TEORÍA Y PRAXIS DE MODELOS GENERALIZADOS: INFIRIENDO PATRONES CON EL PAQUETE ESTADÍSTICO R

Ejemplo de análisis con variables predictoras muy relacionadas.

Curso de la Sociedad de Amigos del Museo Nacional de Ciencias Naturales - CSIC

Luis M. Carrascal

Marzo 2015

CONTENIDO:

Partial Least Squares (PLS) *regression analysis*

Similitudes y diferencias con modelos de regresión lineales

Soluciones robustas mediante el uso de *bootstrapping*.

PLS con los paquetes “plsdepot” y “plsRglm”

Vamos a abordar el análisis de la variación en una variable respuesta, trabajando con pocas unidades muestrales y un gran número de variables predictoras estrechamente relacionadas entre si. Nuestros datos de partida están disponibles en la siguiente URL: www.lmcarrascal.eu/cursos/canarias.xls. La matriz que vamos a analizar se refiere a la riqueza de aves terrestres en el archipiélago de las Islas Canarias, aspecto que queremos explicar en función de una serie de características de 12 islas. Para saber más acerca de este sistema y sus aves podemos consultar http://www.lmcarrascal.eu/pdf/arla02.pdf.

Primeramente vamos a pracitar la **importación de los datos desde una hoja de MS-Excel**. Una vez descargados los datos de la web, los abrimos en MS-Excel e iniciamos el programa RStudio con el que vamos a realizar los análisis.

En la consola de RStudio escribimos la siguiente línea de código que nos servirá para importar los datos desde MS-Excel haciendo uso del portapapeles (`"clipboard"`):

```
canarias <- read.table("clipboard", header=T, sep="\t", dec=".")
```

¡¡ No corráis aun esta línea de código !! Esperad un momento.

Seleccionamos con el ratón los datos en MS-Excel, comenzando desde la celda A1 (codisla) hasta la celda U13 (0.000), y a continuación copiamos esos datos al portapapeles. Volvemos a la consola de RStudio y, ¡ ahora sí !, corremos la línea anterior de código. ¡Acabamos de generar un objeto (*data frame*) denominado `canarias`!

`header=T` hace referencia a que nuestros datos copiados tenían en su primera fila el encabezamiento con el nombre de las columnas – variables. `sep="\t"` indica que el separador entre celdas, o valores numéricos, es el "tabulador". `dec="."` determina que el separador decimal es el "punto"; si nuestro ordenador está configurado para que los decimales se definan con "comas" tendremos que escribir `dec=","` en la anterior línea de código.

A continuación vamos a cargar unos paquetes de R con los que vamos a realizar nuestros análisis:

```
library(plsdepot)
library(car)
library(lmtest)
library(sandwich)
library(psych)
```

Observemos el contenido de nuestro juego de datos (*data frame*) llamado "`canarias`", con un tamaño muestral de 12 islas.

```
names(canarias)
 [1] "codisla"      "isla"         "spp"          "km2"          "altmax"       "distafrica"
 [7] "antiguedad"   "lnkm2"        "lnaltmax"     "lndistafrica" "lnantiguedad" "complveg"
[13] "divhabitats"  "pmonteverde"  "ppinares"     "ptabcard"     "pmatorral"    "ppsammof"
[19] "pcultivos"    "pvolcanico"   "psabinpalm"
```

Y con la siguiente línea, vamos a tener una **descripción de las variables numéricas** que contiene:

```
describe(canarias)
```

También podríamos haber obtenido una tabla de resultados más selectiva escribiendo la siguiente línea que selecciona algunas columnas dentro de la tabla de salida de los resultados (ved qué columnas son): `describe(canarias)[,c(1:4,8:9,11:13)]`.

Columnas	1	2	3	4	5	6	7	8	9	10	11	12	13	esto lo he puesto yo
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se	
codisla	1	12	6.50	3.61	6.50	6.50	4.45	1.00	12.00	11.00	0.00	-1.50	1.04	
isla*	2	12	NaN	NA	NA	NaN	NA	Inf	-Inf	-Inf	NA	NA	NA	
spp	3	12	29.17	16.85	36.50	29.70	21.50	2.00	51.00	49.00	-0.22	-1.60	4.86	
km2	4	12	649.00	766.04	330.50	572.70	487.03	2.00	2059.00	2057.00	0.68	-1.27	221.14	
altmax	5	12	1130.42	1127.15	738.50	977.20	948.86	75.00	3718.00	3643.00	0.93	-0.31	325.38	
distafrica	6	12	235.33	116.82	182.50	231.10	108.23	96.00	417.00	321.00	0.34	-1.72	33.72	
antiguedad	7	12	8.40	9.09	6.00	7.53	8.84	0.03	25.50	25.47	0.47	-1.38	2.62	
lnkm2	8	12	4.68	2.76	5.79	4.79	2.61	0.61	7.63	7.02	-0.35	-1.74	0.80	
lnaltmax	9	12	6.44	1.25	6.60	6.48	1.48	4.32	8.22	3.90	-0.21	-1.45	0.36	
lndistafrica	10	12	5.34	0.51	5.20	5.35	0.63	4.56	6.03	1.47	0.05	-1.70	0.15	
lnantiguedad	11	12	0.41	2.74	1.50	0.52	2.22	-3.51	3.24	6.75	-0.39	-1.77	0.79	
complveg	12	12	1.67	0.31	1.61	1.66	0.21	1.19	2.17	0.98	0.32	-1.17	0.09	
divhabitats	13	12	1.22	0.73	1.14	1.25	0.82	0.00	2.17	2.17	-0.06	-1.58	0.21	
pmonteverde	14	12	0.05	0.07	0.00	0.04	0.00	0.00	0.20	0.20	0.94	-0.95	0.02	
ppinares	15	12	0.05	0.09	0.00	0.04	0.00	0.00	0.27	0.27	1.32	0.49	0.03	
ptabcard	16	12	0.15	0.11	0.13	0.15	0.15	0.00	0.29	0.29	0.07	-1.84	0.03	
pmatorral	17	12	0.46	0.35	0.46	0.44	0.49	0.04	1.00	0.96	0.10	-1.80	0.10	
ppsammof	18	12	0.05	0.11	0.00	0.02	0.01	0.00	0.39	0.39	2.41	4.63	0.03	
pcultivos	19	12	0.19	0.20	0.15	0.18	0.22	0.00	0.50	0.50	0.22	-1.83	0.06	
pvolcanico	20	12	0.04	0.07	0.00	0.03	0.00	0.00	0.20	0.20	1.34	0.29	0.02	
psabinpalm	21	12	0.01	0.03	0.00	0.00	0.00	0.00	0.12	0.12	2.59	5.28	0.01	

	vars	n	mean	sd	min	max	skew	kurtosis	se
codisla	1	12	6.50	3.61	1.00	12.00	0.00	-1.50	1.04
isla*	2	12	NaN	NA	Inf	-Inf	NA	NA	NA
spp	3	12	29.17	16.85	2.00	51.00	-0.22	-1.60	4.86
km2	4	12	649.00	766.04	2.00	2059.00	0.68	-1.27	221.14
altmax	5	12	1130.42	1127.15	75.00	3718.00	0.93	-0.31	325.38
distafrica	6	12	235.33	116.82	96.00	417.00	0.34	-1.72	33.72
antiguedad	7	12	8.40	9.09	0.03	25.50	0.47	-1.38	2.62
lnkm2	8	12	4.68	2.76	0.61	7.63	-0.35	-1.74	0.80
lnaltmax	9	12	6.44	1.25	4.32	8.22	-0.21	-1.45	0.36
lndistafrica	10	12	5.34	0.51	4.56	6.03	0.05	-1.70	0.15
lnantiguedad	11	12	0.41	2.74	-3.51	3.24	-0.39	-1.77	0.79
complveg	12	12	1.67	0.31	1.19	2.17	0.32	-1.17	0.09
divhabitats	13	12	1.22	0.73	0.00	2.17	-0.06	-1.58	0.21
pmonteverde	14	12	0.05	0.07	0.00	0.20	0.94	-0.95	0.02
ppinares	15	12	0.05	0.09	0.00	0.27	1.32	0.49	0.03
ptabcard	16	12	0.15	0.11	0.00	0.29	0.07	-1.84	0.03
pmatorral	17	12	0.46	0.35	0.04	1.00	0.10	-1.80	0.10
ppsammof	18	12	0.05	0.11	0.00	0.39	2.41	4.63	0.03
pcultivos	19	12	0.19	0.20	0.00	0.50	0.22	-1.83	0.06
pvolcanico	20	12	0.04	0.07	0.00	0.20	1.34	0.29	0.02
psabinpalm	21	12	0.01	0.03	0.00	0.12	2.59	5.28	0.01

Esta es la variable respuesta, cuyo número de orden (columna) es el 3.

Estas marcadas en verde son las variables que vamos a utilizar como predictoras; son las columnas 8-13.

NO DEBEMOS CONFUNDIR ESTOS NÚMEROS DE COLUMNAS DE LA MATRIZ ORIGINAL DE DATOS “canarias”, CON LOS ORDENES DE COLUMNAS DE LA TABLA describe QUE HEMOS CREADO ANTES.

Nuestra variable respuesta es la riqueza de especies en cada isla, `canarias$spp`, la tercera variable de nuestros datos `canarias`. Y las variables predictoras son las columnas 8-a-13 de los datos `canarias` (utilizadas en logaritmo neperiano … por aquello de la linealización de su efecto sobre la variable respuesta "riqueza de especies").

Vamos a construir un modelo general lineal (`modelo.lm`) con esas variables, según la ecuación (`eqt`) siguiente:

```
eqt <- as.formula(spp ~ lnkm2+lnaltmax+lndistafrica+lnantiguedad+complveg+divhabitats)
modelo.lm <- lm(eqt, data=canarias)
summary(modelo.lm)

Call:
lm(formula = eqt, data = canarias)
Residuals:
      1       2       3       4       5       6       7       8       9      10      11      12
 0.6876  5.0293 -1.0801 -0.4461  1.5628 -3.5404 -3.1997  0.9019 -0.2349  1.9881 -1.8961  0.2276

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.2076    17.7600   0.518    0.626
lnkm2          2.3176     1.8648   1.243    0.269
lnaltmax       5.1097     4.1074   1.244    0.269
lndistafrica  -5.9300     3.6905  -1.607    0.169
lnantiguedad   0.9181     0.7412   1.239    0.270
complveg       0.7153     7.0354   0.102    0.923
divhabitats    5.1682     3.9149   1.320    0.244

Residual standard error: 3.488 on 5 degrees of freedom
Multiple R-squared:  0.9805,   Adjusted R-squared:  0.9572
F-statistic: 41.97 on 6 and 5 DF,  p-value: 0.000405
```

¡Sorprendente! El modelo es **muy significativo** (p=0.000405), explica un **elevado porcentaje de la varianza** observada en la variable respuesta (98.05%), pero … **NINGUNA VARIABLE PREDICTORA ES SIGNIFICATIVA** (mirad la columna `Pr(>|t|)`).

Veamos qué pasa con la partición de la varianza a través de la suma de cuadrados (SS) obtenida mediante:

```
Anova(modelo.lm, test="F", type=3)
```

```
Anova Table (Type III tests)

Response: spp
             Sum Sq Df F value Pr(>F)
(Intercept)   3.269  1  0.2688 0.6263
lnkm2        18.788  1  1.5447 0.2690
lnaltmax     18.824  1  1.5476 0.2686
lndistafrica 31.404  1  2.5819 0.1690
lnantiguedad 18.664  1  1.5344 0.2704
complveg      0.126  1  0.0103 0.9230
divhabitats  21.197  1  1.7427 0.2440
Residuals    60.816  5
```

Para saber la suma de cuadrados total de la variable respuesta `canarias$spp` calculemos lo siguiente:

```
sum((canarias$spp-mean(canarias$spp))^2)     ## cada datos menos la media de la variable, al cuadrado
[1] 3123.667
```

Si la suma de cuadrados total vale `3123.667` y la suma de cuadrados residual es `60.816`, entonces la suma de cuadrados de nuestro modelo es `3123.667-60.816 = 3062.851` y explica el siguiente porcentaje de varianza (que es lo mismo que observamos antes en `summary(modelo.lm)`):

```
((3123.667-60.816)/3123.667)*100
[1] 98.05306
```

El **problema** surge **cuando** la suma de los cuadrados de los efectos principales (valores azules en `Sum Sq`) no es igual a la suma de cuadrados del modelo; en este ejemplo la suma de los valores `Sum Sq` es 112.3, en vez de 3062.8 que resulta de calcular `3123.667-60.816`. Dicho de otra manera, hay 2950.5 unidades de sumas de cuadrados que se deben a "concomitancias".

Y esto … ¿por qué? Porque las variables predictoras, también llamadas "independientes", **no son independientes entre si**. Esto lo podemos comprobar con la estima del **VIF** (*variance inflation factor*). http://en.wikipedia.org/wiki/Variance_inflation_factor.

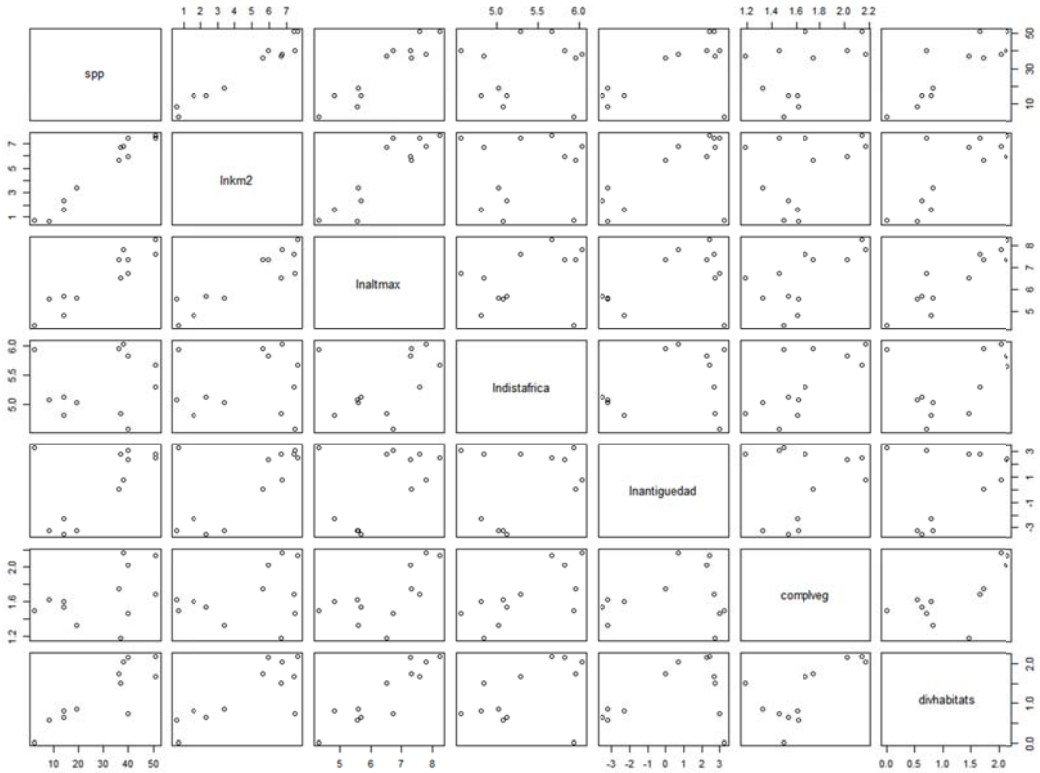
VIF = 1 / (1 - R^2), donde R^2 es lo explicado de cada predictora por todas las restantes. NO INCLUIMOS AQUÍ A LA RESPUESTA.

La raíz cuadrada de VIF indica aproximadamente cuántas veces está aumentado el error estándar del coeficiente de regresión de una variable predictora, debido a su no independencia (o existencia de colinealidad) con las variables predictoras restantes.

```
vif(modelo.lm)
       lnkm2     lnaltmax lndistafrica lnantiguedad     complveg  divhabitats
   23.947121    23.843827     3.224477     3.723799     4.249335     7.324930
```

Salvo para las variables predictoras `lndistafrica`, `lnantiguedad` y `complveg`, estos valores son ¡¡enormes!! Por ejemplo, la raíz cuadrada del VIF de `lnkm2` es 4.9. Por ende, el error estándar de su coeficiente de regresión se ha multiplicado aproximadamente por cinco, y su significación sólo puede disminuir considerablemente. Visualicemos la relación entre las variables de nuestra ecuación `eqt`.

```
pairs (eqt, data=canarias)
```



Podemos ver que hay fuertes relaciones entre las variables predictoras que comienzan en `lnkm2`.

Esto es indicativo de una completa falta de independencia entre ellas (i.e., alta colinealidad).

Dicho de otro modo, las variables predictoras son muy poco independientes entre si.

Además, tengamos en cuenta que estamos trabajando con un N=12 islas y seis variables predictoras, por lo que la potencia de nuestro análisis es pequeña, al ser proporcionalmente muy grande el número de variables explicativas (i.e., predictoras) respecto al número de unidades muestrales. Generalmente se recomienda que el tamaño muestral sea cuarenta veces mayor (x40) el número de variables independientes para tener una buena potencia de análisis.

La regresión por el método de los mínimos cuadrados parciales (*Partial Least Squares regression*, PLS) es una técnica estadística que permite abordar elegantemente estos problemas. En esencia, su objetivo es abordar el análisis del efecto que múltiples variables predictoras relacionadas entre si tienen sobre una variable respuesta. Para ello, encuentra combinaciones lineales de las variables predictoras que simultánemanete maximizan la explicación de la variable respuesta. Esto es, no se obtienen combinaciones lineales entre variables explicativas teniendo en cuenta sólo las relaciones que se establecen entre ellas, como hace por ejemplo el Análisis de las Componentes Principales (PCA), sino que en esa definición de "componentes" del PLS se tiene en cuenta la maximización de la explicación de la variable respuesta. El PLS puede obtener tantas componentes como variables predictoras hay, de manera que cada una de ellas va explicando una secuencia menor de la variabilidad en la variable respuesta. El criterio de parada en la obtención de componentes del PLS puede establecerse sobre la base de la consistencia en la obtención de las componentes PLS de las variables predictoras (obtenida mediante *crossvalidation*), y teniendo en cuenta la significación del efecto de esas componentes explicando la variable respuesta. El PLS manifiesta un grado similar de explicación de la variable respuesta que la regresión múltiple (RM), o la combinación de PCA+RM. Sin embargo, el PLS es más potente y robusto identificando las variables predictoras relevantes que tienen efecto sobre la variable respuesta, así como las magnitudes de sus efectos parciales, especialmente cuando las variables predictoras están muy relacionadas entre si y cuando el tamaño muestral es pequeño en relación con el número de las variables predictoras. Por último, es una herramienta especialmente diseñada para poder abordar el análisis simultáneo de varias variables respuesta que estén relacionadas entre si, en función de múltiples predictoras. Una presentación sobre esta técnica, y un análisis comparado con la regresión múltiple (RM) y con la combinación PCA+RM puede encontrarse en el siguiente documento disponible en internet: http://www.lmcarrascal.eu/pdf/plsr.pdf.

En R podemos contar con varios paquetes para llevar a cabo el PLS. Vamos a comenzar con el paquete "`plsdepot`" y su comando `plsreg1` que aborda el análisis de una variable respuesta. El comando `plsreg2` permite analizar varias variables respuesta.

Tras esta brevísima presentación del PLS prosigamos con el análisis de la riqueza de las aves terrestres canarias y los rasgos ambientales de 12 islas teniendo en cuenta seis variables explicativas.

Trabajando con el comando `plsreg1` definimos las variables predictoras y respuesta de un modo diferente a lo que se suele utilizar en modelos GLM: mediante su orden numérico como "columnas" en el *data frame* de análisis (`canarias` en este caso). La variable respuesta es la columna número 3, mientras que las variables predictoras tienen sus posiciones en las columnas 8 a 13. Por otro lado, al construir el modelo PLS, tenemos que definir *a priori* el número de componentes PLS que queremos obtener (`comps=`; que vamos a definir como un scalar denominado `nc`); lo vamos a establecer en el número máximo posible = número de variables predictoras = 6. Mediante el argumento `crosval=TRUE` especificamos que queremos hacer una valoración de la consistencia del significado de las componentes PLS construidas con las seis variables predictoras mediante *crossvalidation*.

```
nc <- 6
modelo.pls <- plsreg1(canarias[,c(8:13)], canarias[,3], comps=nc, crosval=TRUE)
```

Con la siguiente línea de código podemos obtener qué información contiene el objeto `modelo.pls` que acabamos de crear.

```
names(modelo.pls)
 [1] "x.scores"  "x.loads"   "y.scores"  "y.loads"   "cor.xyt"   "raw.wgs"   "mod.wgs"   "std.coefs"
 [9] "reg.coefs" "R2"        "R2Xy"      "y.pred"    "resid"     "T2"        "Q2"        "y"
```

Obtengamos ahora los primeros resultados, comenzando por la "consistencia" de las componentes del PLS en función de cómo se construyen considerando las variables predictoras (`Q2`) y cuánto explican esas componentes de la variable respuesta (`R2`).

```
modelo.pls$Q2
      PRESS        RSS          Q2  LimQ2     Q2cum
1 1.8906430 11.0000000  0.82812336 0.0975 0.8281234
2 0.3947274  1.2333369  0.67995167 0.0975 0.9449912
3 0.2593823  0.2375681 -0.09182283 0.0975 0.9399401
4 0.3142743  0.2167175 -0.45015691 0.0975 0.9129037
5 0.2820562  0.2146613 -0.31395952 0.0975 0.8855590
6 0.2822863  0.2143688 -0.31682533 0.0975 0.8493012
```

```
round(modelo.pls$R2, 4)
    t1     t2     t3     t4     t5     t6
0.8879 0.0905 0.0019 0.0002 0.0000 0.0000
```

La secuencia de componentes (1…6) obtenidas deben tener unos valores de `Q2` mayores que el umbral crítico `LimQ2` establecido en el valor `0.0975`. Este criterio sólo se cumple con las dos primeras componentes marcadas en verde. Un criterio más "relajado" podría haber sido quedarse con las componentes en las que `Q2` >0 si, y sólo si, el análisis posterior indicase que aquella componente PLS 0 < `Q2` < `LimQ2` es, además, significativa. Según el autor del paquete `plsdepot`, en el proceso de *crossvalidation* "… *the data set is randomly split in 10 segments of approximately equal size. Then, the observations in one of the segments are left outside as a test set. The other nine segments are used as learning set to estimate a model and predict the observations in the test segment. This procedure is applied consecutively for each of the 10 segments*".

El porcentaje de la variabilidad en la variable respuesta explicado por la secuencia de componentes PLS disminuye abruptamente desde la primera componente hasta la última, de manera que sólo las dos primeras componentes (que eran `Q2` > 0 y > `LimQ2`) explican el `0.8879`+`0.0905` = 97.84% de la variación observada en la riqueza de especies de aves terrestres canarias.

Las correlaciones entre las componentes PLS y las variables predictoras y respuesta se pueden obtener llamando a los valores `cor.xyt` del objeto `modelo.pls`.

```
round(modelo.pls$cor.xyt, 4)
                t1      t2      t3      t4      t5      t6
lnkm2       0.9146  0.3759 -0.0399  0.0461  0.0665 -0.1186
lnaltmax    0.9633  0.0459  0.2068  0.0915  0.1202  0.0667
lndistafrica 0.4275 -0.8490 -0.1957 -0.0687  0.2251 -0.0519
lnantiguedad 0.6206  0.1214 -0.7611  0.0532 -0.1240  0.0509
complveg    0.6597 -0.6266  0.2488  0.1809 -0.2783  0.0091
divhabitats 0.9231 -0.0985  0.2483 -0.2744 -0.0132  0.0312
Y           0.9423  0.3009  0.0435  0.0137  0.0052  0.0043
```

Los valores al cuadrado de las correlaciones entre la variable respuesta **Y** y las componentes del PLS **t1 … t6** coinciden con la salida previa de `modelo.pls$R2` (`0.8879 0.0905 0.0019 0.0002 0.0000 0.0000`).

Tras haber observado que sólo dos componentes son claramente importantes y consistentes por *crossvalidation*, también podemos ejecutar un análisis PLS mediante la siguiente línea de código, donde `comps=NULL` especifica que se obtengan las componentes estrictamente necesarias:

```
modelo.pls <- plsreg1(canarias[,c(8:13)], canarias[,3], comps=NULL, crosval=TRUE)
```

Vamos a crear ahora dos matrices de datos derivadas de nuestro modelo `modelo.pls` que contienen las posiciones de las observaciones (islas) en las componentes PLS, referidas a la variable respuesta (`modelo.pls$y.scores`) y a las predictoras (`modelo.pls$x.scores`).

```
modelo.pls.yscores <- as.data.frame(modelo.pls$y.scores)
modelo.pls.xscores <- as.data.frame(modelo.pls$x.scores)
```

Sus contenidos son los siguientes:

```
modelo.pls.yscores
           u1          u2
1   2.5689841 -0.17002370
2   2.5689841  1.83670567
3   1.2746868  1.73941198
4   1.2746868 -1.03956231
5   0.8040332 -0.36278338
6   1.0393600 -1.94840229
7   0.9216966  0.96759889
8  -1.1962445  0.86102143
9  -1.7845615  0.06391536
10 -1.7845615  0.52714382
11 -2.4905418 -0.61430791
12 -3.1965222 -1.86071756
```

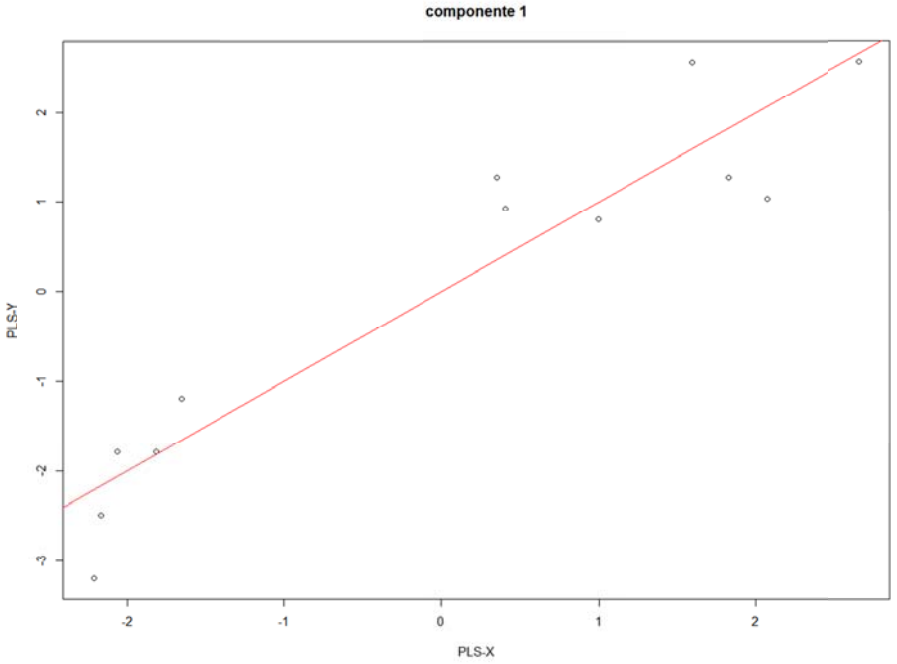
```
modelo.pls.xscores
           t1          t2
1   2.6591646 -0.39626220
2   1.5947959  0.73569411
3   0.3521031  2.00410379
4   1.8260703 -0.85097130
5   0.9964534 -0.75522587
6   2.0727920 -1.25044335
7   0.4084824  1.88417511
8  -1.6529300  0.56429028
9  -1.8184621 -0.08412697
10 -2.0641584  0.08002086
11 -2.1647131 -0.41971894
12 -2.2095981 -1.51153551
```

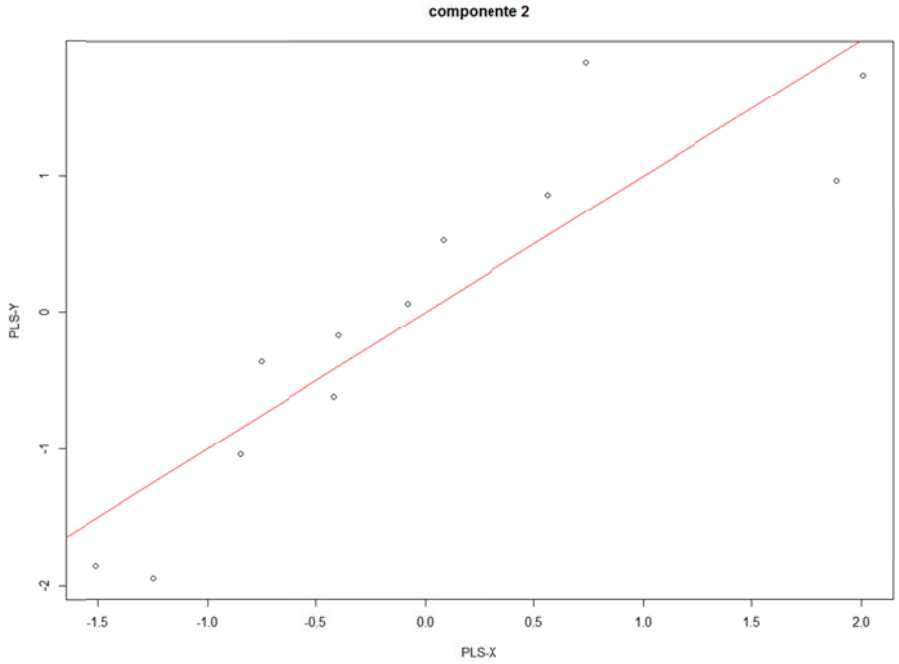
Con estas posiciones de las unidades muestrales en las componentes PLS 1 y 2, teniendo en cuenta la estructura de relaciones entre las variables predictoras (factores `t1` y `t2`) y la variación en la variable respuesta (factores `u1` y `u2`), podemos efectuar representaciones de las relaciones existentes y la **significación de las componentes explicando la respuesta** mediante un sencillo análisis de regresión:

```
plot(modelo.pls.xscores[,1], modelo.pls.yscores[,1], xlab=c("PLS-X"), ylab=c("PLS-Y"), main="componente 1")
abline(lm(modelo.pls.yscores[,1]~modelo.pls.xscores[,1]), col="red")
```

```
plot(modelo.pls.xscores[,2], modelo.pls.yscores[,2], xlab=c("PLS-X"), ylab=c("PLS-Y"), main="componente 2")
abline(lm(modelo.pls.yscores[,2]~modelo.pls.xscores[,2]), col="red")
```

```
modelo.pls.lm <- lm(modelo.pls$y~., data=modelo.pls.xscores)
```



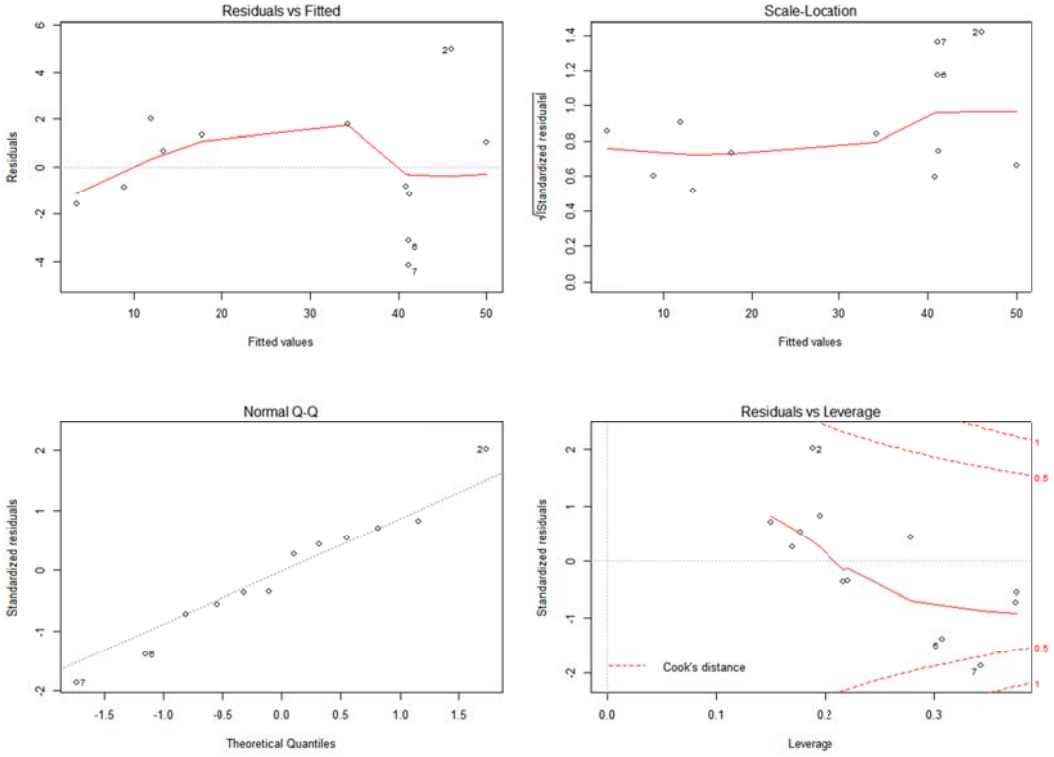



```
summary(modelo.pls.lm)
Call:
lm(formula = modelo.pls$y ~ ., data = modelo.pls.xscores)
Residuals:
    Min      1Q  Median      3Q     Max
-4.1317 -1.2884 -0.0914  1.4455  4.9631
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  29.1667     0.7903  36.904 3.90e-11 ***
t1            8.4988     0.4418  19.235 1.28e-08 ***
t2            4.5078     0.7339   6.142  0.00017 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 2.738 on 9 degrees of freedom
Multiple R-squared:  0.9784,      Adjusted R-squared:  0.9736
F-statistic: 203.9 on 2 and 9 DF,  p-value: 3.197e-08
```

Como podemos ver, las dos componentes PLS de las predictoras (`t1` y `t2`) son muy significativas (valoradlo mirando `Pr(>|t|)`) y se obtiene un modelo con ellas globalmente muy significativo (**p << 0.001**) que explica el **97.8%** de la variación observada en la variable respuesta, riqueza de especies. Esta última cantidad se corresoponde con la obtenida en `modelo.pls$R2` sumando las dos primeras componentes `t1` y `t2`.

Y podemos comprobar cómo el modelo no viola los **supuestos canónicos, explorando sus residuos**.

```
plot(modelo.pls.lm)
```



No hay un claro patrón de heterocedasticidad en los residuos (panel superior izquierdo).

No existe un patente desvío de la normalidad en los residuos (panel inferior izquierdo). El test de Shapiro no identifica un desvío significativo de la normalidad en los residuos del modelo:

```
shapiro.test(residuals(modelo.pls.lm))
        Shapiro-Wilk normality test
data:  residuals(pls.lm)
W = 0.9733, p-value = 0.9419
```

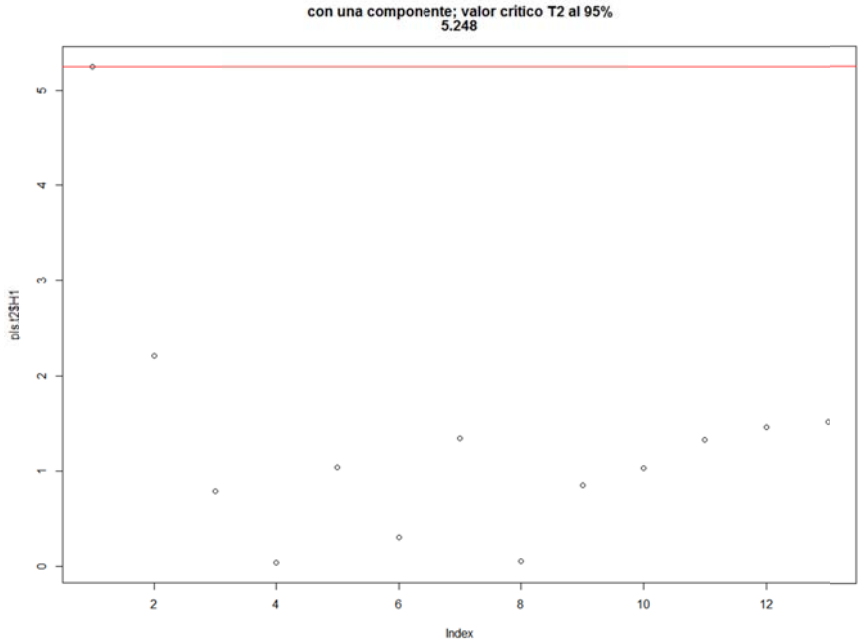
No hay “claros” puntos influyentes o perdidos (panel inferior derecho).

Aparte de esta prueba global de lo adecuado que es nuestro modelo PLS teniendo en cuenta las dos primeras componentes, también es necesario efectuar una exploración de hasta qué punto existen **unidades muestrales muy atípicas** desviadas del patrón multivariante del PLS (*outliers*). Este aspecto lo podemos valorar mediante los valores T2 de Hotelling de cada unidad muestral llamando a los valores de `T2` incluidos en el objeto `modelo.pls`. No encontramos observaciones claramente atípicas.

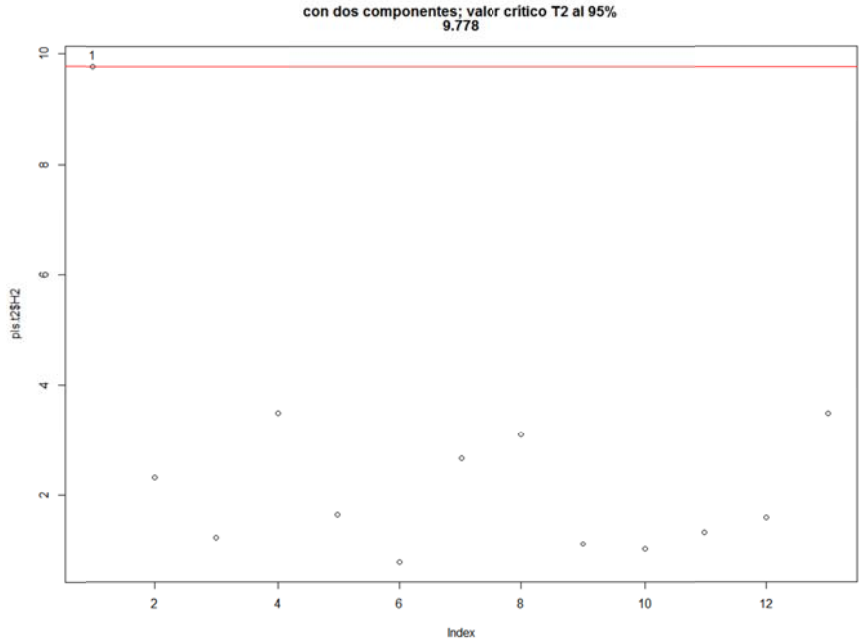
```
pls.t2 <- as.data.frame(modelo.pls$T2)    ## los umbrales críticos de T2 a p=0.05 están en la primera fila

pls.t2[1,]
         H1       H2
T2 5.24803 9.77839                        ## H1 se refiere a la primera componente; H2 a LAS DOS PRIMERAS componentes
```

```
plot(pls.t2$H1, main=c("con una componente; T2 al 95%",
round(pls.t2[1,1],3)))
abline(h=pls.t2[1,1], col="red")
```

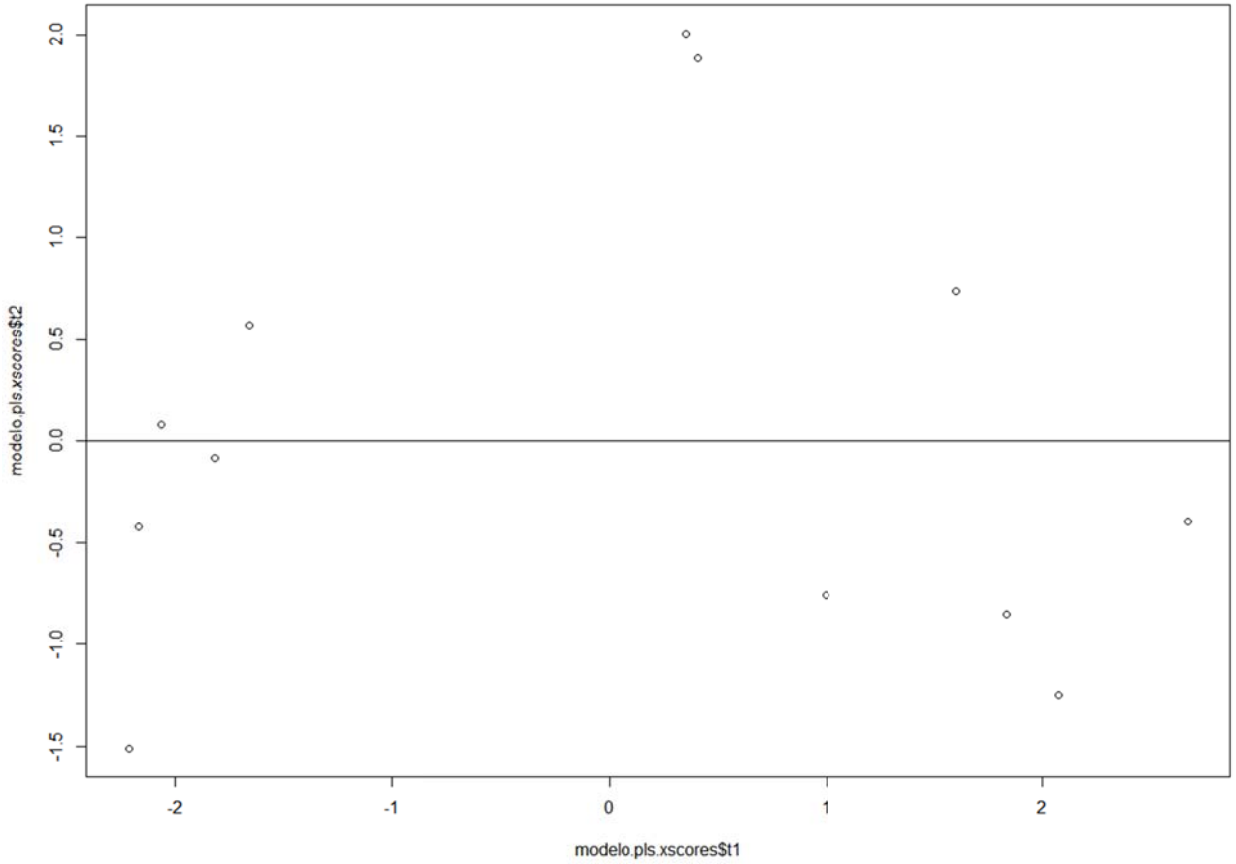



```
plot(pls.t2$H2, main=c("con dos componentes; T2 al 95%",
round(pls.t2[1,2],3)))
abline(h=pls.t2[1,2], col="red"); identify(pls.t2$H2)
```



¡Ah! Y las componentes `t1` y `t2` son perfectamente ortogonales como podemos ver a continuación:

```
plot(modelo.pls.xscores$t1, modelo.pls.xscores$t2)
> abline(lm(modelo.pls.xscores$t2~modelo.pls.xscores$t1, col="red"))
```



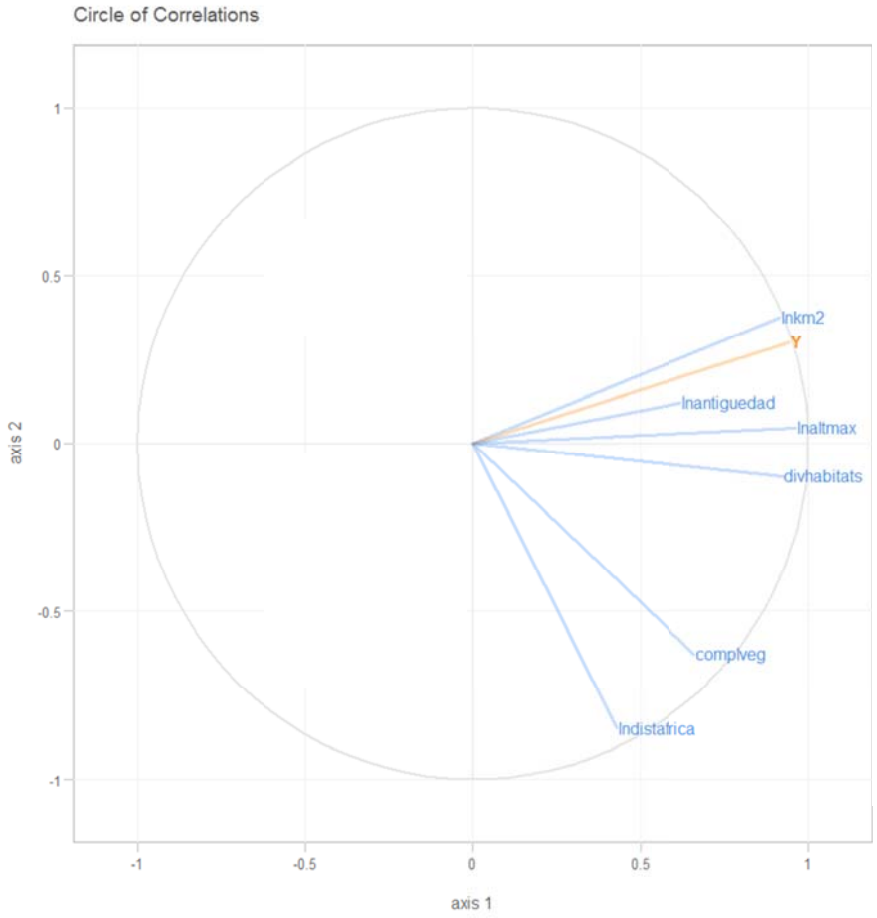
La correlación entre estas componentes es nula:

```
round(cor(modelo.pls.xscores$t1, modelo.
pls.xscores$t2)^2, 10)
[1] 0
```

Esto es, tienen contenido informativo diferente, como veremos a continuación.

Las **asociaciones entre las variables (respuesta y predictoras) y las** dos **componentes** generadas por el PLS pueden visualizarse haciendo uso de las correlaciones previamente obtenidas mediante `modelo.pls$cor.xyt`. Para una representación gráfica corremos la siguiente línea de código:

```
plot(modelo.pls, comps=c(1,2))     ## para las dos primeras componentes
```



Se representan los valores de correlación en el rango -1 / +1 para las componentes PLS1 (axis 1) y PLS2 (axis 2).

Cuanto más cerca esté una variable del límete gris de la circunferencia, mejor estará representada su variación en el modelo PLS. Este aspecto se valora por las líneas azules que salen del valor [0,0] para las predictoras, y la línea naranja para la variable respuesta.

La riqueza de especies de aves (variable respuesta **Y**) se asocia en este espacio, por proximidad, más con `lnkm2` y `lnaltmax`.

La variable respuesta **Y** se explica más por la componente 1 que por la componente 2, ya que su proyección sobre el `axis 1` es mayor que la observada sobre el `axis 2`.

Pero cuantitativamente es mejor trabajar con los **pesos (*weights*) de las variables en el modelo**. Estos pesos oscilan en el rango -1 / +1, denotando el signo y la intensidad del efecto. Pero además tienen una propiedad matemática muy útil: la suma de los cuadrados de los pesos de las variables dentro de cada componente SUMAN LA UNIDAD. De este modo podemos asignar a las componentes una proporción de la información que contienen de cada variable, teniendo en cuenta cómo "pesan" en ellas las distintas variables predictoras. Esta información la sacamos del objeto `modelo.pls` recurriendo a los valores incluidos en `raw.wgs`.

Valores de los pesos "*weights*"

```
round(modelo.pls$raw.wgs,4)
                 w1      w2
lnkm2        0.5530  0.3305
lnaltmax     0.5298  0.0738
lndistafrica 0.0789 -0.7800
lnantiguedad 0.3343  0.0109
complveg     0.2528 -0.5219
divhabitats  0.4813 -0.0666
```

Valores al cuadrado de los pesos "*weights*"

```
round(modelo.pls$raw.wgs^2,4)
                 w1     w2
lnkm2        0.3059 0.1092
lnaltmax     0.2806 0.0054
lndistafrica 0.0062 0.6085
lnantiguedad 0.1117 0.0001
complveg     0.0639 0.2723
divhabitats  0.2316 0.0044
```

La **interpretación** en este caso es sencilla. La componente PLS1 (mirad sus *weights* en `w1`) se asocia de modo positivo e intenso con la superficie (en logaritmo; `lnkm2`), la altitud máxima (en logaritmo; `lnaltmax`) y la diversidad de hábitats (`divhabitats`) de las islas. Sólo estas tres variables predictoras son responsables del `0.3059+0.2806+0.2316` = **`0.8181`** (en tanto por uno) del contenido informativo del PLS1, que a su vez, como ya vimos anteriormente, explicaba el 88.8% de la variación en la variable respuesta (obtenido mediante `modelo.pls$R2`). Esto es, las islas más grandes, que a su vez tienen mayores gradientes altitudinales y mayores diversidades de hábitat son aquellas que más riqueza de especies tienen, estableciéndose por tanto un síndrome ambiental que es imposible disociar de modo independiente, pero cuyos efectos y magnitudes podemos cuantificar con *weights*^{2}.

La segunda componente (PLS2) se asocia mayoritariamente con la distancia de las islas al continente africano (en logaritmo; `lndistafrica`), aunque "arrastra" la covariación positiva con el volumen-desarrollo vertical de la vegetación (`complveg`); de hecho, las islas más occidentales, más alejadas del continente, son más húmedas y la vegetación está mucho más desarrollada. Sólo estas dos variables son responsables del 88% del contenido informativo de esta componente, que a su vez sólo explicaba el 9.1% de la variación en la riqueza de especies (aunque de modo muy significativo; volved atrás a `summary(modelo.pls.lm)`).

Ante la **posibilidad de existencia de datos atípicos por muy influyentes o *outliers***, tal y como nos ha indicado el análisis con los valores de la T2 de Hotelling, podría ser conveniente llevar a cabo una estima robusta del modelo PLS mediante *bootstrapping*.

Preservamos el mismo nombre del modelo PLS (`modelo.pls`), asignamos a un nuevo *data frame* la matriz de datos de análisis (`datos.boot`), y definimos como argumentos dentro de la línea de comando `plsreg1` las variables predictoras y respuesta (marcadas en ROJO como números de orden de las columnas que especifican esas variables).

```
nv <- length(modelo.pls$raw.wgs)                              ## para contar cuántos valores de weights hay
ncomps <- 2                                                   ## especificamos cuántas componentes queremos obtener
datos.boot <- canarias                                        ## "duplicamos" la matriz original para los remuestreos
nparametros <- nv+ncomps
bootstrap <- matrix(99999, nrow=2000, ncol=nparametros)       ## matriz en la que se guardan los resultados
##
for (i in 1:2000)  {
    iboot <- sample(1:nrow(datos.boot), replace=TRUE)
    matboot <- datos.boot[iboot,]
    modboot <- plsreg1(matboot[,c(8:13)], matboot[,3], comps=ncomps, crosval=FALSE)
    wr <- as.vector(modboot$raw.wgs)
    R2r <- as.vector(modboot$R2)
    salida <- append(wr, R2r)
    bootstrap[i,1:nparametros] <- salida
}
##
mbootstrap <- as.data.frame(bootstrap)
nombres <- c(rownames(modelo.pls$raw.wgs), rownames(modelo.pls$raw.wgs), names(modelo.pls$R2))
colnames(mbootstrap) <- nombres
nn <- length(mbootstrap)
##
## QUE LOS SIGUIENTES INTERVALOS NO INCLUYAN EL VALOR CERO
for (j in 1:ncomps) {
  for (i in 1:((nn-ncomps)/ncomps)) {
    ni <- (j-1)*nv/ncomps+i
    print(c("COMPONENTE -", j, nombres[ni], "promedio =", mean(mbootstrap[,ni])), quote=FALSE)
    print(quantile(mbootstrap[,ni], c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995)))
     print("-----------------------------------------------------", quote=FALSE)
  }
  nj <- nn-ncomps+j
  print(c("COMPONENTE -",j,"R2", "promedio =", mean(mbootstrap[,nj])), quote=FALSE)
  print(quantile(mbootstrap[,nj], c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995)))
  print("*****************************************************", quote=FALSE)
  print("*****************************************************", quote=FALSE)
}
```

Para la primera componente:

```
QUE LOS SIGUIENTES INTERVALOS NO INCLUYAN EL VALOR CERO
[1] COMPONENTE -      1                lnkm2             promedio =        0.538108421775554
     0.5%      2.5%        5%       95%     97.5%     99.5%
0.4425896 0.4615352 0.4752851 0.5913237 0.5981219 0.6164230
[1] ----------------------------------------------------
[1] COMPONENTE -      1                lnaltmax          promedio =        0.514858805609538
     0.5%      2.5%        5%       95%     97.5%     99.5%
0.4269471 0.4423025 0.4525489 0.5654319 0.5722809 0.5880867
[1] ----------------------------------------------------
[1] COMPONENTE -      1                lndistafrica      promedio =        0.080883729247060
      0.5%       2.5%         5%        95%      97.5%      99.5%
-0.3450765 -0.2336816 -0.1708920  0.2958871  0.3241925  0.3637186
[1] ----------------------------------------------------
[1] COMPONENTE -      1                lnantiguedad      promedio =        0.328840763307468
       0.5%        2.5%          5%         95%       97.5%       99.5%
-0.11008535  0.01401132  0.07809774  0.51668019  0.52966220  0.54685509
[1] ----------------------------------------------------
[1] COMPONENTE -      1                complveg          promedio =        0.243689756058276
         0.5%          2.5%            5%           95%          97.5%          99.5%
-0.1723156791 -0.0001557215  0.0616436240  0.3924643345  0.4130684542  0.4461081949
[1] ----------------------------------------------------
[1] COMPONENTE -      1                divhabitats       promedio =        0.460424794923873
     0.5%      2.5%        5%       95%     97.5%     99.5%
0.2773221 0.3358979 0.3605023 0.5350024 0.5428612 0.5617249
[1] ----------------------------------------------------
[1] COMPONENTE -      1                R2                promedio =        0.909459291092573
     0.5%      2.5%        5%       95%     97.5%     99.5%
0.8102945 0.8434454 0.8532233 0.9696166 0.9767845 0.9867393
[1] ************************************************************
[1] ************************************************************
```

Con el intervalo de confianza del 95% (alfa = 0.05 con dos colas; cuantiles 2.5% y 97.5%) el peso (weight) de la variable `lnkm2` se encuentra entre `0.4615352` y `0.5981219`, con un promedio de **`0.538108421775554`**. En esencia, este valor medio es muy similar a el observado efectuando el modelo PLS (`lnkm2 = 0.5530`) y es muy significativo incluso a p=0.01 (intervalo azul) porque el valor "cero" (hipótesis nula de ausencia de efecto) no se incluye en los intervalos al 95% y 99%. Otro tanto puede decirse para las variables `lnaltmax` y `divhabitats`. Para `lnantiguedad` el intervalo de confianza al 95% no incluye el "cero" pero sí se incluye en el del 99%. Para las variables **`lndistafrica`** y **`complveg`** los intervalos de confianza de los *weights* al 95% sí incluyen el valor "cero", siendo por tanto no significativas. Por último, el intervalo de confianza al 95% de la variación explicada de la variable respuesta por el PLS1 se encuentra entre el 84.3% y el 97.7%.

Para la segunda componente obtenemos los siguientes resultados que vosotros podéis interpretar de manera similar. Marcamos en **VERDE** aquellas significativas a alfa=0.05 porque no incluyen el valor "cero" de la hipótesis nula, y en **ROJO** los efectos que no serían significativos.

```
[1] COMPONENTE -       2                  lnkm2              promedio =         0.308845856165634
      0.5%       2.5%         5%         95%       97.5%       99.5%
0.08725551 0.15694745 0.19335384 0.42948410 0.46309061 0.53794752
[1] ----------------------------------------------------
[1] COMPONENTE -       2                  lnaltmax           promedio =         0.030967754231126
      0.5%       2.5%         5%         95%       97.5%       99.5%
-0.1934879 -0.1388995 -0.1182445  0.1836274  0.2097945  0.2655472
[1] ----------------------------------------------------
[1] COMPONENTE -       2                  lndistafrica       promedio =        -0.678687092718329
      0.5%       2.5%         5%         95%       97.5%       99.5%
-0.8978035 -0.8407331 -0.8253802 -0.5135410 -0.2892421  0.4984000
[1] ----------------------------------------------------
[1] COMPONENTE -       2                  lnantiguedad       promedio =         0.114549209703925
      0.5%       2.5%         5%         95%       97.5%       99.5%
-0.6439648 -0.4895872 -0.3159317  0.4501647  0.4941963  0.7659204
[1] ----------------------------------------------------
[1] COMPONENTE -       2                  complveg           promedio =        -0.483926477149165
       0.5%        2.5%          5%         95%       97.5%       99.5%
-0.74714711 -0.68436552 -0.65203307 -0.21910711  0.08984604  0.77894205
[1] ----------------------------------------------------
[1] COMPONENTE -       2                  divhabitats        promedio =        -0.124722110951703
      0.5%       2.5%         5%         95%       97.5%       99.5%
-0.5117017 -0.4219664 -0.3775221  0.1737622  0.2236549  0.3098492
[1] ----------------------------------------------------
[1] COMPONENTE -       2                  R2                   promedio =       0.069000420786802
       0.5%        2.5%          5%         95%       97.5%       99.5%
0.003528883 0.010267165 0.016244485 0.123409954 0.135015956 0.161545296
[1] ************************************************************
[1] ************************************************************
```

Fijaos en cómo a pesar del alto valor de `complveg = -0.5219` (obtenido de `modelo.pls$raw.wgs`) en esta segunda componente, su intervalo sí incluye el "cero", siendo su efecto de elevada magnitud pero poco estable en la segunda componente del PLS.
Para terminar, esta segunda componente PLS2 explica una escasa variación en la variable respuesta (oscila entre el 1.0%-13.5%).

El caso anteriormente presentado ilustra el uso del PLS en el contexto del análisis de una variable respuesta para la cual se asume una distribución gausiana. En el caso de contar con una variable respuesta que no siga dicha distribución, podremos utilizar otro paquete que “generaliza” la distribución más allá de la normal: `plsRglm`. Comencemos cargando el paquete:

```
library(plsRglm)
```

Con este paquete podemos trabajar con **OTROS TIPOS DE DISTRIBUCIONES DE LAS VARIABLES RESPUESTA** dentro del argumento "`modele`" en el comando `plsRglm`:

"pls-glm-gaussian" glm gaussian with identity link pls models
"pls-glm-poisson" glm poisson with log link pls models
"pls-glm-Gamma" glm gaussian with inverse link pls models
"pls-glm-logistic" glm binomial with logit link pls models
"pls-glm-polr" glm polr with logit link pls models (modelos multinomiales ordinales asimilables a "proportional odds regression")
"modele=pls-glm-family" allows changing the family and link function (e.g., family=poisson(link = "log"))
"pls" ordinary pls models

Vamos a ejemplificar su uso con el *data frame* `datos` que contiene las siguientes variables:

```
names(datos)
 [1] "ID"          "longitud"    "latitud"     "distmar"     "altmed"      "rangoalt"    "altmax"
 [8] "shannon"     "ice"         "fcaducif"    "fesclerof"   "fconif"      "fagroarb"    "furbano"
[15] "fmatorral"   "fagromos"    "fagua"       "fherbaceo"   "tempmin"     "tempmedia"   "precip"
[22] "ibakm2"      "zepakm2"     "ntransectos" "nspp"        "ptyrup"      "turmer"      "erirub"
[29] "stuuni"      "petpet"      "ptyrup01"    "turmer01"    "erirub01"    "stuuni01"    "petpet01"
[36] "ptyrup02"    "turmer03"    "erirub03"    "nturmer"     "nerirub"     "nstuuni"     "nptyrup"
[43] "npetpet"     "sizepa"      "zepa01"
```

En esta ocasión vamos a proceder a analizar el patrón de distribución invernal en España del Gorrión chillón, teniendo en cuenta su presencia/ausencia (`"petpet01"`). Por tanto, estamos ante una variable respuesta que se ajusta a una binomial [0 vs. 1].

Con el commando `plsRglm` se define primeramente la variable respuesta y a continuación las predictoras, identificando los números de orden de las variables implicadas. También tenemos que definir *a priori* con el argumento `nt` las componentes que queremos extraer. Con `sparse=TRUE, sparseStop=TRUE` podemos obtener componentes que incluyen sólo variables significativas.

Debido a que ya hemos presentado con exhaustividad el uso del PLS con el caso de las aves de las Islas Canarias, en esta ocasión vamos a ilustrar el desarrollo ordenado del análisis sin entrar en mucho detalle con las explicaciones.

La prevalencia de la especie en la muestra (presencia / ausencia en UTM's de 10x10 km^2) tiene un valor medio-bajo:

```
mean(datos$petpet01)
[1] 0.3483483
```

Creamos el modelo PLS, seleccionando primeramente la columna de la respuesta (**rojo** en `names(datos)`) y luego las de las predictoras (**verde** en `names(datos)`), y estableciendo la obtención de cinco componentes PLS (con **`nt=5`**):

```
modelo.pls <- plsRglm(datos[,35], datos[,c(4:6,8:9,14,19,21,23)], nt=5, modele="pls-glm-logistic")
____****************************************____
Family: binomial
Link function: logit
____Component____ 1 ____
____Component____ 2 ____
____Component____ 3 ____
____Component____ 4 ____
____Component____ 5 ____
____Predicting X without NA neither in X nor in Y____
****_____****
```

Tecleando el nombre del modelo en la consola de RStudio (`modelo.pls`) obtenemos sus resultados, de los cuales vamos a destacar los siguientes por su interés:

`Comp_0` indica que no hemos extraído ninguna componente; `Comp_1` que hemos extraido la primera; `Comp_2` que hemos extraido dos componentes; etc.

`AIC` y `BIC` son los valores que usan los criterios de la teoría de la información de Kullback–Leibler para los modelos con 0, 1, 2, 3 ... componentes extraídas. ¡¡¡ OJO !!! no se refieren a la primera, segunda, tercera, ... componentes. Dicho de otra manera, `Comp_0` sería el valor de AIC para un modelo nulo que no extrae ninguna componente PLS.

`R2_Y`: variabilidad explicada por 0, 1, 2, 3 ... componentes extraídas. Para extraer la R2 de la segunda calculad: `R2_Y_2` - `R2_Y_1`.

`Missclassed` proporciona el número de unidades muestrales incorrectamente clasificadas; en esta ocasión sobre un total de N=999.

```
modelo.pls
Number of required components:
[1] 5
Number of successfully computed components:
[1] 5
Coefficients:
                   [,1]
Intercept -4.327635e-01      ## estos coeficientes se calculan para cálculos intermedios
distmar   -5.733991e-07
altmed     1.236353e-03
rangoalt  -1.846715e-03
shannon    3.745556e-01
ice        1.630015e-01
furbano   -3.455473e-01
tempmin   -8.198759e-02
precip    -3.709415e-03
zepakm2   -4.872221e-03
Information criteria and Fit statistics:
               AIC      BIC Missclassed Chi2_Pearson_Y    RSS_Y      R2_Y  R2_residY RSS_residY
Nb_Comp_0 1293.544 1298.450         348        999.000 226.7748        NA         NA   226.7748
Nb_Comp_1 1125.256 1135.069         304       1009.969 191.3267 0.1563139  -8.770921  2215.7984
Nb_Comp_2 1100.043 1114.763         290       1091.558 184.5972 0.1859889 -10.087989  2514.4763
Nb_Comp_3 1099.469 1119.096         289       1107.733 184.1443 0.1879858 -10.269750  2555.6950
Nb_Comp_4 1098.256 1122.790         293       1062.142 183.9100 0.1890189 -10.254137  2552.1545
Nb_Comp_5 1099.669 1129.110         288       1054.726 184.0098 0.1885792 -10.391803  2583.3736
Model with all the required components:

Call:  glm(formula = YwotNA ~ ., family = family, data = tttrain)

Coefficients:
(Intercept)         tt.1         tt.2         tt.3         tt.4         tt.5
    -0.8449       0.7505       0.4568       0.1332       0.2093       0.1162

Degrees of Freedom: 998 Total (i.e. Null);  993 Residual
Null Deviance:     1292
Residual Deviance: 1088        AIC: 1100
```

Selección de componentes: Considerando los anteriores valores de **AIC** podemos percatarnos de que añadir una componente (`Nb_Comp_1`) genera un modelo con un alto grado de evidencia, ya que el valor de AIC disminuye en 168.288 unidades (restando del valor de `Nb_Comp_0` el valor de `Nb_Comp_1`).

```
1293.544 - 1125.256
[1] 168.288
```

Otro tanto resulta al extraer una segunda componente (`Nb_Comp_2`) efectuando la diferencia entre `Nb_Comp_1` y `Nb_Comp_2`: AIC disminuye 25.213 unidades.

```
1125.256 - 1100.043
[1] 25.213
```

Sin embargo, la disminución del valor de AIC al añadir una tercera componente (`Nb_Comp_3`) es muy pequeña, de menos de una unidad. Esto sugiere que deberíamos retener tan solo las dos primeras componentes del PLS.

Por tanto, repetimos el modelo reduciendo las componentes seleccionadas a dos con el argumento **`nt`**.

```
modelo.pls <- plsRglm(datos[,35], datos[,c(4:6,8:9,14,19,21,23)], nt=2, modele="pls-glm-logistic")
```

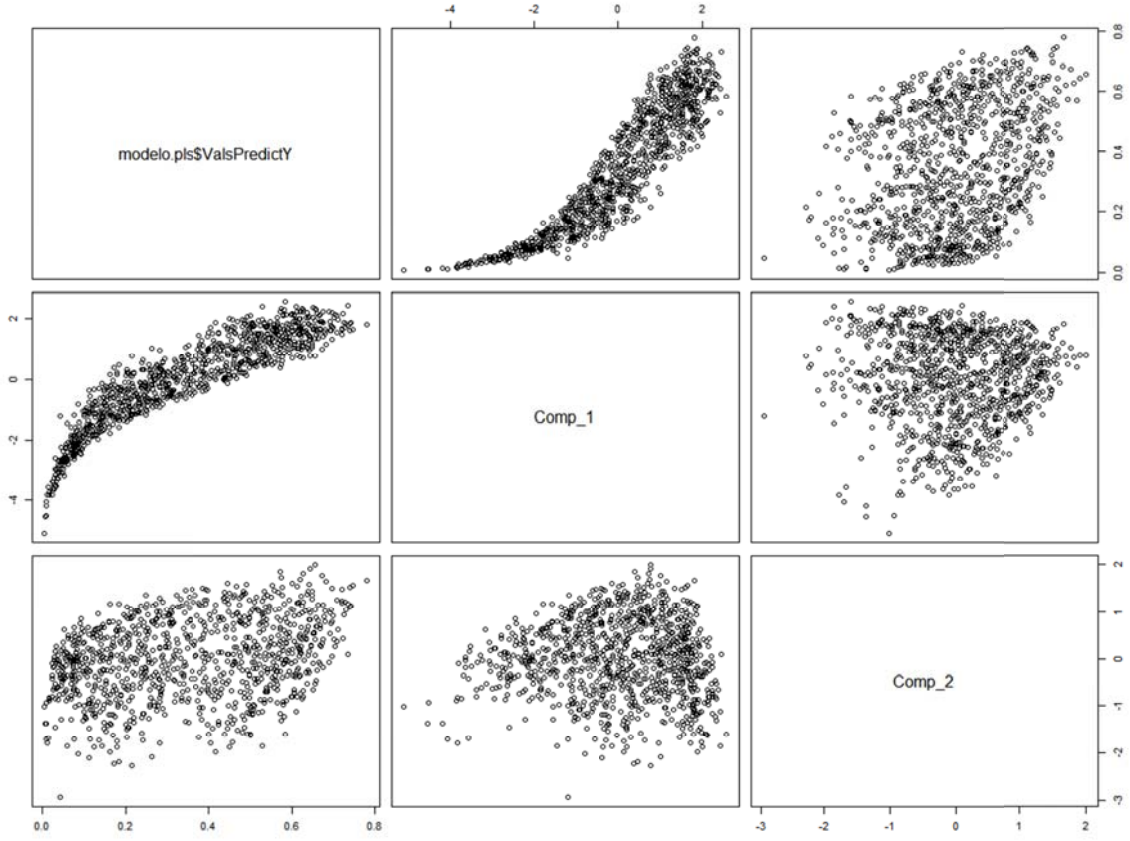
Mediante la siguiente línea de código podemos extraer la posición de las unidades muestrales en las componentes seleccionadas:

```
modelo.pls.xscores <- as.data.frame(modelo.pls$tt)
```

Y en la variable `ValsPredictY` incluida en el objeto `modelo.pls` encontramos las probabilidades de ocurrencia predichas para la especie de estudio (i.e., `modelo.pls$ValsPredictY`).

Las relaciones establecidas entre las dos componentes del PLS seleccionadas, y las predicciones de la variable respuesta, las vemos con:

```
pairs(~ modelo.pls$ValsPredictY +., data=modelo.pls.xscores)   ## +. denota suma la anterior a TODAS las de data
```



La relación es especialmente intensa entre la probabilidad de ocurrencia del Gorrión chillón (`modelo.pls$ValsPredictY`; variable respuesta) y la primera componente (`Comp_1`). Por otro lado, las dos componentes son ortogonales entre si (su R^2 = 0):

```
round(cor(modelo.pls.xscores$Comp_1, modelo.pls.xscores$Comp_2)^2, 8)
[1] 0
```

Otra manera de aproximarnos a la significación de las componentes explicando la variable respuesta (ocurrencia [0,1] del Gorrión chillón) sería construir un Modelo Generalizado Lineal con las dos componentes del PLS seleccionadas teniendo en cuenta el criterio de Akaike (AIC). Esto podemos obtenerlo sencillamente mediante la siguiente línea de código, teniendo en cuenta que la variable respuesta está contenida dentro del objeto `modelo.pls` en `modelo.pls$dataY`:

```
glm.pls.binom <- glm(modelo.pls$dataY~., data=modelo.pls.xscores, family=binomial(link="logit"))
```

Y ahora obtenemos los resultados de este modelo generalizado lineal binomial:

```
summary(glm.pls.binom)

Call:
glm(formula = modelo.pls$dataY ~ ., family = binomial(link = "logit"),
    data = modelo.pls.xscores)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6405  -0.8842  -0.4859   1.0074   2.9181
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.83681    0.08044 -10.403  < 2e-16 ***
Comp_1       0.74358    0.06439  11.549  < 2e-16 ***
Comp_2       0.45397    0.08895   5.104 3.33e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
 (Dispersion parameter for binomial family taken to be 1)
    Null deviance: 1291.5  on 998  degrees of freedom
Residual deviance: 1094.0  on 996  degrees of freedom
AIC: 1100
Number of Fisher Scoring iterations: 4
```

La devianza explicada por el modelo es un 15.3% (obtenido con las devianzas), valor indicativo de que es un modelo poco explicativo:

```
100*(glm.pls.binom$null.deviance - glm.pls.binom$deviance)/glm.pls.binom$null.deviance
[1] 15.29181
```

Y el coeficiente de sobredispersión del modelo es aceptable (aproximadamente 1):

```
sum(residuals(glm.pls.binom, type="pearson")^2)/glm.pls.binom$df.residual
[1] 1.095941
```

El poder del modelo prediciendo la presencia/ausencia del Gorrión chillón (la variable respuesta), independientemente de la probabilidad de corte que defina la presencia vs. la ausencia, podemos estimarlo mediante la estima del valor de AUC en el diagrama ROC. Para ello, antes debemos cargar el paquete `ROCR`:

```
library(ROCR)
```

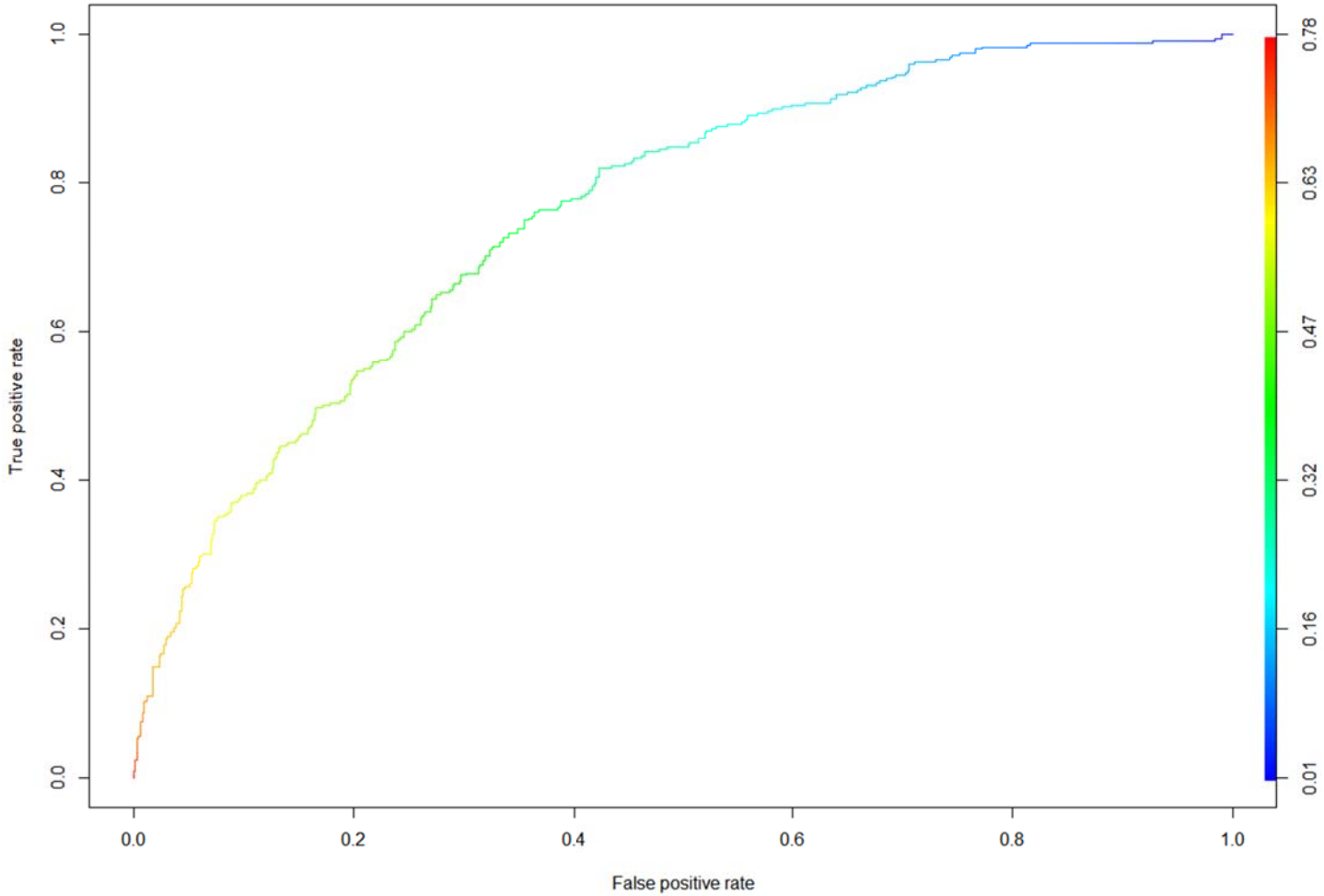
Mediante las siguientes líneas de código obtenemos la gráfica correspondiente y el valor de AUC:

```
pred.modelo <- prediction(fitted(glm.pls.binom), glm.pls.binom$y)
perf.modelo <- performance(pred.modelo,"tpr","fpr", measure="auc")
perf.modelo
An object of class "performance"
Slot "x.name":
[1] "None"
Slot "y.name":
[1] "Area under the ROC curve"
Slot "alpha.name":
[1] "none"
Slot "x.values":
list()
Slot "y.values":
[[1]]
[1] 0.760863                              ## ESTE ES EL VALOR DE AUC
Slot "alpha.values":
list()
```

De nuevo, y al igual que ocurría con la devianza explicada por el modelo generalizado lineal `glm.pls.binom` y con el valor de **R2_Y** generado por `modelo.pls`, el valor "mediano" de **AUC** obtenido informa de que las dos componentes del PLS tienen un papel "limitado" explicando la presencia / ausencia del Gorrión chillón.

El diagrama ROC lo podemos obtener mediante:

```
plot(performance(pred.modelo,"tpr","fpr"),colorize=TRUE)
```


True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0.78
0.63
0.47
0.32
0.16
0.01

Una vez filtradas aquellas componentes significativas, que tienen decrementos sustanciales de **AIC**, y que además retienen proporciones apreciables de la variación existente en la variable respuesta (**R2_Y**), pasamos a interpretar su significado mediante la estima de sus "pesos" (*weights*). Recordemos que el cuadrado de los pesos de las variables en cada componente siempre suma la unidad, con lo cual es fácil asignar la contribución proporcional de cada predictor a cada componente:

Pesos de las variables

```
modelo.pls$wwnorm
              Comp_1      Comp_2
distmar    0.31211341 -0.67604929
altmed     0.27147421  0.09091040
rangoalt  -0.37244331 -0.22614155
shannon   -0.04030065  0.41909947
ice       -0.14984765  0.36897196
furbano   -0.23160726 -0.08292085
tempmin   -0.47393773 -0.04165212
precip    -0.61404015 -0.32934787
zepakm2   -0.10387636 -0.23385294
```

Cuadrados de los pesos de las variables

```
modelo.pls$wwnorm^2
              Comp_1      Comp_2
distmar   0.097414779 0.457042642
altmed    0.073698245 0.008264700
rangoalt  0.138714022 0.051140001
shannon   0.001624142 0.175644367
ice       0.022454317 0.136140304
furbano   0.053641923 0.006875867
tempmin   0.224616972 0.001734899
precip    0.377045300 0.108470023
zepakm2   0.010790299 0.054687197
```

Por medio de la obtención de los cuadrados de los pesos (wi^2) inferimos la importancia de las variables en cada componente. Una aproximación "parsimoniosa", pero aproximada, consiste en enfatizar la importancia de aquellas variables que cumplan con el siguiente criterio:

wi^2 > [1 / nº de variables predictoras incluidas en el modelo]

En este caso, con nueve variables predictoras, que wi^2 sea > 1/9 = 0.111 (en **ROJO**)

La primera componente indica que la probabilidad de ocurrencia del Gorrión chillón se relaciona intensamente, y de modo negativo, con la `precip`, la `tempmin` y la `rangoalt`. Esta componente explica el 15.6% de la variación observada en la respuesta (mirad **R2_Y** en `modelo.pls`). La segunda componente, que sólo contribuye a explicar de la respuesta un 3% de su variación (`0.1859889 - 0.1563139 en modelo.pls`), se asocia principalmente de modo negativo con `distmar`, y positivamente pero de modo menos intenso con la diversidad de hábitats (`shannon`) y el desarrollo de la vegetación (`ice`).

Mediante la opción de **búsqueda de componentes que incluyan sólo variables significativas**, establecida por la inclusión de los argumentos `sparse=TRUE, sparseStop=TRUE` en el comando del modelo, obtenemos un resultado ligeramente diferente.

```
modelo.pls.sig <- plsRglm(datos[,35], datos[,c(4:6,8:9,14,19,21,23)], nt=5, sparse=TRUE, sparseStop=TRUE, modele="pls-glm-logistic")
```

En esta occasion, resultan tres components, que sucesivamente tienen decrementos de **AIC** mayores de 6 unidades:

```
modelo.pls.sig
Number of required components:
[1] 5
Number of successfully computed components:
[1] 3
Coefficients:
                  [,1]
Intercept  1.156613e-01
distmar   -7.584927e-07
altmed     7.528586e-04
rangoalt  -1.534320e-03
shannon    4.397604e-01
ice        6.726135e-02
furbano   -3.061979e-01
tempmin   -1.315629e-01
precip    -4.166544e-03
zepakm2   -3.757822e-03
Information criteria and Fit statistics:
               AIC      BIC Missclassed Chi2_Pearson_Y    RSS_Y      R2_Y  R2_residY RSS_residY
Nb_Comp_0 1293.544 1298.450         348        999.000 226.7748        NA         NA   226.7748
Nb_Comp_1 1123.309 1133.122         301       1012.026 190.8891 0.1582439  -8.852104  2234.2088
Nb_Comp_2 1105.229 1119.949         288       1056.162 185.8975 0.1802551  -9.852372  2461.0442
Nb_Comp_3 1098.562 1118.189         286       1097.301 184.1077 0.1881475 -10.395524  2584.2173
Model with all the required components:

Call:  glm(formula = YwotNA ~ ., family = family, data = tttrain)
Coefficients:
(Intercept)         tt.1         tt.2         tt.3
    -0.8471       0.7605       0.3854       0.2584
Degrees of Freedom: 998 Total (i.e. Null);  995 Residual
Null Deviance:     1292
Residual Deviance: 1091        AIC: 1099
```

Aunque ha salido elegida una tercera componente, su contribución a la explicación de la variación en la ocurrencia del Gorrión chillón es virtualmente despreciable, al retener solamente un 0.8% de la variación observada en la respuesta.

Los pesos de las variables en cada componente ahora sólo se restringen a aquellas variables con efectos significativos a p<0.05 en ellas, existiendo variables con pesos `0.0000000` porque no se incluyen en esas componentes. De cualquier modo, se sigue cumpliendo que la suma de los cuadrados de los pesos de las variables dentro de cada componente es igual a la unidad.

Pesos de las variables

```
modelo.pls.sig$wwnorm
              Comp_1     Comp_2     Comp_3
distmar    0.3123672 -0.8133314  0.0000000
altmed     0.2716949  0.0000000  0.0000000
rangoalt -0.3727461  0.0000000 -0.6607247
shannon    0.0000000  0.4172379  0.0000000
ice       -0.1499695  0.4054682  0.0000000
furbano   -0.2317956  0.0000000  0.0000000
tempmin   -0.4743231  0.0000000  0.7506283
precip    -0.6145394  0.0000000  0.0000000
zepakm2   -0.1039608  0.0000000  0.0000000
```

Cuadrado de los pesos de las variables

```
modelo.pls.sig$wwnorm^2
              Comp_1     Comp_2     Comp_3
distmar   0.09757325 0.6615080 0.0000000
altmed    0.07381814 0.0000000 0.0000000
rangoalt 0.13893968 0.0000000 0.4365571
shannon   0.00000000 0.1740875 0.0000000
ice       0.02249085 0.1644045 0.0000000
furbano   0.05372919 0.0000000 0.0000000
tempmin   0.22498238 0.0000000 0.5634429
precip    0.37765867 0.0000000 0.0000000
zepakm2   0.01080785 0.0000000 0.0000000
```

Estos resultados son muy similares a los previamente expuestos con el modelo `modelo.pls` (marcadas las coincidencias en **ROJO y negrita**), sólo que en esta ocasión se pone el énfasis en la selección de variables dentro de las componentes teniendo en cuenta que su efecto sea significativo en ella.

Otra manera de aproximarnos a la significación de las componentes explicando la variable respuesta (ocurrencia [0,1] del Gorrión chillón) sería construir un Modelo Generalizado Lineal con las tres componentes PLS seleccionadas por el procedimiento `sparse=TRUE, sparseStop=TRUE`. De nuevo, como hicimos para el caso del `modelo.pls` podemos obtenerlo sencillamente mediante las siguientes líneas de código:

```
modelo.pls.xscores2 <- as.data.frame(modelo.pls.sig$tt)
glm.pls.binom2 <- glm(modelo.pls.sig$dataY~., data=modelo.pls.xscores2, family=binomial(link="logit"))
```

Obteniendo estos resultados:

```
summary(glm.pls.binom2)

Call:
glm(formula = modelo.pls.sig$dataY ~ ., family = binomial(link = "logit"),
    data = modelo.pls.xscores2)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6191  -0.8767  -0.4779   1.0093   2.9224
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.84706    0.08120 -10.431  < 2e-16 ***
Comp_1       0.76053    0.06560  11.593  < 2e-16 ***
Comp_2       0.38536    0.08777   4.391 1.13e-05 ***
Comp_3       0.25839    0.09004   2.870  0.00411 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 1291.5  on 998  degrees of freedom
Residual deviance: 1090.6  on 995  degrees of freedom
AIC: 1098.6
Number of Fisher Scoring iterations: 5
```

De nuevo, la devianza explicada por el modelo tiene un valor bajo del 15.6%, indicativo de que es un modelo poco explicativo:

```
100*(glm.pls.binom2$null.deviance - glm.pls.binom2$deviance)/glm.pls.binom2$null.deviance
[1] 15.56135
```

Y el coeficiente de sobredispersión del modelo es aceptable (aproximadamente 1):

```
sum(residuals(glm.pls.binom2, type="pearson")^2)/glm.pls.binom2$df.residual
[1] 1.102815
```

Aunque la tercera componente del PLS explicaba muy poca variación en la variable respuesta (un 0.8%), su efecto es suficientemente significativo como para contener información relevante cuando se opera con la obtención de componentes seleccionando sólo las variables que contienen efectos significativos.

Y ya para terminar, podemos comprobar que la adición de una tercera componente a nuestro modelo PLS no ha mejorado sustancialmente el potencial predictivo de la presencia / ausencia del Gorrión chillón, teniendo en cuenta el valor de **AUC** resultante del diagrama ROC (comparad el valor de AUC con el previo de `0.760863`).

```
pred.modelo <- prediction(fitted(glm.pls.binom2), glm.pls.binom2$y)
perf.modelo <- performance(pred.modelo,"tpr","fpr", measure="auc")
perf.modelo
An object of class "performance"
Slot "x.name":
[1] "None"

Slot "y.name":
[1] "Area under the ROC curve"

Slot "alpha.name":
[1] "none"

Slot "x.values":
list()

Slot "y.values":
[[1]]
[1] 0.7624345          ## ESTE ES EL VALOR DE AUC


Slot "alpha.values":
list()
```