

## **Tarea 1**

Estudiante

**John Daniel hoyos Arias  
Ivan Santiago Rojas Martinez  
Genaro Alfonso Aristizabal Echeverri**

Docente

**Juan Carlos Salazar Uribe**

Asignatura

**Analitica de datos**



Sede Medellín  
17 de septiembre del 2022

# Índice

<b>1. Ejercicio1</b>	<b>4</b>
1.1. Análisis descriptivo de la base de datos bank . . . . .	4
1.2. a) cree un conjunto de datos de entrenamiento del 75 % y el restante 25 % tráteselos como datos de test o prueba. . . . .	6
1.3. b) Con los datos de entrenamiento implemente naive bayes usando loan como el supervisor y las demás como predictores. . . . .	6
1.4. c) Con los datos de entrenamiento, Implemente un modelo Knn con loan como supervisor y las demás como predictoras. utilizar varios valores de k, pero reportar solo uno. . . . .	7
1.5. d) Con los datos de entrenamiento, implemente un modelo Logístico con loan como supervisor y las demás como predictoras. . . . .	9
1.5.1. Planteamiento del Modelo. . . . .	9
1.6. e) Con los datos de entrenamiento, implemente un modelo LDA con loan como supervisor y las demás como predictoras. En LDA se modela la distribución de los predictores X de manera separada en cada una de las categorías de respuesta (es decir, condicionando en Y ) y luego se usa el teorema de Bayes para obtener estimaciones de $Pr(Y = k X = x)$ Cuando estas distribuciones se asumen normales, el modelo es muy similar en forma al de regresión logística. esta probabilidad de clasificación está dada por: . . . . .	10
1.7. f) Con los datos de entrenamiento calcular training MSE, matriz confusión y curva roc para cada uno de los modelos. . . . .	10
1.8. g) Con los datos de test calcular training MSE, matriz confusión y curva roc para cada uno de los modelos. . . . .	12
1.9. h) Con cual modelo observo mejor desempeño y por qué? . . . . .	13
<b>2. Ejercicio2</b>	<b>14</b>
2.1. Breve análisis Exploratorio . . . . .	14
2.2. a) Cree un conjunto de datos de entrenamiento del 75 % y el restante 25 % tráteselos como datos de test o de prueba. . . . .	16
2.3. b) Con los datos de entrenamiento, implemente Knn(con al menos tres valores para k) usando income como el supervisor y debts como predictor. Grafique e interprete. . . . .	16
2.4. c) Con los datos de entrenamiento, implemente regresión lineal simple usando income como el supervisor y debts como predictor. Grafique e interprete. . .	18
2.5. d) Use los respectivos ajustes de cada uno de los modelos anteriores y con el conjunto de prueba, calcule el test-MSE. Qué observa? . . . . .	21

2.6. e) Usando todos los datos y regresión lineal multiple seleccione un modelo usando forward, backward y stepwise . . . . .	22
2.7. f) Seleccione uno de los modelos del paso anterior y responda con argumentación la pregunta: Ajusta bien dicho modelo? . . . . .	23
<b>3. Punto 3</b>	<b>23</b>
3.0.1. DATOS_A . . . . .	23
3.0.2. Análisis descriptivo . . . . .	25
3.1. Ridge . . . . .	25
3.2. Lasso . . . . .	27
3.3. Conclusiones y respuesta a la pregunta planteada . . . . .	30
3.3.1. DATOS_B . . . . .	30
3.3.2. Análisis descriptivo . . . . .	31
3.4. Ridge . . . . .	33
3.5. Lasso . . . . .	34
3.6. Conclusiones y respuesta a la pregunta planteada . . . . .	36
<b>4. Punto 4</b>	<b>37</b>
4.0.1. Análisis descriptivo . . . . .	40
4.1. “Best” Subset . . . . .	40
4.2. Forward . . . . .	45
4.3. Backward . . . . .	49
4.4. Stepwise . . . . .	52
4.5. Validación cruzada (CV) . . . . .	56

## Índice de figuras

# 1. Ejercicio1

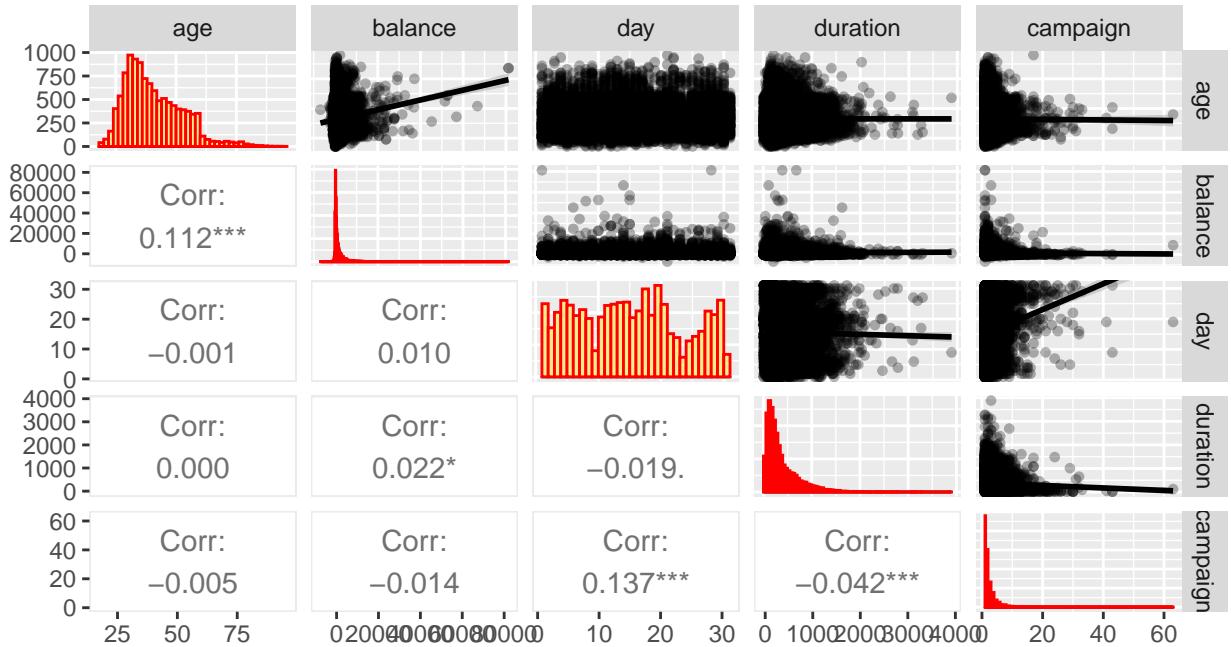
## 1.1. Análisis descriptivo de la base de datos bank

La base de datos bank, contiene un total de 17 variables y 11.162 observaciones, de las cuales 7 son de tipo numéricas y 10 son de tipo categórica, esta base de datos resume algunas características acerca de clientes de un banco en particular tales como la edad(age), el tipo de trabajo que desempeña(job), el estado marital(marital), nivel educativo(education), si ha cometido o no alguna falta pagos(default), el estado de sus fondos económicos(balance), si el cliente tiene o no algún préstamo de vivienda(housing), si el cliente tiene o no algún préstamo (loan), medio de contacto con el cliente(contact), fecha de afiliación(day, month), tiempo de vencimiento (El tiempo de vencimiento). entre otras, a continuación veremos un pequeño resumen de las variables.

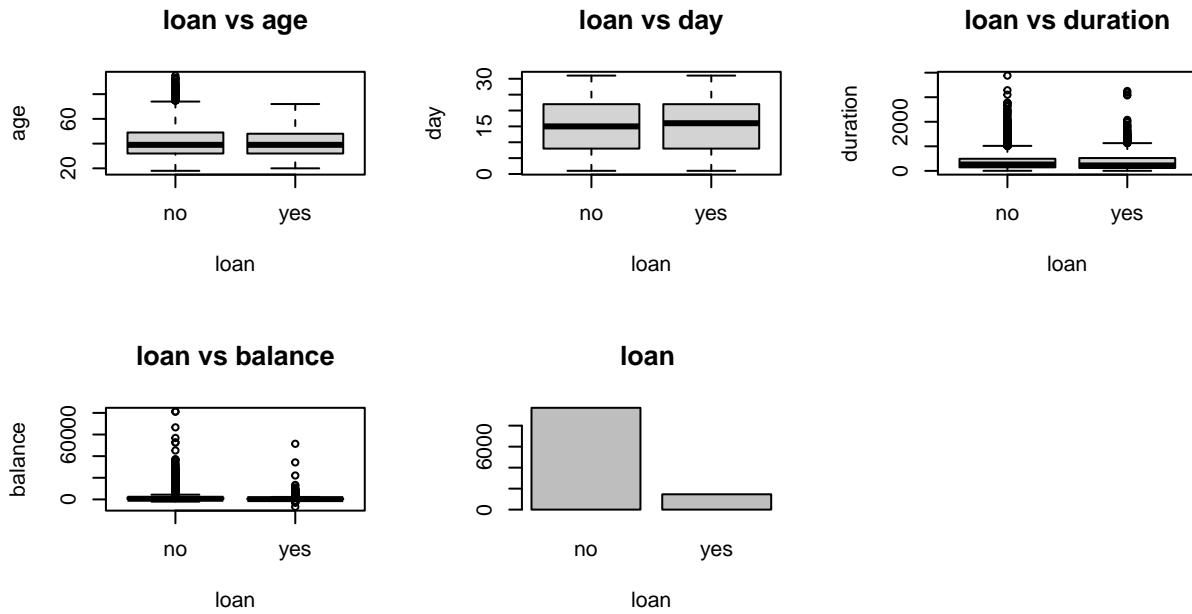
age	job	marital	education	default
Min. :18.00	management :2566	divorced:1293	primary :1500	no :10994
1st Qu.:32.00	blue-collar:1944	married :6351	secondary:5476	yes: 168
Median :39.00	technician :1823	single :3518	tertiary :3689	NA
Mean :41.23	admin. :1334	NA	unknown : 497	NA
3rd Qu.:49.00	services : 923	NA	NA	NA
Max. :95.00	retired : 778	NA	NA	NA
NA	(Other) :1794	NA	NA	NA

balance	housing	loan	contact	day	month
Min. :-6847	no :5881	no :9702	cellular :8042	Min. : 1.00	may :2824
1st Qu.: 122	yes:5281	yes:1460	telephone: 774	1st Qu.: 8.00	aug :1519
Median : 550	NA	NA	unknown :2346	Median :15.00	jul :1514
Mean : 1529	NA	NA	NA	Mean :15.66	jun :1222
3rd Qu.: 1708	NA	NA	NA	3rd Qu.:22.00	nov : 943
Max. :81204	NA	NA	NA	Max. :31.00	apr : 923
NA	NA	NA	NA	NA	(Other):2217

duration	campaign	pdays	previous	poutcome	deposit
Min. : 2	Min. : 1.000	Min. : -1.00	Min. : 0.0000	failure:1228	no :5873
1st Qu.: 138	1st Qu.: 1.000	1st Qu.: -1.00	1st Qu.: 0.0000	other : 537	yes:5289
Median : 255	Median : 2.000	Median : -1.00	Median : 0.0000	success:1071	NA
Mean : 372	Mean : 2.508	Mean : 51.33	Mean : 0.8326	unknown:8326	NA
3rd Qu.: 496	3rd Qu.: 3.000	3rd Qu.: 20.75	3rd Qu.: 1.0000	NA	NA
Max. :3881	Max. :63.000	Max. :854.00	Max. :58.0000	NA	NA



El gráfico anterior es importante para identificar el posible comportamiento de nuestras variables numéricas, en este caso vemos que existe poca relación lineal entre las mismas, situación que nos da una idea de pensar que es poco probable que existan problemas de multicolinealidad.



La variable *loan*, es una de las variables de mayor interés en el estudio, es una variable dicotómica que representa si un cliente tiene o no algún préstamo, del gráfico anterior,

claramente no hay diferencia en mediana para la variable loan con respecto a la edad, día, duración y balance, por otra parte, vemos que la mayoría de personas no tienen ningún tipo de préstamo.

### 1.2. a) cree un conjunto de datos de entrenamiento del 75 % y el restante 25 % tratetelo como datos de test o prueba.

```
df=data.frame(bank)
smp_sz <- floor(0.75 * nrow(df))
train_ind <- sample(seq_len(nrow(df)), size = smp_sz)

train <- df[train_ind, ] #datos de Entrenamiento 75%
test <- df[-train_ind, ] #datos de Test 25%

y_train=df[train_ind,8]
y_test=df[-train_ind,8]
```

### 1.3. b) Con los datos de entrenamiento implemente naive bayes usando loan como el supervisor y las demás como predictores.

El clasificador Naive de Bayes asume que todas las features (componentes del vector x) son independientes y que son igualmente importantes. Con este supuesto, la probabilidad (Likelihood-verosimilitud) de observar el vector de features  $x = (x_1, x_2, \dots, x_p)$  con  $p = 1 \dots 16$  dado que se está en la clase  $j$  es:

$$Pr(x|Y = j) = Pr(x_1|Y = j) * \dots * Pr(x_p|Y = j)$$

Por el teorema de Bayes tenemos:

$$Pr(y = j|x) = \frac{Pr(x|Y = j)(Pr(Y = j))}{Pr(x)}$$

lo anterior representa un modelo bayesiando donde, la distribución posterior, está representada por el producto de la verosimilitud y la probabilidad a priori.

ahora, vamos a implementar un modelo de naivebayes para calcular la probabilidad de que los próximos clientes tengan o no un crédito.

```
#modelo naive bayes
naiveB.fit <- naiveBayes(loan~., data=train, laplace=0.128)

#predict train and testing
```

```

predict_test<-predict(naiveB.fit,test,type="class")
predict_train<-predict(naiveB.fit,train,type="class")

#prediccion probabilidades
predict_train2<-predict(naiveB.fit,train,type="raw")
predict_test2<-predict(naiveB.fit,test,type="raw")

```

#### 1.4. c) Con los datos de entrenamiento, Implemente un modelo Knn con loan como supervisor y las demás como predictoras. utilizar varios valores de k, pero reportar solo uno.

El modelo knn también conocido como k vecinos más cercanos, es una metodología muy eficiente que permite a partir de un valor de k, tomar los k valores más cercanos de una estimación para ajustarse a su comportamiento, a medida que incrementamos el valor de k, se tiende a perder la señal y comenzamos a guarnos por el ruido del modelo, por lo tanto, se debe tener cuidado a la hora de utilizar esta metodología. por otra parte, en este modelo, se deben crear vectores de variables dummy para las variables categóricas, y las variables continuas se deben normalizar. cómo se verá a continuación.

```

#modelo KNN
normalize <- function(x) {
  norm <- ((x - min(x))/(max(x) - min(x)))
  return (norm)
}

#variables categoricas, a vectores de dummy.
suppressWarnings(dummyJob<-dummy(df$job, sep="_"))
suppressWarnings(dummyMarital<-dummy(df$marital, sep="_"))
suppressWarnings(dummyEducation<-dummy(df$education, sep="_"))
suppressWarnings(dummyDefault<-dummy(df$default ,sep="_"))
suppressWarnings(dummyHousing<-dummy(df$housing, sep="_"))
suppressWarnings(dummyContact<-dummy(df$contact ,sep="_"))
suppressWarnings(dummyMonth<-dummy(df$month, sep="_"))
suppressWarnings(dummyPoutcome<-dummy(df$poutcome, sep="_"))
suppressWarnings(dummyDeposit<-dummy(df$deposit, sep="_"))

#normalización de las variables numericas.
age<-normalize(df$age)
balance<-normalize(df$balance)
day<-normalize(df$day)
duration<-normalize(df$duration)
campaign<-normalize(df$campaign)
pdays<-normalize(df$pdays)
previous<-normalize(df$previous)

```

```

#nueva base de datos, con las variables transformadas.
Newdata<-cbind(df,dummyjob,dummyMarital,dummyEducation,
                 dummydefault,dummyhousing,dummycontact,dummymonth,
                 dummpyoutcome,dummydeposit,age,balance,day,duration,
                 campaign,pdays,previous)

#nueva selección de datos de entrenamiento y test
train1 <- Newdata[ train_ind,18:50 ]
test1 <- Newdata[-train_ind,18:50 ]
y_train1=df[train_ind,8]
y_test1=df[-train_ind,8]

#Modelo 1 k=1
#fit.knn_train<-knn(train=train1, test=train1,cl=y_train1, k=1, prob=TRUE)
#fit.knn_Test<-knn(train=train1, test=test1,cl=y_train1, k=1, prob=TRUE)

#Modelo 2 k=2 modelo seleccionado
fit.knn_train<-knn(train=train1, test=train1,cl=y_train1,
                     k=2, prob=TRUE,use.all=TRUE)

fit.knn_Test<-knn(train=train1, test=test1,cl=y_train1, k=2, prob=TRUE)

#modelo 3 k= 5
#fit.knn_train<-knn(train=train1, test=train1,cl=y_train1,
#k=5, prob=TRUE,use.all=TRUE)
#fit.knn_Test<-knn(train=train1, test=test1,cl=y_train1, k=5, prob=TRUE)

#predict para verificar el ajuste de nuestros datos de Test.
Predicted_train<-factor(fit.knn_train)
Predicted_test<-factor(fit.knn_Test)

#probabilidades
prob_train <- attr(fit.knn_train, "prob")
prob_test <- attr(fit.knn_Test, "prob")

```

1.5. d) Con los datos de entrenamiento, implemente un modelo Logístico con loan como supervisor y las demás como predictoras.

#### 1.5.1. Planteamiento del Modelo.

$$\text{Logit}(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_{1xi1} + \beta_{2xi2} + \dots + \beta_{kxik}$$

El modelo logístico, es una variación del modelo de regresión lineal en el que la variable respuesta es una variable dicótoma, es decir solo toma 2 valores 0 ó 1, es por esto que el logit, o el logaritmo de la razón de dos se utiliza como su predictor lineal. este modelo es naturalmente un modelo de clasificación, por lo anterior, el resultado que se va obtener es la probabilidad asociada a si una persona con diversas características tiene o no un crédito. para mayor comprensión, el modelo ajustado entregara el resultado de evaluar la siguiente expresión:

$$\pi_i = \frac{e^{\beta_0 + \beta_{1xi1} + \beta_{2xi2} + \dots + \beta_{kxik}}}{1 + e^{\beta_0 + \beta_{1xi1} + \beta_{2xi2} + \dots + \beta_{kxik}}}$$

Si el valor de la probabilidad es mayor a 0.5 entonces esta persona tiene un crédito(loan=si), en caso contrario, no tiene crédito(loan=no).

```
#Modelo logístico.
lr.fit<-glm(as.factor(loan)~., data = train, family=binomial)

#predict para verificar ajuste.
lrPred_train<-predict(lr.fit,train, type = c("response"))
lrPred_test<-predict(lr.fit,test, type = c("response"))

#clasificación
predict_lr_train<-ifelse(lrPred_train<=0.5,0,1)
predict_lr_test<-ifelse(lrPred_test<=0.5,0,1)
```

1.6. e) Con los datos de entrenamiento, implemente un modelo LDA con loan como supervisor y las demás como predictoras. En LDA se modela la distribución de los predictores X de manera separada en cada una de las categorías de respuesta (es decir, condicionando en Y ) y luego se usa el teorema de Bayes para obtener estimaciones de  $Pr(Y = k|X = x)$  Cuando estas distribuciones se asumen normales, el modelo es muy similar en forma al de regresión logística. esta probabilidad de clasificación esta dada por:

$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^k \pi_l f_l(x)}$$

```
#modelo LDA
lda.fit <- lda(as.factor(loan) ~., data=train)

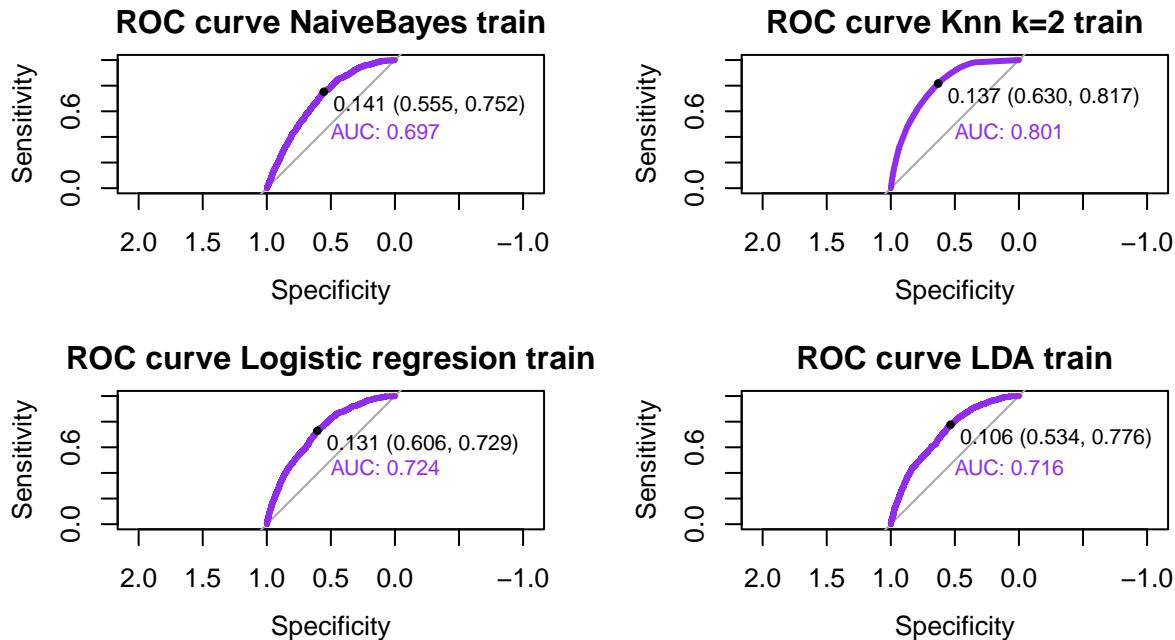
#prediccion
predict_train_lda<-predict(lda.fit,train,type=c("class"))
predict_test_lda<-predict(lda.fit,test,type=c("class"))

#clasificación
train_lda<-ifelse(as.factor(train$loan)==predict_train_lda$class,0,1)
test_lda<-ifelse(as.factor(test$loan)==predict_test_lda$class,0,1)
```

1.7. f) Con los datos de entrenamiento calcular training MSE, matriz confusión y curva roc para cada uno de los modelos.

```
## $NaiveBayes
##           y_train
## predict_train  no  yes
##           no  6729  939
##           yes   511  192
##
## $Train_NaiveBayes_err
## [1] 0.1732171
##
## $Knn
##           y_train
## Predicted_train  no  yes
##           no  7151  991
##           yes   89  140
```

```
##  
## $Train_error_Knn  
## [1] 0.1290168  
##  
## $logistic_reg  
##      predict_lr_train  
##          0     1  
##    no    7232     8  
##    yes   1116    15  
##  
## $lr_train_err  
## [1] 0.1342731  
##  
## $Lda  
##      train_lda  
##          0     1  
##    no    7190    50  
##    yes   38 1093  
##  
## $LDA_train_err  
## [1] 0.01051248  
  
## $sensitividad_NaiveBayes  
## [1] 0.1697613  
##  
## $Especificidad_NaiveBayes  
## [1] 0.9294199  
##  
## $sensitividad_Knn  
## [1] 0.1237843  
##  
## $Especificidad_Knn  
## [1] 0.9877072  
##  
## $sensitividad_logistic_reg  
## [1] 0.6521739  
##  
## $Especificidad_lr  
## [1] 0.8663153  
##  
## $sensitividad_Lda  
## [1] 0.9562555  
##  
## $Especificidad_LDA  
## [1] 0.9947427
```



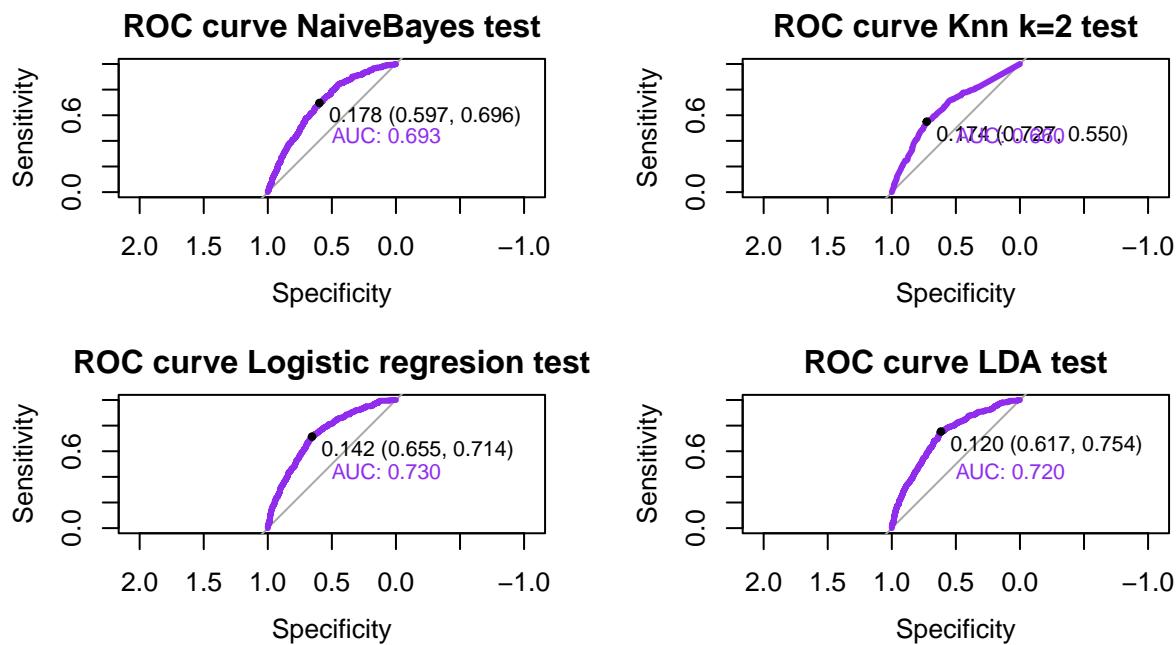
1.8. g) Con los datos de test calcular training MSE, matriz confusión y curva roc para cada uno de los modelos.

```
## $NaiveBayes
##           y_test
## predict_test no yes
##       no    2274  265
##       yes    188   64
##
## $Test_error_NB
## [1] 0.1623074
##
## $Knn
##           y_test1
## Predicted_test no yes
##       no    2416  309
##       yes     46   20
##
## $Test_error_Knn
## [1] 0.1271946
##
## $logistic_reg
##      predict_lr_test
##          0     1
## no    2461     1
```

```

##   yes  324    5
##
## $lr_test_err
## [1] 0.1164457
##
## $Lda
##      test_lda
##          0    1
## no  2449   13
## yes   11  318
##
## $LDA_test_err
## [1] 0.008599068

```



### 1.9. h) Con cual modelo observo mejor desempeño y por qué?

Para seleccionar el mejor modelo, es muy importante tener claro que es lo que se quiere responder. La variable loan como se dijo anteriormente representa si una persona tiene o no un crédito, esto es importante porque cometer un error de darle un crédito a una persona ya tenía uno o no darle un crédito a una persona que no lo tenía, es una decisión que genera un gran impacto negativo en las ganancias del banco, por lo cual, el modelo seleccionado debe cumplir con unos altos índices de sensibilidad y de especificidad. De lo anterior, sin duda alguna el modelo LDA, obtuvo unos índices de sensibilidad y de especificidad superiores al 96 %, lo cual es una excelente tasa de clasificación, por esta razón se selecciona como el mejor modelo, a pesar de que el AUC de las curvas Roc sean similares para todos los demás modelos.

## 2. Ejercicio2

### 2.1. Breve análisis Exploratorio

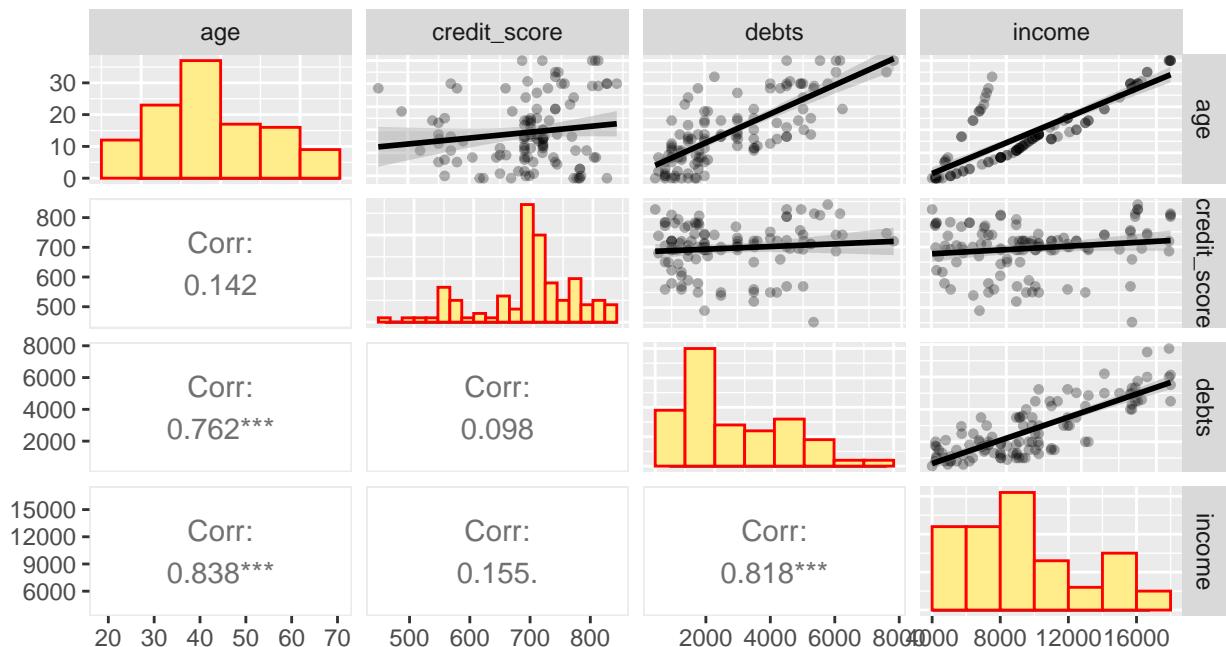
La base de datos costumer\_loan\_details, cuenta con un 12 variables y 114 observaciones de las cuales 4 son variables de tipo numérico y 8 son variables de tipo categóricas. Dicha base de datos tiene características de los cliente como lo son: state, gender, race, marital\_status, occupation, credit\_score, income, debts, loan\_type, loan\_decision\_type.

state	gender	age	race	
OH :11	Female:23	Min. :19.00	American Indian or Alaska Native :26	
CA : 8	Male :91	1st Qu.:28.00	Asian :17	
PA : 8	NA	Median :36.50	Black or African American : 7	
VA : 7	NA	Mean :38.88	Native Hawaiian or Other Pacific Islander: 3	
MI : 6	NA	3rd Qu.:48.00	No co-applicant :24	
NY : 6	NA	Max. :70.00	Not applicable :17	
(Other):68	NA	NA	White :20	
marital_status	occupation	credit_score	income	debts
Divorced:27	Accout :18	Min. :452.0	Min. : 4044	Min. : 500
Married :44	Business:15	1st Qu.:672.5	1st Qu.: 6665	1st Qu.:1300
Single :43	IT :27	Median :702.0	Median : 8869	Median :2000
NA	Manager :26	Mean :695.8	Mean : 9338	Mean :2744
NA	NYPD :28	3rd Qu.:740.0	3rd Qu.:11345	3rd Qu.:4150
NA	NA	Max. :840.0	Max. :16758	Max. :7755
loan_type	loan_decision_type			
Auto :40	Approved :70			
Credit :17	Denied :32			
Home :23	Withdrawn:12			
Personal:34	NA			

Del resumen numérico anterior se tiene:

- El **state** con mayor numero de observaciones es **Other= 68** y el segundo mayor es **OH = 11**.
- El **gender** con mayor numero de observaciones es **Male = 91** y el gender con menor observaciones es **Female = 91**.
- El \*age promedio es de 39 años\*\*.
- El \*race con mayor numero de observaciones es American Indian or Alaska Native = 26\*\*.
- El **marital\_status** con mayor numero de observaciones es **Married = 44** y el siguiente con mayor numero de observaciones es **Single = 43**.

- La **occupation** con mayor numero de observaciones es **NYPD = 28** y el siguiente con mayor numero de observaciones es **IT = 27**.
- El **credit\_score** promedio es de **695.8**.
- El **income** promedio es de **9338**.
- El **debts** promedio es de **2744**.
- El **loan\_type** con mayor numero de observaciones es **Auto = 40** y el siguiente con mayor numero de observaciones es **Personal = 34**.
- El **loan\_decision\_type** con mayor numero de observaciones es **Approved = 70** y el siguiente con mayor numero de observaciones es **Denied = 32**.



Este gráfico nos permite mirar y evidenciar relaciones lineales entre las variables.

Se puede observar relaciones lineales de interés entre las variables como los son:

- Se observa una alta correlación entre las variables **income** y **age** igual a **0.838**.
- Se observa una alta correlación entre las variables **income** y **debts** igual a **0.818**.
- Se observa una moderada correlación entre las variables **age** y **debts** igual a **0.762**.
- Se observa una baja correlación entre las variables **age** y **credit\_score** igual a **0.142**.

## 2.2. a) Cree un conjunto de datos de entrenamiento del 75 % y el restante 25 % tráteslo como datos de test o de prueba.

Se fija una **semilla = 123** con el fin de permitir replicabilidad del trabajo. Luego procedemos a crear el conjunto de datos para entrenamiento **train** de un valor del **75 %** de los datos para un total de **85 observaciones** y el conjunto de datos para prueba **test** de un valor del **25 %** de los datos para un total de **29 observaciones**. Se define **income** como el supervisor “**Y**” y la característica **debts** como predictor.

```
library(caret)
data <- read.csv("Data/costumer_loan_details.csv",
                 stringsAsFactors=TRUE, header = TRUE, sep = ",")  
  
set.seed(123)  
  
  
  
smp_sz <- floor(0.75 * nrow(data))
train_indx <- sample(seq_len(nrow(data)), size = smp_sz)  
  
train <- data[train_indx, 10]
train_scale <- scale(data[train_indx, 10])
test <- data[-train_indx, 10]
test_scale <- scale(data[-train_indx, 10])  
  
y_train <- data[train_indx, 9]
y_test <- data[-train_indx, 9]
```

## 2.3. b) Con los datos de entrenamiento, implemente Knn(con al menos tres valores para k) usando income como el supervisor y debts como predictor. Grafique e interprete.

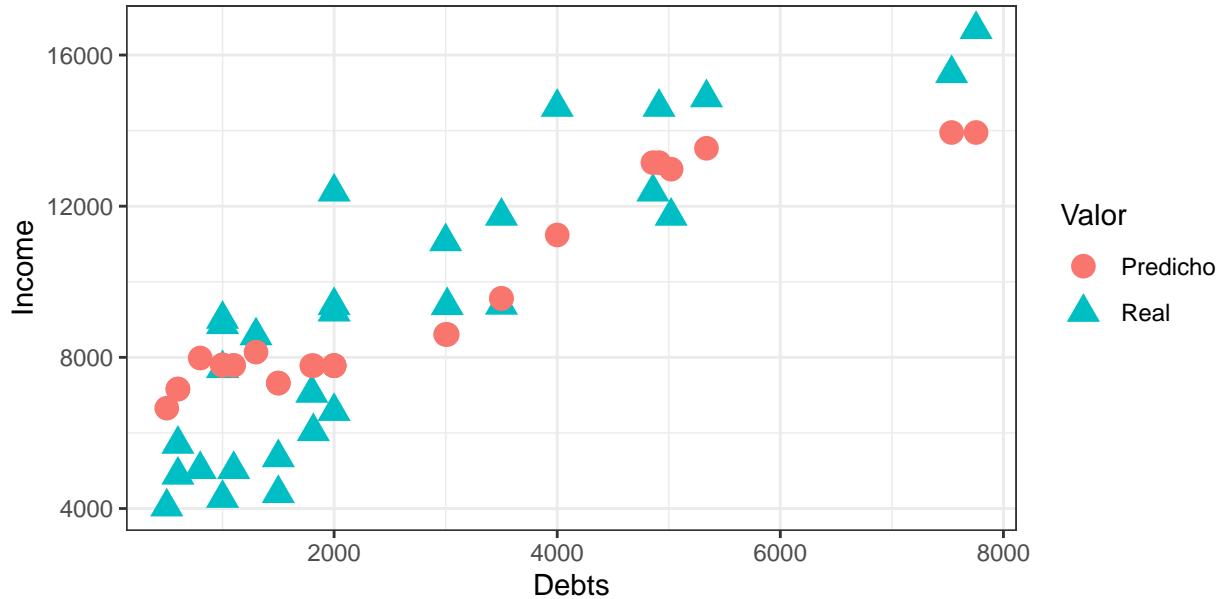
Con la ayuda de la librería **caret** se realiza una regresión knn con un parametro de k = 16. Mas adelante se explica por que se obta por usar ese k.

```
knnmodel = knnreg(train_scale, y_train, k = 16)  
  
pred_y = predict(knnmodel, data.frame(test_scale))  
  
mse = mean((y_test - pred_y)^2)
mae = caret::MAE(y_test, pred_y)
rmse = caret::RMSE(y_test, pred_y)
```

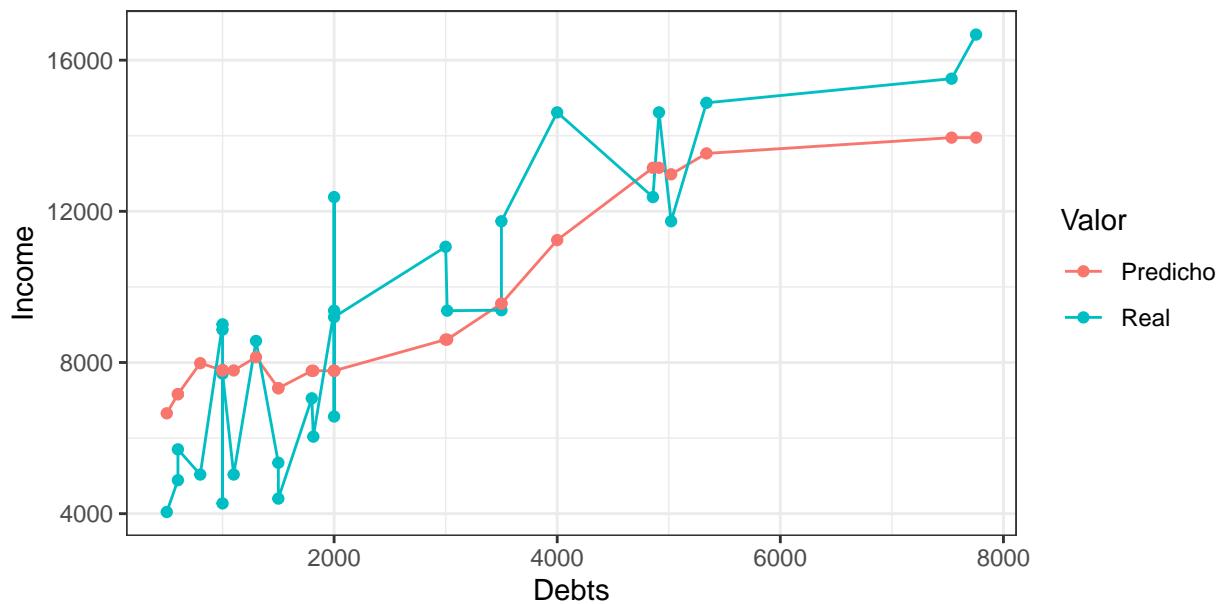
```
#cat("MSE: ", mse, "MAE: ", mae, " RMSE: ", rmse)
```

MSE	MAE	RMSE
4398259	1814.891	2097.203

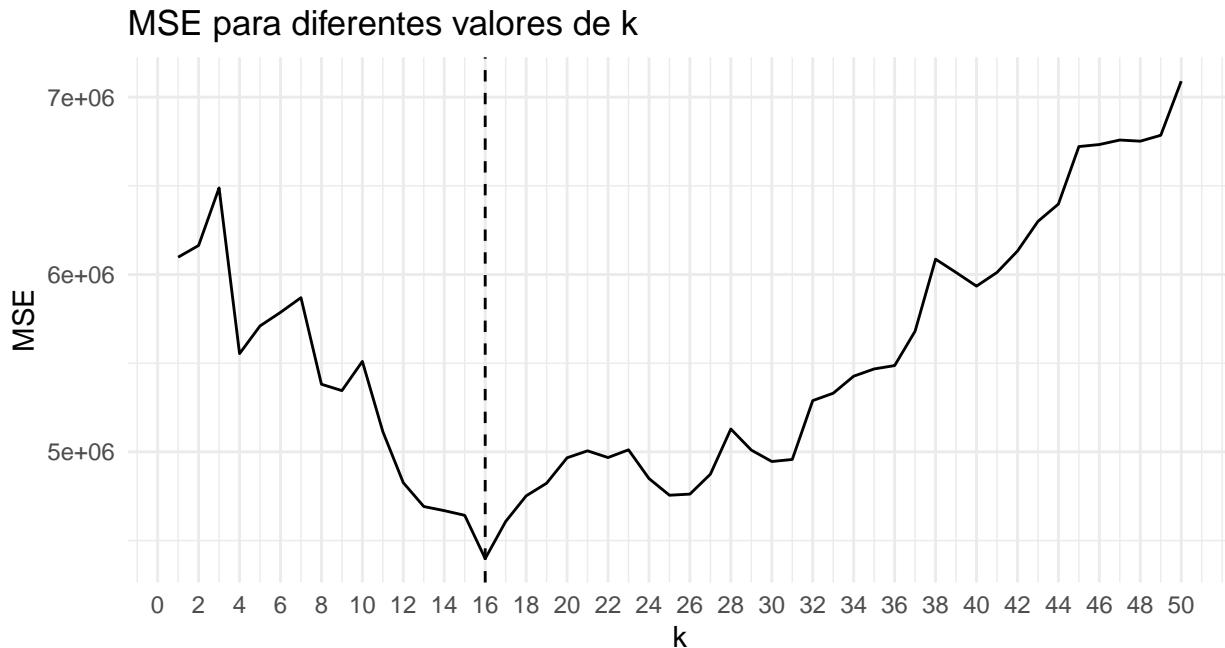
Datos de prueba (Test)



Datos de prueba (Test)



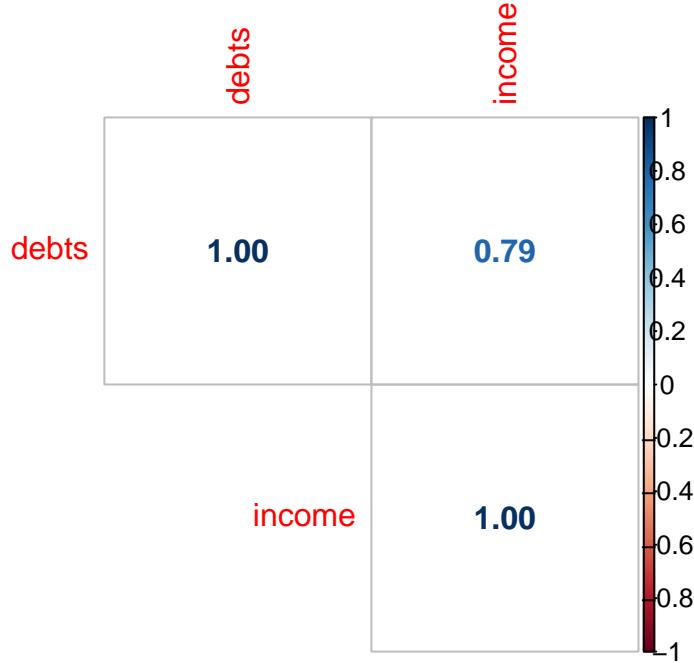
De la anteriores gráficas se puede observar que las predicciones dadas por nuestro modelo de regresión knn son relativamente buenas ya que el modelo trata de capturar la tendencia y no el ruido, si es comparado entre las predicciones que puede arrojar otros modelos de knn.



Nuestro modelo de regresión knn fue entrenado con 85 observaciones de un total de 114 y se toma  $k = 16$  por que es el modelo con menor **MSE = 4398259**, **MAE = 1814.891**, **RMSE = 2097.203** y cumple con ser el más parsimonioso entre los modelos de  $k = 17$  hasta  $k = 50$ .

## 2.4. c) Con los datos de entrenamiento, implemente regresión lineal simple usando income como el supervisor y debts como predictor. Grafique e interprete.

2.4.0.1. Correlación entre el supervisor (Income) y el predictor (debts). Datos de entrenamiento



Se observa una relación lineal **media-alta** entre el supervisor **income** y el predictor **debt**.

#### 2.4.0.2. Modelo lineal

$$Y \approx \beta_0 + \beta_1 X_i + \epsilon_i; \epsilon_i \sim N(0, \sigma^2)$$

```
model <- lm(y_train ~ debt, data=train_df)
```

$$\hat{Y} \approx 4584.5801 + 1.7285X_i + \epsilon_i; \epsilon_i \sim N(0, \sigma^2)$$

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	4584.580082	466.8251791	9.820764	0
debt	1.728516	0.1453318	11.893582	0

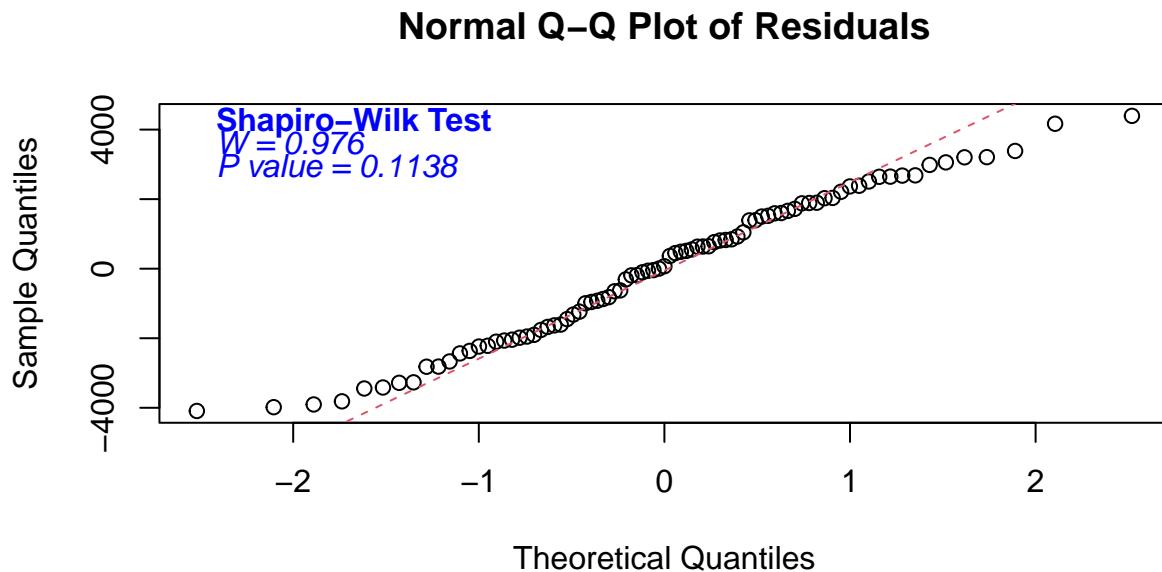
#### 2.4.0.4. Pruebas de hipótesis: Significancia del modelo.

$$H_0 : \beta_i = 0 \text{ vs } H_i : \beta_i \neq 0$$

Con una significancia de  $\alpha = 0.05$  y observando los p-valores para  $\beta_0$  y para  $\beta_1$  hay evidencia suficiente para rechazar a  $H_0$  esto quiere decir que el modelo es significativo.

#### 2.4.0.5. Pruebas de hipótesis: Supuesto de normalidad.

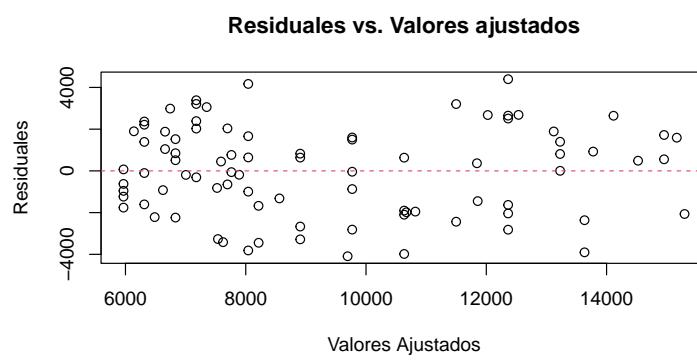
$$H_0: \varepsilon_i \sim \text{Normal.} \text{ vs } H_1: \varepsilon_i \not\sim \text{Normal}$$



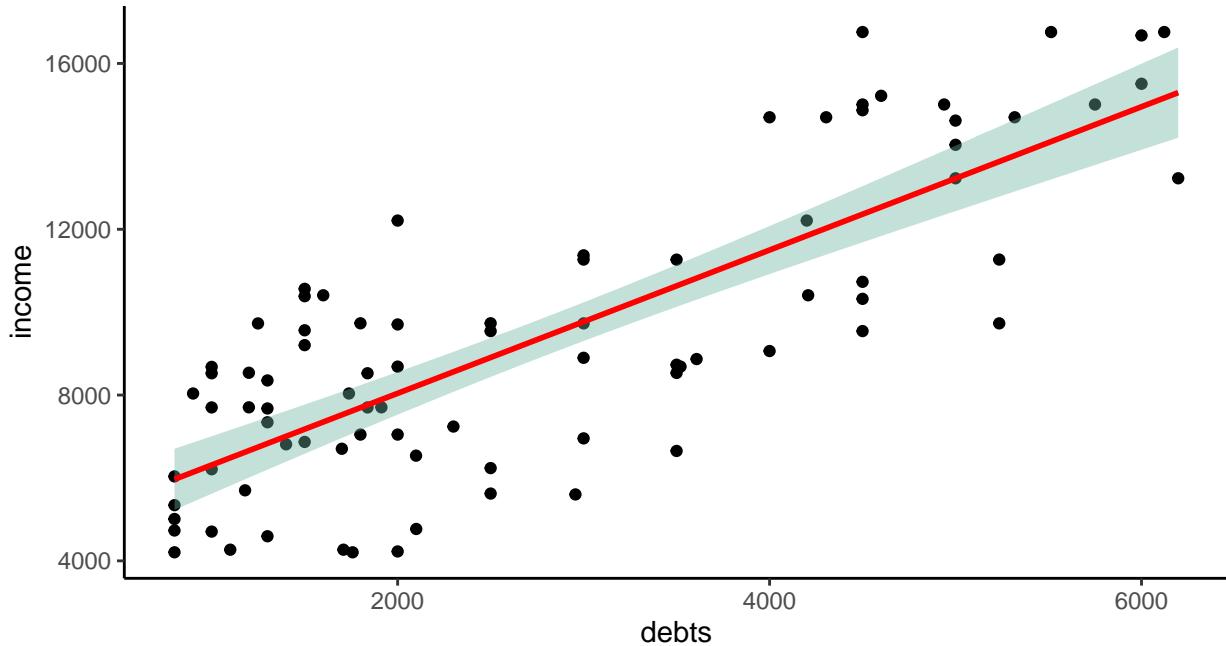
Como el patrón de los residuales sigue en su mayoría la línea roja que representa el ajuste de la distribución de los residuales a una distribución normal, se concluye que el supuesto de normalidad se cumple. Lo cual se ratifica en el resultado de la prueba de normalidad de Shapiro-Wilk con un valor-P mayor a 0.05, por lo cual no se rechaza  $H_0$  y se concluye que los residuales se distribuyen normal.

#### 2.4.0.6. Supuesto de varianza constante.

Se realizó una prueba gráfica comparando los residuales con los valores ajustados para analizar su distribución.



De la gráfica se observa que el patrón formado por la nube de puntos no se aleja mucho de un patrón rectangular. Lo que nos da un indicio de homocedasticidad de varianza.



Del gráfico anterior y de las pruebas realizada nos permite concluir que una regresión lineal puede ser un modelo adecuado para poder explicar **income** dado que un cliente tiene una determinadad caracteristica **debts**.

**2.5. d) Use los respectivos ajuste de cada uno de los modelos anteriores y con el conjunto de prueba, calcule el test-MSE. Qué observa?**

```
test_df <- cbind(test, y_test) %>% as.data.frame()
colnames(test_df) <- c("debts", "income")

#testMSE Regresion lineal
preds <- predict(model, test_df)
modelEval <- cbind(test_df$income, preds) %>% as.data.frame()
mse_ml <- mean((modelEval$V1 - modelEval$preds)^2)
mse_ml

#testMSE Regresion KNN
pred_y = predict(knnmodel, data.frame(test_scale))
mse_knn = mean((y_test - pred_y)^2)
mse_knn
```

testMSE Regresión Lineal	testMSE Regresión Knn
3224949	4398259

Se observa que el modelo con menor **test MSE** es el de la regresión lineal. Dicho modelo a parte de hacer una mejor predicción también permite hacer inferencia, se recomienda dicho modelo como el modelo mas adecuado comparado respecto a los modelos obtenidos usando el método de aprendizaje estadístico KNN.

## 2.6. e) Usando todos los datos y regresión lineal multiple seleccione un modelo usando forward, backward y stepwise

### 2.6.0.1. Forward

```
##                                     Selection Summary
## -----
##   Variable      AIC      Sum Sq      RSS      R-Sq      Adj. R-Sq
##   age          2054.821  1006482664.253  426207906.976  0.70251  0.69986
##   occupation   1867.919  1355578734.167  77111837.063  0.94618  0.94369
##   debts         1857.825  1363340054.177  69350517.052  0.95159  0.94888
##   marital_status 1856.570  1366464650.085  66225921.144  0.95378  0.95025
## -----
```

### 2.6.0.2. Backward

```
##                                     Backward Elimination Summary
## -----
##   Variable      AIC      RSS      Sum Sq      R-Sq      Adj. R-Sq
##   Full Model   1878.828  35920283.953  1396770287.276  0.97493  0.95198
##   state        1866.024  57279038.724  1375411532.505  0.96002  0.95089
##   race         1859.074  59873471.024  1372817100.205  0.95821  0.95181
##   gender       1857.090  59881686.253  1372808884.976  0.95820  0.95229
##   credit_score 1855.697  60201373.493  1372489197.737  0.95798  0.95252
## -----
```

### 2.6.0.3. Stepwise

```

## 
## 
##                               Stepwise Summary
## -----
##   ## Variable      Method     AIC      RSS      Sum Sq      R-Sq
##   ## -----
##   ## age           addition  2054.821  426207906.976  1006482664.253  0.70251
##   ## occupation   addition  1867.919  77111837.063   1355578734.167  0.94618
##   ## debts        addition  1857.825  69350517.052   1363340054.177  0.95159
##   ## marital_status addition  1856.570  66225921.144   1366464650.085  0.95378
##   ##

```

## 2.7. f) Seleccione uno de los modelos del paso anterior y responda con argumentación la pregunta: Ajusta bien dicho modelo?

Dado que los métodos de selección automáticos como: Forward y Stepwise nos indican que dos modelo plausibles para explicar **income** son el modelo con la covariable **marital\_status** y el modelo con la covariable **debts** obtamos por seleccionar el modelo con la covariable **debts** ya que cuenta con una de las mejores metricas(AIC, R-sq, Adj.R-sq) dadas por el metodo de selección automático de variables y como vimos anteriormente es un modelo que cumple con los supuesto de un modelo lineal ademas de tener una mejor predicción que los modelos de regresión Knn visto en este trabajo. Lo que nos permite concluir que dicho modelo es adecuado y tiene un buen ajuste.

## 3. Punto 3

Se utilizan las técnicas **ridge** y **lasso** para regularizar las bases de datos **DATOS\_A** y **DATOS\_B**. Se busca saber ¿Cuáles variables aparentemente muestran no ser relevantes para explicar la variable aleatoria **Y**?

Primeramente, es menester cargar las bases de datos y descubrir la estructura y comportamiento de dichas bases:

### 3.0.1. DATOS\_A

Se procede a cargar y examinar la base de datos:

```

##      Y      X1      X2      X3      X4      X5      X6      X7      X8
## 1 11.19162 -1.08114  9.1861 -0.0937  2.8957  1.5078  4.34157  5.2942  5.9344
## 2 22.60943 -1.28549  8.6449  6.1533  0.1205  3.0802  5.24621  6.8235  5.1159
## 3 29.71620  0.85479  6.6983  5.8730 10.6547 10.0062  4.93574  3.3363  3.6772
## 4 10.33344  0.80922 13.0474  8.6096  3.6979 10.2647  6.25590 11.2681  7.4279

```

```

## 5 28.84943 -0.26611 9.3007 0.5411 11.7838 8.2244 2.88766 2.3122 2.6440
## 6 29.63682 1.31320 3.2235 0.6024 3.2813 3.9245 4.97632 5.0332 3.6060
##          X9      X10      X11      X12      X13      X14
## 1 11.1092 8.46643 11.1092 1.9148 5.2417 11.0799
## 2 13.2634 17.64208 13.2634 -2.3003 8.4040 7.0935
## 3 10.0689 29.27286 10.0689 -2.7403 8.8594 5.5088
## 4 11.2968 9.52291 11.2968 0.4165 4.7300 5.6324
## 5 8.9499 19.12384 8.9499 8.7977 9.6763 7.7176
## 6 10.6468 17.81611 10.6468 0.1442 7.1415 6.5541

##          Y      X1      X2      X3      X4      X5      X6      X7      X8
## 1762 7.01276 -2.32589 4.1752 3.0142 -1.2306 7.3331 4.67011 6.0340 3.0386
## 1763 4.59556 -1.75741 13.0582 3.1893 0.6289 8.8093 7.29293 3.0454 0.7160
## 1764 30.45841 1.31920 12.2524 7.0773 4.0676 8.6815 4.05778 10.2287 3.3240
## 1765 43.26226 -0.18903 14.5321 7.9787 8.7490 9.8863 1.16916 8.5789 0.6577
## 1766 11.80182 -0.26799 6.8936 0.2812 13.7735 3.0945 7.97053 4.5097 -1.9575
## 1767 35.24450 0.37281 2.9637 6.8167 -0.4087 10.5603 3.49900 4.3150 7.8607
##          X9      X10      X11      X12      X13      X14
## 1762 8.9507 7.31456 8.9507 4.1019 5.3981 3.6470
## 1763 7.3183 10.59786 7.3183 2.3539 3.7079 8.0791
## 1764 6.6937 25.50611 6.6937 -2.1939 5.0697 2.3640
## 1765 6.3572 7.66677 6.3572 9.6995 6.2937 3.7943
## 1766 6.1171 10.85878 6.1171 0.5539 10.8068 5.3767
## 1767 10.4717 25.87824 10.4717 1.2825 3.4358 4.1977

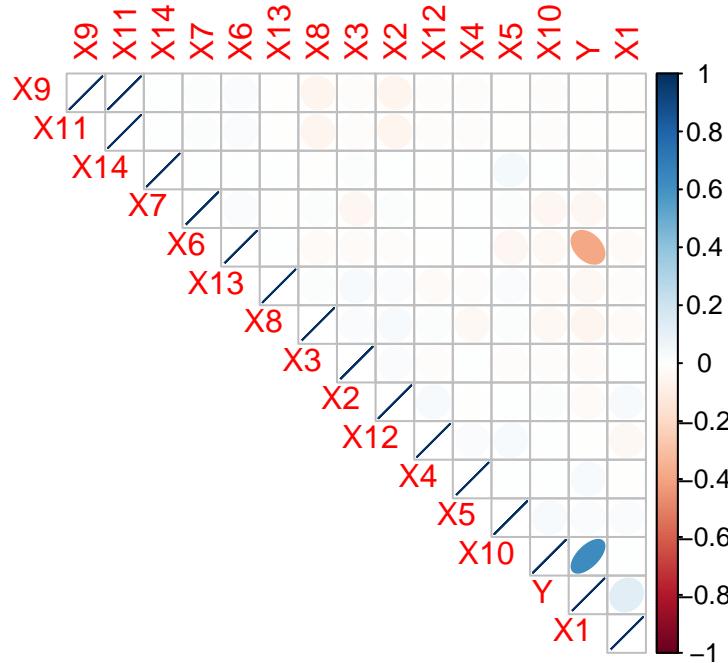
## 'data.frame': 1767 obs. of 15 variables:
## $ Y : num 11.2 22.6 29.7 10.3 28.8 ...
## $ X1 : num -1.081 -1.285 0.855 0.809 -0.266 ...
## $ X2 : num 9.19 8.64 6.7 13.05 9.3 ...
## $ X3 : num -0.0937 6.1533 5.873 8.6096 0.5411 ...
## $ X4 : num 2.9 0.12 10.65 3.7 11.78 ...
## $ X5 : num 1.51 3.08 10.01 10.26 8.22 ...
## $ X6 : num 4.34 5.25 4.94 6.26 2.89 ...
## $ X7 : num 5.29 6.82 3.34 11.27 2.31 ...
## $ X8 : num 5.93 5.12 3.68 7.43 2.64 ...
## $ X9 : num 11.11 13.26 10.07 11.3 8.95 ...
## $ X10: num 8.47 17.64 29.27 9.52 19.12 ...
## $ X11: num 11.11 13.26 10.07 11.3 8.95 ...
## $ X12: num 1.915 -2.3 -2.74 0.416 8.798 ...
## $ X13: num 5.24 8.4 8.86 4.73 9.68 ...
## $ X14: num 11.08 7.09 5.51 5.63 7.72 ...

```

Se observa como la base de datos **datos\_a** no presenta valores *NA*, además, todas las variables presentes en ella son de tipo numérico, cuenta con 15 variables (*Y*, *X1*, *X2*, ..., *X14*) y 1767 observaciones.

### 3.0.2. Análisis descriptivo

A manera descriptiva se realizará una matriz de correlación de los datos.



Se observa como las covariables que tienen más correlación con la variable respuesta **Y** son **X10** y **X6** con correlación positiva y negativa respectivamente.

Luego de haber realizado el análisis descriptivo y haber examinado la base de datos, se procede con las técnicas de regularización mediante regresión *ridge* y *lasso*.

### 3.1. Ridge

Los métodos de **subset** emplean mínimos cuadrados ordinarios (OLS) para ajustar un modelo lineal que contiene únicamente un subconjunto de predictores. Otra alternativa, conocida como regularización o **shrinkage**, consiste en ajustar el modelo incluyendo todos los predictores pero aplicando una penalización que fuerce a que las estimaciones de los coeficientes de regresión tiendan a cero. Con esto se intenta evitar **overfitting**, reducir varianza, atenuar el efecto de la correlación entre predictores y minimizar la influencia en el modelo de los predictores menos relevantes. Por lo general, aplicando regularización se consigue modelos con mayor poder predictivo (generalización). En esta ocasión, se utilizan los métodos de regularización Ridge y Lasso.

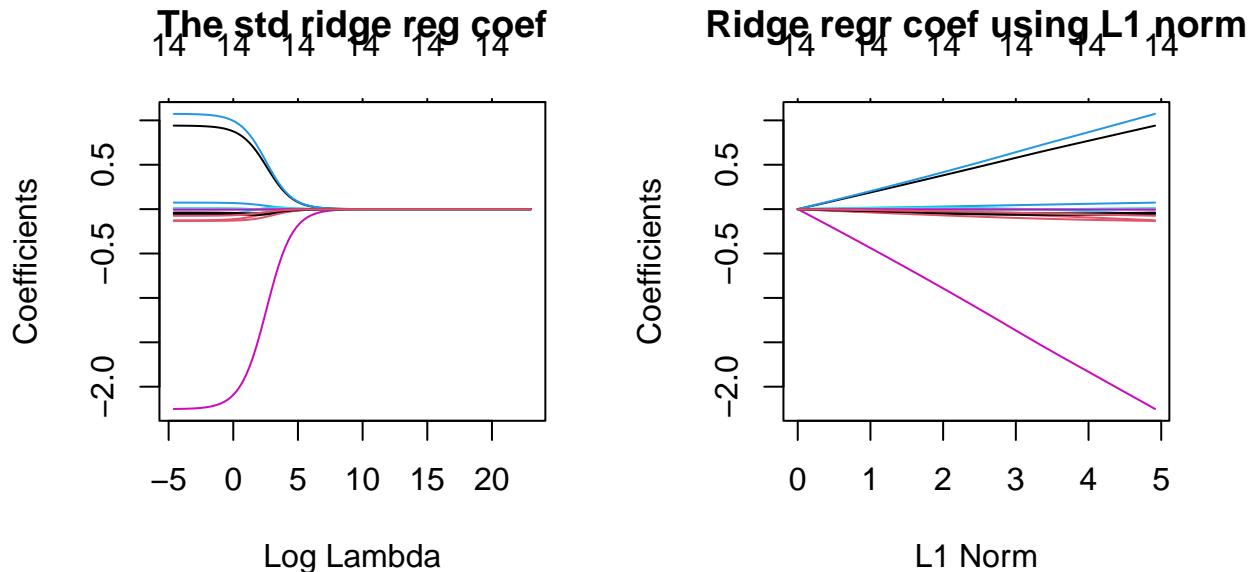
La regularización Ridge penaliza la suma de los coeficientes elevados al cuadrado ( $\|\beta\|^2 = \sum_{j=1}^p |\beta_j|^2$ ), La cual tiene el efecto de reducir de forma proporcional el valor de todos los coeficientes del modelo pero sin que estos lleguen a cero. El grado de penalización está controlado por el hiperparámetro  $\lambda$ . Cuando  $\lambda = 0$ , la penalización es nula y el resultado

es equivalente al de un modelo lineal por mínimos cuadrados ordinarios (OLS). A medida que  $\lambda$  aumenta, mayor es la penalización y menor el valor de los predictores.

A continuación, se presenta la realización de una regresión **Ridge** para la base **datos\_a**:

Primero se crea el modelo de regresión **ridge**:

```
## [1] 15 100
```

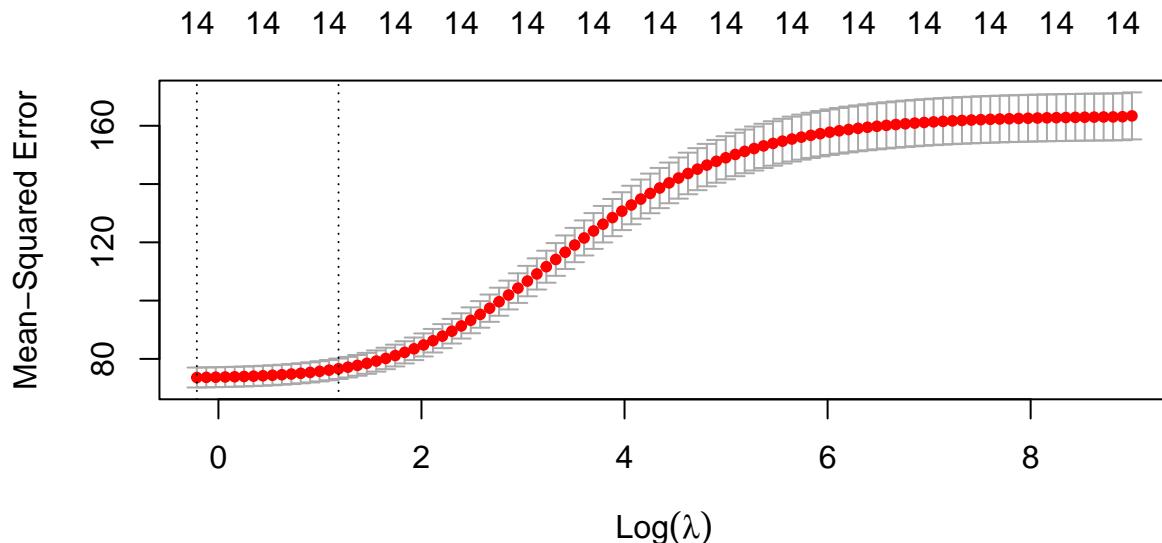


De los dos gráficos, se observa como en el de los coeficientes estandarizados los valores convergen a cero a un valor loglambda aproximadamente igual a siete. Mientras que, en el segundo gráfico de los coeficientes usando L1 norm, los valores de los coeficientes de la regresión ridge convergen cuando este valor L1 es 0.

Luego, con los resultados obtenidos anteriormente, se procede a realizar la validación cruzada para cada valor de  $\lambda$  y así estimar el error de validación cruzada.

```
##
## Call: cv.glmnet(x = x[train], y = y[train], alpha = 0)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min   0.809    100    73.61 3.423       14
## 1se   3.264     85    76.61 3.502       14
```

Después, se escoge el “mejor”  $\lambda$ , es decir, el que produzca el menor error.



```
## [1] 0.8085394
## [1] -0.2125259
```

En este caso, se obtuvo que  $\lambda = 0.8085394$  es el “mejor”, es decir, el que produce el menor error.

Y finalmente, se realizan las predicciones del modelo de regresión **ridge** para los coeficientes de las variables predictoras.

```
## [1] 68.21606
## (Intercept)          X1          X2          X3          X4          X5
## 18.442283956  0.887863861 -0.115586713 -0.059517250  0.069355939 -0.049739631
##          X6          X7          X8          X9          X10         X11
## -2.121483883 -0.057218016 -0.129631318  0.007041305  1.008890012  0.007280381
##          X12         X13         X14
## -0.005780133 -0.042258105 -0.069982641
```

### 3.2. Lasso

La regularización Lasso penaliza la suma del valor absoluto de los coeficientes de regresión ( $\|\beta\| = \sum_{j=1}^p |\beta_j|$ ). Lo cual tiene el efecto de forzar a que los coeficientes de los

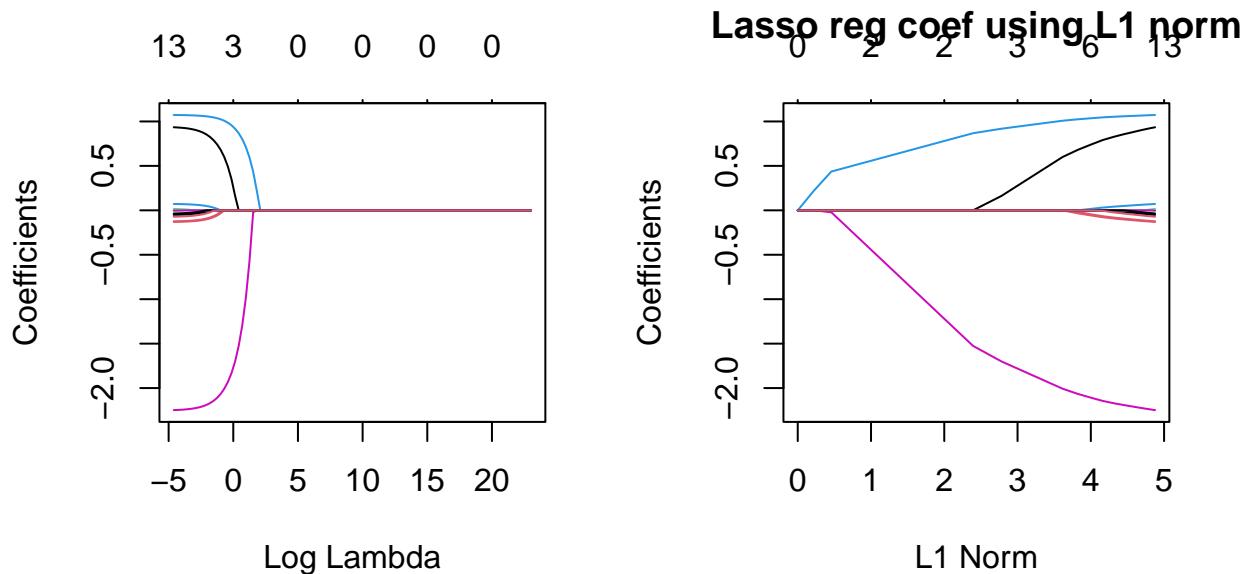
predictores tiendan a cero. Dado que un predictor con coeficiente de regresión cero no influye en el modelo, lasso consigue excluir los predictores menos relevantes. Al igual que en ridge, el grado de penalización está controlado por el hiperparámetro  $\lambda$ . Cuando  $\lambda = 0$ , el resultado es equivalente al de un modelo lineal por mínimos cuadrados ordinarios. A medida que  $\lambda$  aumenta, mayor es la penalización y más predictores quedan excluidos.

Seguidamente, se presenta la realización de una regresión **Lasso** para la base **datos\_a**:

Primero se crea el modelo de regresión **lasso**:

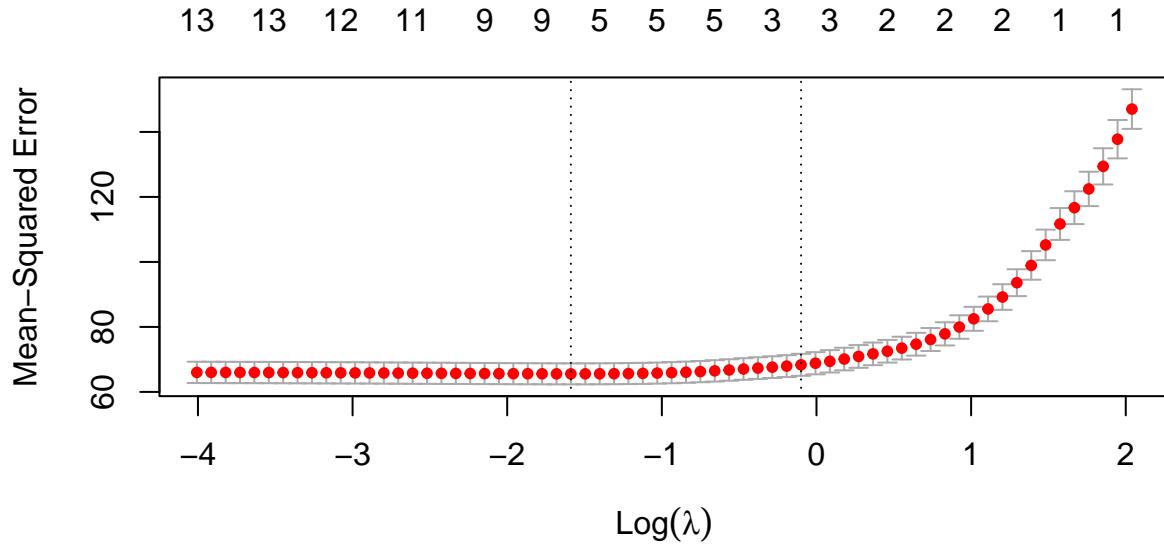
```
## [1] 15 100
```

Luego se grafican la regresión buscando observar el valor de lambda adecuado:



De los dos gráficos, se observa como en el de los coeficientes estandarizados los valores convergen a cero a un valor loglambda entre 0 y 1. Mientras que, en el segundo gráfico de los coeficientes usando L1 norm, los valores de los coeficientes de la regresión lasso convergen cuando este valor L1 es 4 aproximadamente.

Luego, con los resultados obtenidos anteriormente, se procede a realizar la validación cruzada para cada valor de  $\lambda$  y así estimar el error de validación cruzada.



```
## [1] 0.2042918
```

```
## [1] -1.588206
```

Se obtuvo que  $\lambda = 0.2042918$  es el “mejor”, es decir, el que produce el menor error.

Y finalmente, se realizan las predicciones del modelo de regresión lasso para los coeficientes de las variables predictoras.

```
## [1] 76.15713
```

```
## (Intercept)          X1          X2          X3          X4          X5
## 16.066817488  0.798032232 -0.069718592 -0.004294575  0.033077479  0.000000000
##           X6          X7          X8          X9          X10         X11
## -2.149978765  0.000000000 -0.080835898  0.000000000  1.047160803  0.000000000
##          X12          X13          X14
## 0.000000000  0.000000000 -0.009306422

## (Intercept)          X1          X2          X3          X4          X6
## 16.066817488  0.798032232 -0.069718592 -0.004294575  0.033077479 -2.149978765
##          X8          X10          X14
## -0.080835898  1.047160803 -0.009306422
```

Note que, en estos últimos resultados arrojados por el R, se muestra como a diferencia de la regresión ridge, lasso fuerza y los hace cero a los coeficientes que con este criterio son los que menos aportan a explicar la variable **Y**. Por lo cual, las variables que si se toman en cuenta por la regresión lasso son las variables **X1**, **X2**, **X3**, **X4**, **X6**, **X8**, **X10** y **X14**.

### 3.3. Conclusiones y respuesta a la pregunta planteada

- Se observa como usando la regresión ridge, esta nos arrojó un  $\lambda = 0.80$  el cual es mejor para minimizar el RSS. Además, los coeficientes arrojados por este tipo de regresión después de haber aplicado validación cruzada muestra aparentemente que, las variables que muestran no ser relevantes para explicar la variable respuesta **Y** son las variables **X9**, **X11** y **X12**, ya que son las que presentan los valores de los coeficientes más cercanos a cero en comparación con las demás.
- Se observa como usando la regresión lasso, esta nos arrojó un  $\lambda = 0.20$  el cual es mejor para minimizar el RSS. Además, los coeficientes arrojados por este tipo de regresión después de haber aplicado validación cruzada muestra aparentemente que, las variables que muestran no ser relevantes para explicar la variable respuesta **Y** son las variables que no se toman en cuenta por la regresión lasso en la estimación de coeficientes mediante Cv. Es decir, las variables **X5**, **X6**, **X7**, **X9**, **X11**, **X12** y **X13**.

#### 3.3.1. DATOS\_B

Se procede a cargar y examinar la base de datos:

```
##      Y     X1 X2  X3
## 1 1112 22548  1 LUN
## 2  399 19075  1 MAR
## 3   103 15425  0 MIE
## 4   502 19859  0 JUE
## 5   181 16774  0 VIE
## 6   280 18223  0 SAB

##      Y     X1 X2  X3
## 723  591 20519  0 MAR
## 724  307 18532  0 MIE
## 725  317 18724  0 JUE
## 726  998 22295  1 VIE
## 727 1168 23021  1 SAB
## 728    90 16082  1 DOM

## 'data.frame': 728 obs. of  4 variables:
##   $ Y : int  1112 399 103 502 181 280 158 1380 216 301 ...
##   $ X1: int  22548 19075 15425 19859 16774 18223 17810 23312 17447 18159 ...
##   $ X2: int  1 1 0 0 0 0 0 0 1 1 ...
##   $ X3: chr  "LUN" "MAR" "MIE" "JUE" ...
```

Se observa como la base de datos **datos\_b** no presenta valores *NA*, además, tres de sus variables son de tipo int (entero), aunque la variable **X2** podría ser de tipo factor, ya

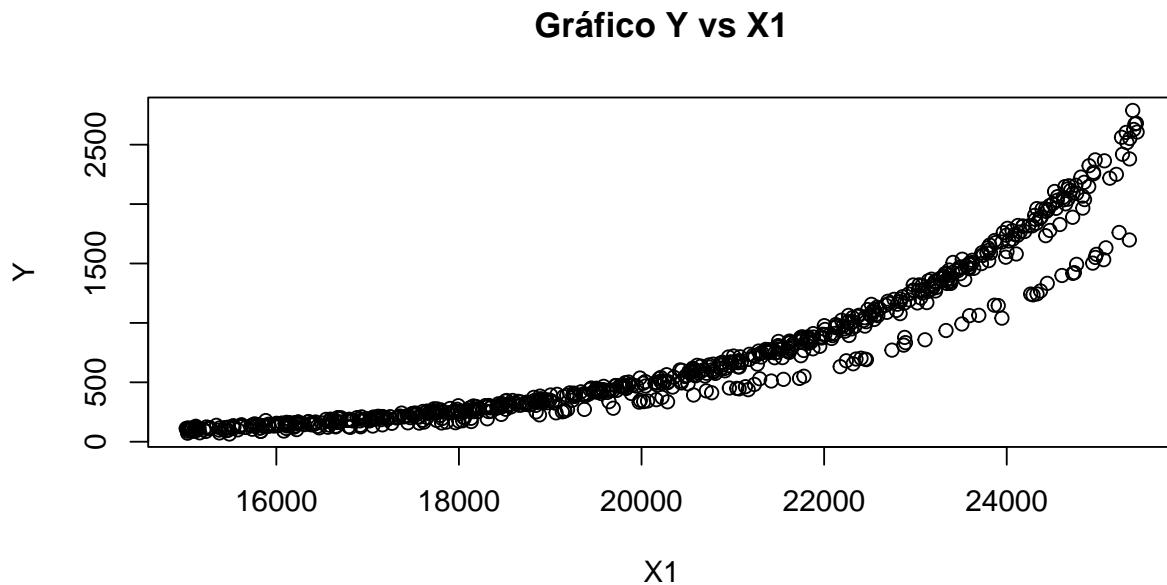
que solo cuenta con valores de 0 y 1. Y la variable **X3** es de tipo carácter, que cuenta con los 7 días de la semana. La base cuenta con 4 variables (Y, X1, X2, X3) y 728 observaciones.

```
## 'data.frame': 728 obs. of 4 variables:
## $ Y : int 1112 399 103 502 181 280 158 1380 216 301 ...
## $ X1: int 22548 19075 15425 19859 16774 18223 17810 23312 17447 18159 ...
## $ X2: Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 2 2 ...
## $ X3: Factor w/ 7 levels "DOM","JUE","LUN",...: 3 4 5 2 7 6 1 3 4 5 ...
```

### 3.3.2. Análisis descriptivo

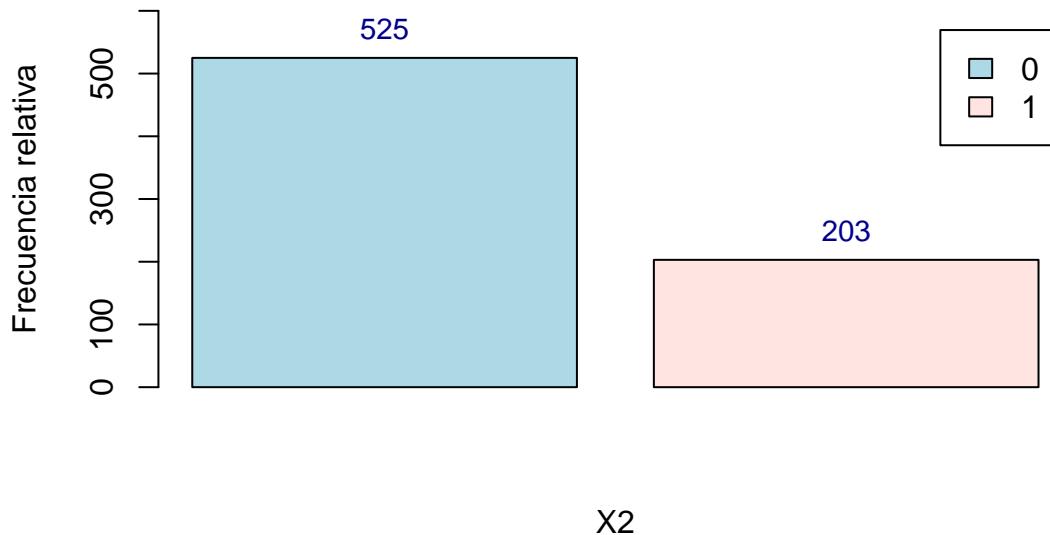
Se presentan algunos análisis descriptivos que ayudan a dilucidar el comportamiento de las covaraibles X's con respecto a la variable respuesta Y.

```
## [1] 0.9127479
```



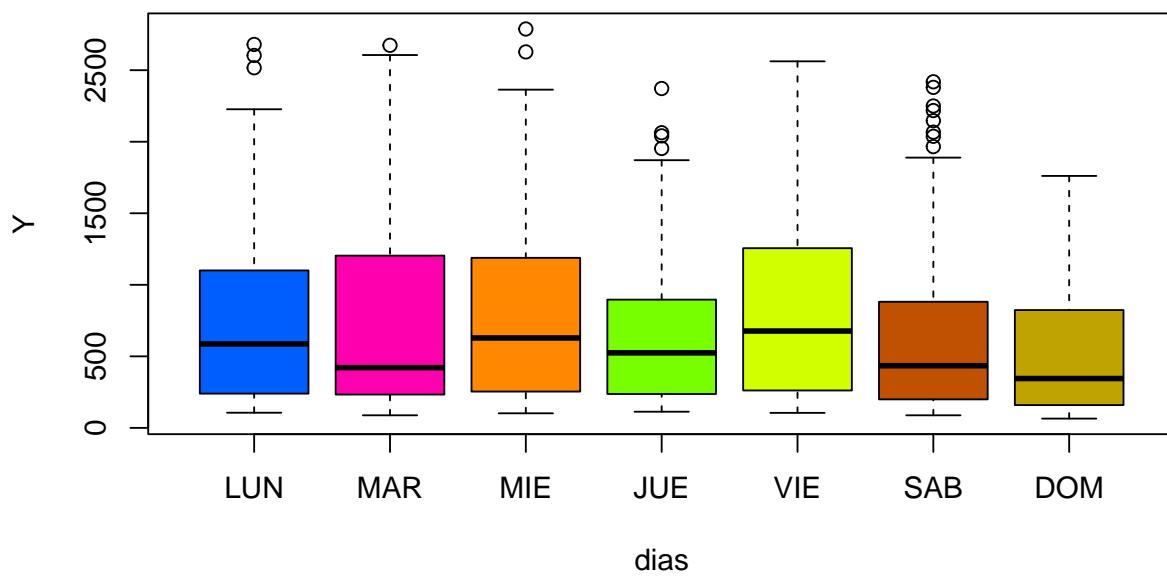
Se observa como la  $\text{Corr}(Y, X1) = 0.91$ , lo cual implica que existe alta correlación positiva entre las variables. Por otro lado, el gráfico entre estas variables, muestra como existe una relación cuadrática o cúbica entre estas dos variables.

### Barplot de la variable X2



Del gráfico del barplot de la variable X1 con respecto a la variable Y, se observa la diferencia entre las frecuencias en los valores “0” y “1” de la variable X2.

### Boxplot por días



Se observa en el gráfico de boxplot por días, como todas las cajas correspondientes a los días de la semana se encuentran centradas alrededor del valor de 500 de la variable Y. Además, los días que presentan mayor variabilidad en sus cajas son los lunes, martes, miércoles y viernes.

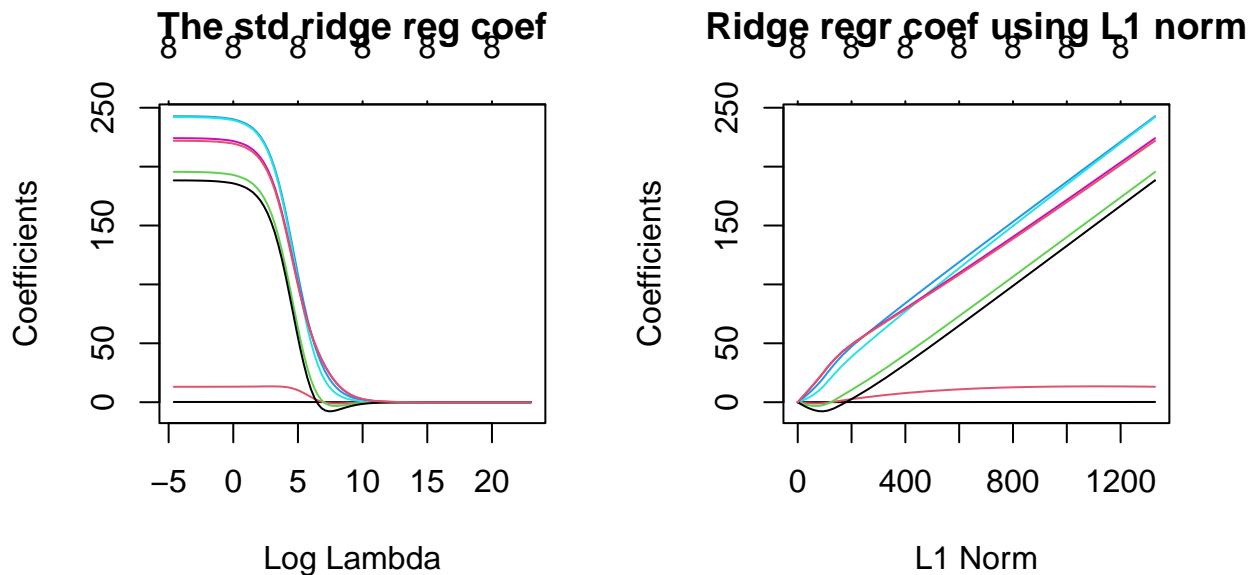
Luego de haber realizado el análisis descriptivo y haber examinado la base de datos, se procede con las técnicas de regularización mediante regresión *ridge* y *lasso*.

Recordar que, el objetivo principal de estas metodologías es reducir la varianza de los estimadores de los parámetros, y con ello, dar respuesta a la pregunta de interés.

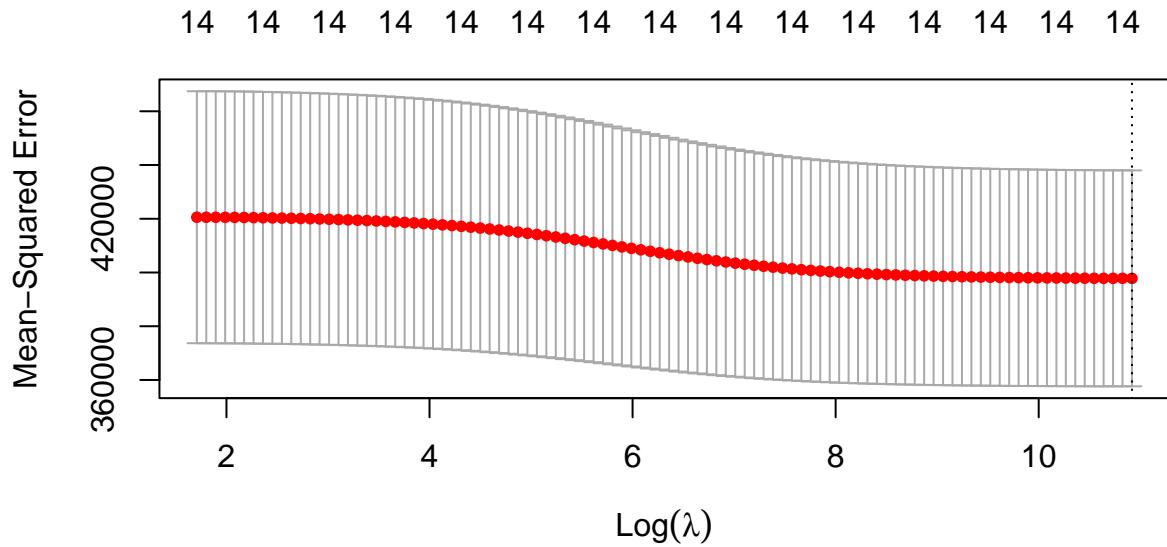
### 3.4. Ridge

Como se pudo notar ya se detalló anteriormente el uso y funcionamiento del método, así que ahora solo se procede a mostrar los resultados arrojados en este caso para la base **datos\_b**.

```
## [1] 9 100
```



```
##
## Call: cv.glmnet(x = x[train_b], y = y_b[train_b], alpha = 0)
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min   55194     1 397823 40180       14
## 1se   55194     1 397823 40180       14
```



```

## [1] 55194.23

## [1] 10.91861

## [1] 353828.9

## (Intercept)          X1          X2          X3          X4
## 2.378860e+01 1.049046e-36 -7.617115e-38 -7.382806e-38 8.175169e-38
##          X5          X6          X7          X8          X9
## 1.159490e-37 -2.417240e-36 -1.901651e-37 -2.172279e-37 -4.959508e-38
##          X10         X11         X12         X13         X14
## 1.111890e-36 -4.959508e-38 -2.610940e-39 -1.403576e-37 -5.946633e-38

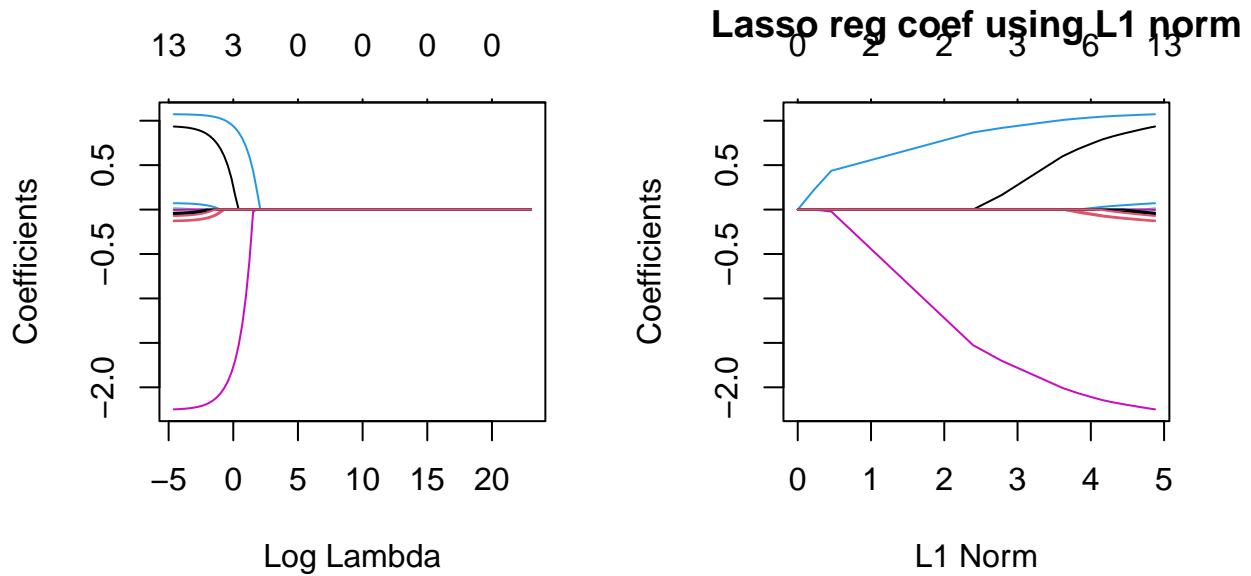
```

### 3.5. Lasso

Como se pudo notar ya se detalló anteriormente el uso y funcionamiento del método, así que ahora solo se procede a mostrar los resultados arrojados en este caso para la base **datos\_b**.

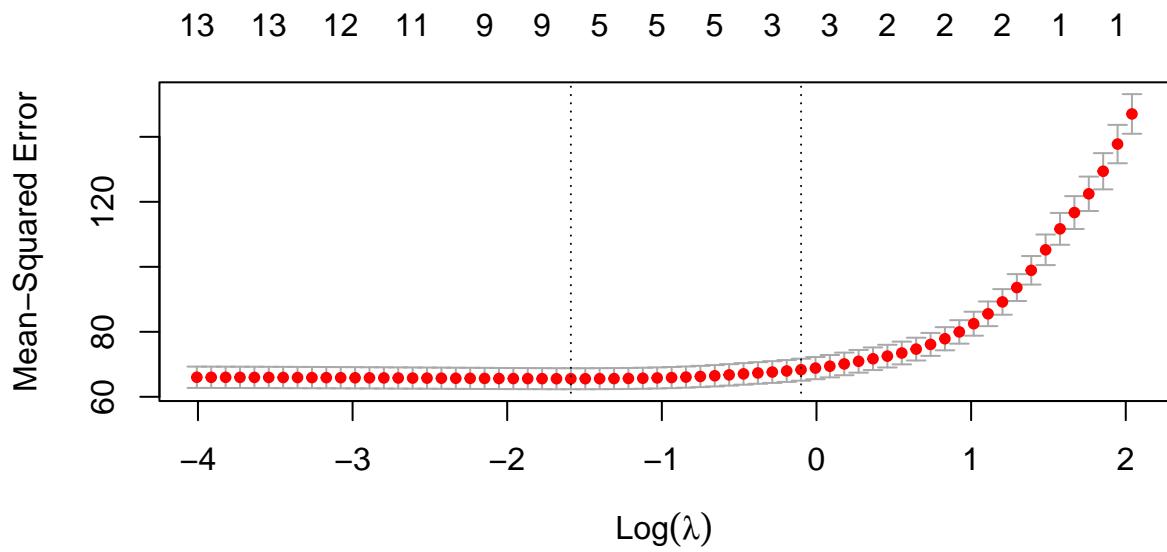
```
## [1] 15 100
```

Luego se grafican la regresión buscando observar el valor de lambda adecuado:



De los dos gráficos, se observa como en el de los coeficientes estandarizados los valores convergen a cero a un valor loglambda entre 0 y 1. Mientras que, en el segundo gráfico de los coeficientes usando L1 norm, los valores de los coeficientes de la regresión lasso convergen cuando este valor L1 es 4 aproximadamente.

Luego, con los resultados obtenidos anteriormente, se procede a realizar la validación cruzada para cada valor de  $\lambda$  y así estimar el error de validación cruzada.



```
## [1] 0.2042918
## [1] -1.588206
```

Se obtuvo que  $\lambda = 0.2042918$  es el “mejor”, es decir, el que produce el menor error.

Y finalmente, se realizan las predicciones del modelo de regresión lasso para los coeficientes de las variables predictoras.

```
## [1] 76.15713
```

	X1	X2	X3	X4	X5
## (Intercept)					
## 16.066817488	0.798032232	-0.069718592	-0.004294575	0.033077479	0.000000000
## X6	X7	X8	X9	X10	X11
## -2.149978765	0.000000000	-0.080835898	0.000000000	1.047160803	0.000000000
## X12	X13	X14			
## 0.000000000	0.000000000	-0.009306422			

	X1	X2	X3	X4	X6
## (Intercept)					
## 16.066817488	0.798032232	-0.069718592	-0.004294575	0.033077479	-2.149978765
## X8	X10	X14			
## -0.080835898	1.047160803	-0.009306422			

Note que, en estos últimos resultados arrojados por el R, se muestra como a diferencia de la regresión ridge, lasso fuerza y los hace cero a los coeficientes que con este criterio son los que menos aportan a explicar la variable **Y**. Por lo cual, las variables que si se toman en cuenta por la regresión lasso son las variables **X1**, **X2**, **X3**, **X4**, **X6**, **X8**, **X10** y **X14**.

### 3.6. Conclusiones y respuesta a la pregunta planteada

- Se observa como usando la regresión ridge, esta nos arrojó un  $\lambda = 0.80$  el cual es mejor para minimizar el RSS. Además, los coeficientes arrojados por este tipo de regresión después de haber aplicado validación cruzada muestra aparentemente que, las variables que muestran no ser relevantes para explicar la variable respuesta **Y** son las variables **X9**, **X11** y **X12**, ya que son las que presentan los valores de los coeficientes más cercanos a cero en comparación con las demás.
- Se observa como usando la regresión lasso, esta nos arrojó un  $\lambda = 0.20$  el cual es mejor para minimizar el RSS. Además, los coeficientes arrojados por este tipo de regresión después de haber aplicado validación cruzada muestra aparentemente que, las variables que muestran no ser relevantes para explicar la variable respuesta **Y** son las variables que no se toman en cuenta por la regresión lasso en la estimación de coeficientes mediante Cv. Es decir, las variables **X5**, **X6**, **X7**, **X9**, **X11**, **X12** y **X13**.

## 4. Punto 4

Utilizando los métodos de selección y validación cruzada, se realiza un análisis de la base de datos ***surgical*** del paquete ***olsrr*** donde se seleccionen las variables más importantes para explicar la variabilidad del tiempo de supervivencia.

Primero, es menester poner en contexto acerca de la base ***surgical***. Dicha base cuenta con información sobre la supervivencia de los pacientes que se someten a una operación de hígado. Está compuesta por 54 registros y 9 variables, las cuales se detallan a continuación:

- **bcs**: puntaje de coagulación de la sangre.
- **pindex**: índice de pronóstico.
- **enzyme\_test**: puntaje de la prueba de función enzimática.
- **liver\_test**: puntuación de la prueba de función hepática.
- **age**: edad en años.
- **gender**: variable indicadora de género (0 = masculino, 1 = femenino).
- **alc\_mod**: variable indicadora de antecedentes de consumo de alcohol (0 = ninguno, 1 = moderado).
- **alc\_heavy**: variable indicadora de antecedentes de consumo de alcohol fuerte (0 = Ninguno, 1 = Mucho).
- **y**: tiempo de supervivencia.

se procede a examinar la base de datos:

```
## [1] "bcs"          "pindex"        "enzyme_test"   "liver_test"    "age"
## [6] "gender"        "alc_mod"        "alc_heavy"     "y"

## [1] 54 9

##   bcs pindex enzyme_test liver_test age gender alc_mod alc_heavy   y
## 1 6.7    62      81     2.59  50      0      1      0 695
## 2 5.1    59      66     1.70  39      0      0      0 403
## 3 7.4    57      83     2.16  55      0      0      0 710
## 4 6.5    73      41     2.01  48      0      0      0 349
## 5 7.8    65     115     4.30  45      0      0      1 2343
## 6 5.8    38      72     1.42  65      1      1      0 348
```

```

##   bcs pindex enzyme_test liver_test age gender alc_mod alc_heavy   y
## 49  5.1     67         77      2.86   66     1       0       0  581
## 50  3.9     82        103      4.55   50     0       1       0 1078
## 51  6.6     77         46      1.95   50     0       1       0  405
## 52  6.4     85         40      1.21   58     0       0       1  579
## 53  6.4     59         85      2.33   63     0       1       0  550
## 54  8.8     78         72      3.20   56     0       0       0  651

## 'data.frame':   54 obs. of  9 variables:
## $ bcs       : num  6.7 5.1 7.4 6.5 7.8 ...
## $ pindex    : int  62 59 57 73 65 ...
## $ enzyme_test: int  81 66 83 41 115 ...
## $ liver_test : num  2.59 1.7 2.16 2.01 4.3 ...
## $ age        : int  50 39 55 48 45 ...
## $ gender     : int  0 0 0 0 1 ...
## $ alc_mod    : int  1 0 0 0 1 ...
## $ alc_heavy  : int  0 0 0 1 0 ...
## $ y          : int  695 403 710 349 2343 ...

```

Note que la base cuenta con todas sus variables numéricas, aunque se sabe que las variables **gender**, **alc\_mod** y **alc\_heavy** son de tipo indicadoras y cualitativas, correspondientes con el género e antecedentes de consumo de alcohol respectivamente. La base no cuenta con valores *NA*.

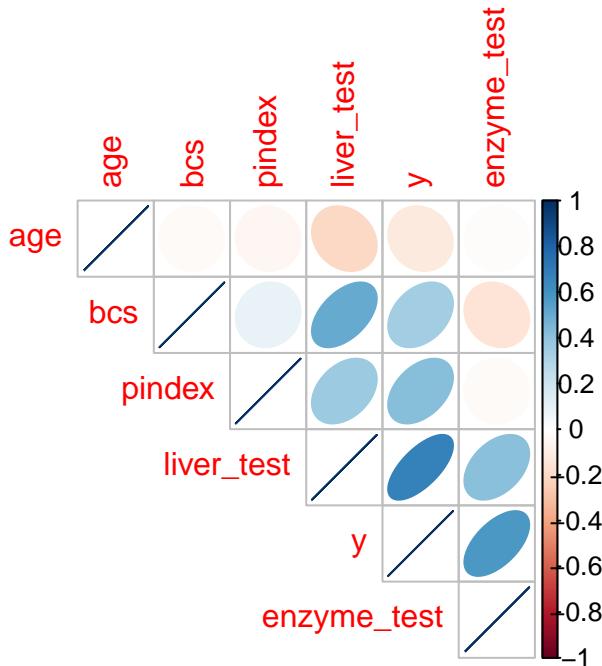
Adicionalmente, se presenta una tabla con los datos que contiene la base *surgical*.

Cuadro 3: Base Surgical

bcs	pindex	enzyme_test	liver_test	age	gender	alc_mod	alc_heavy	y
6.7	62	81	2.59	50	0	1	0	695
5.1	59	66	1.70	39	0	0	0	403
7.4	57	83	2.16	55	0	0	0	710
6.5	73	41	2.01	48	0	0	0	349
7.8	65	115	4.30	45	0	0	1	2343
5.8	38	72	1.42	65	1	1	0	348
5.7	46	63	1.91	49	1	0	1	518
3.7	68	81	2.57	69	1	1	0	749
6.0	67	93	2.50	58	0	1	0	1056
3.7	76	94	2.40	48	0	1	0	968
6.3	84	83	4.13	37	0	1	0	745
6.7	51	43	1.86	57	0	1	0	257
5.8	96	114	3.95	63	1	0	0	1573
5.8	83	88	3.95	52	1	0	0	858
7.7	62	67	3.40	58	0	0	1	702
7.4	74	68	2.40	64	1	1	0	809
6.0	85	28	2.98	36	1	1	0	682
3.7	51	41	1.55	39	0	0	0	205
7.3	68	74	3.56	59	1	0	0	550
5.6	57	87	3.02	63	0	0	1	838
5.2	52	76	2.85	39	0	0	0	359
3.4	83	53	1.12	67	1	1	0	353
6.7	26	68	2.10	30	0	0	1	599
5.8	67	86	3.40	49	1	1	0	562
6.3	59	100	2.95	36	1	1	0	651
5.8	61	73	3.50	62	1	1	0	751
5.2	52	86	2.45	70	0	1	0	545
11.2	76	90	5.59	58	1	0	1	1965
5.2	54	56	2.71	44	1	0	0	477
5.8	76	59	2.58	61	1	1	0	600
3.2	64	65	0.74	53	0	1	0	443
8.7	45	23	2.52	68	0	1	0	181
5.0	59	73	3.50	57	0	1	0	411
5.8	72	93	3.30	39	1	0	1	1037
5.4	58	70	2.64	31	1	1	0	482
5.3	51	99	2.60	48	0	1	0	634
2.6	74	86	2.05	45	0	0	0	678
4.3	8	119	2.85	65	1	0	0	362
4.8	61	76	2.45	51	1	1	0	637
5.4	52	88	1.81	40	1	0	0	705
5.2	49	72	1.84	46	0	0	0	536
3.6	28	99	1.30	55	0	0	1	582
8.8	86	88	6.40	30	1	1	0	1270

#### 4.0.1. Análisis descriptivo

Como un análisis descriptiva se realiza una matriz de correlación de los datos.



Del gráfico de correlaciones, se observa como la variable **liver\_test** es la que presenta mayor correlación positiva con respecto a la variable **y**. Es decir que, la puntuación de la prueba de función hepática está muy relacionada de manera directamente proporcional con el tiempo de supervivencia de los pacientes que se someten a una operación de hígado.

Después de poner a punto los datos y haber analizado los mismos, se realiza el proceso de selección y validación cruzada usando los métodos de **best subset**, **backward**, **forward** y **stepwise**. Además de aplicar la validación cruzada.

#### 4.1. “Best” Subset

Primero se procede a seleccionar el “mejor” subconjunto para estos datos:

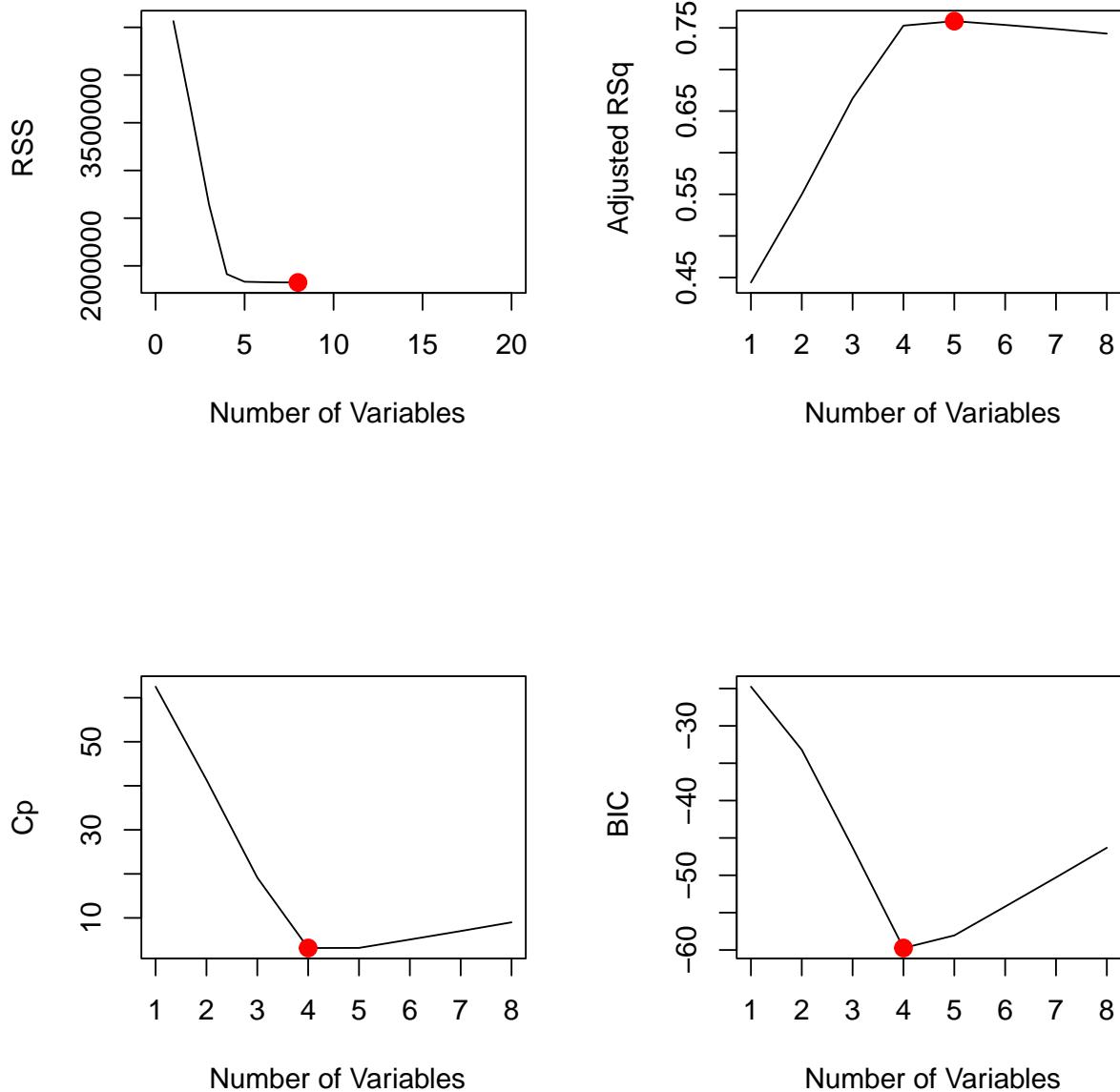
```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = surgical, nvmax = NULL, method = "exhaustive")
## 8 Variables  (and intercept)
##          Forced in    Forced out
## bcs          FALSE      FALSE
## pindex       FALSE      FALSE
## enzyme_test  FALSE      FALSE
## liver_test   FALSE      FALSE
```

```

## age          FALSE    FALSE
## gender1      FALSE    FALSE
## alc_mod1     FALSE    FALSE
## alc_heavy1   FALSE    FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##           bcs pindex enzyme_test liver_test age gender1 alc_mod1 alc_heavy1
## 1  ( 1 ) " " " " " " "*" " " " " " "
## 2  ( 1 ) " " " " " " "*" " " " " " "
## 3  ( 1 ) "*" "*" "*" " " " " " " " "
## 4  ( 1 ) "*" "*" "*" "*" " " " " " " "
## 5  ( 1 ) "*" "*" "*" "*" "*" " " " " "
## 6  ( 1 ) "*" "*" "*" "*" "*" " " " " "
## 7  ( 1 ) "*" "*" "*" "*" "*" "*" " " "
## 8  ( 1 ) "*" "*" "*" "*" "*" "*" "*" " "
##           rsq    adjr2      cp      rss      bic bcs pindex enzyme_test
## 1  ( 1 ) 0.4545389 0.4440492 62.511923 4565248 -24.75271
## 2  ( 1 ) 0.5667409 0.5497504 41.368078 3626171 -33.19970
## 3  ( 1 ) 0.6841275 0.6651752 19.154821 2643701 -46.27456 *
## 4  ( 1 ) 0.7713565 0.7526917 3.162146 1913636 -59.73701 *
## 5  ( 1 ) 0.7809054 0.7580831 3.192498 1833716 -58.05170 *
## 6  ( 1 ) 0.7814169 0.7535127 5.086996 1829436 -54.18892 *
## 7  ( 1 ) 0.7817703 0.7485615 7.014100 1826478 -50.28731 *
## 8  ( 1 ) 0.7818387 0.7430544 9.000000 1825906 -46.31525 *
##           liver_test age gender1 alc_mod1 alc_heavy1
## 1  ( 1 )          *
## 2  ( 1 )          *
## 3  ( 1 )
## 4  ( 1 )
## 5  ( 1 )          *
## 6  ( 1 )          *  *
## 7  ( 1 )          *  *  *
## 8  ( 1 )          *  *  *  *
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"

```

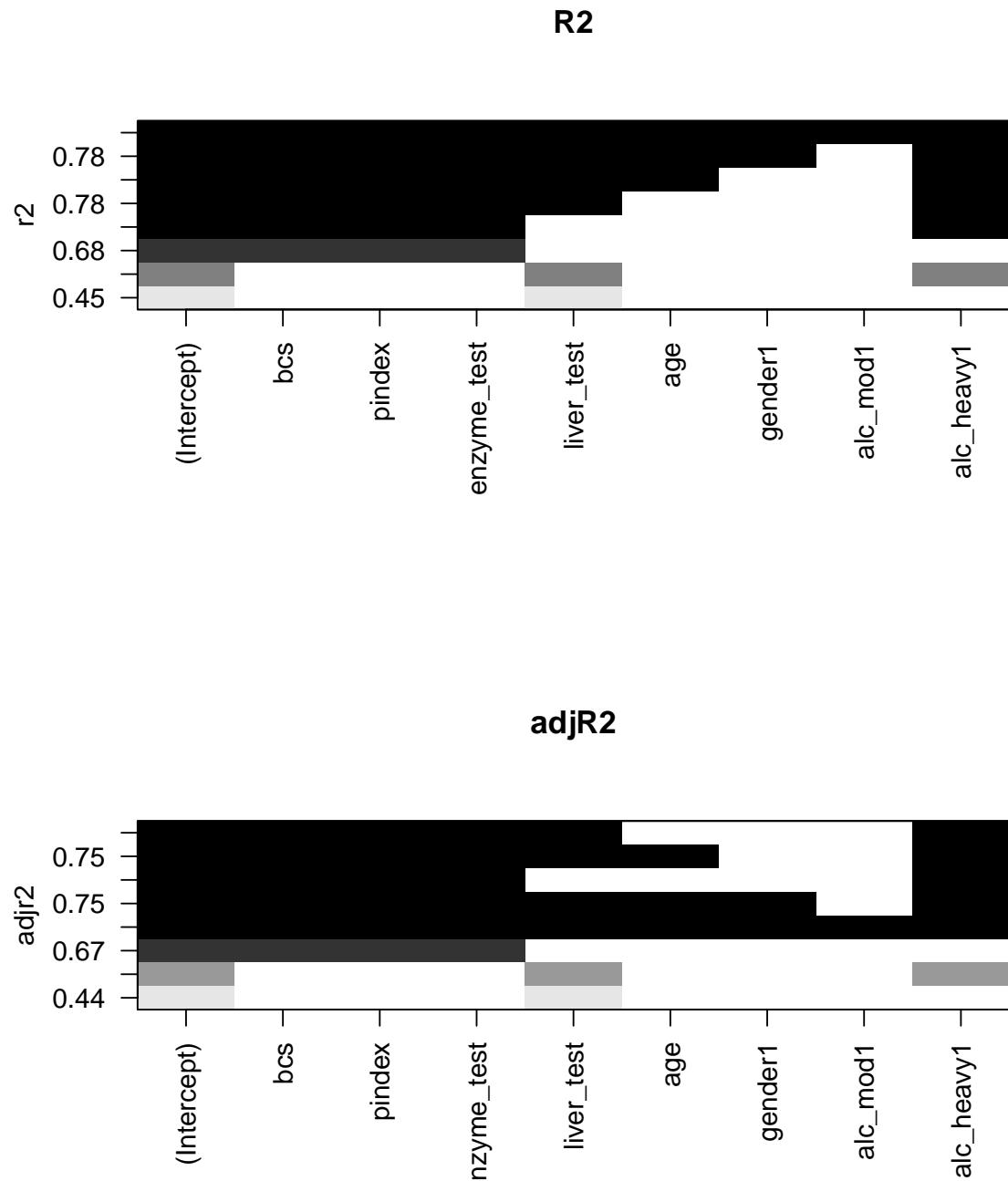
Luego, se presentan los gráficos con las 4 medidas de referencia que sugieren el número de parámetros a tomar.

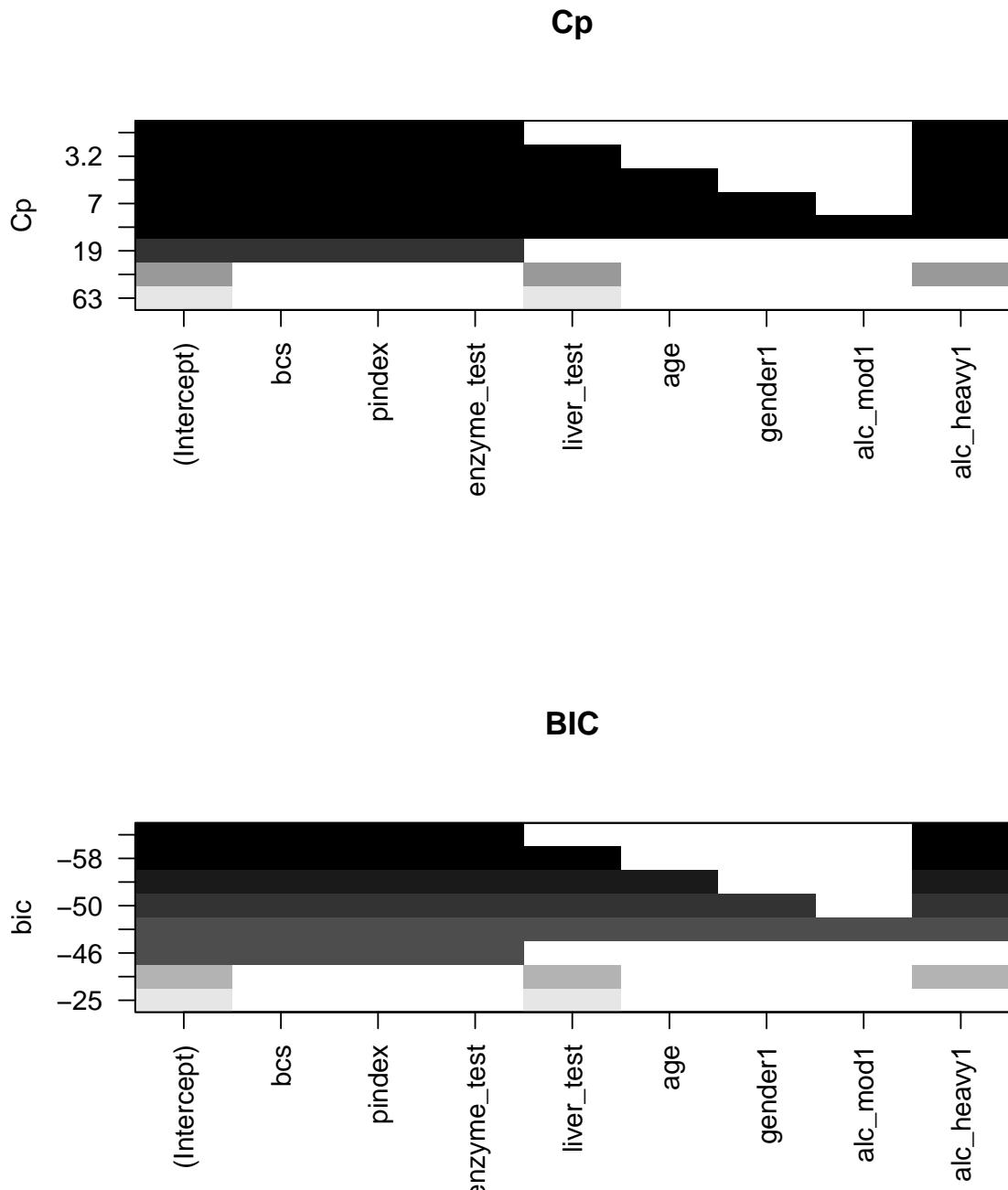


Se puede observar como, usando el método de “the best subset” se proponen las siguientes cantidades a tomar según los 4 estimadores estadísticos:

- **RSS:** propone 8 parámetros.
- **R<sub>2adj</sub>:** propone 5 parámetros.
- **C<sub>p</sub>:** propone 4 parámetros.
- **bIC:** propone 4 parámetros.

Ahora, se presentan también estos 4 graficos que nos ayudarán a dilucidar los parámetros que mejor ayudan a explicar la variable **y**





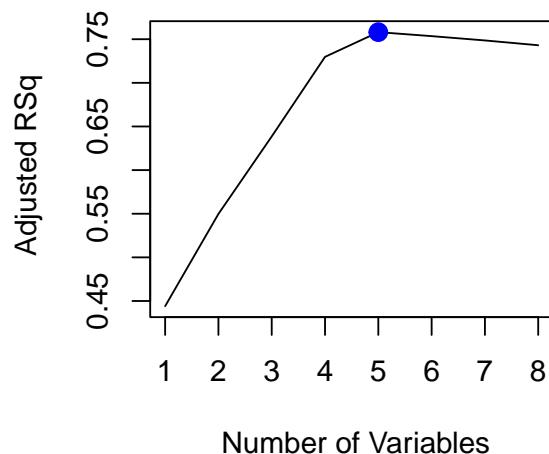
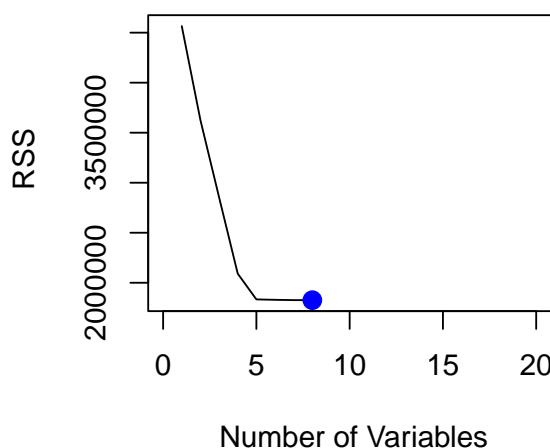
Se observa como, usando estas 4 medidas y el método “the best subset”, las variables que mejor ayudan a explicar la variable respuesta **y** son **bcs**, **pindex**, **enzyme\_test** y **alc\_heavy1**.

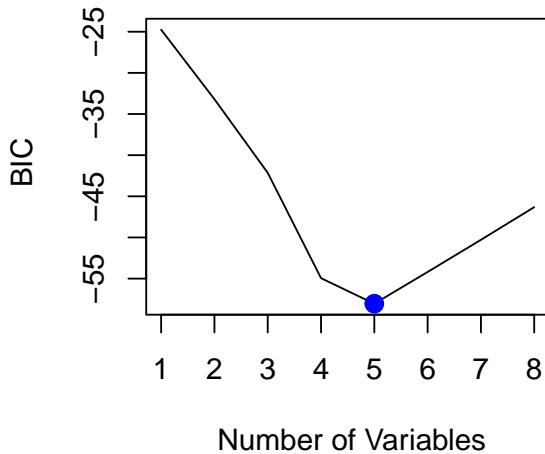
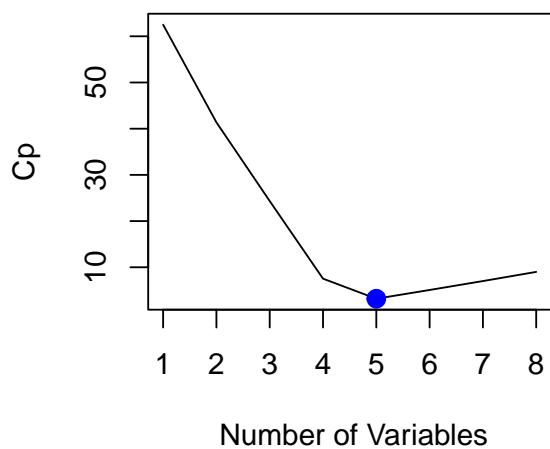
## 4.2. Forward

Primero se procede a seleccionar el “mejor” subconjunto para estos datos:

```
##          rsq      adjr2       cp      rss      bic bcs pindex enzyme_test
## 1  ( 1 ) 0.4545389 0.4440492 62.511923 4565248 -24.75271
## 2  ( 1 ) 0.5667409 0.5497504 41.368078 3626171 -33.19970
## 3  ( 1 ) 0.6590000 0.6385400 24.337853 2854006 -42.14120
## 4  ( 1 ) 0.7501457 0.7297495 7.537284 2091160 -54.94646      *
## 5  ( 1 ) 0.7809054 0.7580831 3.192498 1833716 -58.05170      *
## 6  ( 1 ) 0.7814169 0.7535127 5.086996 1829436 -54.18892      *
## 7  ( 1 ) 0.7817703 0.7485615 7.014100 1826478 -50.28731      *
## 8  ( 1 ) 0.7818387 0.7430544 9.000000 1825906 -46.31525      *
##          liver_test age gender1 alc_mod1 alc_heavy1
## 1          *           *
## 2          *           *
## 3          *           *
## 4          *           *
## 5          *           *
## 6          *   *
## 7          *   *       *
## 8          *   *       *       *
```

Luego, se presentan los gráficos con las 4 medidas de referencia que sugieren el número de parámetros a tomar.

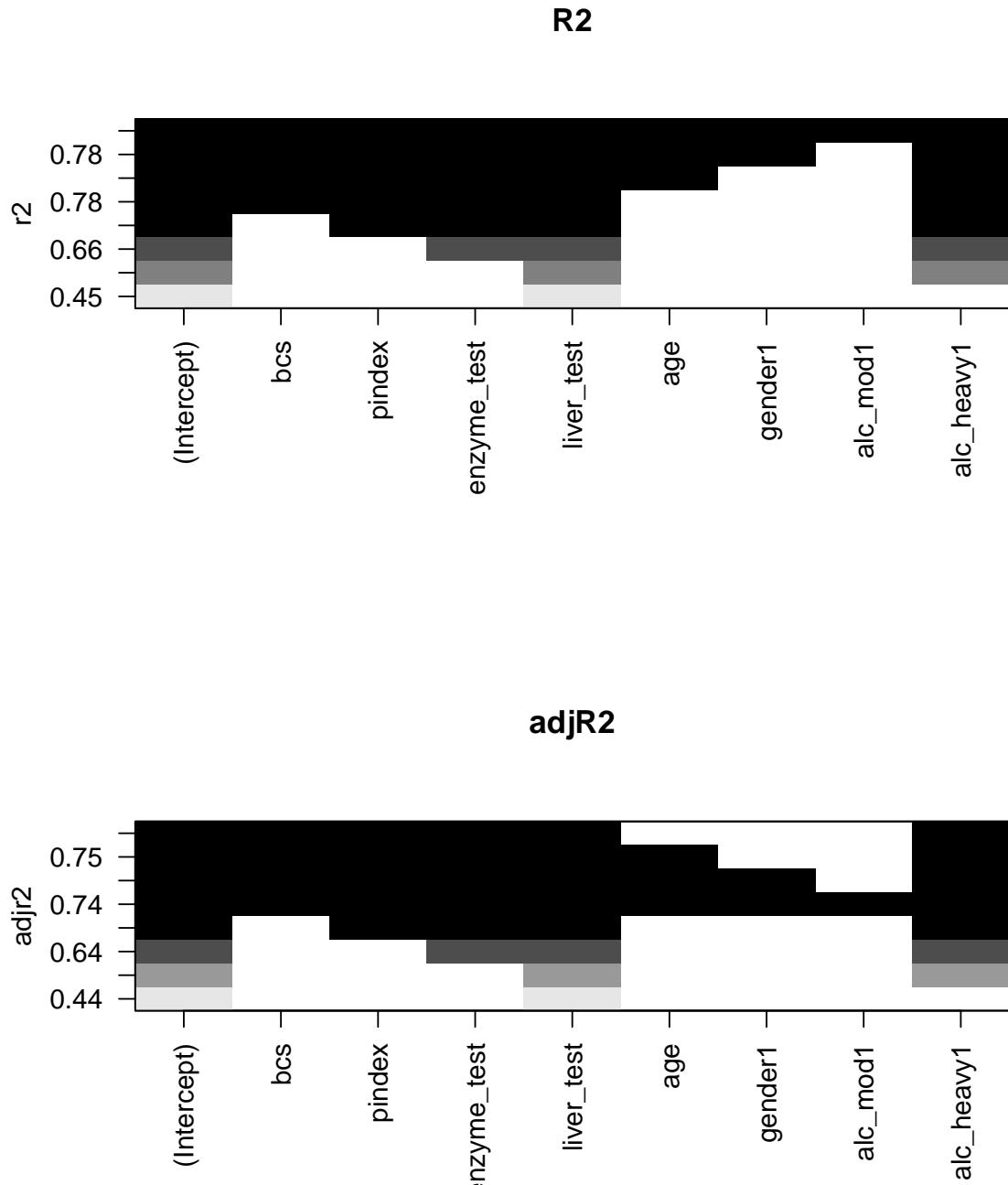


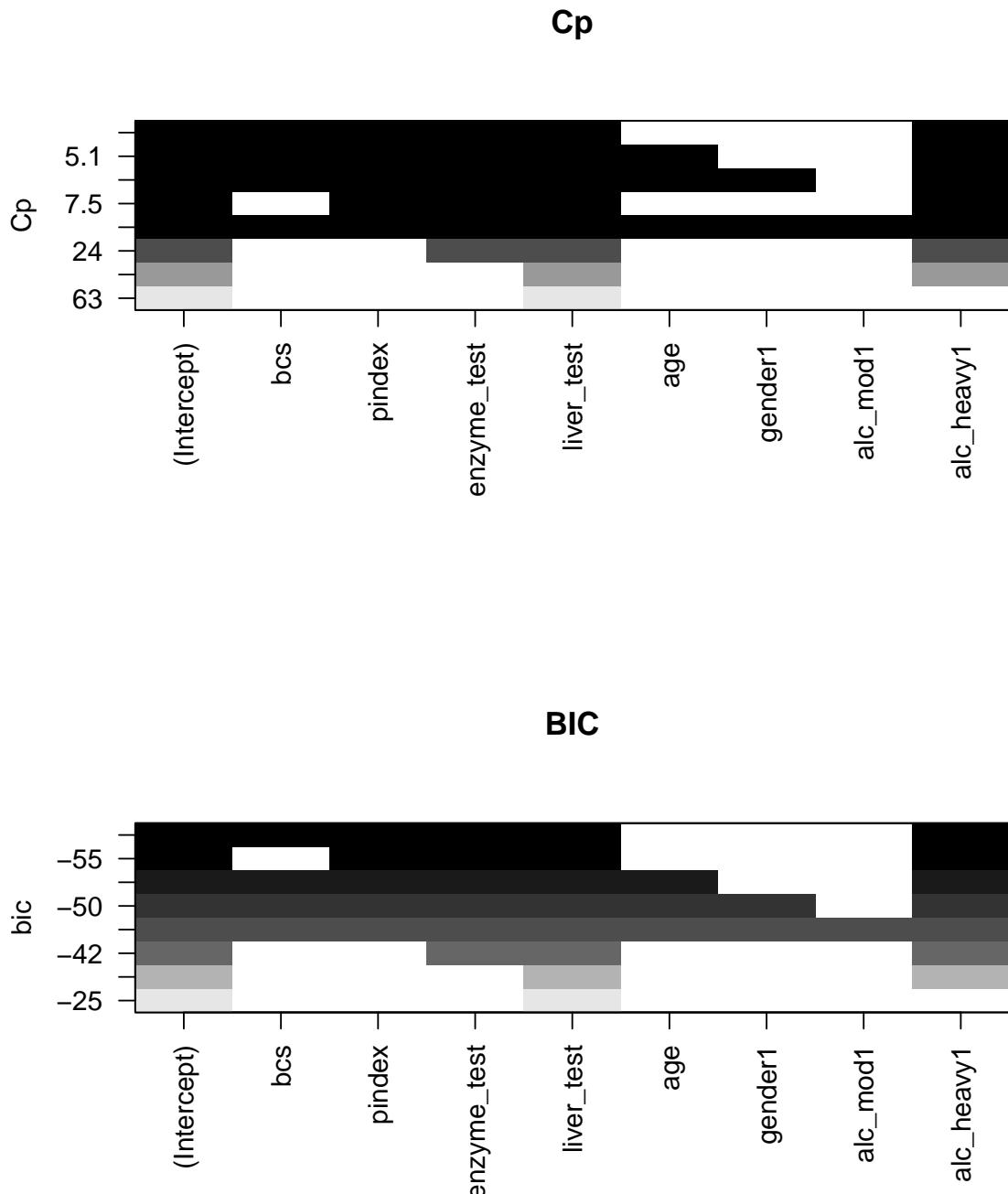


Se puede observar como, usando el método de **forward** se proponen las siguientes cantidades a tomar según los 4 estimadores estadísticos:

- **RSS**: propone 8 parámetros.
- **R2adj**: propone 5 parámetros.
- **Cp**: propone 5 parámetros.
- **bIC**: propone 5 parámetros.

Ahora, se presentan también estos 4 graficos que nos ayudarán a dilucidar los parámetros que mejor ayudan a explicar la variable **y**





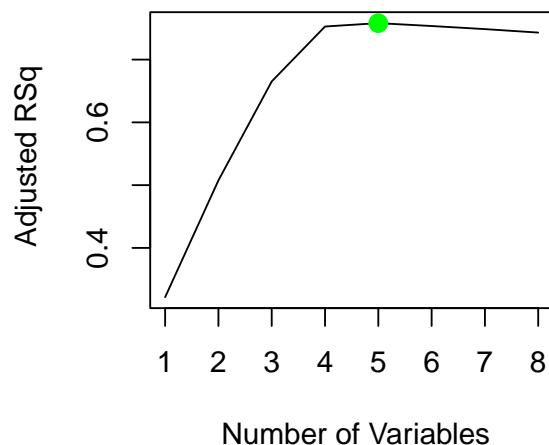
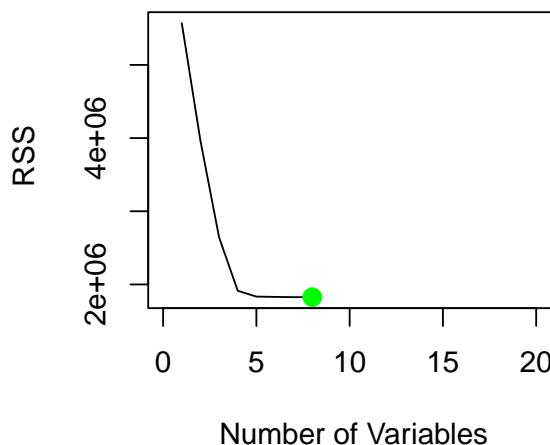
Se observa como, usando estas 4 medidas y el método **forward**, las variables que mejor ayudan a explicar la variable respuesta **y** son **bcs**, **pindex**, **enzyme\_test**, **liver\_test** y **alc\_heavy1**.

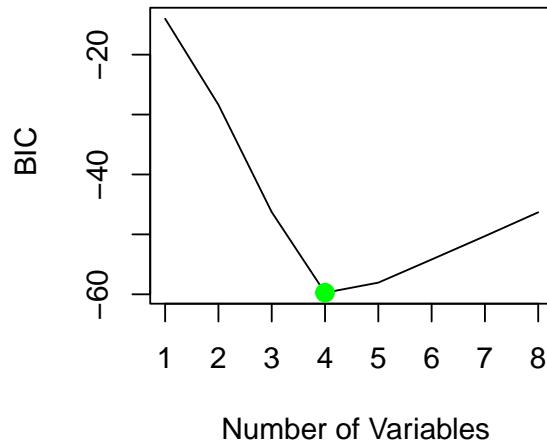
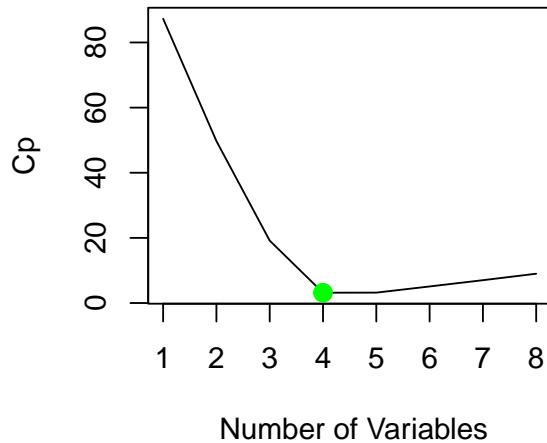
### 4.3. Backward

Primero se procede a seleccionar el “mejor” subconjunto para estos datos:

```
##          rsq      adjr2       cp      rss      bic bcs pindex enzyme_test
## 1  ( 1 ) 0.3343453 0.3215443 87.304176 5571211 -13.99918      *
## 2  ( 1 ) 0.5261922 0.5076115 49.732038 3965544 -28.36854      *
## 3  ( 1 ) 0.6841275 0.6651752 19.154821 2643701 -46.27456      *      *
## 4  ( 1 ) 0.7713565 0.7526917  3.162146 1913636 -59.73701      *      *
## 5  ( 1 ) 0.7809054 0.7580831  3.192498 1833716 -58.05170      *      *
## 6  ( 1 ) 0.7814169 0.7535127  5.086996 1829436 -54.18892      *      *
## 7  ( 1 ) 0.7817703 0.7485615  7.014100 1826478 -50.28731      *      *
## 8  ( 1 ) 0.7818387 0.7430544 9.000000 1825906 -46.31525      *      *
##          liver_test age gender1 alc_mod1 alc_heavy1
## 1  ( 1 )
## 2  ( 1 )
## 3  ( 1 )
## 4  ( 1 )                  *
## 5  ( 1 )                *
## 6  ( 1 )                *   *
## 7  ( 1 )                *   *
## 8  ( 1 )                *   *   *
```

Luego, se presentan los gráficos con las 4 medidas de referencia que sugieren el número de parámetros a tomar.

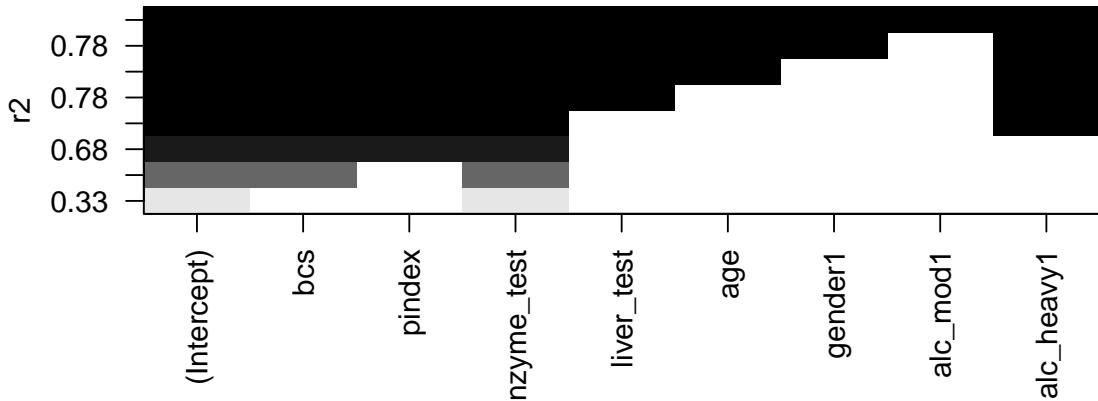
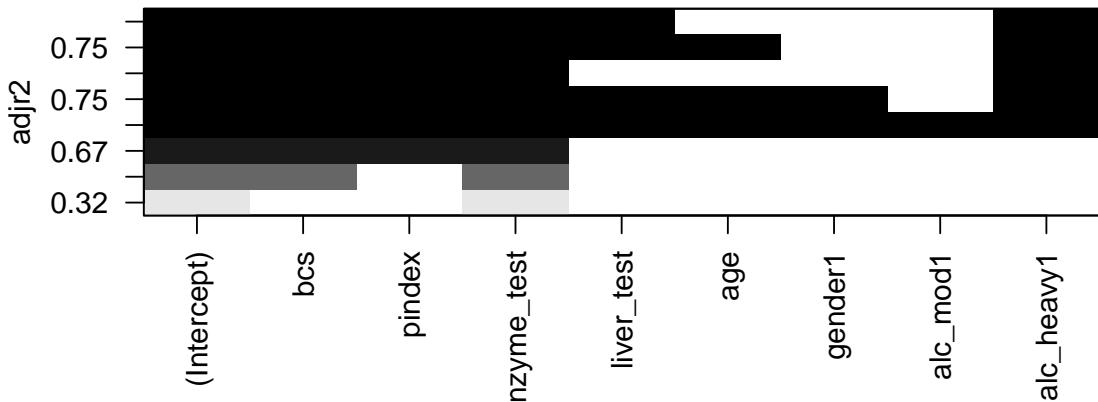


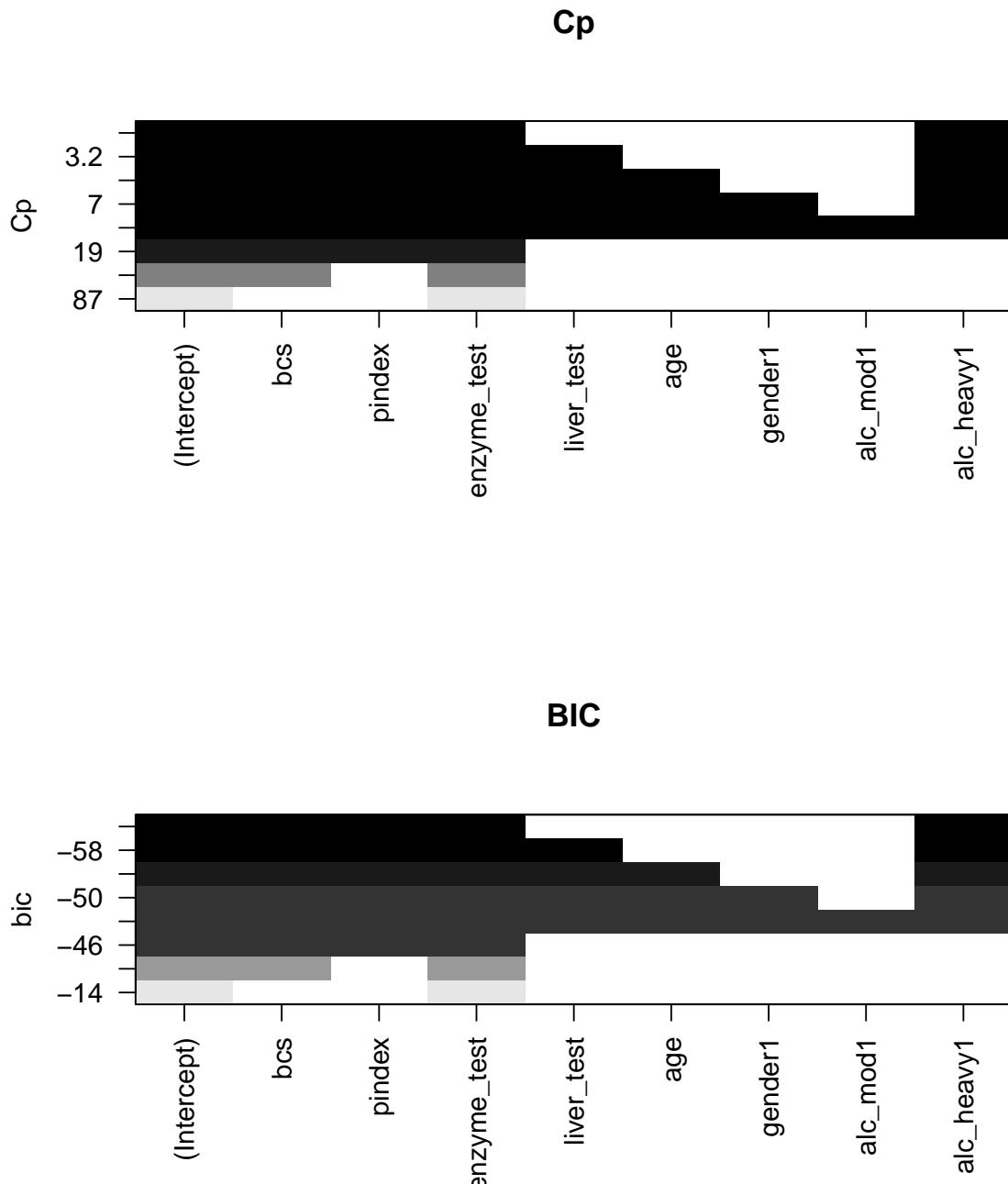


Se puede observar como, usando el método de **backward** se proponen las siguientes cantidades a tomar según los 4 estimadores estadísticos:

- **RSS**: propone 8 parámetros.
- **R2adj**: propone 5 parámetros.
- **Cp**: propone 4 parámetros.
- **bIC**: propone 4 parámetros.

Ahora, se presentan también estos 4 graficos que nos ayudarán a dilucidar los parámetros que mejor ayudan a explicar la variable **y**

**R2****adjR2**



Se observa como, usando estas 4 medidas y el método **backward**, las variables que mejor ayudan a explicar la variable respuesta y son **bcs**, **pindex**, **enzyme\_test** y **alc\_heavy**.

#### 4.4. Stepwise

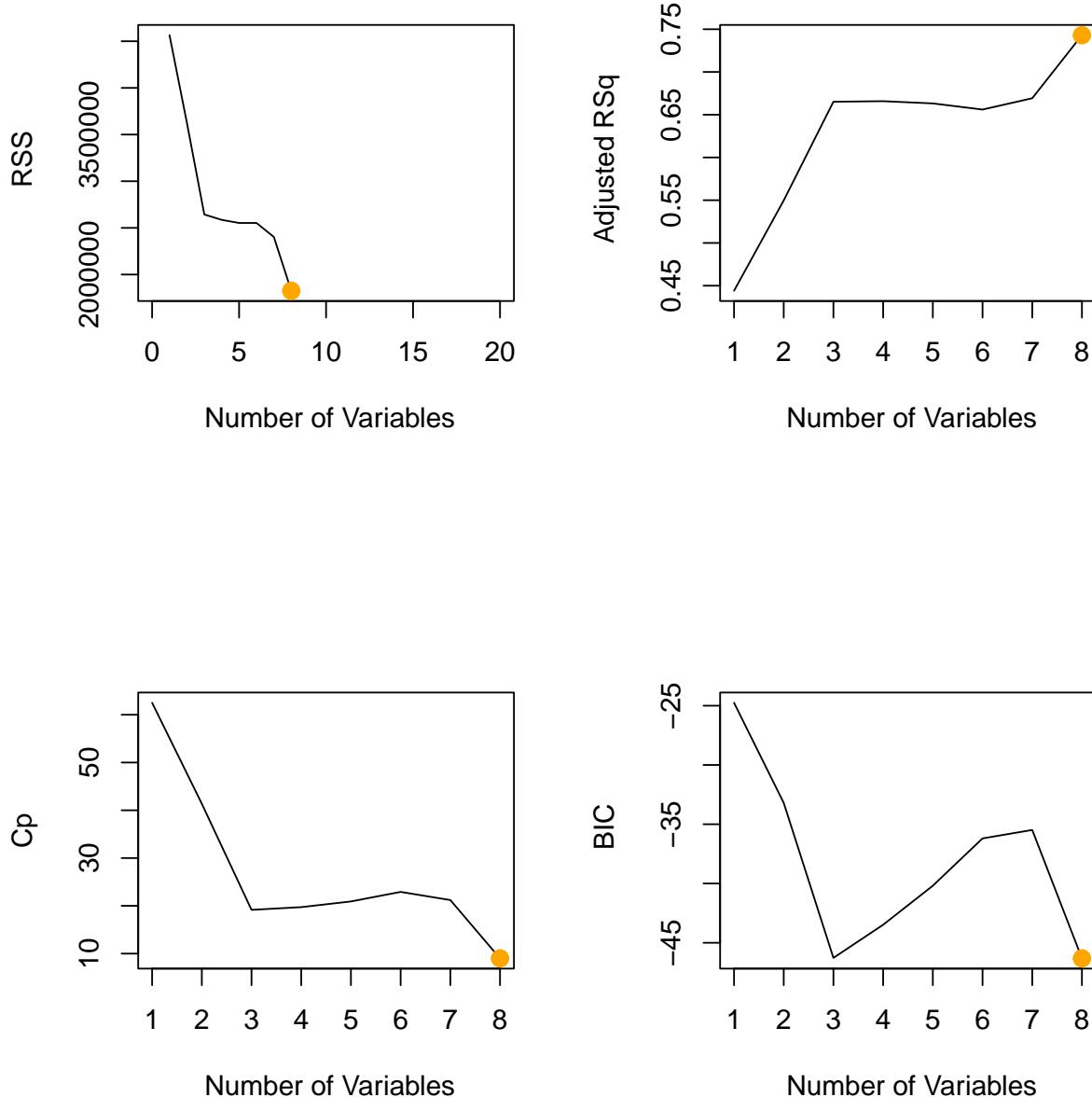
Primero se procede a seleccionar el “mejor” subconjunto para estos datos:

```

## Subset selection object
## Call: regsubsets.formula(y ~ ., data = surgical, nvmax = NULL, method = "seqrep")
## 8 Variables (and intercept)
##          Forced in Forced out
## bcs          FALSE      FALSE
## pindex        FALSE      FALSE
## enzyme_test   FALSE      FALSE
## liver_test    FALSE      FALSE
## age           FALSE      FALSE
## gender1       FALSE      FALSE
## alc_mod1      FALSE      FALSE
## alc_heavy1    FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: 'sequential replacement'
##          bcs pindex enzyme_test liver_test age gender1 alc_mod1 alc_heavy1
## 1 ( 1 ) " " " " " " "*" " " " " " " " "
## 2 ( 1 ) " " " " " " "*" " " " " " " " *
## 3 ( 1 ) "*" "*" "*" " " " " " " " " " "
## 4 ( 1 ) "*" "*" "*" "*" " " " " " " " "
## 5 ( 1 ) "*" "*" "*" "*" " " " " " " " "
## 6 ( 1 ) "*" "*" "*" "*" " " " " " " " "
## 7 ( 1 ) "*" "*" "*" "*" " " " " " " " "
## 8 ( 1 ) "*" "*" "*" "*" " " " " " " " "
##          rsq     adjr2      cp      rss      bic bcs pindex enzyme_test
## 1 ( 1 ) 0.4545389 0.4440492 62.51192 4565248 -24.75271
## 2 ( 1 ) 0.5667409 0.5497504 41.36808 3626171 -33.19970
## 3 ( 1 ) 0.6841275 0.6651752 19.15482 2643701 -46.27456   *   *   *
## 4 ( 1 ) 0.6910409 0.6658198 19.72879 2585839 -43.48059   *   *   *
## 5 ( 1 ) 0.6949877 0.6632155 20.91470 2552807 -40.18586   *   *   *
## 6 ( 1 ) 0.6949877 0.6560500 22.91469 2552807 -36.19689   *   *   *
## 7 ( 1 ) 0.7129415 0.6692587 21.21137 2402542 -35.48387   *   *   *
## 8 ( 1 ) 0.7818387 0.7430544  9.00000 1825906 -46.31525   *   *   *
##          liver_test age gender1 alc_mod1 alc_heavy1
## 1 ( 1 )          *
## 2 ( 1 )          *          *
## 3 ( 1 )
## 4 ( 1 )          *
## 5 ( 1 )          *   *
## 6 ( 1 )          *   *   *
## 7 ( 1 )          *   *   *   *
## 8 ( 1 )          *   *   *   *   *

```

Luego, se presentan los gráficos con las 4 medidas de referencia que sugieren el número de parámetros a tomar.

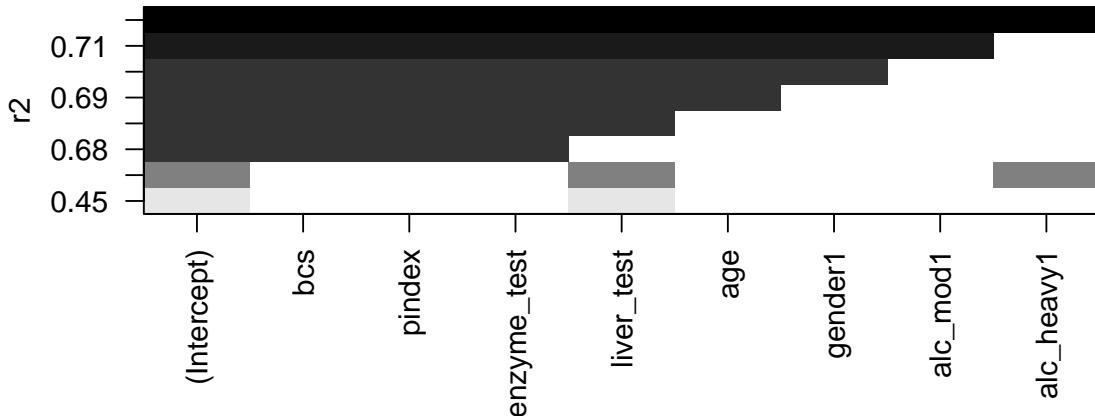


Se puede observar como, usando el método de **stepwise** se proponen las siguientes cantidades a tomar según los 4 estimadores estadísticos:

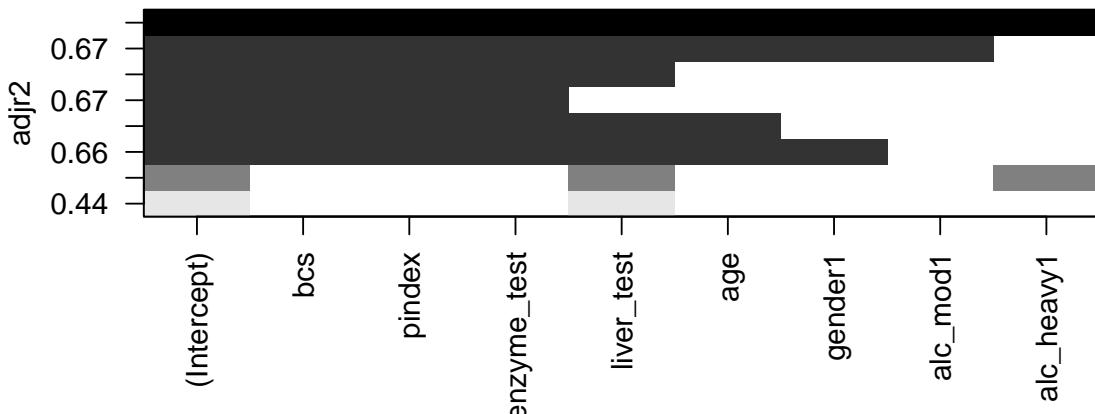
- **RSS:** propone 8 paraméetros.
- **R<sup>2</sup>adj:** propone 8 paraméetros.
- **C<sub>p</sub>:** propone 8 paraméetros.
- **bIC:** propone 8 paraméetros.

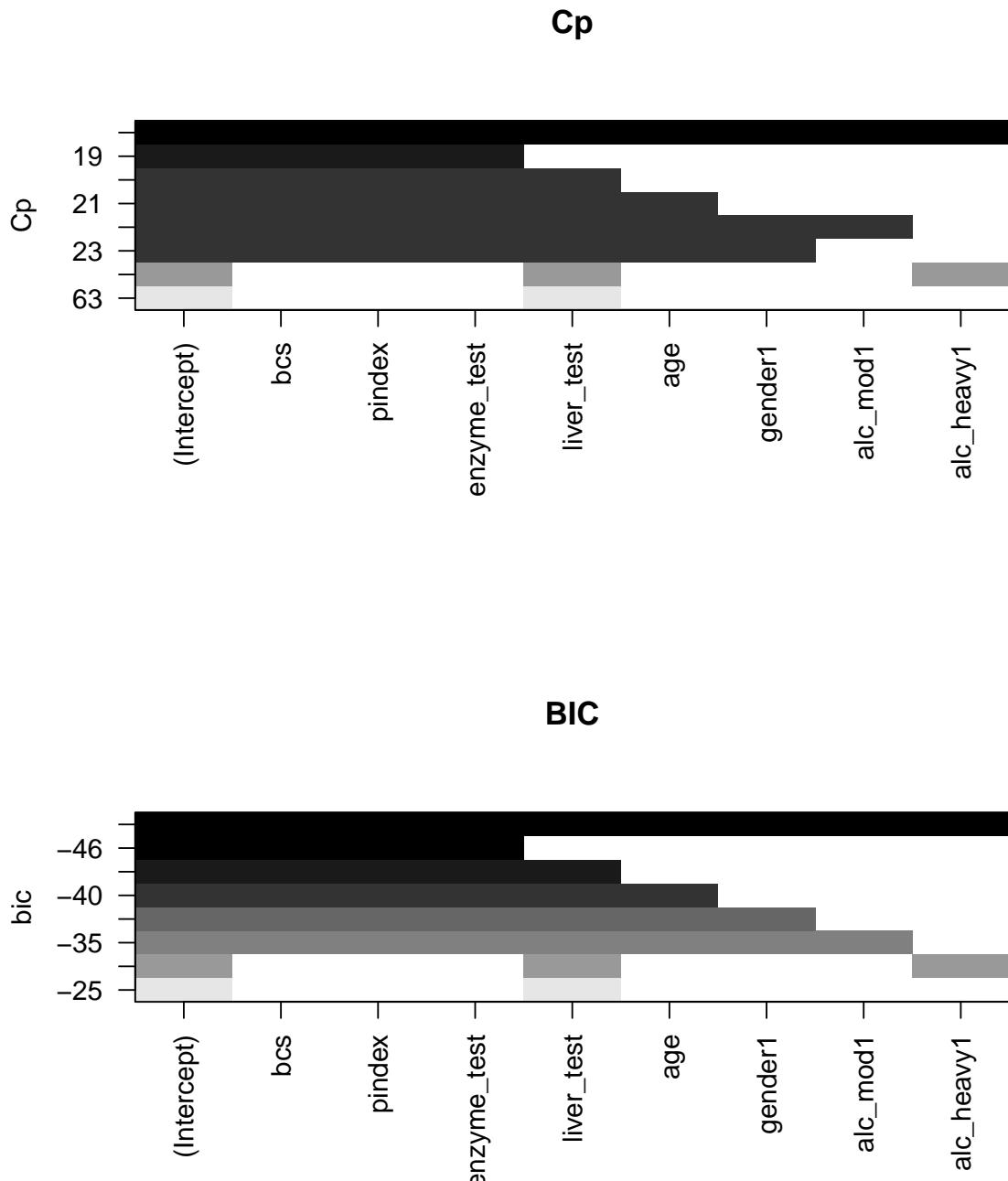
Ahora, se presentan también estos 4 graficos que nos ayudarán a dilucidar los parámetros que mejor ayudan a explicar la variable y.

R2



adjR2





Se observa como, usando estas 4 medidas y el método **stepwise**, considera que todas las variables son buenas para explicar a la variable y.

#### 4.5. Validación cruzada (CV)

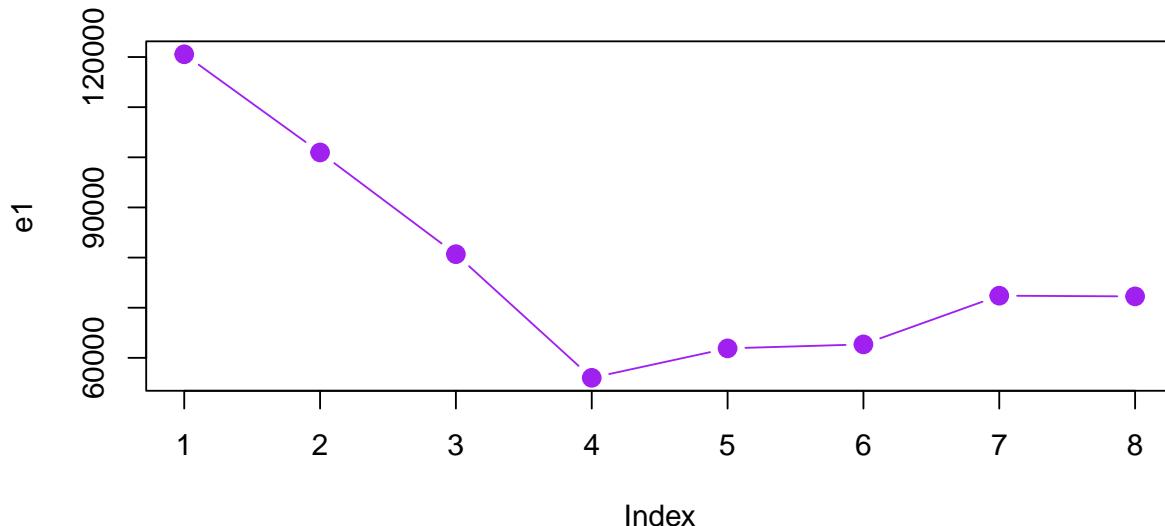
Primero se procede a preparar los datos:

Luego se realizan las predicciones.

```
## [1] 120550.73 100968.36 80676.62 56014.11 61890.38 62686.38 72396.52
## [8] 72258.60

## [1] 4

## (Intercept)          bcs         pindex      enzyme_test      liver_test
## -1148.8230115    62.3903579   8.9731346    9.8880787   50.4127395
##           age       gender1      alc_mod1      alc_heavy1
## -0.9510052    15.8742926   7.7131259   320.6969061
```



Luego el gráfico nos muestra que, usando **CV**, un modelo que ayuda explicar de mejor manera la variable respuesta **y** es un que cuenta con 4 parámetros (sin contar el intercepto), y corresponde con las variables **bcs**, **pindex**, **enzyme\_test** y **alc\_heavy**.