

PAC 1 Regresión Lineal

Maria Lucas

2023-04-22

Índice

Ejercicio 1	2
(a) Ajuste del modelo	2
(b) Intervalos de confianza para AmphipodDensity	3
(c) Multicolinealidad	3
(d) Modelo reducido	3
(e) Gráfico región de confianza	4
(f) Predicción	5
Ejercicio 2	7
(a) Gráfico de dispersión	7
(b) Modelos según área	9
(c) Modelo leones macho	11
(d) Predicción de la edad de una leona	12

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 H_0 que todos los coeficientes de regresión son 0, y como H_1 que al menos uno es distinto de 0.

- $H_0: \beta_1 = \beta_2 = \dots = \beta_p = 0$ (donde $\beta_1, \beta_2, \dots, \beta_p$ son los coeficientes de regresión de las variables predictoras del modelo)
- H_1 : al menos un β_i es diferente a 0, donde $i = 1, 2, \dots, p$

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 H_0 .

Tal y como se ve en la tabla de coeficientes, tanto SPS (Sand Particle Size, p -valor = 0.01) como AD (Amphipod Density, p -valor = 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 p -valor 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 varía 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 p -valor.

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 $BD \sim SPS + AD$ 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.

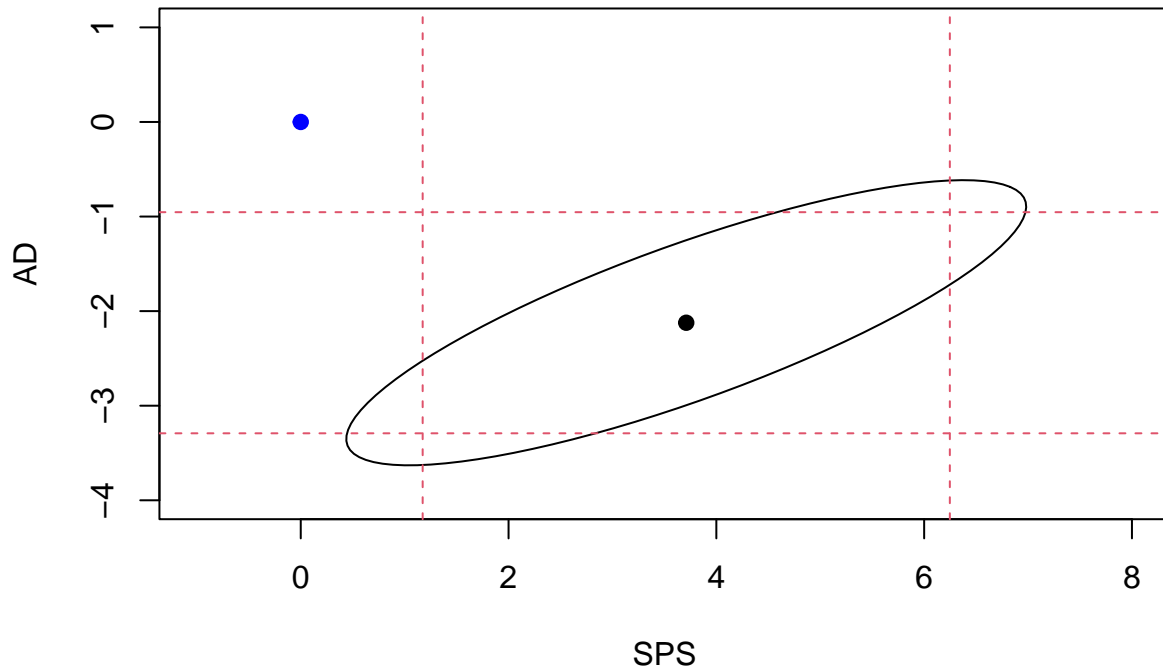
Por otro lado, en el modelo reducido todas las variables explican de manera significativa la variable respuesta. Además, el valor de R ajustado es similar en ambos modelos (0.93), este valor indica el porcentaje de la variable respuesta que es explicado por el modelo.

(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:

- $H_0: \beta_1 = \beta_2 = 0$. Los coeficientes de ambas variables son 0
- $H_1: \beta_1 \neq 0$ 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
```

```

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

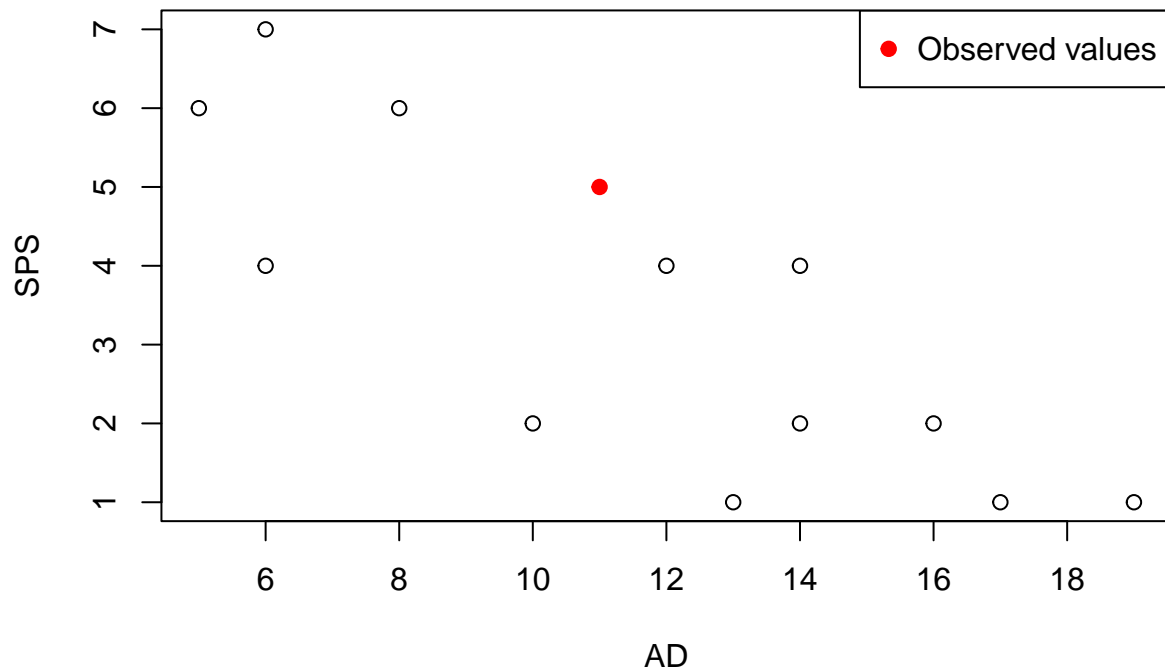
## Loading required package: bestglm
## Loading required package: leaps
## Loading required package: VGAM
## Loading required package: stats4
## Loading required package: splines
##
## Attaching package: 'VGAM'
## The following object is masked from 'package:car':
##
##      logit
## Loading required 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
# 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

(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':
##
```

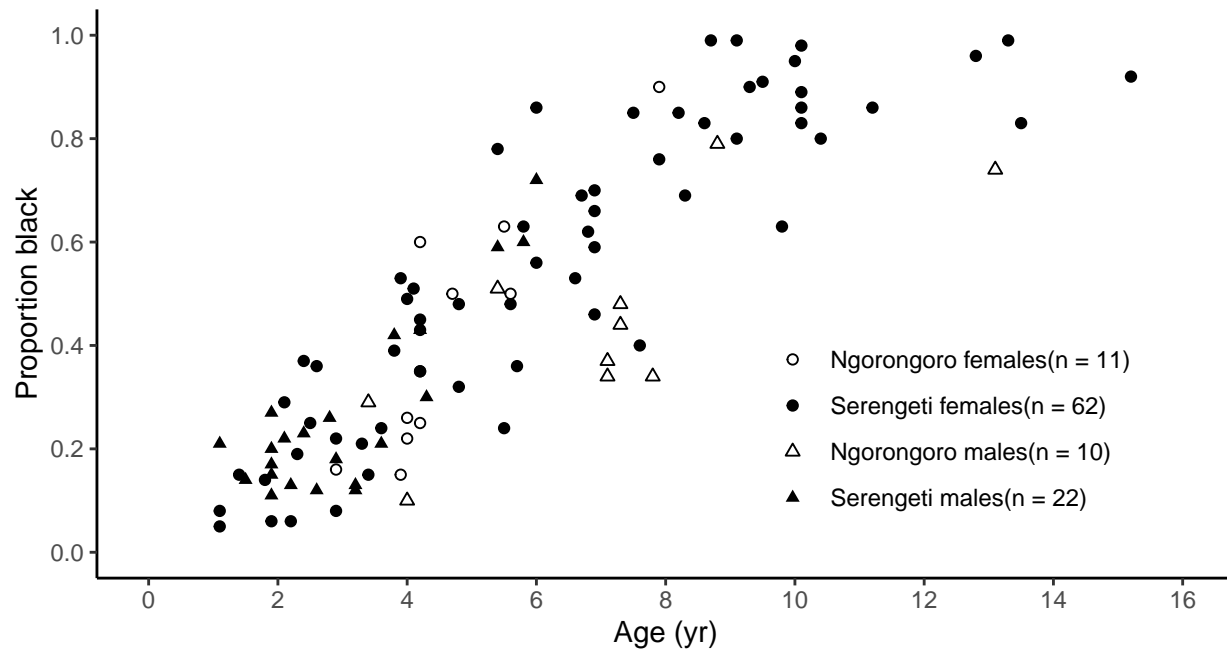
```

##      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", "Ngorongoro", "Serengeti")),
  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"), " ", ifelse(sex == "M", "males", "females")) +
      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
```

```
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

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
```

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 $p\text{-value} = 0.0279$.

Al separar por área, obtenemos los mismos resultados que en el artículo, en el que en la población Ngorongoro el sexo influye significativamente en la variable respuesta ($p\text{-value} = 0.04$), mientras que en los Serengeti no influye ($p\text{-value} = 0.4$). 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
```

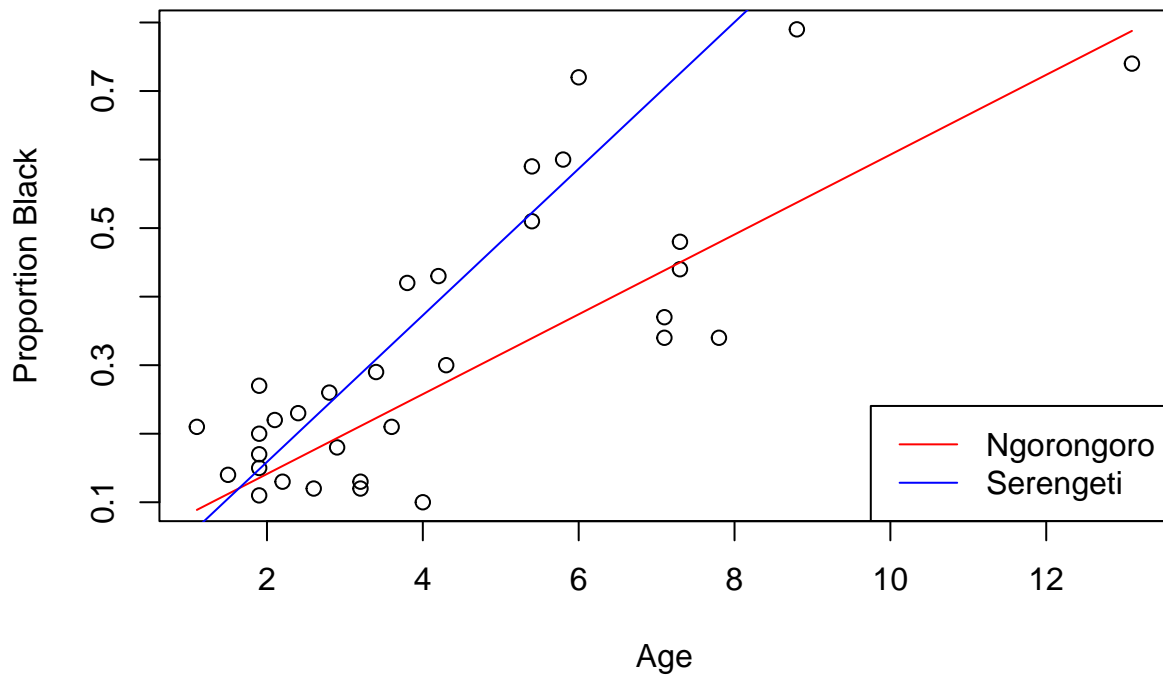
Confirmamos que para los machos existen diferencias significativas según el área ($p_v = 0.03$), teniendo los Serengeti una nariz más oscura que los Ngorongoro ($\text{coef} = 0.14$).

```
# 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.

(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 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 siguen una distribución normal, gracias a esto se puede realizar una regresión lineal. Se define de la siguiente manera: $\arcsin(x) = \arcsin(x) = \sin^{-1}(x)$. Cabe destacar que no aplicamos una transformación sqrt porque nuestros datos sabemos que están seguros entre 0 y 1, por lo tanto no se requiere esta transformación.

```
library(stats)

# Aplicamos la transformación sólo a
data2_split_sex$F$prop.black.transformed = asin(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(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), interval = "95%")
ci_75 <- predict(lmod_age, newdata = data.frame(prop.black.transformed = prop_black_transformed), interval = "75%")
ci_50 <- predict(lmod_age, newdata = data.frame(prop.black.transformed = prop_black_transformed), interval = "50%")

# 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_transformed - summary_lmod_age$prop_black)^2 / var(prop_black_transformed))

# 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_predicted_age, 2), ")",
                           "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	5.4 (1.65)	2.21-8.6	3.55-7.26	4.32-6.49

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.