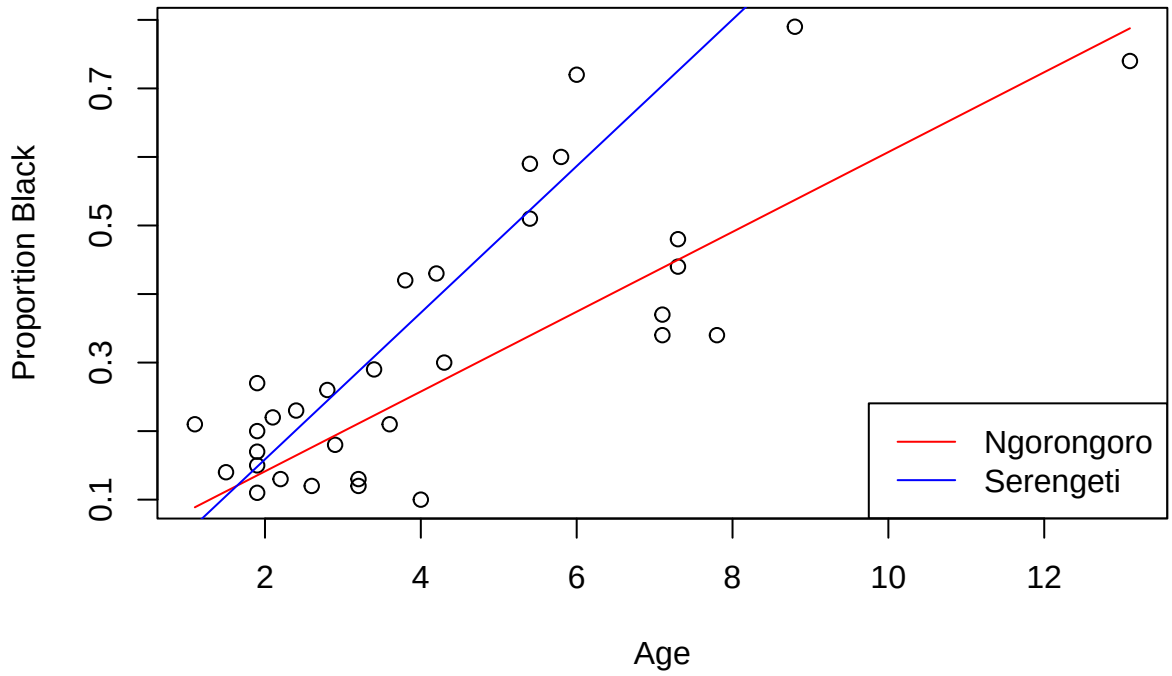
```
data2_split_sex = split(data2, f = data2$sex)
lmod_male = lm(prop.black ~ age * area, data = data2_split_sex$M)
summary(lmod_male)
```

```
##
## Call:
## lm(formula = prop.black ~ age * area, data = data2_split_sex$M)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16711 -0.08083  0.01880  0.05576  0.25274
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02474    0.10178   0.243  0.80971
## age          0.05824    0.01343   4.335  0.00017 ***
## areaS       -0.07990    0.11646  -0.686  0.49831
## age:areaS    0.04872    0.02171   2.244  0.03292 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1088 on 28 degrees of freedom
## Multiple R-squared:  0.7286, Adjusted R-squared:  0.6995
## F-statistic: 25.05 on 3 and 28 DF,  p-value: 4.408e-08
```

Para los machos, no existen diferencias significativas según el área (pv = 0.49). Similarmente al apartado anterior, sí existe significación en la interacción de la edad y el área (pv = 0.03). Esto quiere decir, que según la edad del león sí encontraremos diferencias por área. En específico, los machos Serengeti tienen la nariz más oscura que los Ngorongoro (coef age:areaS = 0.04). Podemos ver este efecto más claramente en el siguiente gráfico. Para un león de 2 años no existen diferencias según el área, pero para un león de 8 años sí lo hay, siendo más oscura la de los Serengeti (mayor proporción de negro).

```
# 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, newdata = data.frame(age = age_seq, area = "N"))
pred_S <- 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, col = "red")
```

```
lines(age_seq, pred_S, 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))
```

```r
# 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), interv
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

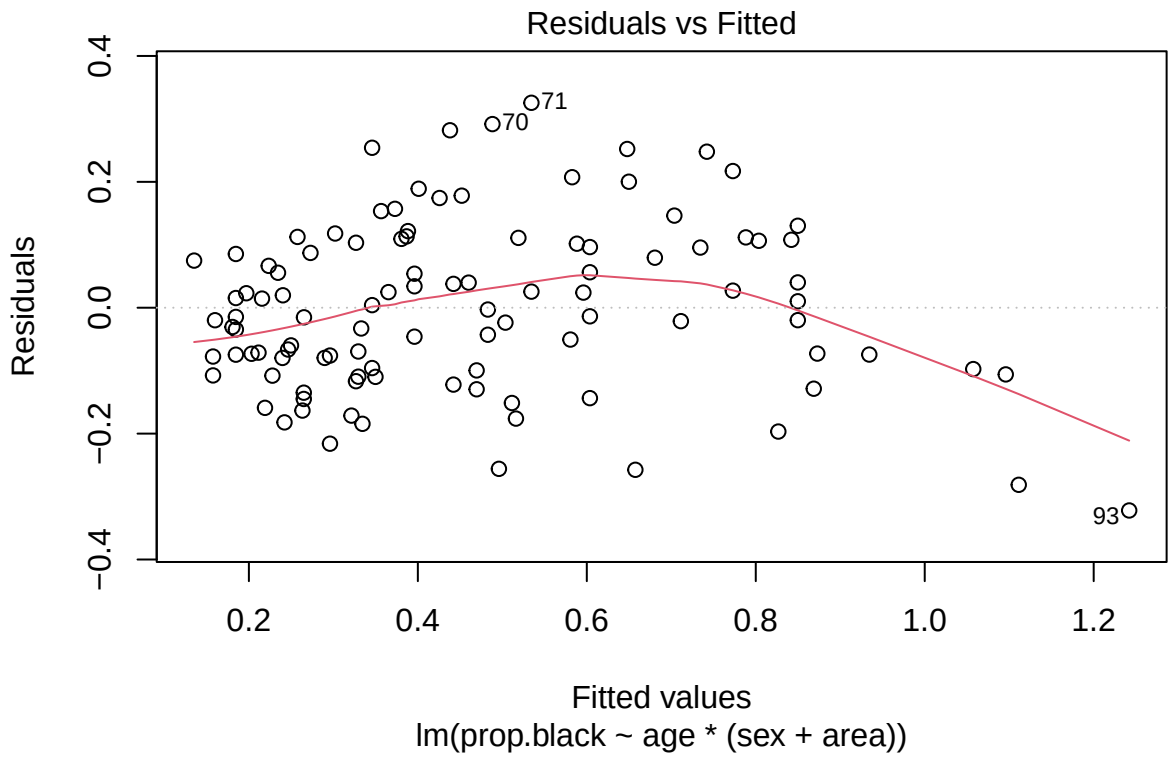
(a) Gauss-Markov y condiciones del modelo re regresión

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)
```

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



Linealidad

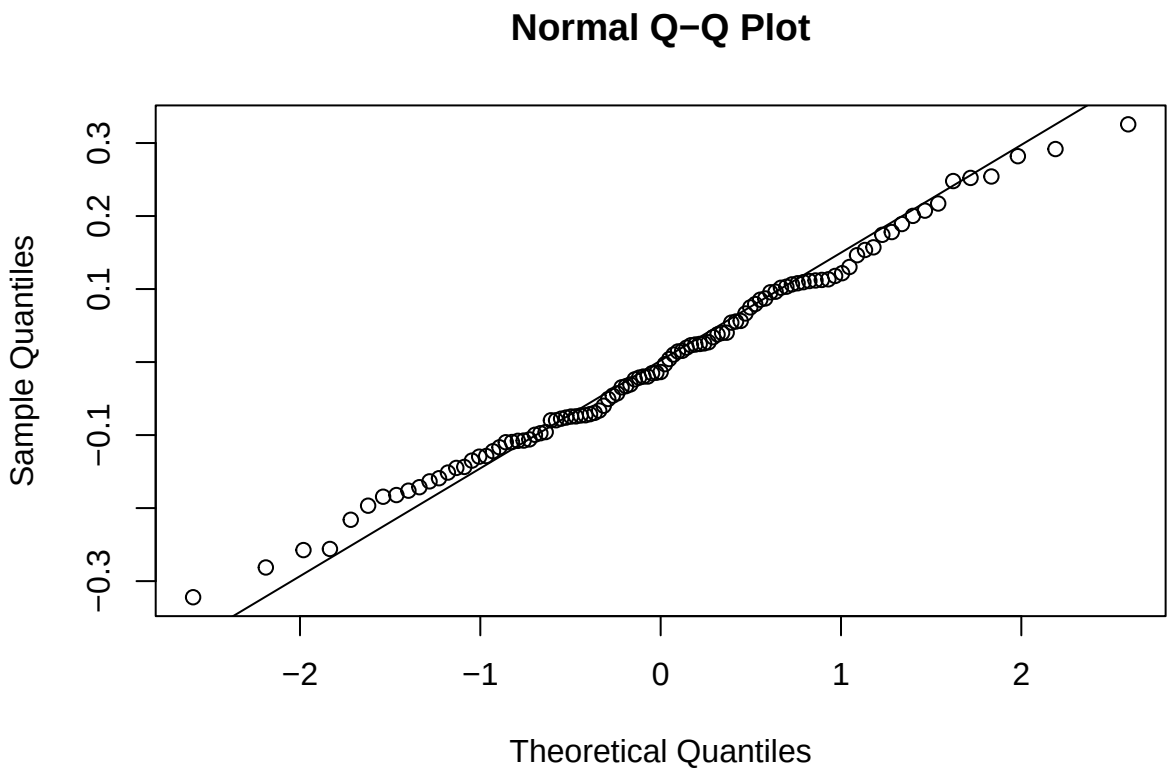
```r
# Create dummy variables for sex and area
dummy_sex <- model.matrix(~ sex, data = data2)
dummy_area <- model.matrix(~ area, data = data2)

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 * (dummy_sex + dummy_area) + I(age^2) * (I(dummy_sex^2) + I(dummy_area^

# 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 * (dummy_sex + dummy_area) + I(age^2) * (I(dummy_sex^2) +
##     I(dummy_area^2))
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1     99 1.8662
## 2     96 1.4942  3   0.37201 7.967 8.484e-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. Para acabar de testear linealidad, creamos un modelo añadiendo las variables cuadráticas 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))
```



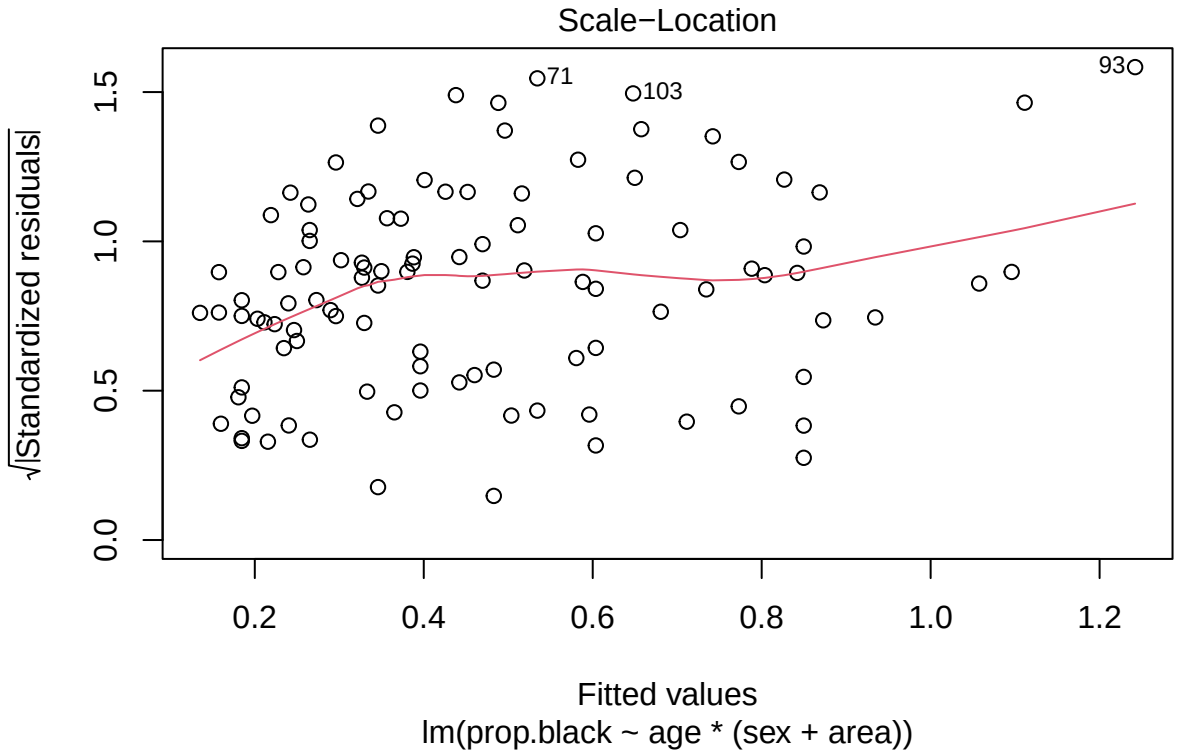
Normalidad

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

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lmod_all)
## W = 0.99248, p-value = 0.8337
```

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 = 0.83) y en el gráfico los valores siguen que manera bastante ajustada la recta.

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



Homocedasticidad

```r
# 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
```

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

```
##
##  studentized Breusch-Pagan test
##
```

```
## data:  lmod_all
## BP = 10.316, df = 5, p-value = 0.06677
```

```
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
## -0.255173 -0.089350  0.003797  0.096214  0.256041
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.25467    0.02478  10.276   <2e-16 ***
## fitted(lmod_all)  0.11205    0.04672   2.398   0.0183 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1181 on 103 degrees of freedom
## Multiple R-squared:  0.05289,    Adjusted R-squared:  0.04369
## F-statistic: 5.752 on 1 and 103 DF,  p-value: 0.01827
```

Parece ser que el modelo cumple homocedasticidad. En el gráfico podemos ver como los valores tienen una forma bastante rectangular, y casi no aumenta su dispersión a medida que aumenta el valor ajustado, aunque sí lo hace ligeramente dando una ligera forma de cono.

De manera similar, el test de Breusch-Pagan nos indica que hay homocedasticidad, ya que con pv = 0.06 aceptamos la hipótesis nula: la varianza de los residuos es constante.

Por otro lado, 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. Esta discrepancia puede deberse a múltiples motivos, ya que el test de Breush-Pagan hace ciertas asunciones como que los errores están normalmente distribuidos y tienen varianza constante.

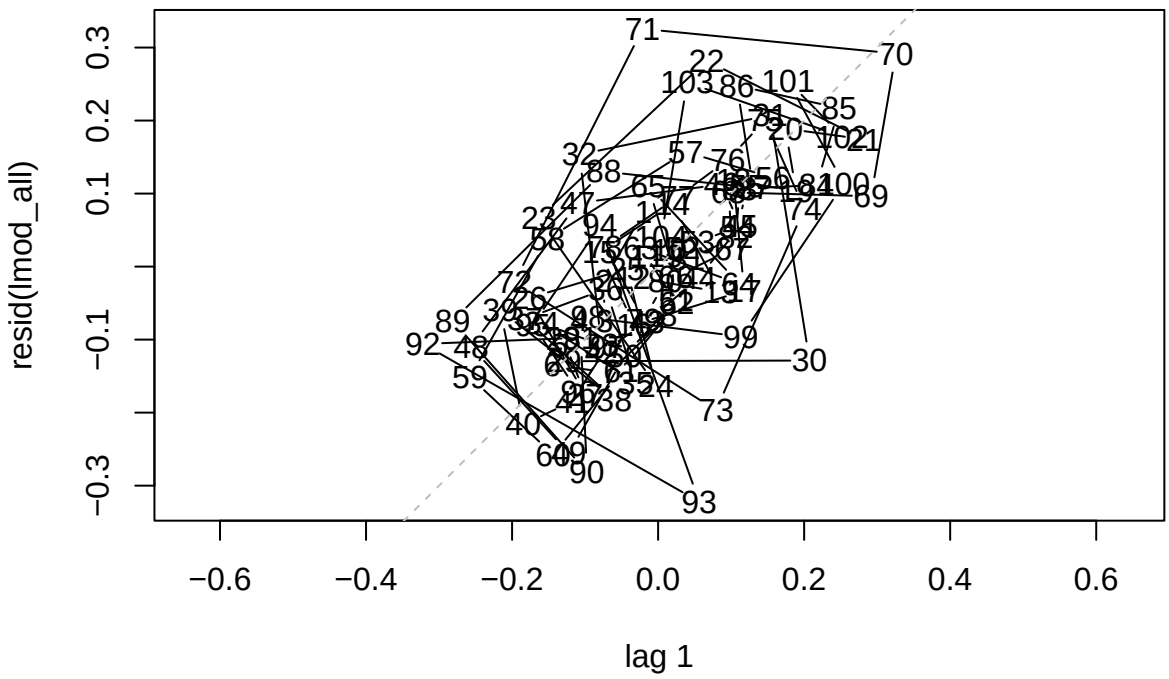
```
# Load the lmtest package
library(lmtest)

# 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 = 0.83845, p-value = 5.739e-11
## alternative hypothesis: true autocorrelation is greater than 0
```

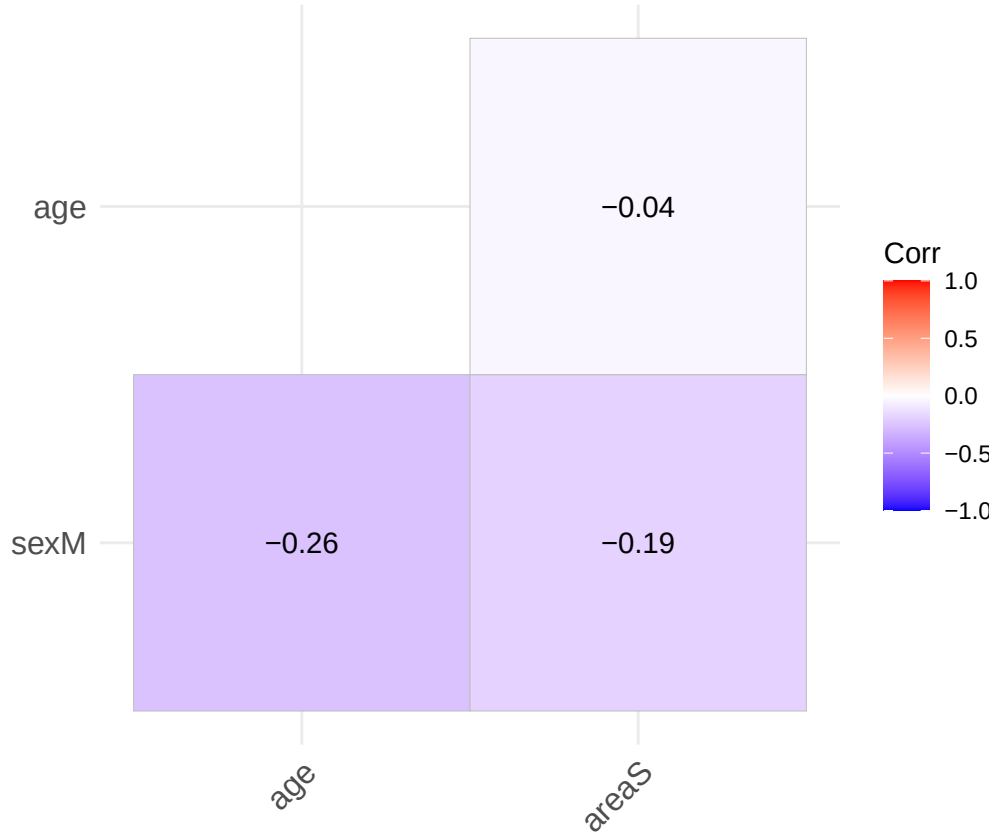
```
# Load the graphics package
library(graphics)

# 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
## -0.32206 -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
```

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 sesgadas.

```
# Leverage
```

```r
# Calculate the threshold based on the rule of thumb
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: 11
```

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

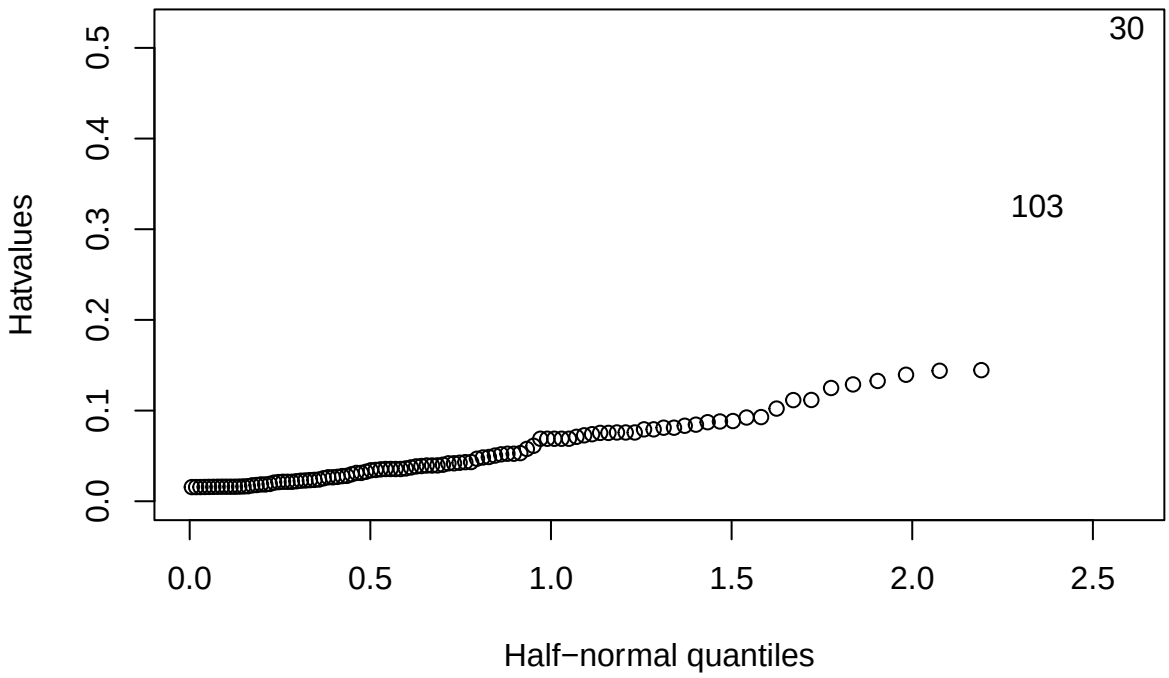
```
##
## Attaching package: 'faraway'
```

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

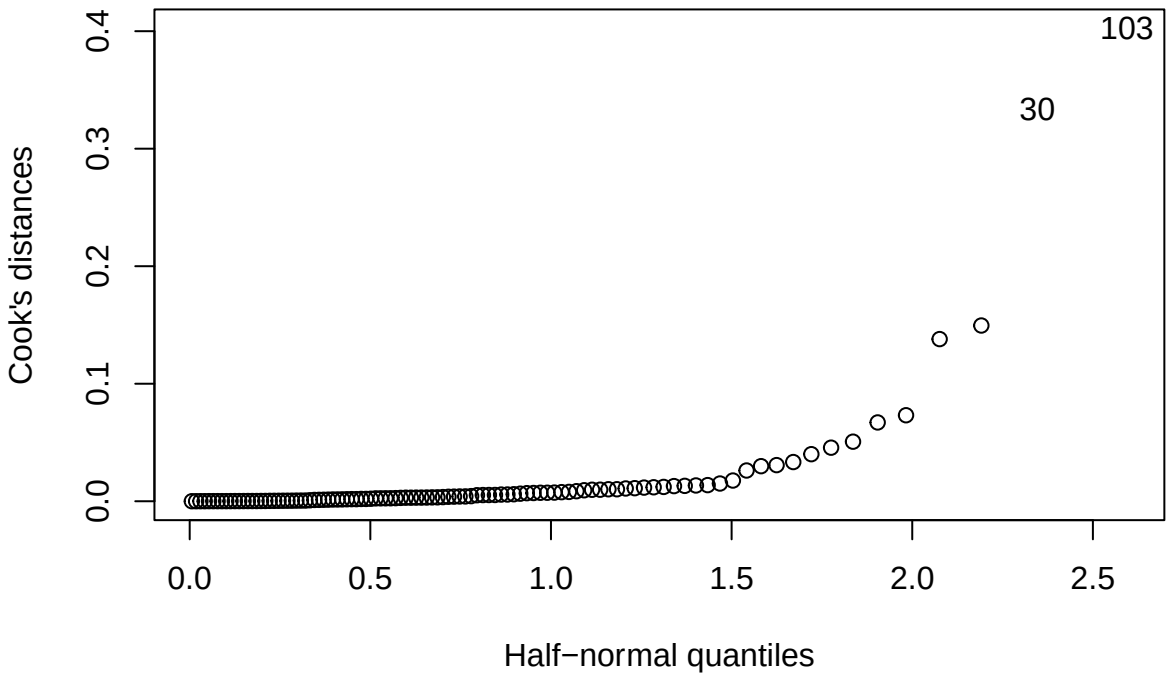
```
## The following objects are masked from 'package:VGAM':
##
##     hormone, logit, pneumo, prplot
```

```
## 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 11 puntos influyentes. Mirando el gráfico Half-normal de hatvalues, podemos ver que efectivamente son aproximadamente 11 los puntos que se desvían de la recta principal.

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, parece son 13 los puntos más influyentes.

```
# 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 A pesar que nuestro modelo cumple con muchas de las características, no podemos afirmar que sea un buen modelo. El motivo principal es la falta de linealidad. No podemos ajustar un buen modelo lineal a variables que no tienen una relación lineal. Sería adecuado realizar algún tipo de transformación para poder ajustar un modelo lineal, o bien ajustar un modelo no-lineal.

Cabe destacar que a pesar que el modelo no cumpla todas las asunciones, no significa que no nos proporcione información útil y buenas predicciones, pero es extremadamente importante interpretar los resultados con cautela y tener en mente las limitaciones del modelo.

(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 homocedasticidad 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).

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 debería llegar 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 dependiente

Dado que la variable respuesta es una proporción, y no acaba de ajustarse a una relación lineal, realizaremos una transformación arcsin(sqrt) a la variable respuesta. He escogido esta transformación antes que la logit, porque la transformación logit se suele usar en casos donde la proporción representa un resultado binario (ej: proporción de leones con nariz negra). Mientras que arcsin se suele usar para representar una variable continua (ej: proporción negra de la nariz).

```
# Comprobamos que los datos están entre 0 y 1, si no no podemos aplicar la transformación
min(data2$prop.black)
```

```
## [1] 0.05
```

```
max(data2$prop.black)
```

```
## [1] 0.99
```

```
# Aplicamos la transformación
prop.black.transformed = asin(sqrt(data2$prop.black))
# Rehacemos el modelo
lmod_all_transformed = lm(prop.black.transformed ~ age * (sex + area), data = data2)

summary(lmod_all_transformed)
```

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

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37787 -0.10177 -0.01389  0.10657  0.39591
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.220586   0.122869   1.795 0.075657 .
## age          0.094754   0.024796   3.821 0.000232 ***
## sexM         0.023239   0.082852   0.280 0.779691
## areaS        0.068191   0.123663   0.551 0.582588
## age:sexM    -0.023067   0.020074  -1.149 0.253298
## age:areaS   -0.004416   0.024657  -0.179 0.858234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1598 on 99 degrees of freedom
## Multiple R-squared:  0.7716, Adjusted R-squared:   0.76
## F-statistic: 66.87 on 5 and 99 DF,  p-value: < 2.2e-16
```

```r
library(knitr)

# Calculate the goodness-of-fit metrics
library(broom) # Usaremos la función augment para calcular los residuos (diferencia entre valores predi

R = summary(lmod_all)$r.squared
Rad = summary(lmod_all)$adj.r.squared
rmse = sqrt(mean(augment(lmod_all)$.resid^2))
aic = AIC(lmod_all)
R_t = summary(lmod_all_transformed)$r.squared
Rad_t = summary(lmod_all_transformed)$adj.r.squared
rmse_t = sqrt(mean(augment(lmod_all_transformed)$.resid^2))
aic_t = AIC(lmod_all_transformed)

# Create table
metrics_df <- bind_rows(
    augment(lmod_all) %>% summarise(Modelo = "Sin transformar", R_squared = R, Adj_R_squared = Rad, rms
    augment(lmod_all_transformed) %>% summarise(Modelo = "Arcsin(sqrt)", R_squared = R_t, Adj_R_squared
)

# Print the table
kable(metrics_df, format = "markdown")
```

Modelo	R_squared	Adj_R_squared	rmse	AIC
Sin transformar	0.7738200	0.7623968	0.1333169	-111.17838
Arcsin(sqrt)	0.7715525	0.7600147	0.1551784	-79.29063

En cuanto a significación del modelo no vemos ninguna diferencia. Ambos cuentan con un pvalor significativo. En cuanto a la R, ésta indica el porcentaje de variabilidad de la varianle dependiente que es explicado por el modelo. En ambos casos es 0.77, este resultado implica que aproximadamente un 77% de la proporción de negro en las narices de los leones es explicada por las variables y modelo usado. Es un poco pobre, ya que hay un 23% de variabilidad que no podemos explicar.

El rmse o error cuadrático medio calcula la cantidad de error entre los valores predichos por el modelo y los

valores reales observados. Por lo tanto, cuanto menor sea este error, mejor predice nuestro modelo los datos usados. En este caso, parece que el modelo sin transformar predice ligeramente mejor los datos.

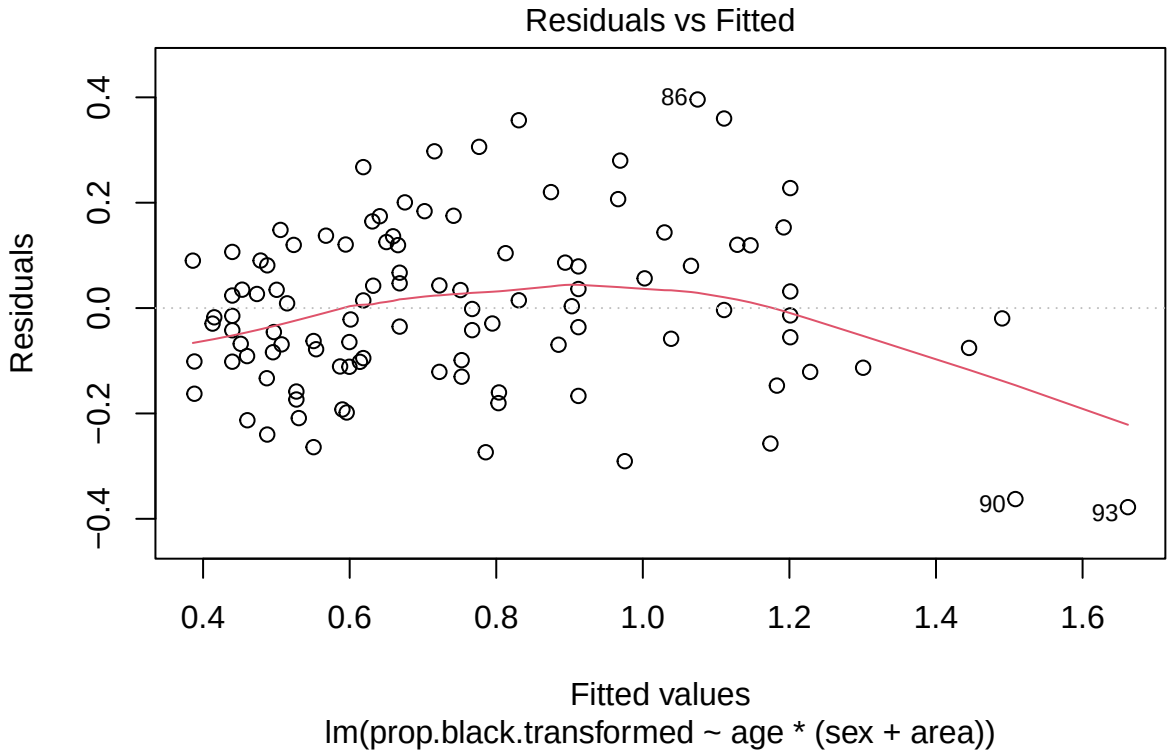
El AIC es una medida de la calidad relativa del modelo. Permite comparar modelos teniendo en cuenta su complejidad. Nuevamente, valores menores de AIC indican mejor ajuste, de manera que el modelo sin transformar se ajusta mejor a los datos que el transformado.

Tras estudiar estos parámetros, concluimos que el modelo sin transformar es mejor, puesto que es el más sencillo y se ajusta igual de bien o mejor que el modelo transformado.

(d) Diagnóstico rápido

Puesto que nuestra principal preocupación era la no-relación lineal entre las variables independientes y la dependiente, vale la pena estudiar si ésta ha mejorado.

```
# Load required packages
library(ggplot2)

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

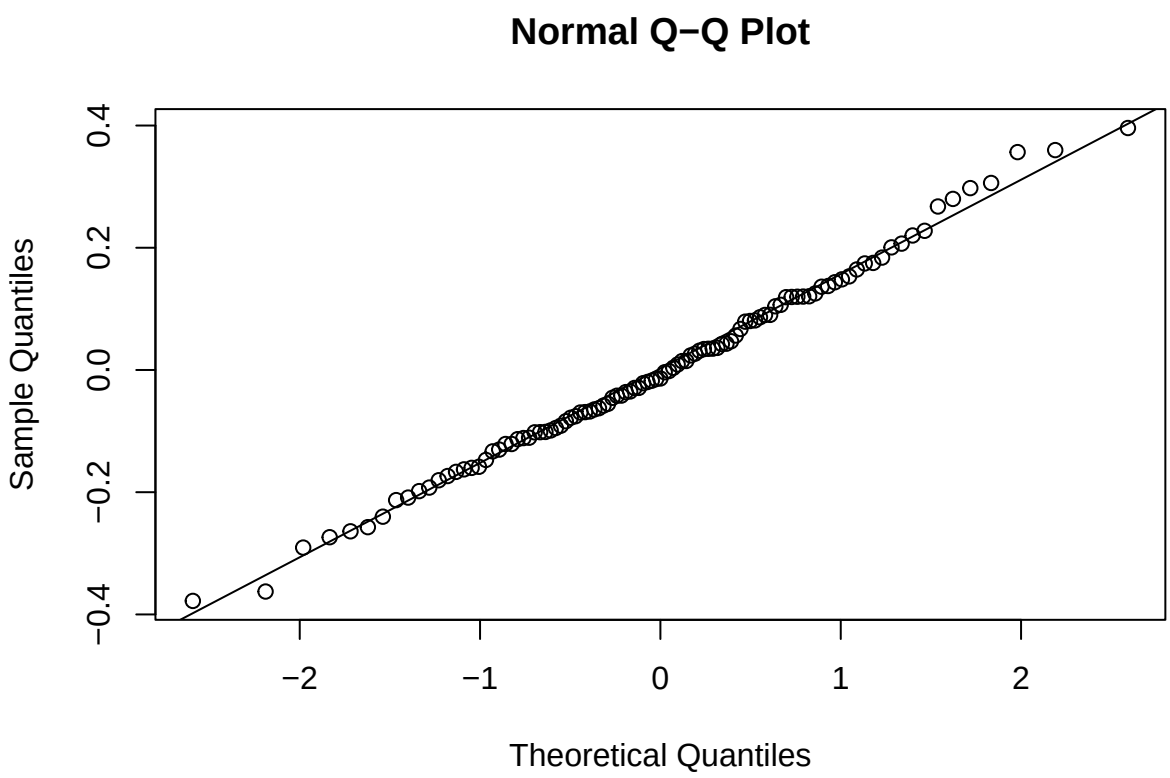



```
# Add quadratic terms for age, sex, and area to lmod_all
lmod_quad_transformed = lm(prop.black.transformed ~ age * (dummy_sex + dummy_area) + I(age^2) * (I(dummy

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

```
## Analysis of Variance Table
##
```

```
## Model 1: prop.black.transformed ~ age * (sex + area)
## Model 2: prop.black.transformed ~ age * (dummy_sex + dummy_area) + I(age^2) *
##     (I(dummy_sex^2) + I(dummy_area^2))
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     99 2.5284
## 2     96 2.0914  3   0.43703 6.6869 0.0003795 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

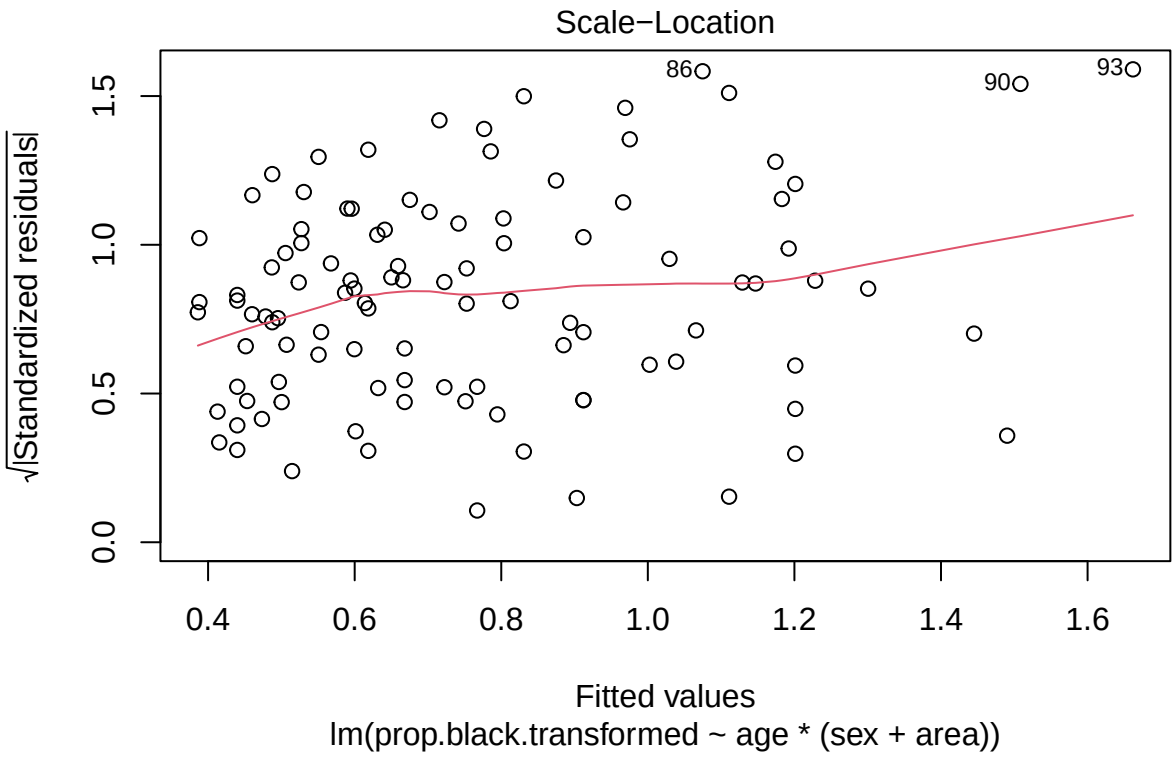
```
# Normalidad
# Normality of residuals
qqnorm(resid(lmod_all_transformed))
qqline(resid(lmod_all_transformed))
```




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

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(lmod_all_transformed)
## W = 0.99408, p-value = 0.9332
```

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




```
# Load required package
library(lmtest)
# Perform Breusch-Pagan test
bptest(lmod_all_transformed)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lmod_all_transformed
## BP = 11.571, df = 5, p-value = 0.04117
```

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

```
##
## Call:
## lm(formula = sqrt(abs(residuals(lmod_all_transformed))) ~ fitted(lmod_all_transformed))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.295560 -0.109175  0.004272  0.096188  0.276074
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.25610    0.03727   6.871    5e-10 ***
## fitted(lmod_all_transformed)   0.09030    0.04601   1.963   0.0524 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1344 on 103 degrees of freedom
## Multiple R-squared:  0.03605,    Adjusted R-squared:  0.02669
## F-statistic: 3.852 on 1 and 103 DF,  p-value: 0.05239
```

Parece ser que no ha mejorado la linealidad a pesar de la transformación. Sí ha mejorado la normalidad (como podemos ver en el QQ-plot, ya que se ajusta mejor a la línea recta), aunque en el modelo sin transformar ya se seguía normalidad. En cuanto a la homocedasticidad, aunque ahora el test de Breusch-Pagan indica heterocedasticidad, en el modelo de regresión sobre los residuos podemos ver que sí ha mejorado la heterocedasticidad, ya que ahora el pvalor no es significativo.

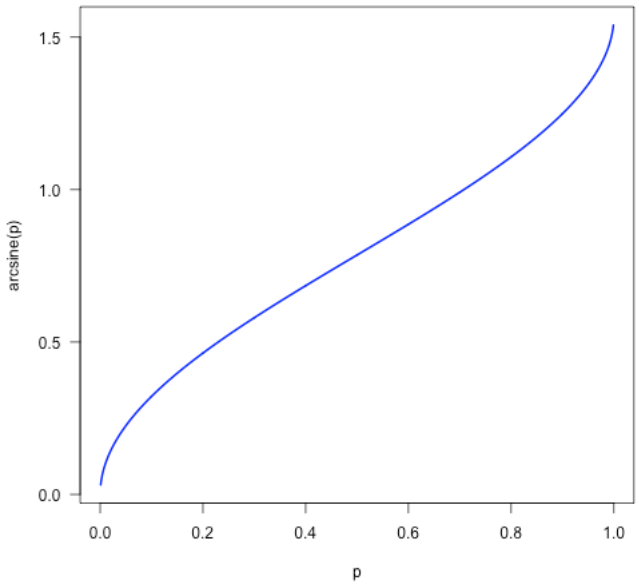
A pesar de las mejoras en normalidad y homocedasticidad, seguimos insatisfechos con el modelo ajustado, ya que sigue sin cumplir linealidad.

Por otro lado, la adición de interacciones o variables podría ayudar el ajuste del modelo, pero también aportaría complejidad y dificultad a la hora de analizar los resultados. Realizar un pre-procesamiento adicional de los datos también sería posible, como una estandarización o una transformación distinta de la variable dependiente, ya que arcsin(sqrt) no ha funcionado. Si esto falla, deberíamos considerar usar otro tipo de regresión. Si la relación entre las variables independientes y la variable dependiente no es lineal, debemos considerar realizar otro tipo de regresión como la logística.

(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(sqrt(x)) = sin^(-1)($\sqrt{x}$). La transformación sqrt estabiliza la varianza, haciendo 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