

Regresión Avanzada

Universidad Austral

PhD. Débora Chan
Junio- Julio 2023

Facultad de Ingeniería

Organización

1 El Modelo de Regresión Lineal Múltiple

2 Evaluación del modelo

3 Selección de Variables

4 Multicolinealidad

5 Métodos de Regularización

6 Reducción de la Dimensión

Como generalización del ML Simple

Con k variables predictoras

La variable respuesta se expresa como la suma del error y una combinación lineal de variables explicativas y el error.



$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_k X_{ki} + \varepsilon_i$$

Otra forma de expresar la relación del modelo es:



$$E(Y/X_1, X_2, \cdots, X_k) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

Supuestos del Modelo

Igual que en el ML Simple

Los supuestos distribucionales de los errores son los mismos que en el caso del modelo de regresión lineal simple: independencia, normalidad y homocedasticidad $\varepsilon_i \sim N(0; \sigma^2)$.

Valor diferencial del ML Múltiple

Es importante destacar que una contribución fundamental de los modelos de regresión la constituye el poder explicativo del efecto de cada una de las variables así como de su capacidad en conjunto.

Facultad de Ingeniería

Interpretación de los Coeficientes

Importancia de la Interpretación de los coeficientes

- ★ β_j = efecto medio que tiene sobre el valor esperado de Y el incremento en una unidad de X_j , manteniendo fijos el resto de predictores.
- ★ β_0 = ordenada en el origen, valor esperado de Y cuando todos los predictores tienen nivel nulo.

También destaquemos que los efectos de las variables en el modelo propuesto son aditivos, es decir se suman entre sí para poder explicar el comportamiento de la variable respuesta.

Linealidad con la Variable Respuesta

Cómo verificarla?

- 🧐 Cada variable predictora debe tener una relación lineal con la variable respuesta, considerando fijas las restantes variables predictoras.
- 🧐 Se pueden graficar los residuos del modelo versus cada una de las variables predictoras. Si la asociación es lineal, los residuos presentarán una distribución aleatoria alrededor de cero.
- 🧐 En el caso que para alguna variable predictora se evidencie estructura, se debería intentar con alguna transformación de ésta.
- 🧐 La selección de las variables predictoras a incluir, así como de sus eventuales transformaciones o interacciones constituyen uno de los pasos más importantes de la modelización estadística.

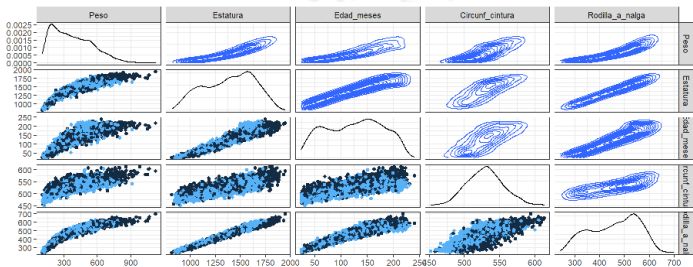
Ejemplo 9: Medidas Antropométricas

La base de datos **antropom.xlsx**, disponible en shorturl.at/gHJV1, contiene medidas antropométricas de niños. Interesa vincular la distancia de la rodilla a la nalga con otras variables de la base.



Visualizamos la forma de relación de las variables de la base

```
ggpairs(chicos, columns=c(2:6), mapping=aes(colour=Sexo), upper =  
list(continuous = 'density', combo = 'box_no_facet'), lower =  
list(continuous = 'points', combo = 'dot_no_facet'))+theme_bw()
```



OLS: Rodilla a nalga en función de peso y estatura

```
mod_chicos=lm(Rodilla_a_nalga ~ Peso+Estatura,data=chicos)
summary(mod_chicos)
residuos=mod_chicos$residuals# guardamos los residuos
```

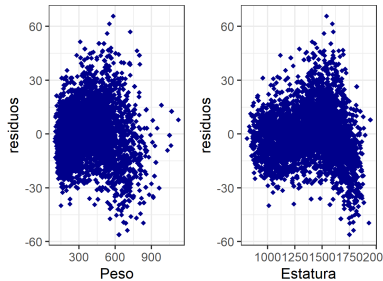
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-47.1003	2.4633	-19.12	0.0000
Peso	0.0442	0.0037	11.96	0.0000
Estatura	0.3531	0.0027	133.03	0.0000

Residual standard error: 14.79 on 3772 degrees of freedom

Multiple R-squared: 0.9763, Adjusted R-squared: 0.9763

F-statistic: 7.77e+04 on 2 and 3772 DF, p-value: < 2.2e - 16

Visualizamos Residuos versus predictoras



El modelo parece adecuado, todas sus variables tienen un aporte significativo. Pero se aprecia cierta estructura en el gráfico de residuos. La estatura y la longitud de rodilla a nalga pueden ser lineales, no así el peso con la estatura!

Ajustamos un modelo con Peso Transformado

```
mod_chicos1=lm(Rodilla_a_nalga~ sqrt(Peso)+Estatura,data=chicos)
summary(mod_chicos1)
residuos1=mod_chicos1$residuals# guardamos los residuos
# Visualizamos residuos vs las predictoras
p4=ggplot(chicos,aes(x=sqrt(Peso),y=residuos1))+
  geom_point(pch=18,color='darkblue')+theme_bw()
p5=ggplot(chicos,aes(x=Estatura,y=residuos1))+
  geom_point(pch=18,color='darkblue')+theme_bw()
p6= ggarrange(p4,p5)
```

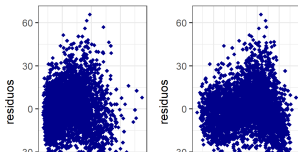
La Salida: disminuye la estructura, se mantiene heterocedasticidad

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-235.0219	6.0616	-38.77	0.0000
log(Peso)	52.9003	1.9182	27.58	0.0000
Estatura	0.2792	0.0039	72.38	0.0000

Residual standard error: 448.7 on 3772 degrees of freedom

Multiple R-squared: 0.9782, Adjusted R-squared: 0.9781

F-statistic: 8.445e+04 on 2 and 3772 DF, p-value: $< 2.2e - 16$



Transformamos la variable respuesta

```
mod_chicos2=lm(l(Rodilla_a_nalga1.5)  
sqrt(Peso)+Estatura,data=chicos)  
summary(mod_chicos2)  
residuos2=mod_chicos2$residuals
```

```
p7=ggplot(chicos,aes(x=sqrt(Peso),y=residuos2))+  
geom_point(pch=18,color='darkblue')+theme_bw()  
p8=ggplot(chicos,aes(x=Estatura,y=residuos2))+  
geom_point(pch=18,color='darkblue')+theme_bw()  
p9= ggarrange(p7,p8)
```

La Salida

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5813.0974	52.7922	-110.11	0.0000
sqrt(Peso)	161.7226	5.6159	28.80	0.0000
Estatura	9.2164	0.1037	88.89	0.0000

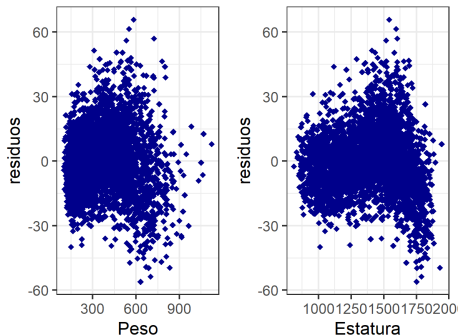
Residual standard error: 79.06 on 3772 degrees of freedom

Multiple R-squared: 0.9794, Adjusted R-squared: 0.9794

F-statistic: 8.445e+04 on 2 and 3772 DF: $< 2.2e - 16$

Facultad de Ingeniería

La estructura de los residuos



Mejoró la forma de la estructura de los residuos.

Expresión Matricial: Modelo Lineal Múltiple

La expresión matricial del modelo de regresión lineal múltiple puede expresarse del mismo modo que el modelo de regresión lineal simple:



$$Y = X\beta + \varepsilon$$

pero ahora:





$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & X_{11} & X_{12} & \dots & X_{1k} \\ 1 & X_{21} & X_{22} & \dots & X_{2k} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & X_{n1} & X_{n2} & \dots & X_{nk} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_n \end{pmatrix}$$

Propiedades


Nuevas Dimensiones


Valen en este caso también las siguientes propiedades:



$$E(Y/X) = X\beta$$


$$V(Y/X) = \sigma^2 I$$

Y la deducción de los estimadores de máxima verosimilitud y momentos que hemos realizado también vale. Sin embargo, debemos destacar que:


$$\dim(Y) = \dim(\varepsilon) = n \times 1$$


$$\dim(\beta) = (k + 1) \times 1$$


$$\dim(X) = n \times (k + 1)$$

Organización

1 El Modelo de Regresión Lineal Múltiple

2 Evaluación del modelo

3 Selección de Variables

4 Multicolinealidad

5 Métodos de Regularización


6 Reducción de la Dimensión

Contribución de las Variables

Cuando se tiene un modelo de regresión con k variables predictoras, usualmente nos interesa analizar hipótesis relativas al valor de los coeficientes de las variables. Para ello se pueden plantear para el coeficiente β_j con $j \in \{0, 1, \dots, k\}$ hipótesis del tipo:

$$H_0 : \beta_j = \beta_{j0} \quad \text{versus} \quad \beta_j \neq \beta_{j0}$$

Para estas pruebas el estadístico de contraste es:


$$t_{obs} = \frac{\hat{\beta}_j - \beta_{j0}}{\sqrt{\widehat{Var}\beta_j}}$$

Prueba para los coeficientes

Bajo la hipótesis de normalidad de los residuos del modelo se cumple que:



$$t_{obs} \sim t_{n-k-1}$$

En general nos interesa el caso $\beta_{j0} = 0$ es decir si esa variable contribuye significativamente o no a la explicación de la variable respuesta.

Es importante destacar que este contraste es una prueba marginal dado que se supone que las restantes variables permanecen en el modelo.

En la salida del *summary* del modelo se realiza esta prueba para el caso de $\beta_{j0} = 0$, para todos los coeficientes del modelo, se muestra el valor estimado del parámetro, su error estándar, el estadístico de contraste y el p valor de la prueba.

Evaluación Global del Modelo Lineal

Validación Cruzada y Métricas

Dado que uno de los objetivos de la regresión es predecir con precisión el valor de la variable respuesta, una vez construido el modelo es adecuado cuantificar la precisión de sus predicciones.

Una metodología muy habitual es dividir el conjunto de datos en dos partes con una de ellas estimar los coeficientes del modelo (conjunto de entrenamiento) y la otra utilizarla para estimar la precisión del modelo estimado (conjunto de validación).

Luego se dispone de diferentes métricas para la evaluación del modelo de regresión. todas ellas hacen foco en la medición de la similitud entre los valores ajustados y los valores reales u observados.

Facultad de Ingeniería

Métricas para Evaluar un Modelo Lineal

El coeficiente de determinación R^2

En los modelos lineales simples cuantifica la bondad de ajuste, indicando el porcentaje de variabilidad explicada de la respuesta a partir de la predictora.

En los modelos lineales múltiples

No puede utilizarse R^2 , pues a mayor cantidad de variables predictoras que se incluyan mayor es el valor del estadístico R^2 , puesto que cada predictor va a explicar alguna parte de la variabilidad observada en Y .

Existen varias métricas que de alguna forma penalizan la cantidad de variables predictoras incluidas en el modelo, es decir basadas en el principio de **parsimonia**.

de la cantidad total de predictores incluidos en el modelo y el tamaño de muestra, cantidad de observaciones o los grados de libertad. Cuanto mayor sea el tamaño muestral mayor cantidad de predictores pueden incluirse en el modelo, pero la idea es elegir un modelo que con la menor cantidad posible de variables explicativas logre explicar mejor la variabilidad de Y .

Coeficiente de Determinación Ajustado R^2_{adj}

En síntesis:

Este estadístico penaliza la inclusión de variables innecesarias.



$$R^2 = 1 - \frac{SCRes/(n - p - 1)}{SCT/(n - 1)}$$

donde:

- n es el número de observaciones
- p es el número de predictores.

Un valor alto de este estadístico indica un valor bajo de error. Si bien R^2_{adj} es un método muy utilizado para evaluar la bondad de ajuste de un modelo, existen otras alternativas.

Criterio de Información de Akaike (AIC)

En síntesis:

Se puede aplicar a modelos ajustados mediante el **criterio de máxima verosimilitud** (OLS es ajustado por máxima verosimilitud que para el modelo lineal es equivalente a mínimos cuadrados). Su expresión es:



$$AIC = \frac{1}{n\hat{\sigma}^2} \sum_{i=1}^n (y_i - \hat{y}_i)^2 - N + 2P$$

Cuanto menor sea el valor de este estadístico menor es el error de ajuste del modelo.

Cp de Mallows

En síntesis:

El estadístico Cp de Mallows es un estimador del error cuadrático medio de un modelo ajustado por **mínimos cuadrados** con p predictores.



$$C_p = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 - n + 2p$$

siendo n la cantidad de observaciones y p la cantidad de predictores. Este estadístico añade una penalización que aumenta conforme aumenta la cantidad de predictores incluidos en el modelo.

El mejor modelo corresponde a aquel que alcance un menor Cp.

Criterio de Información Bayesiano (BIC)

La expresión del BIC es



$$BIC = p.\log(n) + n.\log \left\{ \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right\}$$

donde n es el número de observaciones, y p es el número de predictores.

Como $\log(n) > 2$ para $n > 7$, el estadístico BIC aumenta la penalidad en modelos con muchas variables, y selecciona **modelos de menor tamaño en relación a la selección del C_p** .

Valores pequeños de BIC corresponde a un test error bajo, por lo que el mejor modelo será aquel con menor BIC..

Los recientemente descritos se encuadran dentro de **Indirectos** que pretenden reducir el sesgo y el sobreajuste.

Cuidado!

Sin embargo, C_p , AIC, BIC y R^2 ajustado, no son recomendables en el caso en que el número de variables predictoras sea superior al número de observaciones $p > n$.

Los **métodos directos** utilizan validación simple o **validación cruzada**.

Organización

1 El Modelo de Regresión Lineal Múltiple

2 Evaluación del modelo

3 Selección de Variables

4 Multicolinealidad

5 Métodos de Regularización

6 Reducción de la Dimensión

Existen diferentes estrategias para seleccionar los predictores a incluir:

- Ⓐ El **criterio del analista**, que introduce las variables predictoras en un orden determinado generalmente orientado por modelos previos disponibles o bien en la relevancia en el contexto del problema.
- Ⓑ **Métodos paso a paso** requieren de algún criterio de los indirectos presentados para determinar si el modelo mejora o empeora con cada incorporación o extracción de variables explicativas.

El criterio de Akaike(AIC) tiende a ser más restrictivo e introducir menos predictores que el R^2_{adj} y que sobre un mismo set de datos, distintos modelos podrían seleccionar distintos subconjuntos de variables explicativas.

Estrategias

Los algoritmos

- ➡ **Best subset o mejor subconjunto:** entra dentro de los métodos de entrada forzados. Prueba todos los subconjuntos posibles para cada cantidad de predictores entre los disponibles y señala el mejor subconjunto para esa cantidad de predictores, dado un criterio establecido.
- ➡ **Foward o hacia adelante:** El modelo inicial no contiene ningún predictor, solo el intercepto. Luego introduce una sola variable y entre todas las disponibles selecciona la que optimiza el criterio establecido. En el segundo paso, incorpora una segunda variable si la incorporación es significativa y se repite hasta que no pueden introducirse nuevas variables.

Estrategias

Los algoritmos

- ➡ **Backward o hacia atrás:** El modelo se inicia con todas las variables disponibles incluidas como variables predictoras. Se prueba a eliminar una a una cada variable, si se mejora el modelo, queda excluida. Este método facilita la evaluación de cada una de las variables presencia de las otras.
- ➡ **Stepwise o paso a paso mixto:** combina la selección forward y backward. Se inicia igual que el forward pero tras cada nueva incorporación se prueba la extracción de predictores como en el backward. Presenta la ventaja de que si a medida que se añaden predictores, alguno de los ya presentes deja de contribuir al modelo, es posible eliminarlo.

Observaciones

- Es frecuente que la selección de predictores se fundamente en el p-valor asociado a cada una de las variables explicativas.
- Si bien el algoritmo es sencillo, tiene algunos inconvenientes destacables tales como la eliminación de los predictores menos significativos tiende y el aumento de la significación de otros predictores.
- Esta metodología conduce a múltiples errores por lo cual no es recomendable excepto en casos muy sencillos (pocas v. explicativas).
- Para el caso particular de variables categóricas (factores), si alguno de sus niveles es significativo, se considera que la variable lo es.
- Una variable excluida por alguno de los métodos de selección puede incluirse por la conexión con la variable respuesta sin tener un aporte significativo en el contexto.

Ejemplo 10: Carseats

Los Datos

Vamos a trabajar con una base de datos simulada que contiene las ventas de asientos de seguridad infantiles para autos de 400 tiendas distintas. La base está disponible en R con el nombre de Carseats y contiene las siguientes 11 variables:



- 📖 **Sales:** cantidad de ventas en miles por cada ubicación.
- 📖 **CompPrice:** precio cobrado por cada competidor en cada ubicación.
- 📖 **Income:** Nivel de ingresos de la comunidad (en miles de dólares)
- 📖 **Advertising:** Presupuesto de publicidad local para la empresa en cada ubicación (en miles de dólares).
- 📖 **Population:** tamaño poblacional de la región (en miles).
- 📖 **Price:** precio.
- 📖 **ShelveLoc:** factor con tres niveles Malo, Bueno y Medio que indica la calidad de e las estanterías de los asientos de seguridad.
- 📖 **Age:** promedio de edades de la población local.
- 📖 **Education:** nivel educativo de la población local.
- 📖 **Urban:** factor con dos niveles: Si y No para indicar si la tienda está en una locación urbana o rural.
- 📖 **US:** factor con dos niveles: Si y No para indicar si la tienda es de Estados Unidos o no.

El Código

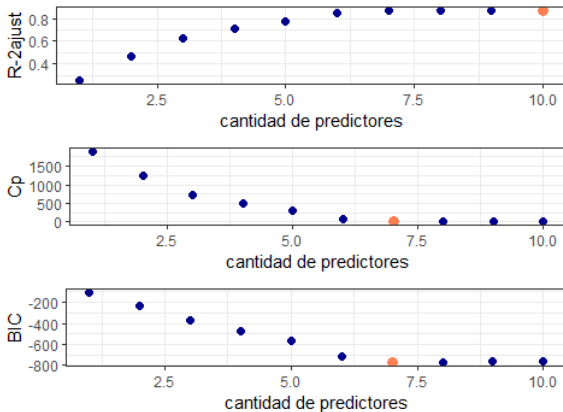
```
library(leaps); library(sp); library(glmnet); library(pls)
library(ISLR); library(ggpubr)
# consideramos todos los modelos hasta 10 predictoras
regfit.todos <- regsubsets(Sales ~ ., Carseats, nvmax = 10)
reg.summary <- summary(regfit.todos)
reg.summary
# definimos puntos óptimos para c/ criterio, guardamos su posición
max_adjr2=which.max(reg.summary$adjr2)
min_cp=which.min(reg.summary$cp)
min_bic=which.min(reg.summary$bic)
which.min(reg.summary$aic)
sal_reg=data.frame(orden=1:10,adjr2=reg.summary$rsq,
Cp=reg.summary$cp,Bic=reg.summary$bic)
```

graficamos los valores de c/ criterio en función de la cant. de predictores

```
pl1=ggplot(sal_reg,aes(orden,adjr2))+geom_point(size=2,col='darkblue')-
theme_bw()+xlab('cantidad de predictores')+ylab('R-2ajust')+
geom_point(aes(max_adj2,reg.summary$adj2[max_adj2]),
size=3,color='coral')
pl2=ggplot(sal_reg,aes(orden,Cp))+geom_point(size=2,col='darkblue')+
theme_bw()+xlab('cantidad de predictores')+ylab('Cp')+
geom_point(aes(min_cp,reg.summary$cp[min_cp]),size=3,color='coral')
pl3=ggplot(sal_reg,aes(orden,Bic))+geom_point(size=2,col='darkblue')+
theme_bw()+xlab('cantidad de predictores')+ylab('BIC')+
geom_point(aes(min_bic,reg.summary$bic[min_bic]),size=3,color='coral')

ggarrange(pl1,pl2,pl3,ncol=1)
```

Cantidad de variables



Se ve que con 7 predictoras se optimizan los criterios de C_p y Bic.

Coeficientes estimados

```
# buscamos los coeficientes del modelo con 7 predictores  
coef(regfit.full, 7)
```

	coeff
(Intercept)	5.48
CompPrice	0.09
Income	0.02
Advertising	0.12
Price	-0.10
ShelveLocGood	4.84
ShelveLocMedium	1.95
Age	-0.05

Coinciden con los modelos paso a paso?

fijamos una semilla y elegimos un 80% de los datos para entrenar el modelo y el 20% para validarlo

```
set.seed(1123)
```

```
n = nrow(Carseats)
```

```
index = sample(n, n*0.80, replace = FALSE)
```

```
train_set = Carseats[index,]
```

```
dim(train_set)
```

```
test_set = Carseats[-index,]
```

```
dim(test_set)
```

modelo completo con todas las variables

```
model_full = lm(Sales ~ ., data = train_set)
```

modelo sólo con intercepto

```
model_int = lm(Sales ~ -. , data = train_set)
```

Modelo Foward

```
n=400  
dim(train_set)  
320 11  
dim(test_set)  
80 11
```

```
scopeformula = formula(model_full)  
scopeformula  
fwd_sel      = step(object=model_int,    scope=scopeformula,    direc-  
tion='forward')  
summary(fwd_sel)  
FwdSelection_AIC = AIC(fwd_sel)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.7534	0.6394	9.00	0.0000
ShelveLocGood	4.8301	0.1661	29.08	0.0000
ShelveLocMedium	1.9239	0.1406	13.69	0.0000
Price	-0.0940	0.0029	-32.12	0.0000
CompPrice	0.0944	0.0046	20.56	0.0000
Advertising	0.1296	0.0118	11.01	0.0000
Age	-0.0451	0.0034	-13.10	0.0000
Income	0.0157	0.0020	7.66	0.0000
USYes	-0.4238	0.1632	-2.60	0.0099
Education	-0.0378	0.0216	-1.75	0.0808

Residual standard error: 1.006 on 310 degrees of freedom

Multiple R-squared: 0.8775, Adjusted R-squared: 0.8739

F-statistic: 246.7 on 9 and 310 DF, p-value: < 2.2e - 16

FwdSelection_AIC 923.5469

Backward

```
model_full = lm(Sales ~ ., data = train_set)
scopeformula = formula(model_full)

back_sel = step(object=model_full, scope=scopeformula, direc-
tion='backward')
summary(back_sel)
BackSelection_AIC = AIC(back_sel)
BackSelection_AIC
```

Facultad de Ingeniería

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.7534	0.6394	9.00	0.0000
CompPrice	0.0944	0.0046	20.56	0.0000
Income	0.0157	0.0020	7.66	0.0000
Advertising	0.1296	0.0118	11.01	0.0000
Price	-0.0940	0.0029	-32.12	0.0000
ShelveLocGood	4.8301	0.1661	29.08	0.0000
ShelveLocMedium	1.9239	0.1406	13.69	0.0000
Age	-0.0451	0.0034	-13.10	0.0000
Education	-0.0378	0.0216	-1.75	0.0808
USYes	-0.4238	0.1632	-2.60	0.0099

Residual standard error: 1.006 on 310 degrees of freedom

Multiple R-squared: 0.8775, Adjusted R-squared: 0.8739

F-statistic: 246.7 on 9 and 310 DF, p-value: < 2.2e - 16

BackSelection_AIC 923.5469

Stepwise

```
both_sel = step(object=model_full, scope=scopeformula, direc-
tion='both')
summary(both_sel)
BidirSelection_AIC = AIC(both_sel)
BidirSelection_AIC
AIC_df = data.frame(FwdSelection=FwdSelection_AIC, BackSelec-
tion=BackSelection_AIC, BidirSelection=BidirSelection_AIC)
rownames(AIC_df) = c("AIC")
AIC_df
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.7534	0.6394	9.00	0.0000
CompPrice	0.0944	0.0046	20.56	0.0000
Income	0.0157	0.0020	7.66	0.0000
Advertising	0.1296	0.0118	11.01	0.0000
Price	-0.0940	0.0029	-32.12	0.0000
ShelveLocGood	4.8301	0.1661	29.08	0.0000
ShelveLocMedium	1.9239	0.1406	13.69	0.0000
Age	-0.0451	0.0034	-13.10	0.0000
Education	-0.0378	0.0216	-1.75	0.0808
USYes	-0.4238	0.1632	-2.60	0.0099

Residual standard error: 1.006 on 310 degrees of freedom

Multiple R-squared: 0.8775, Adjusted R-squared: 0.8739

F-statistic: 246.7 on 9 and 310 DF, p-value: < 2.2e - 16

BidirSelection_AIC 923.5469

Comparando los distintos modelos ajustados

- ▶ Los tres métodos de selección paso a paso concluyen en el mismo modelo. Es decir seleccionan el mismo subconjunto de variables.
- ▶ Esto no es necesariamente así en todos los casos.
- ▶ Además este modelo incluye más variables que el elegido por los criterios indirectos.

Facultad de Ingeniería

Organización

1 El Modelo de Regresión Lineal Múltiple

2 Evaluación del modelo

3 Selección de Variables

4 Multicolinealidad

5 Métodos de Regularización

6 Reducción de la Dimensión

Mal condicionamiento vs Multicolinealidad

Mal Condicionado

Es deseable que los predictores de un modelo lineal múltiple sean independientes, pero esto frecuentemente no ocurre. Cuando uno de los predictores es combinación lineal de otros se dice que el modelo está mal especificado.

Multicolinealidad

Cuando esto no ocurre en forma exacta, sino más bien aproximada, se dice que existe fuerte asociación lineal entre un subconjunto de las variables explicativas y el modelo presenta multicolinealidad.




La multicolinealidad puede provenir del método de recolección de los datos, de la sobredefinición del modelo o bien de las restricciones del mismo.

Efectos de la presencia de Multicolinealidad

- ➡ No se puede identificar el efecto individual de cada una de las variables sobre la variable respuesta, se incrementa la varianza de los coeficientes del modelo de regresión estimados y por lo tanto se dificulta establecer su significación.
- ➡ Pequeños cambios en los datos generan cambios notables en la estimación de coeficientes.
- ➡ Las estimaciones de los coeficientes pueden aparecer con signos opuestos a los esperados o razonables o los coeficientes estimados son muy grandes en valor absoluto.
- ➡ La estimación de los coeficientes es muy sensible a pequeños cambios en los datos o a la eliminación/incorporación de una variable.
- ➡ Para determinar la presencia de multicolinealidad, existen diversas estrategias para detectar si está afectando a un modelo.

Otros Indicios

Efectos adversos

-  El coeficiente de determinación es significativo, sin embargo ninguna de las variables predictoras lo es.
-  Si en este caso, se construyen los modelos de regresión simple con estas variables alguno de ellos tendrá un coeficiente de determinación significativo.
-  Es posible detectar la multicolinealidad mediante la matriz de correlación de las variables predictoras; aunque podría no aparecer en esta representación ya que la asociación podría tener una estructura un poco más compleja. Es decir, es posible que la correlación no se aprecie de a pares y sin embargo sí en una terna o cuaterna.

Tolerancia y VIF

Puede ser útil el cálculo del factor de inflación de la varianza(VIF).

Cuando la multicolinealidad no se debe a la simple asociación de pares de variables.

Para cada variable puede calcularse como:



$$VIF(\beta_j) = \frac{1}{1 - R_{X_j|X_{-j}}^2}$$

Siendo el denominador de esta expresión función del coeficiente de determinación de la regresión de X_j en función del resto de los predictores.

Interpretación del VIF

Para interpretar los valores del VIF pueden considerarse los siguientes valores de corte:

- ✗ $VIF = 1$ entonces hay ausencia de colinealidad.
- ✗ $1 < VIF < 5$ entonces hay algo de colinealidad.
- ✗ $5 < VIF < 10$ entonces hay fuerte colinealidad.

Facultad de Ingeniería

Tolerancia

Tolerancia

La tolerancia (TOL) es el inverso del VIF y sus valores de interpretación surgen de los presentados para el VIF.

- ✂ Como la multicolinealidad indica redundancia de la información de las predictoras, una opción es eliminar una variable explicativa.
- ✂ Se podría eliminar una de las variables problemáticas ya que la presencia de colinealidad implica que la información que esta variable proporciona sobre la respuesta Y es redundante en presencia de otra/s variable/s.
- ✂ Otra alternativa sería combinar las variables colineales en un solo

Indice de Condición

La multicolinealidad entorpece la interpretación de los coeficientes.

Objetivo

Cuantificar la sensibilidad de las estimaciones realizadas por OLS ante pequeños cambios en los datos. Se define como la raíz cuadrada del cociente entre el máximo y el mínimo autovalor de la matriz de correlación $R = X^t X$, donde X debe estar normalizada.

Simbólicamente:



$$K = \frac{\lambda_{max}}{\lambda_{min}}$$

Donde λ es un autovalor de la matriz R .

Indice de Condición

Definición e Interpretación

El Índice de Condición para cada variable se define como el cociente entre su autovalor y el máximo autovalor de la matriz R:



$$IC_i = \sqrt{\frac{\lambda_i}{\lambda_{max}}}$$

Para interpretar el IC podemos utilizar la siguiente guía:

Si $\kappa < 20$ entonces la multicolinealidad es leve.

Si $20 < \kappa < 30$ entonces la multicolinealidad es moderada.

Si $\kappa > 30$ entonces la multicolinealidad es severa.

Prueba de Farrar-Glaubar

Objetivo

Identificar si a nivel poblacional, las variables regresoras presentan independencia estadística (ortogonalidad), a través de la matriz de correlación muestral. Se analiza si dicha matriz de correlación corresponde a una matriz identidad, las hipótesis de la prueba son las siguientes:

$$H_0 : R \sim I \quad \text{versus} \quad H_1 : R \not\sim I$$

El rechazo de la hipótesis de nulidad es indicador de multicolinealidad.

El estadístico de contraste de esta prueba es:

$$i \quad \chi^2_{FG} = - \left(n - 1 - \frac{2m + 5}{6} \right) \ln(|R|)$$

Ejemplo 11: mtcars

En este ejemplo vamos a usar los datos **mtcars** que se extrajeron de la revista estadounidense **Motor Trend** de 1974 y comprenden el consumo de combustible y 10 aspectos del diseño y rendimiento del automóvil para 32 automóviles (con modelos correspondientes a 1973–74).

Hay 32 observaciones son 11 variables numéricas:



Ejemplo 11: Las variables

- ✈ **mpg** Millas recorridas con un galón de combustible.
- ✈ **cyl** Número de cilindros.
- ✈ **disp** Cilindrada del motor en cm^3 .
- ✈ **hp** Potencia del motor en caballos
- ✈ **drat** Relación del eje trasero.
- ✈ **wt** Peso del automóvil.
- ✈ **qsec** Tpo que tarda en recorrer 1/4 de milla desde reposo (Aceleración).
- ✈ **vs.** Disposición de los cilindros.
- ✈ **am** Transmisión (0 = automática, 1 = manual).
- ✈ **gear** Número de marchas hacia adelante.
- ✈ **carb** Número de carburadores.

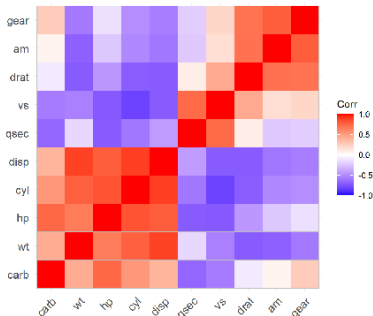
Análisis de colinealidad entre predictoras de a pares

```
library(mctest) library(ggcorrplot) library(car)
data(mtcars)
cor_matrix <- cor(mtcars[, -1])
cor_matrix
ggcorrplot(cor_matrix, hc.order = TRUE) # Graficamos la matriz de
correlación separando las variables
```

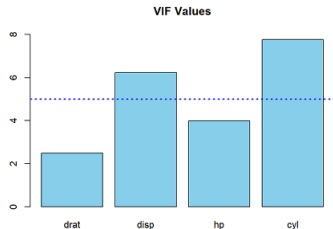
Facultad de Ingeniería

Graficamos el correlograma separando las asociaciones positivas de las negativas

```
ggcorrplot(cor_matrix, hc.order = TRUE)
```



Graficamos los VIF



Ajustamos un modelo lineal con variable respuesta consumo (mpg) y predictoras drat, disp, hp y cyl

```
modelo1 <- lm(mpg ~ drat + disp + hp + cyl, data=mtcars)
summary(modelo1)
```

Salida del Modelo

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	23.9852	7.9890	3.00	0.0057
drat	2.1540	1.5987	1.35	0.1890
disp	-0.0139	0.0109	-1.28	0.2129
hp	-0.0232	0.0158	-1.47	0.1530
cyl	-0.8140	0.8437	-0.96	0.3432

Residual standard error: 3.012 on 27 degrees of freedom

Multiple R-squared: 0.7825, Adjusted R-squared: 0.7503

F-statistic: 24.29 on 4 and 27 DF, p-value: 1.314e-08

Indicios de Multicolinealidad

Si bien el modelo tiene un R^2_{adj} alto, ninguna de las variables resulta estadísticamente significativa excepto el intercepto. Esto puede deberse a la presencia de multicolinealidad. Vamos a calcular el VIF para las variables del modelo y a graficarlo para visualizar cuál/es son superiores al valor de corte.

```
vif(modelo1)
vif_values <- vif(modelo1)
vif_df=data.frame(variables=c('drat' , 'disp','hp' , 'cyl'),vif_values)
barplot(vif_values, main = 'VIF Values', col = 'skyblue',ylim =
c(0.0,8.0))
abline(h = 5, lwd = 2, lty = 'dotted',col = 'blue')
```

Detección de multicolinealidad

drat	disp	hp	cyl
2.497	6.228	3.989	7.759

Como $VIF > 5$ indica que hay fuerte multicolinealidad.

Por simplicidad en este caso, podríamos intentar un modelo que no considere cyl dado que está muy correlacionado con las demás variables del modelo.

Luego abordaremos soluciones más abarcativas.

```
modelo2 <- lm(mpg~ drat + disp +hp, data=mtcars)
summary(modelo2)
```

Salida del nuevo modelo

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	19.3443	6.3709	3.04	0.0051
drat	2.7150	1.4874	1.83	0.0786
disp	-0.0192	0.0094	-2.05	0.0496
hp	-0.0312	0.0133	-2.34	0.0266
Residual standard error: 3.008 on 28 degrees of freedom				
Multiple R-squared: 0.775, Adjusted R-squared: 0.7509				
F-statistic: 32.15 on 3 and 28 DF, p-value: 3.28e-09				

En ese nuevo modelo la significación global es similar pero son significativas hp y disp y levemente significativa drat. Veamos si se resolvió el problema de multicolinealidad, analizando los VIF del nuevo modelo.

VIF de las variables del nuevo modelo

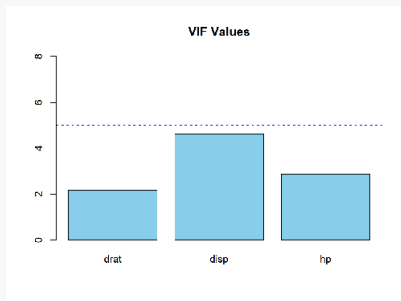
```
vif_values2 <- vif(modelo2)
```

```
barplot(vif_values2, main = 'VIF Values', col = 'skyblue', ylim =  
c(0.0, 8.0))  
abline(h = bad_vif, lwd = 2, lty = 'dotted', col = 'blue')
```

drat	disp	hp
2.167	4.622	2.868

Qué ocurrió con la multicolinealidad?

Multicolinealidad en el nuevo modelo



Se corrigió el problema de la multicolinealidad! Si bien hay otras opciones para los alcances de esta sección nos contentaremos con esta solución.

Organización

1 El Modelo de Regresión Lineal Múltiple

2 Evaluación del modelo

3 Selección de Variables

4 Multicolinealidad

5 Métodos de Regularización

6 Reducción de la Dimensión

Los métodos de selección de variables presentados en el capítulo anterior usan OLS para estimar los coeficientes del modelo lineal que contiene únicamente un subconjunto de predictores de todos los disponibles. La metodología de OLS funciona bien cuando se cumplen una serie de supuestos como la linealidad, la normalidad de los residuos, la ausencia de autocorrelación de las observaciones, la homocedasticidad de los residuos, la ausencia de multicolinealidad y que la cantidad de observaciones supera la cantidad de variables.

Sin embargo, cuando alguno de estos supuesto no se satisface, es adecuado recurrir a alguna alternativa. La regularización o penalización es una metodología que ofrece controlar el valor los parámetros estimados. La idea fundamental es agregar restricciones a la estimación de los coeficientes que logren contraerlos progresivamente hacia 0 y de este modo reducir la varianza del modelo.

Todos los métodos de regularización minimizan la suma de cuadrados residuales y de una restricción, lo que varía de una propuesta a otra es la expresión de esta función de penalización.

Ridge

Este método de regularización propone minimizar la suma de cuadrados residuales con una penalización definida por la suma de cuadrados de los coeficientes [?], simbólicamente:



$$\hat{\beta}_{Ridge} = \min_{\beta} \left\{ SCR + \lambda \sum_{i=1}^k (\beta_i)^2 \right\}$$

El valor λ recibe el nombre de parámetro de penalización que como no constituye un parámetro del modelo se denomina hiperparámetro.

Para seleccionar el valor del parámetro de regularización se utilizan dos enfoques: el indirecto es seleccionar el valor de λ que minimice algún criterio como AIC o BIC y el directo que es elegirlo por validación cruzada. La primera propuesta prioriza el ajuste del modelo a los datos y la segunda enfatiza su desempeño para la predicción.

En el caso particular en el cual $\lambda = 0$ se aprecia que OLS resulta un caso

LASSO

Este método Lasso (Least Absolute Shrinkage and Selection Operator) de regularización introduce otra penalización, en este caso se penaliza con la suma de los valores absolutos de los coeficientes [?], simbólicamente:

$$\hat{\beta}_{Lasso} = \min_{\beta} \left\{ SCR + \lambda \sum_{i=1}^k |\beta_i| \right\}$$


Nuevamente observemos que si el parámetro de regularización $\lambda = 0$, se aprecia que OLS es un caso particular de Lasso.

La sutil diferencia entre las dos funciones de penalización propuestas conlleva propiedades interesantes que permiten distinguir entre ambas propuestas. La regresión de Ridge tiende a castigar los coeficientes grandes, pero en realidad no logra eliminar variables del modelo.

Por su parte, la regresión de Lasso, sí lo logra; lo cual puede resultar bastante útil en el caso de que algunos predictores no tengan una

Elastic Net

La función a minimizar para Elastic Net combina las penalizaciones propuestas para Ridge y Lasso [?]:


$$\hat{\beta}_{E.N} = \min_{\beta} \left\{ SCR + \lambda \left[\frac{1-\alpha}{2} \sum_{i=1}^k (\beta_i)^2 + \alpha \sum_{i=1}^k |\beta_i| \right] \right\}$$

La ventaja de Elastic Net es que permite equilibrar ambas penalizaciones, lo que puede resultar en un mejor desempeño de un modelo.

K-Fold Cross-Validation

Este método propone dividir los datos de forma aleatoria en k grupos de aproximadamente el mismo tamaño, con $k-1$ grupos se entrena el modelo y uno de los grupos se emplea como validación. Este proceso se repite k veces utilizando un grupo distinto como validación en cada iteración. En el siguiente ejemplo, vamos a utilizar esta metodología para elegir el valor óptimo del hiperparámetro λ para el ajuste de los métodos de regularización.

Facultad de Ingeniería

Ejemplo 12: mtcars

Vamos a utilizar nuevamente la base de datos mtcars, ahora para ajustar modelos de regularización y comparar los resultados. Ya hemos visto en el apartado anterior la presencia de multicolinealidad entre los predictores.

```
library(glmnet)
library(coefplot)
set.seed(100)
```

```
train_ind <- sample(1:nrow(mtcars), size = 0.75*(nrow(mtcars)))
train <- mtcars[train_ind, ]
test <- mtcars[-train_ind, ]
# Ajustamos un modelo lineal completo
lm_comp <- lm(mpg~., data=train)
summary(lm_comp)
```

	Estimate	Std. Error	t value	Pr(> t)	Residual
(Intercept)	17.7955	20.0572	0.89	0.3911	
cyl	-0.1151	1.1631	-0.10	0.9227	
disp	0.0130	0.0198	0.65	0.5248	
hp	-0.0291	0.0270	-1.08	0.3005	
drat	0.2973	1.8175	0.16	0.8726	
wt	-3.4075	2.2510	-1.51	0.1540	
qsec	0.7750	0.7942	0.98	0.3470	
vs	0.5450	2.3934	0.23	0.8234	
am	4.7852	2.9393	1.63	0.1275	
gear	-0.5405	1.7719	-0.31	0.7651	
carb	0.1610	1.0203	0.16	0.8770	

standard error: 2.722 on 13 degrees of freedom

Multiple R-squared: 0.8786, Adjusted R-squared: 0.7852

F-statistic: 9.407 on 10 and 13 DF, p-value: 0.0001899

```
lm_comp.pred <- predict(lm_comp, new_data=test)
lm_mse <- mean((lm_comp.pred - test$mpg)^2)
lm_mse
```

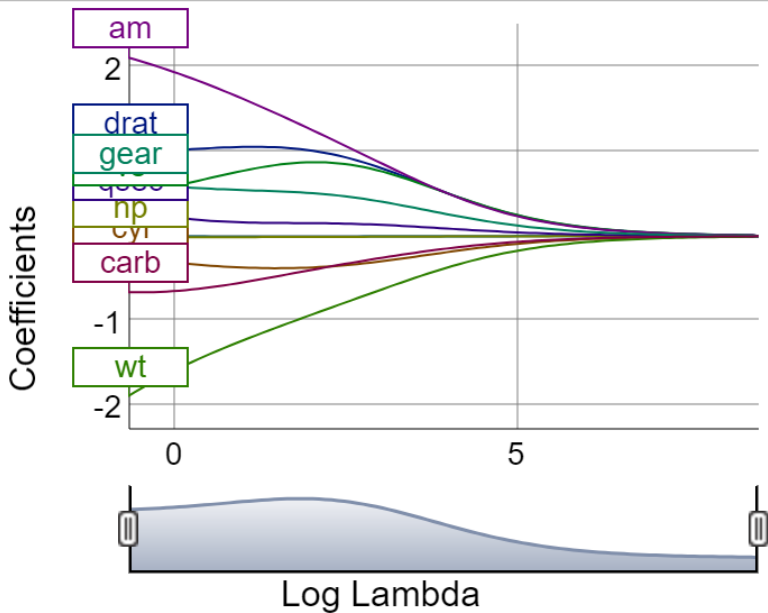
71.36648

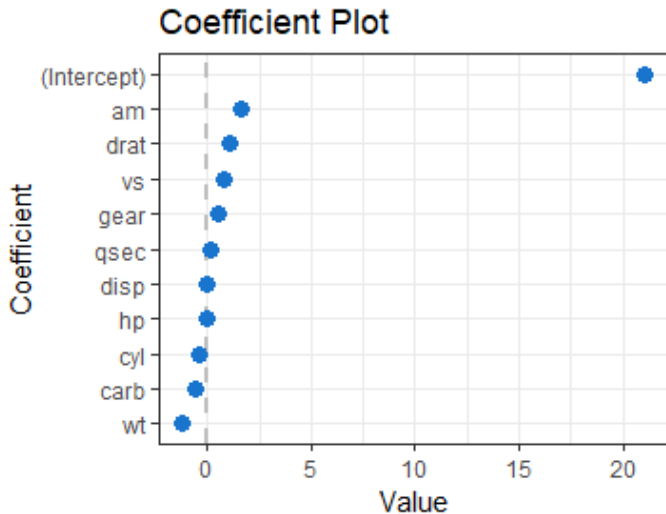
```
X = model.matrix(mpg ~ .-1, data=mtcars) # matriz de diseño
Y = mpg # nuestra variable dependiente
std <- TRUE
# ajustamos una Regresión de Ridge

# Valor óptimo de lambda por 10-fold CV
ridge.cv=cv.glmnet(X, Y, alpha=0, type='mse', family='gaussian',
standardize=std, nfolds=10)
# Ajustamos Ridge con 100 opciones de valores para lambda
```

3.014598

(Intercept)	21.0513
cyl	-0.3741
disp	-0.0053
hp	-0.0115
drat	1.0556
wt	-1.2046
qsec	0.1604
vs	0.7871
am	1.5915
gear	0.5418
carb	-0.5346





Este gráfico muestra los coeficientes estimados por Ridge, ordenados por magnitud en el valor lambda óptimo.

ajustamos una Regresión de LASSO

Valor óptimo de lambda por 10-fold CV

```
lasso.cv=cv.glmnet(X,Y,alpha=1, type='mse', family='gaussian',
standardize=std, nfolds=10)
```

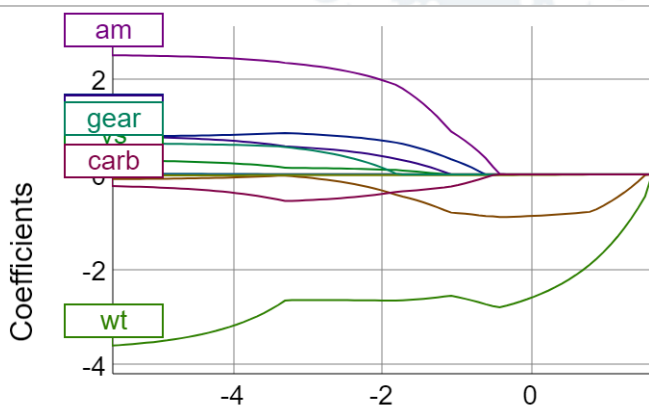
```
lasso_md1 = glmnet(X,Y,alpha=1,family='gaussian', standard-  
ize=std, nlambda=100)
```

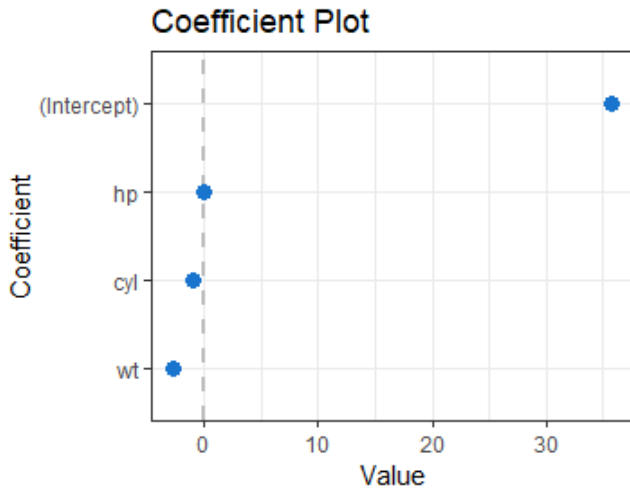
```
# Extraemos los coeficientes del modelo con lambda óptimo
coef(lasso_mdl,s=lasso.cv$lambda.min)
```

0.8787712

(Intercept)	35.7308062
cyl	-0.8801087
disp	.
hp	-0.0110762
drat	.
wt	-2.6636936
qsec	.
vs	.
am	.
gear	.
carb	.

```
coefpath(lasso_mdl)
coefplot(lasso_mdl, lambda=lasso.cv$lambda.min,
col='dodgerblue3',sort="magnitude")+theme_bw()
```





Finalmente, ajustamos un modelo por Elastic Net, elegimos para eso $\alpha = 0.5$ en este caso.

```
# ajustamos una Regresión de Elastic Net
```

```
# Valor óptimo de lambda por 10-fold CV
```

```
enet.cv=cv.glmnet(X,Y,alpha=0.5, type='mse', family='gaussian',  
standardize=std, nfolds=10)
```

```
enet.mdl = glmnet(X,Y,alpha=0.5,standardize=std,nlambda=100)
```

```
# Extraemos los coeficientes del modelo con lambda óptimo
```

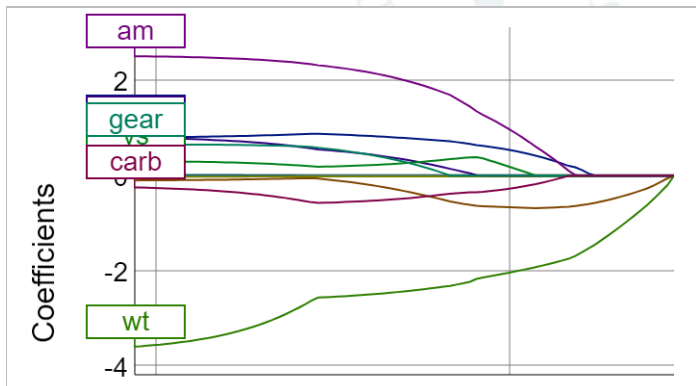
```
enet.cv$lambda.min
```

```
coef(enet.mdl,s=enet.cv$lambda.min)
```

1.00573	
(Intercept)	31.5738
cyl	-0.6614
disp	-0.0022
hp	-0.0137
drat	0.5393
wt	-2.0313
qsec	.
vs	0.1866
am	0.9728
gear	.
carb	-0.2719

```
coefplot(enet_mdl, lambda=enet.cv$lambda.min,  
col='dodgerblue3',sort='magnitude')+theme_bw()
```

```
coefpath(enet_mdl)
```



Comparemos los resultados finales de los tres modelos ajustados por métodos de regularización para los datos de mtcars.

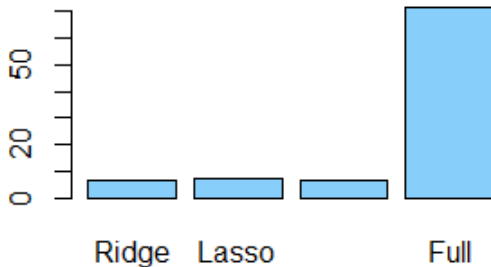
```
tmatrix <- model.matrix(mpg~.,test)[-1]
# Hallamos el Error Cuadrático Medio para cada uno de los modelos
# y luego los comparamos analítica y gráficamente
ridge.pred = predict(ridge_mdl, s=ridge.cv$lambda.min,
newx=tmatrix)
ridge_mse = mean((ridge.pred-test$mpg)2)

lasso.pred = predict(lasso_mdl, s=lasso.cv$lambda.min,
newx=tmatrix)
lasso_mse = mean((lasso.pred-test$mpg)2)

enet.pred = predict(enet_mdl, s=enet.cv$lambda.min,
newx=tmatrix)
enet_mse = mean((enet.pred-test$mpg)2)
```

Ridge	LASSO	Elastic Net	Completo
6.4387	6.9875	6.5520	71.3665

Mean Squared Error



Organización

1 El Modelo de Regresión Lineal Múltiple

2 Evaluación del modelo

3 Selección de Variables

4 Multicolinealidad

5 Métodos de Regularización

6 Reducción de la Dimensión

El contexto

- 🌐 En muchos estudios observacionales, se dispone de un gran número de variables que no necesariamente implica que se dispone de la información relevante o necesaria.
- 🌐 Si las variables predictoras están fuertemente asociadas y la información que brindan adolece de redundancia, OLS no va a ser apropiado.
- 🌐 Los métodos de regularización controlan la varianza. En ambos casos se emplean las variables originales sin ser modificadas o a lo sumo habiendo sido estandarizadas.

Objetivo

- Las técnicas de reducción de la dimensionalidad crean un número reducido de nuevas variables (componentes) a partir de combinaciones lineales de las variables originales y con ellas se ajusta el modelo.
- Así, se generan modelos con menor número de predictores, ahora independientes entre sí, pero que abarcan casi la misma información que la que aportan todas las variables originales.
- Los métodos más conocidos de reducción de la dimensionalidad son PCR (Principal Component regression) y PLS (Partial Least Squares).

PCR

- 😊 PCR utiliza como variables predictoras o explicativas las componentes principales (combinaciones lineales de las variables originales que captan la mayor varianza posible y son independientes entre sí).
- 😊 Cuando el número de componentes es igual al número original de variables explicativas, el resultado de PCR equivale al de OLS. Sin embargo, la idea es reducir la cantidad de predictores, como en el Análisis de Componentes Principales.
- 😊 Para ajustar un modelo por PCR es necesario **estandarizar los predictores** previamente, de lo contrario, las variables que se miden en una escala mayor o las que presenten mayor varianza tendrán más peso.

PLS

Relación con PCR

El método Partial Least Squares (PLS) es muy similar al PCR en cuanto que ambos emplean el resultado del análisis de componentes principales PCA como variables predictoras.

Mientras PCR ignora la variable respuesta Y para determinar las combinaciones lineales, PLS busca aquellas que, además de explicar la varianza observada, predicen Y lo mejor posible. La propuesta PLS puede entonces considerarse como una versión supervisada de PCR.

Similitud con OLS

Relación con PCR

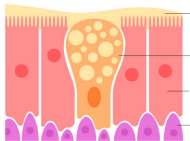
PCA elimina información redundante y puede mejorar el modelo de regresión. Pero, si bien el PCR reduce el número de predictores del modelo, no es un método de selección de variables ya que todas ellas se necesitan para el cálculo de las componentes.

El método PCR no deja de ser un OLS que emplea componentes principales como predictores, para que sea válido se tienen que cumplir las condiciones requeridas para la regresión por mínimos cuadrados.

Facultad de Ingeniería

Ejemplo 13: Permeabilidad

Utilizaremos la base de datos **permeability**. La capacidad de un compuesto para penetrar en las membranas biológicas es fundamental para la farmacología. Dado un número suficiente de compuestos examinados, se puede desarrollar un modelo predictivo de permeabilidad. En este proyecto se examinaron 165 compuestos únicos; determinando 1107 huellas dactilares moleculares para cada uno. Una huella digital molecular es una secuencia binaria de números que representa la presencia o ausencia de una subestructura molecular específica, la respuesta es sesgada y muchos predictores están fuertemente asociados.



```
data('permeability') fingerprint = fingerprints%>%data.frame()
# identificamos predictores con baja variabilidad
baja_var <- nearZeroVar(fingerprints)

# eliminamos los predictores de baja variabilidad
def_Pred <- fingerprints[,-baja_var]

# inspeccionamos cuántos registros quedaron
str(def_Pred)
```

Quedaron 165 registros de 388 variables.

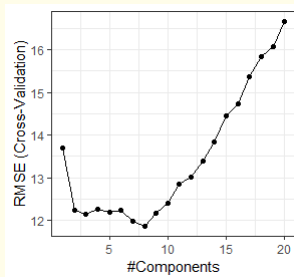
```

set.seed(1325)
# pre-procesamos los datos centrando y escalando
pred_prep <- preProcess(def_Pred, method = c('center','scale'))
resp_prep <- predict(pred_prep, def_Pred)
# elegimos aleatoriamente el 75% para entrenar los modelos
train_filas <- createDataPartition(permeability,p=0.75,list=FALSE)
# partimos las predictoras y la variable respuesta en train- test
fing_train <- resp_prep[train_filas,]
fing_test <- resp_prep[-train_filas,]
perm_train <- permeability[train_filas,]
perm_test <- permeability[-train_filas,]
# Ajustamos PLS a train usando 10-fold CV y RMSE
pls_model <- train(fing_train, perm_train , method = 'pls' , metric
= 'RMSE' , tuneLength = 20 , trControl = trainControl(method =
"cv" , number = 10))
pls_model
    
```

ncomp	RMSE	Rsquared	MAE
1	13.698	0.323	10.593
2	12.246	0.441	8.725
3	12.155	0.457	9.248
4	12.265	0.447	9.138
5	12.201	0.451	8.928
6	12.254	0.454	9.084
7	11.984	0.475	9.132
8	11.863	0.487	9.032
9	12.171	0.468	9.353
10	12.418	0.459	9.462
11	12.854	0.438	9.592
12	13.027	0.432	9.806
13	13.398	0.420	9.971
14	13.831	0.391	10.196
15	14.448	0.367	10.587
16	14.725	0.353	10.813

Visualizamos

R^2 was used to select the optimal model using the largest value.



El mínimo del RMSE se verifica para 8 predictores en el modelo PLS.

predict the test set response for the PLS model

```
pls_test <- predict(pls_model, newdata = fing_test)
postResample(pred=pls_test, obs = perm_test)
```

RMSE	Rsquared	MAE
9.437	0.559	7.193

```
pcr_model <- train(x=fing_train, y=perm_train,
method='pcr', metric = 'RMSE', tuneLength = 10,
trControl = trainControl('cv'), preProc=c('center','scale'))
pcr_model
ggplot(pcr_model)
```

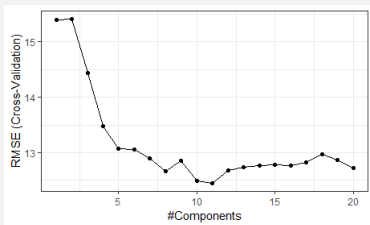
ncomp	RMSE	Rsquared	MAE
1.000	15.400	0.195	12.174
2.000	15.413	0.190	12.176
3.000	14.443	0.277	11.200
4.000	13.474	0.374	10.619
5.000	13.067	0.410	9.970
6.000	13.051	0.394	9.831
7.000	12.898	0.418	9.584
8.000	12.668	0.432	9.349
9.000	12.856	0.411	9.561
10.000	12.492	0.457	9.208
11.000	12.442	0.464	9.125
12.000	12.678	0.440	9.300
13.000	12.736	0.433	9.378
14.000	12.763	0.432	9.331
15.000	12.787	0.434	9.294

La cantidad de componentes

RMSE was used to select the optimal model using the smallest value.

The final value used for the model was $ncomp = 11$.

El mínimo del RMSE se verifica para 11 predictores en el modelo PCR.



Comparando los resultados de los métodos de reducción de la dimension, PLS produce mejores resultados usando los criterios de RMSE y MAE.

Ajustamos ahora los métodos de regularización presentados.

```
RidgeGrid <- data.frame(.lambda = seq(0.1, .1, length = 15))

Ridge_model = train(x = fing_train, y = perm_train, method =
'ridge',
trControl = trainControl('cv', number = 10), metric = 'RMSE',
tuneGrid = RidgeGrid, preProc = c('center', 'scale'))
Ridge_model$bestTune

data.frame(rsquared = Ridge_model[['results']]
[['Rsquared']][as.numeric(rownames(Ridge_model$bestTune))],
rmse=Ridge_model[['results']][['RMSE']]
[as.numeric(rownames(Ridge_model$bestTune))])
```

Ajuste por lasso

lambda 0.1

```
Lasso_model = train(x = fing_train, y = perm_train, method =  
'glmnet',  
trControl = trainControl('cv', number = 10), metric = 'RMSE',  
preProc = c('center', 'scale'))
```

```
Lasso_model$bestTune
```

Ajuste por elastic net

alpha	lambda
0.55	0.6064

```
enetGrid <- expand.grid(.lambda = c(0.01, 0.01, .1), .fraction =
seq(.05, 1, length = 20))
```

```
enet_model <- train(x = fing_train, y = perm_train,
method = 'enet', tuneGrid = enetGrid, center = TRUE,
metric = 'RMSE', trControl = trainControl('cv', number = 10),
preProc = c('center', 'scale'))
enet_model$bestTune
```

Comparación de los ajustes

fraction	lambda
0.3	0.1

Comparamos los tres modelos de regularización ajustados

	Rsquared	RMSE
RIDGE	0.40	13.84
LASSO	0.50	11.44
ELASTICNET	0.43	12.14

Entre los tres de regularización, considerando el RMSE, el mejor sería LASSO, pero PLS logra un valor menor de RMSE que cualquiera de estos tres.