Regresión, modelos y métodos - PEC 2

Alejandro Keymer

1/6/2020

Ejercicio 1 (25 pt.) Mecanismo neural de la nocicepción.

(a)

En una primera aproximación para comparar los porcentajes de reacción de los gusanos mutantes con los salvajes, nos planteamos hacer un test de comparación de dos muestras independientes y un gráfico adecuado, sin tener en cuenta los pulsos. Para ello debemos observar que la variable respuesta es un porcentaje y el test t de Student habitual no sirve. Entre las posibles soluciones tenemos las siguientes:

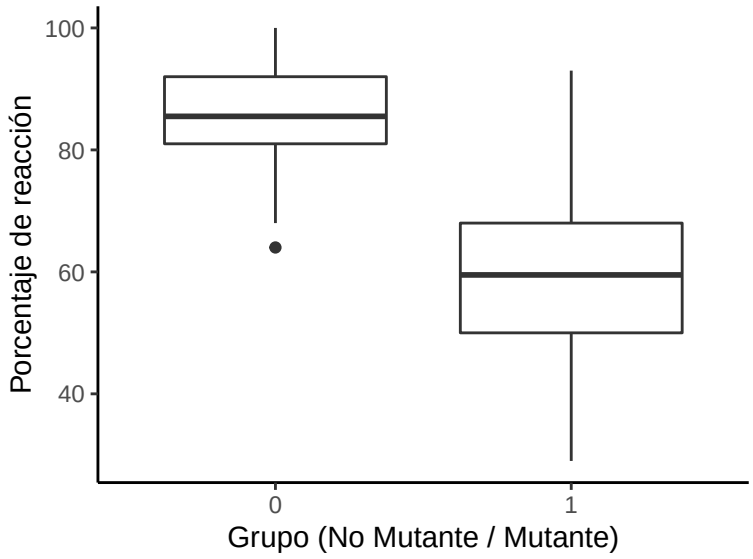
1. **Aplicar un test de Welch a los datos transformados con la función normalizante $\arcsin(\sqrt{p})$.**
2. **Hacer un test no paramétrico como el de Mann-Whitney-Wilcoxon.**
3. **Aplicar un test de permutaciones para comparar las dos muestras.**
4. **Realizar una regresión logística binomial.**
5. **Calcular una regresión beta que es especialmente apropiada para proporciones.**

Las dos últimas propuestas son las más acertadas, ya que las otras contemplan comparar medias o medianas de porcentajes. En todo caso, realizar tres de las cinco soluciones con una de ellas entre las dos últimas y comentar su resultado.

```
df <-
    read_csv2("C_elegans.csv") %>%
    mutate(p_reac = Perc_Reacting/100,
           IndMutant = factor(IndMutant))
```

Una vez cargada la base de datos, hacemos un gráfico de cajas entre los dos grupos de gusanos, para valorar la diferencia entre los porcentajes de reacción.

```
# boxplot de los datos
ggplot(df, aes(x = IndMutant, y = Perc_Reacting)) +
    geom_boxplot() + labs(x="Grupo (No Mutante / Mutante)", y = "Porcentaje de reacción") +
    theme_classic()
```



regresión logista binomial

```
mod <-
    glm(IndMutant ~ p_reac, df, family = binomial(link = "logit"))
pander(mod)
```

Table 1: Fitting generalized (binomial/logit) linear model: IndMutant ~ p_reac

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	11.07	3.343	3.313	0.0009244
p_reac	-15.1	4.483	-3.369	0.000755

Test de permutaciones

```
mod <- coin::independence_test(p_reac ~ IndMutant, df)
```

```
mod <- wilcox.test(Perc_Reacting ~ IndMutant, df)

pander(mod)
```

Table 2: Wilcoxon rank sum test with continuity correction: `Perc_Reacting` by `IndMutant`

Test statistic	P value	Alternative hypothesis
368	5.733e-06 * * *	two.sided

(b)

Dibujar el gráfico de dispersión del porcentaje de reacción en función del número de pulsos de luz con dos rectas que representen las dos sub poblaciones por separado, sin modelo común. Describir la forma de la relación. ¿Qué sugiere sobre el patrón de respuestas de abstinencia en las dos sub-poblaciones de *C. elegans* para un número creciente de pulsos de luz? ¿Cómo encaja

su respuestaen el contexto de habituación? Explique por qué estos resultados sugieren una participación de las neuronas sensoriales de PVD en la nocicepción y la habituación. Nota: En este apartado consideramos la variable respuesta el porcentaje de reacción sin ninguna transformación.

```
mod_0 <- lm(Perc_Reacting ~ Pulses, filter(df, IndMutant == 0))
mod_1 <- lm(Perc_Reacting ~ Pulses, filter(df, IndMutant == 1))

# modelo de NO MUTANTES
pander(mod_0)
```

Table 3: Fitting linear model: Perc_Reacting ~ Pulses

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	100.5	2.491	40.37	4.135e-19
Pulses	-1.471	0.2079	-7.073	1.351e-06

```
# modelo de MUTANTES
pander(mod_1)
```

Table 4: Fitting linear model: Perc_Reacting ~ Pulses

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	82.97	3.242	25.59	1.318e-15
Pulses	-2.259	0.2707	-8.347	1.331e-07

```
q1 <-
ggplot(df, aes(Pulses,Perc_Reacting,  color = IndMutant)) +
    geom_point() +
    geom_abline(intercept = coef(mod_0)[1],
                slope = coef(mod_0)[2],
                linetype = 1) +
    geom_abline(intercept = coef(mod_1)[1],
                slope = coef(mod_1)[2],
                linetype = 1) +
    labs(title = "Dos rectas por separado", y = "Porcentaje de reacción", color = "Mutante") +
    theme_classic() +
    theme(legend.position = "bottom")
```

(c)

Proponer un modelo lineal que permita estudiar la asociación de la variable respuesta *porcentaje de reacción* (sin ninguna transformación) con la variable Pulses y el factor IndMutant.

¿Cual es la ecuación del modelo final propuesto?

¿Es un modelo significativo?

¿Qué tanto por ciento de la variabilidad del porcentaje de reacción queda explicado por el modelo?

¿Todos los coeficientes de las variables explicativas del modelo son significativos?

Dibujar el gráfico de dispersión del apartado anterior pero con las rectas resultantes del modelo. Nota: En este apartado no haremos ninguna transformación de la variable respuesta, ni utilizaremos ningún modelo generalizado, ya que confiamos en la validez del modelo balanceado.

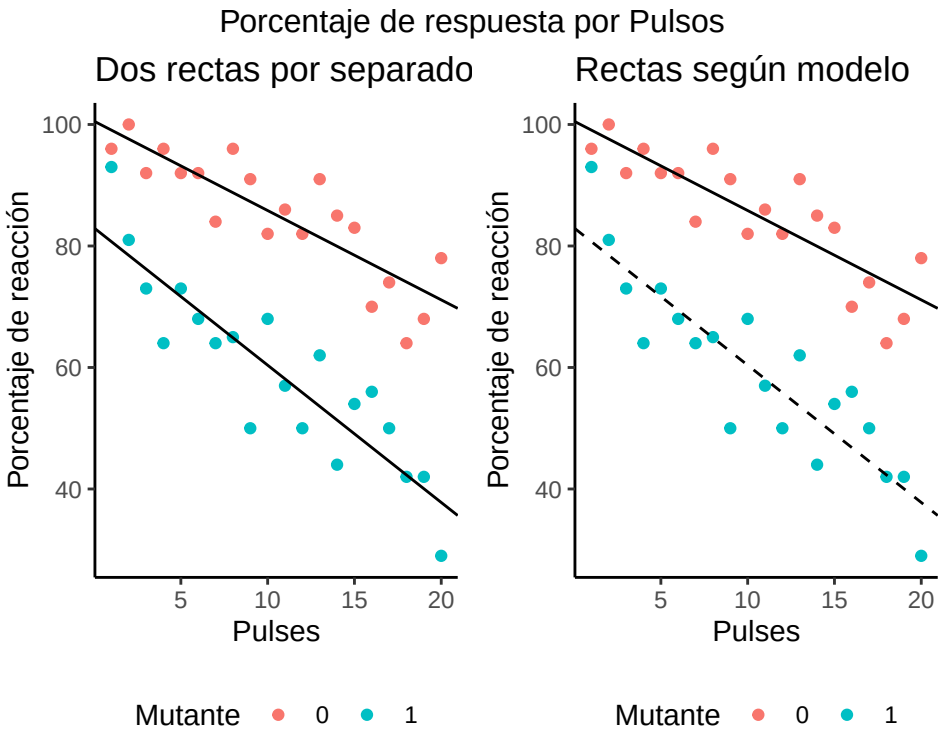
```
mod_2 <- lm(Perc_Reacting ~ Pulses * IndMutant, df)
pander(summary(mod_2))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	100.5	2.891	34.78	2.66e-29
Pulses	-1.471	0.2413	-6.094	5.212e-07
IndMutant1	-17.57	4.089	-4.297	0.0001257
Pulses:IndMutant1	-0.7887	0.3413	-2.311	0.02668

Table 6: Fitting linear model: Perc_Reacting ~ Pulses * IndMutant

Observations	Residual Std. Error	R^2	Adjusted R^2
40	6.224	0.892	0.883

```
q2<- ggplot(df, aes(Pulses, Perc_Reacting, color = IndMutant)) +
    geom_point() +
    geom_abline(intercept = coef(mod_2)[1],
                slope = coef(mod_2)[2],
                linetype = 1) +

    geom_abline(intercept = (coef(mod_2)[1] + coef(mod_2)[3]),
                slope = coef(mod_2)[2] + coef(mod_2)[4],
                linetype = 2) +
    labs(title = "Rectas según modelo", y = "Porcentaje de reacción", color = "Mutante") +
    theme_classic() +
    theme(legend.position = "bottom")

grid.arrange(q1,q2, nrow=1, top= "Porcentaje de respuesta por Pulsos")
```



(d)

Ahora contestaremos las preguntas y dibujaremos el gráfico del apartado anterior, si realizamos la transformación normalizante $arcsin(\sqrt{p})$ sobre la variable respuesta. Hallar el intervalo de confianza al 95 % para la media del porcentaje de mutantes que reaccionan a 10 pulsos de luz en condiciones experimentales parecidas. Nota: Habrá que deshacer la transformación para dibujar "las rectas".

```
df_adj <-
    df %>%
    mutate(p_arcsin = asin(sqrt(Perc_Reacting/100)))

df_adj
```

```
## # A tibble: 40 x 5
##    Perc_Reacting Pulses IndMutant p_reac p_arcsin
##            <dbl>  <dbl> <fct>      <dbl>    <dbl>
##  1            93      1 1           0.93    1.30
##  2            81      2 1           0.81    1.12
##  3            73      3 1           0.73    1.02
##  4            64      4 1           0.64    0.927
##  5            73      5 1           0.73    1.02
##  6            68      6 1           0.68    0.970
##  7            64      7 1           0.64    0.927
##  8            65      8 1           0.65    0.938
##  9            50      9 1           0.5     0.785
## 10            68     10 1           0.68    0.970
## # ... with 30 more rows
```

```r
mod_4 <- lm(p_arcsin ~ Pulses * IndMutant, df_adj)
mod_5 <- lm(p_arcsin ~ Pulses + IndMutant, df_adj)

pander(mod_4)
```

Table 7: Fitting linear model: p_arcsin ~ Pulses * IndMutant

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.442	0.03831	37.63	1.677e-30
Pulses	-0.023	0.003198	-7.192	1.839e-08
IndMutant1	-0.2995	0.05418	-5.527	2.977e-06
Pulses:IndMutant1	-0.001499	0.004523	-0.3314	0.7423

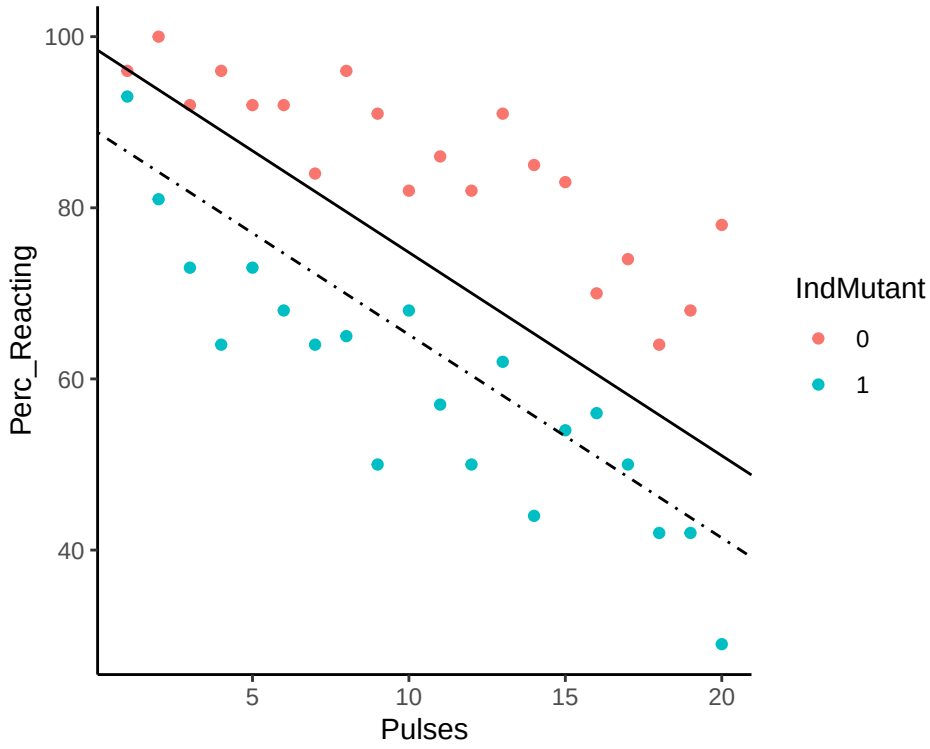
```r
pander(summary(mod_5))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.449	0.0297	48.8	3.421e-35
Pulses	-0.02375	0.002234	-10.63	8.417e-13
IndMutant1	-0.3152	0.02576	-12.23	1.434e-14

Table 9: Fitting linear model: p_arcsin ~ Pulses + IndMutant

Observations	Residual Std. Error	R^2	Adjusted R^2
40	0.08147	0.8765	0.8699

```r
org_cf <- sin(coef(mod_5))^2 * 100

df %>%
    ggplot(aes(Pulses, Perc_Reacting, color = IndMutant)) +
    geom_point() +
    geom_abline(intercept = org_cf[1],
                slope = coef(mod_5)[2] * 100,
                linetype = 1) +
    geom_abline(intercept = org_cf[1] - org_cf[3],
                slope = coef(mod_5)[2] * 100,
                linetype = 4) +
    theme_classic()
```




```
# prediccion
new_df <- tibble(Pulses = 10, IndMutant = "1")
ypred <- predict.lm(mod_5, new_df, interval = "conf")

pander(sin(ypred)^2)
```

fit	lwr	upr
0.6104	0.574	0.6461

Ejercicio 2 (50 pt.)

```
# cargamos los datos
penta <-
    read.table("penta.dat", header = T, na.strings = ".") %>%
    na.omit()

# funcion para caluclo de RMSE
rmse <- function(x,y) sqrt(mean((x-y)^2))
```

```
set.seed(321)
idx <- sample(1:30, size = 20, replace = F)
ptrain <- penta[idx,]
ptest <- penta[-idx,]
```

(a)

Comprobar con los factores de inflación de la varianza que el modelo lineal completo y con todos los datos padece un problema grave de multicolinealidad. Esto justifica que debamos

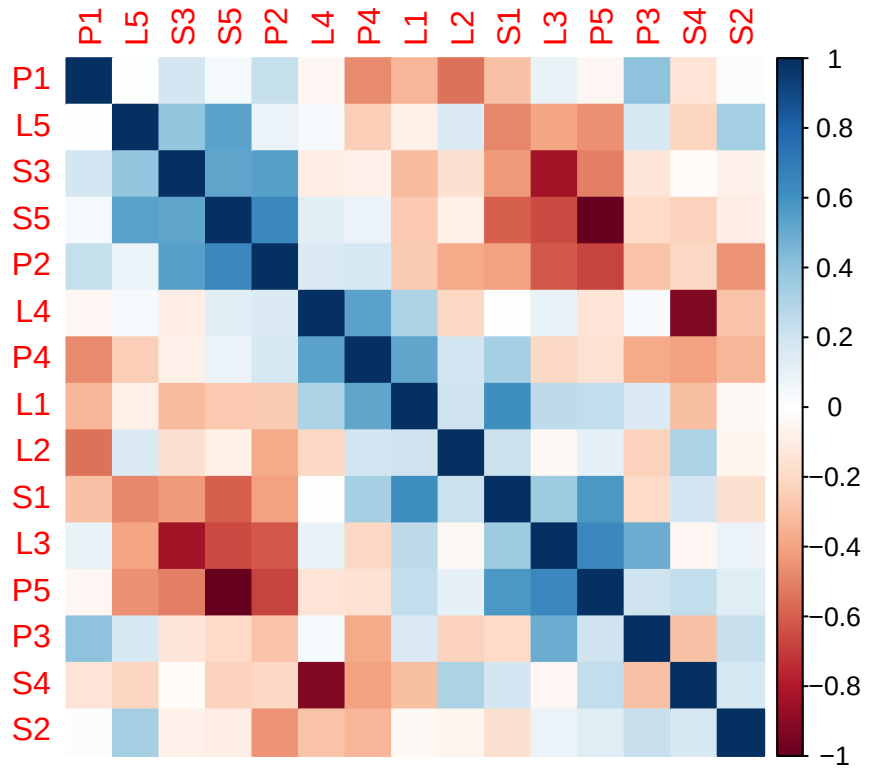
reducir el número de variables predictoras o utilizar algún método alternativo.

```
mod <- lm(log.RAI ~ ., ptrain[-1])
pander(mod)
```

Table 11: Fitting linear model: log.RAI ~ .

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.069	1.492	0.7169	0.5131
S1	-0.06191	0.1043	-0.5935	0.5848
L1	-0.02261	0.1511	-0.1497	0.8883
P1	-0.161	0.1562	-1.031	0.3609
S2	0.05369	0.06536	0.8216	0.4575
L2	0.04448	0.0988	0.4502	0.6759
P2	-0.07596	0.1505	-0.5046	0.6404
S3	-0.08589	0.09257	-0.9278	0.406
L3	0.1564	0.1383	1.131	0.3212
P3	0.1674	0.3226	0.5189	0.6312
S4	0.3242	0.4097	0.7912	0.4731
L4	0.3572	0.3484	1.025	0.3632
P4	-0.3223	0.5807	-0.5551	0.6084
S5	-1.344	3.695	-0.3638	0.7344
L5	0.1931	0.4431	0.4357	0.6855
P5	-0.9953	2.762	-0.3603	0.7368

```
X <- model.matrix(mod)[,-1]

# hacemos un plot de correlacion
corrplot::corrplot(cor(X), method = "color", order = 'AOE')
```




```
# Chequeamos las vifs
vifs_x <- vif(X)
pander(rbind(vifs_x, SE = sqrt(vifs_x)), digits = 3)
```

Table 12: Table continues below

	S1	L1	P1	S2	L2	P2	S3	L3	P3
vifs_x	9.18	10.7	3.83	2.97	3.87	6.42	9.05	22.2	7.88
SE	3.03	3.26	1.96	1.72	1.97	2.53	3.01	4.71	2.81

	S4	L4	P4	S5	L5	P5
vifs_x	69.8	55.4	13.1	6375	70.4	5913
SE	8.36	7.44	3.62	79.8	8.39	76.9

(b)

Estudiar el modelo reducido que proporciona el método `stepwise`. Se trata de comparar los RMSE del modelo reducido y del modelo completo para el grupo de entrenamiento y para el grupo de prueba. También de ver si el modelo reducido salva el problema de multicolinealidad.

```
mod_step <- step(lm(log.RAI ~ ., ptrain[-1]), trace = 0)
anova(mod, mod_step)
```

```
## Analysis of Variance Table
##
## Model 1: log.RAI ~ S1 + L1 + P1 + S2 + L2 + P2 + S3 + L3 + P3 + S4 + L4 +
```

```
##     P4 + S5 + L5 + P5
## Model 2: log.RAI ~ S1 + P1 + S2 + L2 + S3 + L3 + L4 + P4
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1      4 0.48628
## 2     11 0.59796 -7  -0.11169 0.1312 0.9892
```

```
ypred <- predict(mod, newdata = ptest)

rmse(ypred, ptest$log.RAI)
```

```
## [1] 0.582237
```

```
ypred_step <- predict(mod_step, newdata = ptest)

rmse(ypred_step, ptest$log.RAI)
```

```
## [1] 0.5183494
```

```
# Chequeamos las vifs
X <- model.matrix(mod_step)[,-1]
vifs_x <- vif(X)
pander(rbind(vifs_x, SE = sqrt(vifs_x)), digits = 2)
```

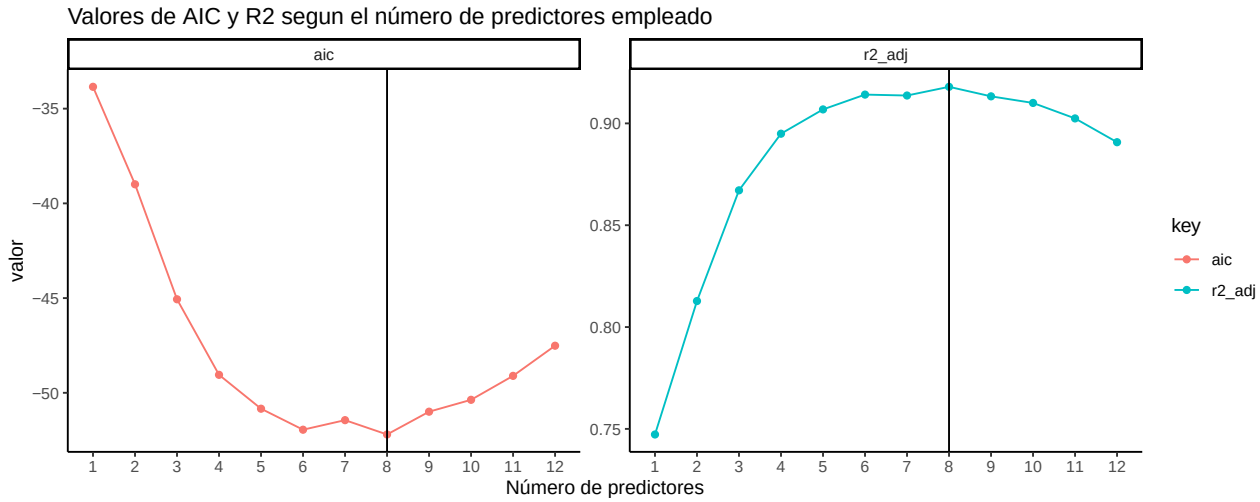
	S1	P1	S2	L2	S3	L3	L4	P4
vifs_x	1.8	2	1.2	1.6	5.1	6.1	2.2	3.4
SE	1.3	1.4	1.1	1.3	2.3	2.5	1.5	1.9

(c)

Hallar el mejor modelo por el criterio del AIC y por el R2 ajustado. ¿Coinciden? ¿Es el mismo modelo que selecciona el stepwise?

```
# creo los sub modelos
rs <-
    regsubsets(log.RAI ~ ., ptrain[-1], nvmax = 12) %>%
    summary()

n <- dim(ptrain)[1]
p <- length(rs$rss)


rs_crit <-
    rs$which %>%
    as_tibble() %>%
    rowid_to_column() %>%
    mutate(
        rowid = factor(rowid),
        rss = rs$rss,
        r2_adj = rs$adjr2,
        aic = n * log(rss/n) + (2:(p+1)) * 2)

# Creamos el plot de AIC y R2
rs_crit %>%
    select(rowid, r2_adj, aic) %>%
```

```
    gather(key, value, -rowid) %>%
    ggplot(aes(rowid, value, color = key, group = key)) +
    geom_point() +
    geom_line() +
    geom_vline(xintercept = 8) +
    facet_wrap(~key, scales = "free") +
    labs(title = "Valores de AIC y R2 segun el número de predictores empleado",
         x = "Número de predictores", y = "valor") +
    theme_classic()
```




```
# modelo elegido
vars_aic <- rs$obj$xnames[rs$which[8,]]
vars_step <- colnames(model.matrix(mod_step))


pander(rbind(vars_aic, vars_step))
```

vars_aic	(Intercept)	S1	P1	S2	L2	S3	L3	L4	P4
vars_step	(Intercept)	S1	P1	S2	L2	S3	L3	L4	P4

En este caso los criterios de AIC y de R2 ajustado, coinciden en que el mejor modelo es el que utiliza 8 parámetros (+ el intercepto). El modelo elegido por le criterio AIC / R2, resulta igual al modelo elegido mediante el procedimiento *stepwise*

(d)

Estudiar una regresión por componentes principales con estos datos. ¿Qué tal funciona con 8 componentes para el grupo de entrenamiento? ¿Cual es el número de componentes que se recomienda si hacemos una validación cruzada? Con ese número mínimo de componentes, ¿cual es el RMSE para el conjunto de prueba?

```
pca_1 <-
    ptrain %>%
    select(S1:P5) %>%
    FactoMineR::PCA(graph = F)
```

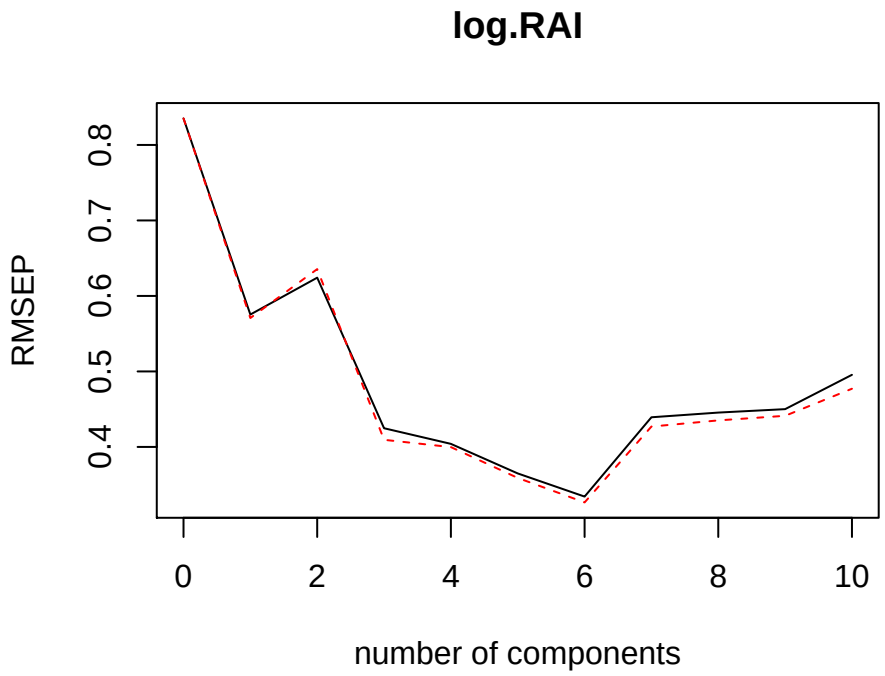
```
pander(t(pca_1$eig)[,1:8], digits = 2)
```

Table 16: Table continues below

	comp 1	comp 2	comp 3	comp 4	comp 5
eigenvalue	4.6	2.9	2.4	1.7	0.84
percentage of variance	31	19	16	11	5.6
cumulative percentage of variance	31	50	66	78	83

	comp 6	comp 7	comp 8
eigenvalue	0.74	0.6	0.39
percentage of variance	4.9	4	2.6
cumulative percentage of variance	88	92	95

```
mod_pca <- pcr(log.RAI ~ ., data = ptrain[-1], validation = "CV", ncomp = 10)

plot(RMSEP(mod_pca, estimate = c("CV", "adjCV")))
```




```
ypred <- predict(mod_pca, ptest, ncomp=8)

rmse(ypred, ptest$log.RAI)
```

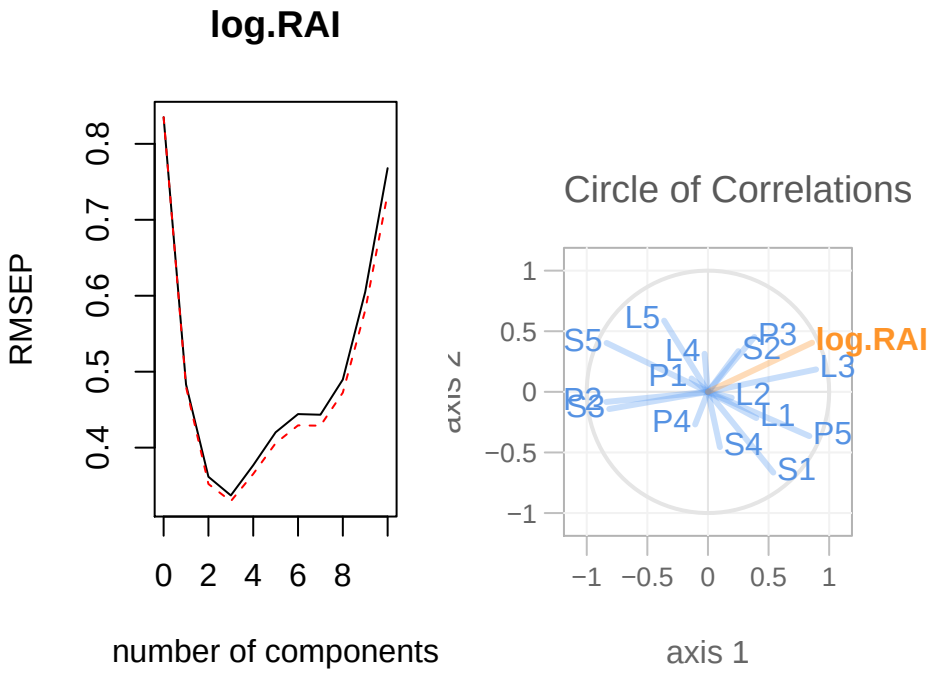
```
## [1] 0.5548939
```

(e)

Estudiar una regresión PLS. ¿Cuantas componentes selecciona la validación cruzada? ¿Cual es la variabilidad de las variables predictoras explicada por 3 componentes? Con el paquete `plsdepot` se pueden estudiar las correlaciones entre las variables y las componentes y realizar el gráfico llamado *círculo de correlaciones*. ¿Cual es la variable predictora mejor correlacionada con la primera componente?

```
mod_plr <- plsr(log.RAI ~ ., data = ptrain[,-1], validation = "CV", ncomp = 10)




pls_reg <- plsdepot::plsreg1(ptrain[2:16], ptrain[17])

par(mfrow = c(1,2))
plot(RMSEP(mod_plr))
plot(pls_reg)



ypred <- predict(mod_pca, ptest, ncomp=3)

rmse(ypred, ptest$log.RAI)
```

```
## [1] 0.6055861
```



(f) Estudiar también el método de Ridge Regression con la función `lm.ridge()`. Además de los valores RMSE para el grupo de entrenamiento y de prueba, debemos acompañar el análisis con un gráfico de los coeficientes. Para hallar el lambda óptimo podemos empezar por un límite superior bajo y aumentar sucesivamente ese límite hasta que el valor óptimo no sea el límite superior. Aunque siempre es mejor estandarizar los datos, en este ejercicio no lo haremos para no complicar más los cálculos del RMSE del conjunto de prueba. Internamente, lo hace la propia función `lm.ridge()`.

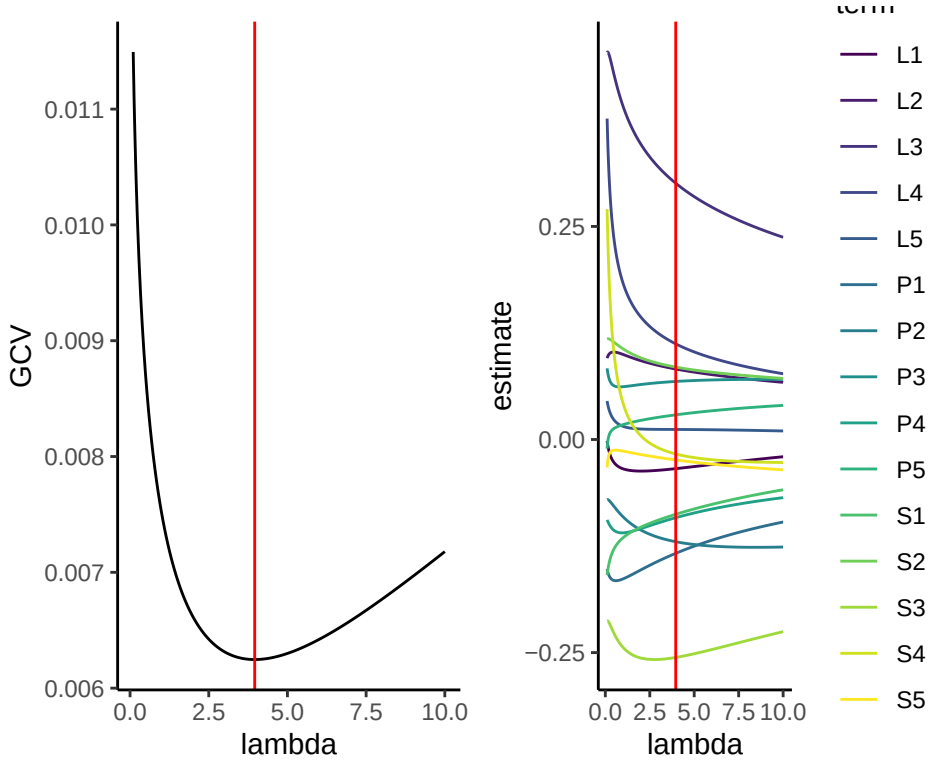
Nota: Si el resultado del ajuste de las predicciones a los datos es decepcionante podríamos optarpor utilizar la función `v.glmnet(...,alpha=0)` del paquete `glmnet`. Esta función no necesita que le indiquemos una secuencia de valores de lambda, aunque sí requiere que le pasemos los datos como objeto matrix. Además podemos utilizar la validación leave one out simplemente haciendo que el número de carpetas (folds) sea exactamente el número de observaciones.

Nota2: Aunque no hace falta, si se utilizan las funcines `cv.glmnet()` y `glmnet()` para el cálculo del modelo Ridge Regression, se puede comprobar que los resultados difieren notablemente de los que tenemos con la función `lm.ridge()`. Una sencilla explicación se puede leer en https://stats.stackexchange.com/questions/74206/ridge-regression-results-different-in-using-lm-ridge-and-glmnet

```
mod_rg <- lm.ridge(log.RAI ~ ., data = ptrain[,-1], lambda = seq(0.1, 10, .005))

lbd <-
    tidy(mod_rg) %>%
    distinct(lambda, GCV) %>%
    rowid_to_column() %>%
    top_n(1, -GCV)

lbd
```

```
## # A tibble: 1 x 3
##   rowid lambda     GCV
##   <int>  <dbl>   <dbl>
## 1   773   3.96 0.00625
```

```
# plot de los GCV

p1 <- tidy(mod_rg) %>%
    ggplot(aes(lambda, GCV)) +
    geom_line() +
    geom_vline(xintercept = lbd$lambda, color = "red") +
    theme_classic()


p2<- tidy(mod_rg) %>%
    ggplot(aes(lambda, estimate, color = term)) +
    geom_line() +
    geom_vline(xintercept = lbd$lambda, color = "red") +
    scale_colour_viridis_d() +
    theme_classic()

grid.arrange(p1,p2,nrow = 1)
```

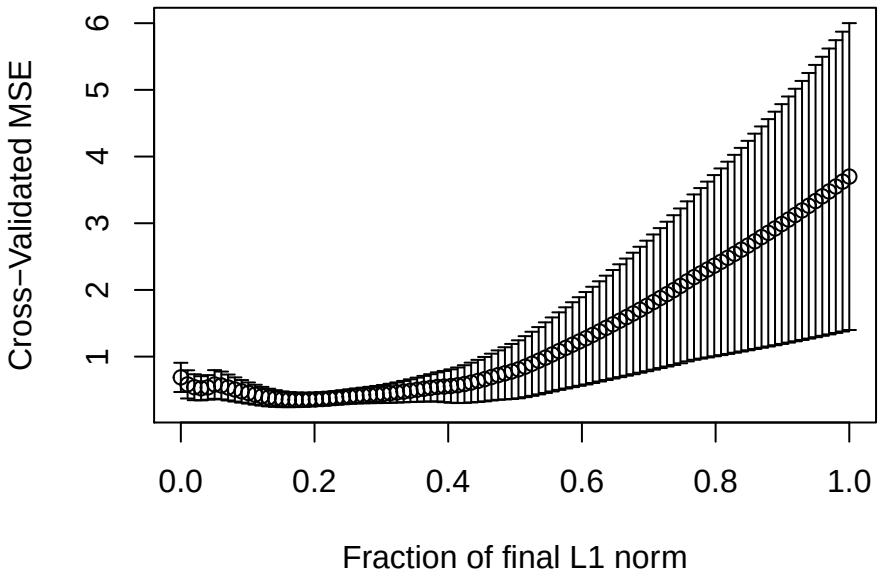



```
# hacemos la prediccion con el valor de lambda que minimiza GCV

ypred <- cbind(1, as.matrix(ptest[2:16])) %*% coef(mod_rg)[lbd$rowid,]



# rendimiento del modelo
rmse(ypred, ptest$log.RAI)
```

```
## [1] 0.5967337
```

(g) Finalmente estudiaremos el método LASSO con la función `lars()` del paquete del mismo nombre. Nota: Tal vez el resultado es demasiado restrictivo y hay que aumentar el número de variables seleccionadas, lo que equivale a aumentar ligeramente el valor de lambda.

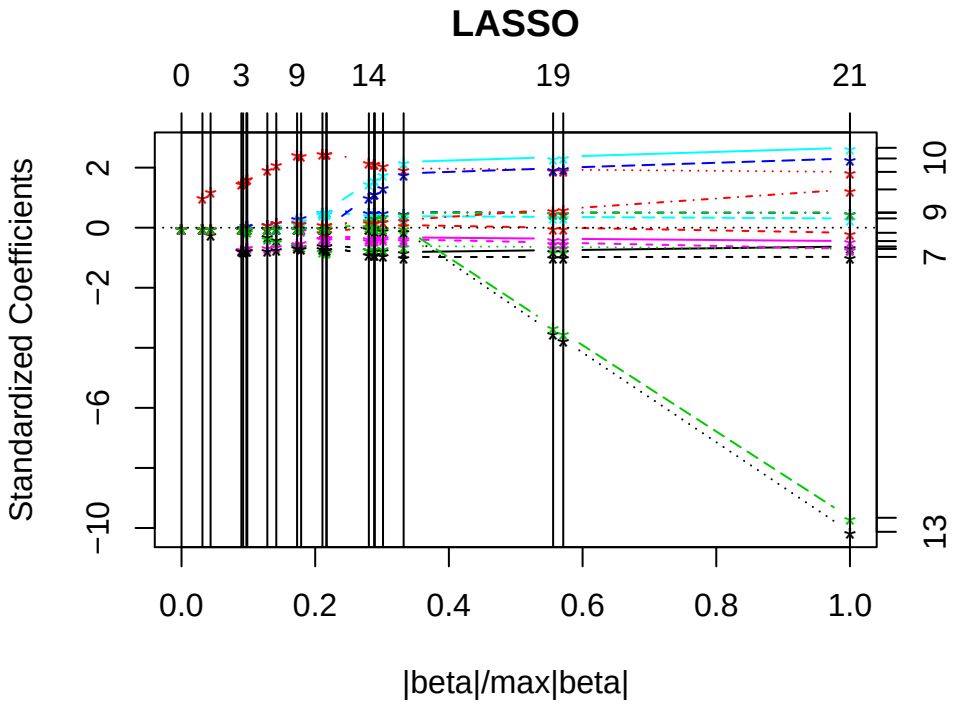
```
mod_lar <- lars(as.matrix(ptrain[2:16]), ptrain$log.RAI)
mod_lar_cv <- cv.lars(as.matrix(ptrain[2:16]), ptrain$log.RAI)
```




```
summary(mod_lar)
```

```
## LARS/LASSO
## Call: lars(x = as.matrix(ptrain[2:16]), y = ptrain$log.RAI)
##    Df     Rss      Cp
## 0   1 12.5888 85.5527
## 1   2  7.2826 43.9055
## 2   3  5.7803 33.5475
## 3   4  2.3851  7.6190
## 4   5  2.2948  8.8766
## 5   6  2.1410  9.6112
## 6   7  2.1008 11.2806
## 7   8  1.3668  7.2430
## 8   9  1.1280  7.2790
## 9  10  0.7622  6.2697
## 10 11  0.7178  7.9048
## 11 12  0.5787  8.7600
## 12 13  0.5695 10.6850
## 13 14  0.5690 12.6803
## 14 15  0.5157 14.2422
## 15 14  0.5120 12.2113
## 16 15  0.5115 14.2077
## 17 14  0.5072 12.1724
## 18 15  0.5022 14.1311
## 19 16  0.4932 16.0571
## 20 15  0.4927 14.0532
## 21 16  0.4863 16.0000
```

```
plot(mod_lar)
```




```
ind <-
    as_tibble(mod_lar_cv) %>%
    top_n(1, -cv) %>%
    pull(index)



predict(mod_lar, s = 0.1, type="coef", mode="fraction")$coef
```

```
##           S1           L1           P1           S2           L2           P2
## -0.001316898  0.000000000 -0.022792134  0.011926276  0.000000000 -0.108034690
##           S3           L3           P3           S4           L4           P4
## -0.064645586  0.140743816  0.000000000  0.000000000  0.000000000  0.000000000
##           S5           L5           P5
##  0.000000000  0.003123518  0.000000000
```

Ejercicio 3 (25 pt.)

Es posible utilizar la regresión logística para estudiar la discriminación entre dos grupos dado un conjunto de variables explicativas. Para más detalles podéis consultar el apartado 8.10 del libro de Manly.

(a) Realizar un análisis discriminante con ayuda de la regresión logística con los datos de los 49 gorriones hembra después de una tormenta. Reproducir con todo detalle la tabla 8.6 de la página 153 del libro de Manly.

```
load("gorriones.RData")

mod_lg <- glm(superviv ~ x1 + x2 + x3 + x4 + x5, gorriones, family = binomial(link = "logit"))
mod_NULL <- glm(superviv ~ 1, gorriones, family=binomial(link="logit"))
```

```
# Para reporducir *con todo detalle* se puede utilizar la sintaxis de dplyr y crear la tabla
tbl <-
```

```
    tidy(mod_lg) %>%
    mutate(
        chi_2 = (estimate / std.error) ^ 2,
        Variable = c(
            "Constant",
            "Total length",
            "Alar length",
            "Length beak and head",
            "Length humerus",
            "Length keel of sternum"
        ),
        chi_2 = round(chi_2, 2)
    ) %>%
    mutate_at(vars(estimate:p.value), list( ~ round(., 3))) %>%
    select(
        Variable,
        '  estimate' = estimate,
        'Standard \\\\ error' = std.error,
        'Chi-squared' = chi_2,
        'p-Value' = p.value
    )


tbl[1,4:5] <- "--"


pander(tbl, justify = "lcccc")
```

Variable	estimate	Standard \ error	Chi-squared	p-Value
Constant	13.58	15.87	–	–
Total length	-0.163	0.14	1.36	0.244
Alar length	-0.028	0.106	0.07	0.794
Length beak and head	-0.084	0.629	0.02	0.894
Length humerus	1.062	1.023	1.08	0.299
Length keel of sternum	0.072	0.417	0.03	0.864

(b) Contrastar con un test 2 la significación de la regresión y explicar su resultado. xc

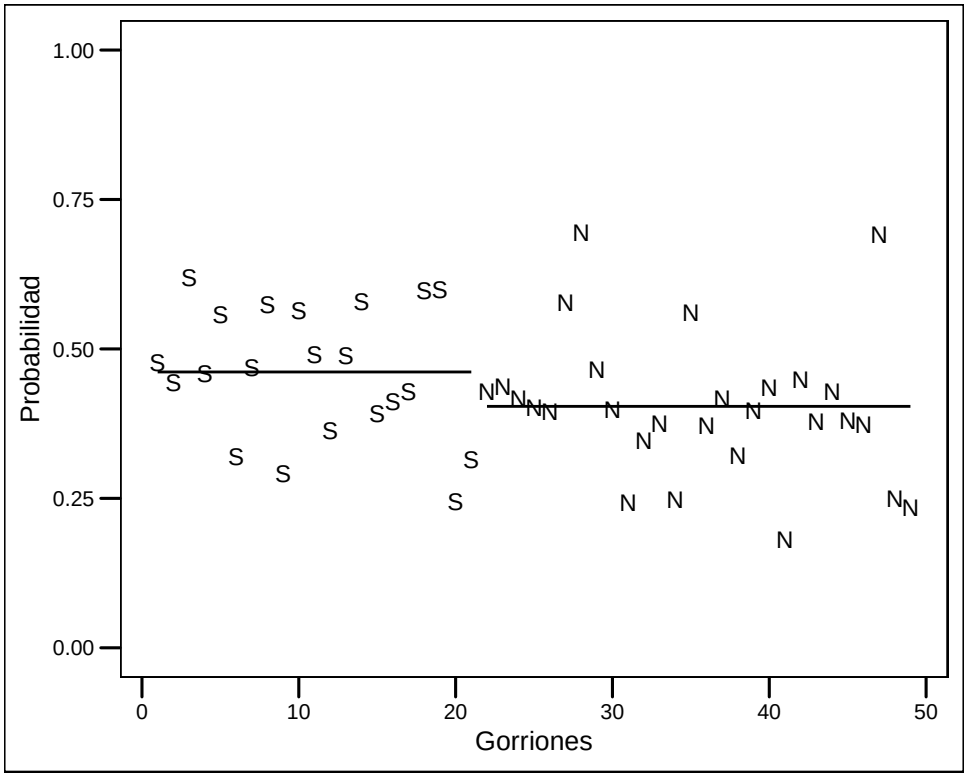
```
anova(mod_NULL, mod_lg, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: superviv ~ 1
## Model 2: superviv ~ x1 + x2 + x3 + x4 + x5
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        48     66.925
## 2        43     64.071  5   2.8536   0.7225
```

(c) Realizar un gráfico como el de la figura 8.2 de la página 156 del libro de Manly pero con los datos de este ejercicio y valorar el resultado.

```
tibble(
    fitted = mod_lg$fitted.values,
    class = augment(mod_lg)$superviv
```

```
) %>%
    rowid_to_column() %>%
    group_by(class) %>%
    mutate(mean = mean(fitted)) %>%
    ggplot(aes(rowid, fitted, shape = class)) +
    geom_point(size = 3) +
    geom_line(aes(y=mean)) +
    scale_shape_manual(values = c("N","S")) +
    theme_base(base_size = 10) +
    labs(x="Gorriones", y="Probabilidad") +
    scale_y_continuous(limits = c(0,1)) +
    guides(shape = F)
```



(d) Calcular un intervalo de confianza para el parámetro de la variable x4=length humerus y para su odds ratio.

```
confint(mod_lg)[5,]
```

```
## Waiting for profiling to be done...
```

```
##      2.5 %     97.5 %
## -0.9118587  3.1664865
```

```
tidy(mod_lg)
```

```
## # A tibble: 6 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)  13.6       15.9       0.856   0.392
## 2 x1           -0.163      0.140    -1.16    0.244
## 3 x2           -0.0276     0.106    -0.261   0.794
```

```
## 4 x3           -0.0837     0.629    -0.133   0.894
## 5 x4            1.06       1.02      1.04    0.299
## 6 x5            0.0716     0.417     0.172   0.864
```

(e) Calcular la tabla de confusión de la clasificación obtenida con la regresión logística.

```
ypred <-
    ifelse(
        predict.glm(mod_lg, type = "response") > 0.5,
        'S', 'N') %>%
    factor()


# para la matriz de confusión se puede utilizar la función de
# la librería `caret`
cm <- confusionMatrix(gorriones$superviv, ypred)


pander(cm$table)
```

	N	S
N	24	4
S	14	7

```
pander(cm$overall)
```

Table 20: Table continues below

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
0.6327	0.2025	0.4829	0.7658	0.7755

AccuracyPValue	McnemarPValue
0.9926	0.03389