TRABAJO REGRESIÓN
GRUPO 13

INTEGRANTES:

-César de Diego Morales

-Ignacio Fernández Sánchez-Pascuala

-Javier Castellano Soria

-Lucas Ruiz Bacete

Índice

1.Introducción

2.Inspección de las variables

2.1. Tipo de variables

2.2. Relación entre variables

2.3. Histograma

3.Construcción de modelos (con las interacciones de la variable cualitativa)

3.1. Procedimientos basados en pruebas

3.2. Procedimientos basados en criterios

3.3. Modelos finales a considerar

4.Diagnóstico de modelos (con las interacciones de la variable cualitativa)

4.1. Diagnóstico inicial

4.2. Outliers

4.3. Estudio de las hipótesis de regresión lineal múltiple

5.Construcción de modelos (sin las interacciones de la variable cualitativa)

5.1. Procedimientos basados en pruebas

5.2. Procedimientos basados en criterios

5.3. Modelos finales a considerar

6.Diagnóstico de modelos (sin las interacciones de la variable cualitativa)

6.1. Diagnóstico inicial

6.2. Outliers

6.3. Estudio de las hipótesis de regresión lineal múltiple

7.Elección del modelo

8.Conclusión

9.Anexo

1.Introducción

En este trabajo, vamos a tratar de construir el mejor modelo de regresión lineal múltiple que mejor prediga el valor de la variable respuesta.

Estudiaremos los datos de una serie de individuos con cáncer de próstata que van a recibir una prostatectomía. Estos se encuentran en el data-frame *Prostate* de la librería *faraway* de *R.* En él se recoge la información de las siguientes variables:

1. **lcavol:** volumen del cáncer (aplicado el logaritmo)
2. **lweight:** peso de la próstata (aplicado el logaritmo)
3. **age:** edad del paciente
4. **lbph:** cantidad de hiperplasia prostática benigna, es decir, agrandamiento de la glándula prostática (aplicado el logaritmo)
5. **svi:** invasión de la vesícula seminal (0 si no está invadida 1 caso contrario)
6. **lcp:** penetración capsular (aplicado el logaritmo). Mide la superficie que ha alcanzado el cáncer de la pared exterior de la próstata
7. **gleasson:** puntaje de Gleason (toma valores entre 2 y 10). Es la suma de los grados más comunes en el tejido canceroso (cada grado se puntúa del 1 al 5, y cuanto más alto sea el valor, más peligrosas serán las células cancerígenas)
8. **pgg45:** porcentaje de tejido que tiene grado de Gleason 4 ó 5
9. **lpsa:** indica la concentración de PSA en la sangre (aplicado el logaritmo). La PSA es una proteína producida por células sanas y células malignas de la glándula prostática (La concentración de PSA en sangre es frecuente en hombres con cáncer de próstata)

En nuestro caso, la variable respuesta será **lpsa.**

Vemos que nos ofrecen datos de 97 individuos. Podemos ver los 6 primeros datos.

```
> head(prostate)
      lcavol lweight age      lbph svi      lcp gleason pgg45     lpsa
1 -0.5798185  2.7695  50 -1.386294   0 -1.38629       6     0 -0.43078
2 -0.9942523  3.3196  58 -1.386294   0 -1.38629       6     0 -0.16252
3 -0.5108256  2.6912  74 -1.386294   0 -1.38629       7    20 -0.16252
4 -1.2039728  3.2828  58 -1.386294   0 -1.38629       6     0 -0.16252
5  0.7514161  3.4324  62 -1.386294   0 -1.38629       6     0  0.37156
6 -1.0498221  3.2288  50 -1.386294   0 -1.38629       6     0  0.76547
```

Alguna información útil de nuestras variables como la media, el valor máximo, el valor mínimo o la mediana se recoge en la siguiente imagen.

```
> summary(prostate) #Nos da informacion util sobre cada variable de primeras
     lcavol           lweight           age              lbph              svi              lcp        
 Min.   :-1.3471   Min.   :2.375   Min.   :41.00   Min.   :-1.3863   Min.   :0.0000   Min.   :-1.3863  
 1st Qu.: 0.5128   1st Qu.:3.376   1st Qu.:60.00   1st Qu.:-1.3863   1st Qu.:0.0000   1st Qu.:-1.3863  
 Median : 1.4469   Median :3.623   Median :65.00   Median : 0.3001   Median :0.0000   Median :-0.7985  
 Mean   : 1.3500   Mean   :3.653   Mean   :63.87   Mean   : 0.1004   Mean   :0.2165   Mean   :-0.1794  
 3rd Qu.: 2.1270   3rd Qu.:3.878   3rd Qu.:68.00   3rd Qu.: 1.5581   3rd Qu.:0.0000   3rd Qu.: 1.1786  
 Max.   : 3.8210   Max.   :6.108   Max.   :79.00   Max.   : 2.3263   Max.   :1.0000   Max.   : 2.9042  
    gleason          pgg45            lpsa        
 Min.   :6.000   Min.   :  0.00   Min.   :-0.4308  
 1st Qu.:6.000   1st Qu.:  0.00   1st Qu.: 1.7317  
 Median :7.000   Median : 15.00   Median : 2.5915  
 Mean   :6.753   Mean   : 24.38   Mean   : 2.4784  
 3rd Qu.:7.000   3rd Qu.: 40.00   3rd Qu.: 3.0564  
 Max.   :9.000   Max.   :100.00   Max.   : 5.5829
```

2.Inspección de las variables

2.1. Tipo de variables

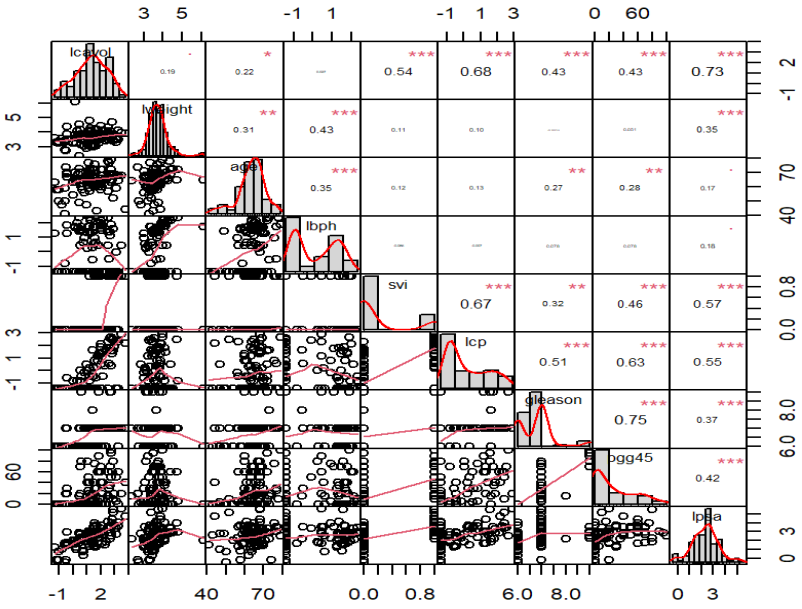
Veamos ahora de qué tipo son nuestras variables.

```
> str(prostate)
'data.frame':   97 obs. of  9 variables:
 $ lcavol : num  -0.58 -0.994 -0.511 -1.204 0.751 ...
 $ lweight: num  2.77 3.32 2.69 3.28 3.43 ...
 $ age    : int  50 58 74 58 62 50 64 58 47 63 ...
 $ lbph   : num  -1.39 -1.39 -1.39 -1.39 -1.39 ...
 $ svi    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lcp    : num  -1.39 -1.39 -1.39 -1.39 -1.39 ...
 $ gleason: int  6 6 7 6 6 6 6 6 6 6 ...
 $ pgg45  : int  0 0 20 0 0 0 0 0 0 0 ...
 $ lpsa   : num  -0.431 -0.163 -0.163 -0.163 0.372 ...
>
```

Observamos que las variables **lcavol, lweight, lbph, lcp** y **lpsa** son variables cuantitativas continuas. Por otro lado, las variables **age, gleason** y **pgg45** son variables cuantitativas discretas. Por último, **svi** es una variable categórica con dos categorías explicadas en la introducción.

2.2. Relación entre variables

Un primer paso para estudiar la colinealidad entre nuestras covariables es analizar las correlaciones. Para ello nos ayudamos también de un scatter-plot.

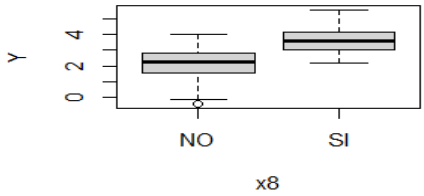


Con esta información concluimos que:

-La variable **lcavol** tiene una alta correlación con **lpsa** por lo que será interesante considerarla en nuestro modelo.

-Las variables **pgg45** y **gleason** (0.7519); **pgg45** y **lcp** (0.6315); **lcavol** y lcp (0.6753) están correlacionadas. Hay que tener cuidado con estas variables ya que podrían dar problemas de colinealidad si aparecen ambas en el modelo.

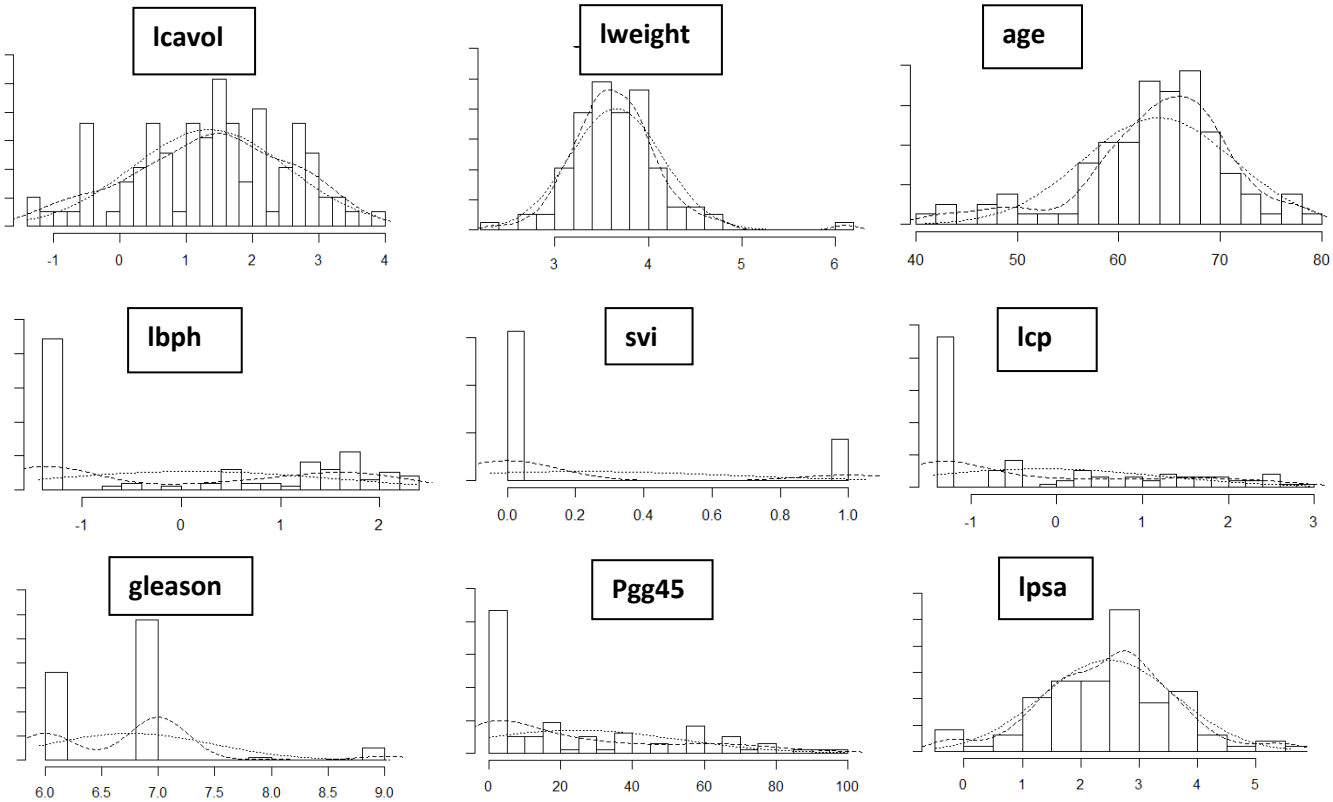
Por último, vamos a ver la relevancia de incluir la variable cualitativa (**svi**) en nuestro modelo:



Observamos que si no hay invasión seminal por lo general se tiene un menor valor de **lpsa**. Por lo tanto, será interesante incluirla en nuestro estudio.

2.3. Histograma

Realizamos un histograma que nos permitirá tener una primera vista de cómo se distribuyen las variables.



3.Construcción de modelos (con las interacciones de la variable cualitativa)

Antes de todo, construimos el modelo completo con la cualitativa y sus respectivas interacciones: **lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi + lcavol:svi + lweight:svi + age:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi.** Para obtener una información inicial hacemos el summary y el ANOVA de este modelo.

SUMMARY

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 0.6981 on 81 degrees of freedom
Multiple R-squared: 0.6914,
Adjusted R-squared: 0.6342
F-statistic:  12.1 on 15 and 81 DF,  p-value: 5.439e-15

Residuals:
     Min       1Q   Median       3Q      Max
-1.44793 -0.34778 -0.03352  0.35349  1.67838

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.109987   1.421834  -0.077   0.9385
lcavol       0.529821   0.092733   5.713 1.78e-07 ***
lweight      0.461621   0.180929   2.551   0.0126 *
age         -0.017300   0.012839  -1.347   0.1816
lbph         0.146500   0.065935   2.222   0.0291 *
lcp         -0.104578   0.104435  -1.001   0.3196
gleason      0.141251   0.179847   0.785   0.4345
pgg45        0.005667   0.005148   1.101   0.2742
svi          6.902929   3.694815   1.868   0.0653 .
lcavol:svi   0.217337   0.315474   0.689   0.4928
lweight:svi -0.607264   0.552415  -1.099   0.2749
age:svi     -0.002163   0.026423  -0.082   0.9350
lbph:svi    -0.153765   0.144979  -1.061   0.2920
lcp:svi     -0.098772   0.222261  -0.444   0.6579
gleason:svi -0.566293   0.380819  -1.487   0.1409
pgg45:svi   -0.001184   0.009855  -0.120   0.9047
```

ANOVA

```
Analysis of Variance Table

Response: lpsa
            Df Sum Sq Mean Sq  F value    Pr(>F)
lcavol       1 69.003  69.003 141.5718 < 2.2e-16 ***
lweight      1  5.949   5.949  12.2044 0.0007755 ***
age          1  0.420   0.420   0.8617 0.3560304
lbph         1  1.069   1.069   2.1933 0.1424892
lcp          1  0.833   0.833   1.7091 0.1947969
gleason      1  0.516   0.516   1.0589 0.3065334
pgg45        1  1.030   1.030   2.1123 0.1499815
svi          1  4.936   4.936  10.1261 0.0020735 **
lcavol:svi   1  0.130   0.130   0.2665 0.6071209
lweight:svi  1  0.779   0.779   1.5984 0.2097535
age:svi      1  0.370   0.370   0.7583 0.3864299
lbph:svi     1  1.205   1.205   2.4723 0.1197630
lcp:svi      1  0.516   0.516   1.0586 0.3065889
gleason:svi  1  1.677   1.677   3.4399 0.0672803 .
pgg45:svi    1  0.007   0.007   0.0144 0.9046930
Residuals   81 39.480   0.487
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Debido a que algunas covariables tienen p-valores muy elevados, deberemos construir modelos que se ajusten mejor a las hipótesis de regresión lineal múltiple.

3.1. Procedimientos basados en pruebas.

Utilizaremos el método Backward aplicado al modelo completo. Como el modelo que obtengamos mediante dicho proceso puede depender de los datos que tomemos, hemos decidido dividir de forma aleatoria el conjunto de datos dado entre train (70%) y test (30%) y realizar el proceso partiendo del modelo completo con los datos de la parte train. Para tres particiones de los datos distintas hemos obtenido los siguientes modelos con grado de significación 0.20 :

->Train 1: Después de realizar nuestras iteraciones llegamos al siguiente modelo: **lpsa ~ lcavol + lweight + lbph + gleason + svi + lweight:svi + lbph:svi + lcp:svi + gleason: svi + pgg45:svi**

->Train 2: Después de realizar nuestras iteraciones llegamos al siguiente modelo: **lpsa ~ lcavol + lbph + pgg45 + svi + lbph:svi + pgg45:svi**

->Train 3: Después de realizar nuestras iteraciones llegamos al siguiente modelo: **lpsa ~ lcavol + lweight + age + lbph + lcavol:svi + lweight:svi + age:svi + lcp:svi + pgg45:svi**

Con el objetivo de obtener un modelo con el método Backward, decidimos quedarnos con aquel que tiene el menor error de predicción en su respectiva parte de test. Por tanto, nos quedamos con el modelo siguiente:

lpsa ~ lcavol + lweight + lbph + gleason + svi + lweight:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi.

3.2. Procedimientos basados en criterios.

3.2.1. Criterio Cp de Mallow: Siguiendo la misma idea de antes, volvemos a tomar tres particiones distintas del conjunto de datos . Ahora por cada partición, de los modelos que nos da la función *regsubsets()* de R (con train como datos de entrada) cogemos aquel que tenga el mejor Cp. Obtenemos así, los siguientes modelos:

->Train 1: El modelo obtenido es: **lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi.**

->Train 2: El modelo obtenido es: **lpsa ~ lcavol + lweight + pgg45 + svi + lweight:svi + age:svi + lcp:svi.**

->Train 3: El modelo obtenido es: **lpsa ~ lcavol + age + lbph + lcp + pgg45 + svi + lbph:svi + gleason:svi.**

Con el objetivo de obtener un modelo mediante el criterio Cp, decidimos quedarnos con aquel que tiene el menor error de predicción en su respectiva parte de test. Por tanto, nos quedamos con el siguiente modelo:

lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi

3.2.2. Método de entrenamiento y test con regsubset: Ahora train constará de todos los datos salvo uno que será la parte de test. Vamos a considerar las **97 particiones distintas** que se pueden hacer de esta manera y con *regsubsets()* obtendremos para cada train un total de 8 modelos en función del número de variables que lo conformen. Finalmente, nos quedaremos con aquel modelo que menor error medio tenga (es exactamente lo mismo que lo hecho en un ejercicio de clase). El modelo obtenido es el siguiente:

lpsa ~ lcavol + lweight + svi

Observación: Este modelo es el mismo que se obtiene con el criterio BIC teniendo en cuenta todos los datos de la muestra.

3.3. Modelos finales a considerar

->Modelo 1: Modelo obtenido por el método Backward:
lpsa ~ lcavol + lweight + lbph + gleason + svi + lweight:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi.

->Modelo 2: Modelo obtenido por el criterio Cp de Mallow:
lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi.

->Modelo 3: Modelo obtenido por el método **de entrenamiento y test con regsubset:**
lpsa ~ lcavol + lweight + svi.

4.Diagnóstico de modelos (con las interacciones de la variable cualitativa)

4.1. Diagnóstico inicial.

Realizamos el contraste de hipótesis para ver si podemos suponer la eliminación de las variables excluidas en cada modelo. Para cada uno obtenemos los siguientes p-valores:

->Modelo 1: p-valor = 0.4375

->Modelo 2: p-valor = 0.4027

->Modelo 3: p-valor = 0.174

Podemos suponer la eliminación de las correspondientes covariables con hasta un grado de significación del 0.15 .

Ahora realizaremos un *summary* de cada uno de ellos para un primer diagnóstico.

->Modelo 1: El mayor p-valor es 0.9 y se alcanza en la variable **pgg45:svi.** Al tener algunos p-valores elevados y un error estándar en el coeficiente de la variable **svi** muy superior al resto, nos hace pensar que no será un buen modelo, a priori.

```
Call:
lm(formula = lpsa ~ lcavol + lweight + lbph + gleason + svi +
    lweight:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi,
    data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.47439 -0.40361 -0.07559  0.44844  1.55011

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.1739121  1.0739332  -1.093   0.2774
lcavol       0.5028703  0.0802106   6.269  1.4e-08 ***
lweight      0.4268739  0.1779409   2.399   0.0186 *
lbph         0.1233934  0.0635529   1.942   0.0555 .
gleason      0.1856328  0.1237881   1.500   0.1374
svi          7.4951365  3.3556276   2.234   0.0281 *
lweight:svi -0.6387341  0.5296102  -1.206   0.2311
lbph:svi    -0.1614863  0.1407136  -1.148   0.2543
svi:lcp     -0.0890016  0.1638964  -0.543   0.5885
gleason:svi -0.6034774  0.3574198  -1.688   0.0950 .
svi:pgg45    0.0009888  0.0079881   0.124   0.9018

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 0.6977 on 86 degrees of freedom
Multiple R-squared: 0.6728,
Adjusted R-squared: 0.6347
F-statistic: 17.68 on 10 and 86 DF,  p-value: < 2.2e-16
```

->Modelo 2: El mayor p-valor es 0.13 y se alcanza en la variable **lcp:svi.** Errores estándar y p-valores inferiores al anterior.

```
Call:
lm(formula = lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi +
    lcp:svi, data = prostate)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5140 -0.4355 -0.0334  0.4078  1.5806

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.36110    0.60191   0.600  0.55007
lcavol       0.52879    0.07613   6.946  5.7e-10 ***
lweight      0.33679    0.16695   2.017  0.04665 *
lbph         0.14312    0.06267   2.284  0.02474 *
lcavol:svi   0.44284    0.13881   3.190  0.00196 **
lbph:svi    -0.22106    0.13098  -1.688  0.09493 .
svi:lcp     -0.26606    0.17596  -1.512  0.13402
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 0.7002 on 90 degrees of freedom
Multiple R-squared: 0.6551,
Adjusted R-squared: 0.6321
F-statistic: 28.49 on 6 and 90 DF,  p-value: < 2.2e-16
```

->Modelo 3: Los p-valores en este modelo son muy pequeños lo que indica que todas las covariables son significativas.

```
Call:
lm(formula = lpsa ~ lcavol + lweight + svi, data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.72964 -0.45764  0.02812  0.46403  1.57013

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.26809    0.54350  -0.493  0.62298
lcavol       0.55164    0.07467   7.388  6.3e-11 ***
lweight      0.50854    0.15017   3.386  0.00104 **
svi          0.66616    0.20978   3.176  0.00203 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 0.7168 on 93 degrees of freedom
Multiple R-squared: 0.6264,
Adjusted R-squared: 0.6144
F-statistic: 51.99 on 3 and 93 DF,  p-value: < 2.2e-16
```

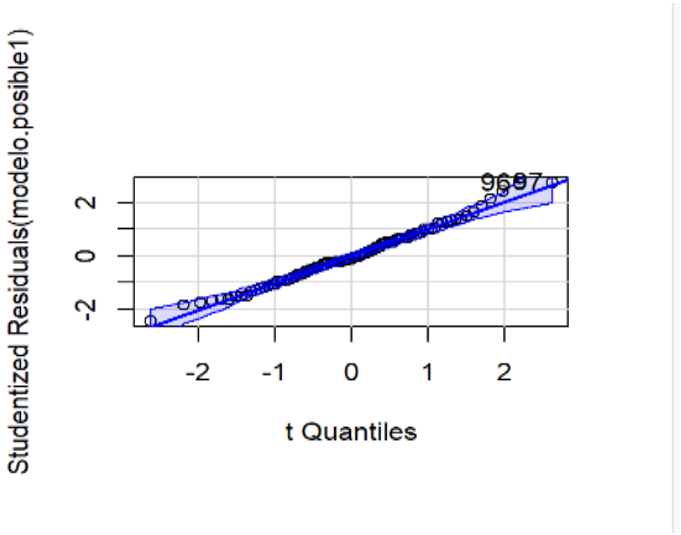
4.2. Outliers.

4.2.1. Outliers del modelo 1.

Outliers respecto a Y: Comenzamos viendo outliers respecto a Y para comparar si al quitarlos se mejoran las hipótesis que debe cumplir nuestro modelo de regresión lineal múltiple. Empezamos tomando las tres observaciones con mayor residuo estandarizado en valor absoluto. Obtenemos así las observaciones: 97 (2.652), 96 (2.634) y 39 (2.361).

Aplicando el método de Bonferroni con $\alpha = 0.20$, se obtiene que ninguna de las observaciones puede considerarse outlier.

Por otro lado, el *qqplot* de los residuos estudentizados nos da dos observaciones inusuales a considerar: 96 y 97.



Outliers respecto las covariables: Tomaremos varias observaciones con alto leverage. Estudiando la diagonal de nuestra matriz *hat* H, obtenemos:

```
> which(hii>hcv)#47,74,89,92 las mayores
32 47 64 73 74 76 79 89 90 92 95 96 97
32 47 64 73 74 76 79 89 90 92 95 96 97
>
```

Consideraremos las observaciones 47, 74, 89 y 92; pues son las mayores de las que superan el valor crítico, aunque **puede que no todas sean influyentes.**

Distancia de Cook: Los valores con mayor distancia de Cook son 97 y 96. Sin embargo, ninguna de nuestras observaciones será influyente pues Di < $F_{o.5;p,n-p}$ con p=11 y n= 97.

DFFITSi: Realizando ahora la prueba de DFFITSi, tomaremos otras cinco observaciones 97, 96, 95, 39 y 32 (las cinco mayores de todas las que superan el valor crítico).

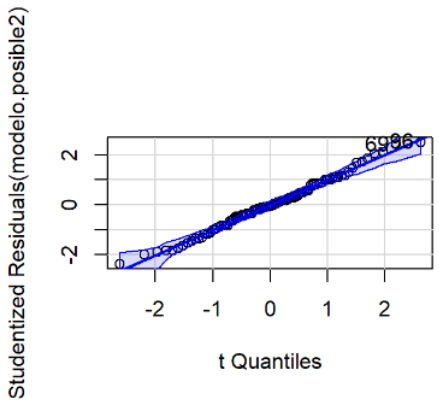
DFBETAS: Consideraremos las observaciones 32, 69, 96, 47, 95 y 87. De cada covariable que tenga asociada varias observaciones influyentes hemos considerado las que mayor *DFBETAS* tiene.

Observaciones a eliminar: Tras varias pruebas decidimos eliminar únicamente la observación 97 del modelo 1. Si eliminábamos otro grupo de observaciones empeoraban las hipótesis.

4.2.2. Outliers del modelo 2.

Outliers respecto a Y: Empezamos tomando las tres observaciones con mayor residuo estandarizado en valor absoluto: 96(2.445), 69 (2.362) y 39 (2.305). Aplicando el método de Bonferroni con $\alpha = 0.20$, se obtiene que ninguna de las observaciones puede considerarse outlier.

Por otro lado, el qqplot de los residuos estudentizados nos da dos observaciones inusuales a considerar: 96 y 69.



Outliers respecto las covariables: Estudiando la diagonal de nuestra matriz *hat* H, obtenemos varias observaciones con alto leverage:

```
> which(hii>hcv)#47,32,92 las mayores de las seis
32 47 64 89 92 96
32 47 64 89 92 96
>
```

Consideraremos las observaciones 47, 32 y 92; pues son las mayores, aunque puede que no todas sean influyentes.

Distancia de Cook: Los valores con mayor distancia de Cook son 32 y 96. Sin embargo, ninguna de nuestras observaciones será influyente pues Di < $F_{o.5;p,n-p}$ con p=7 y n= 97.

DFFITSi: Realizando ahora la prueba de DFFITSi, tomaremos otras cinco observaciones: 96, 32, 69, 39 y 95; que son las más altas de todas las que superan el valor crítico.

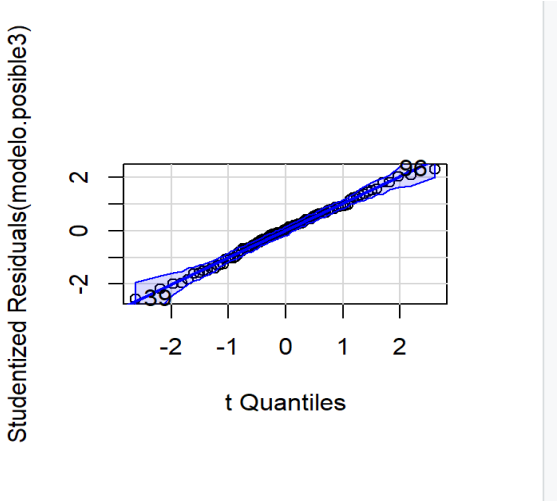
DFBETAS: Siguiendo el mismo criterio que antes consideramos las observaciones 32, 69 y 38.

Observaciones a eliminar: Tras varias pruebas decidimos eliminar la observación 96 del modelo 2.

4.2.3. Outliers del modelo 3.

Outliers respecto a Y: Empezamos tomando las tres observaciones con mayor residuo estandarizado en valor absoluto: 96(2.24), 39 (2.47) y 5 (2.13). Aplicando el método de Bonferroni con α = 0.20, se obtiene que ninguna de las observaciones puede considerarse outlier.

Por otro lado, el *qqplot* de los residuos estudentizados nos da dos observaciones inusuales a considerar: 96 y 39.



Outliers respecto las covariables:Estudiando la diagonal de nuestra matriz *hat* H, obtenemos varias observaciones con alto leverage: 32 y 89. No tienen porqué ser influyentes.

Distancia de Cook: El valor con mayor distancia de Cook es 32. Sin embargo, ninguna de nuestras observaciones será influyente pues Di < $F_{o.5;p,n-p}$ con p=4 y n= 97.

DFFITSi: Tenemos otras dos observaciones 32 y 39, que son las más altas.

DFBETAS: Siguiendo el criterio de antes tomamos las observaciones 32 y 38.

Observaciones a eliminar: Tras varias pruebas decidimos eliminar las observaciones 39 y 96 del modelo 3.

4.3. Estudio de las hipótesis de regresión lineal múltiple

4.3.1. Modelo 1

Primero estudiamos la colinealidad del modelo 1, comparando si mejora al quitar las observaciones señaladas anteriormente.

El número de condición en ambos casos supera la cifra de 1900 lo que indica graves problemas de colinealidad.

Esta imagen muestra los índices de condición del modelo 1 sin eliminar ningún outlier. Observamos que hay índices de condición muy elevados, algunos superando los valores entre 500 y 1000. Por tanto, habrá graves problemas de colinealidad.

```
+ print(indice.condicion1[j])} #muy elevados en general
[1] 1
[1] 13.9463
[1] 235.86
[1] 339.3777
[1] 802.2371
[1] 2013.715
[1] 3094.696
[1] 3677.34
[1] 24987.79
[1] 140153.4
[1] 1573435
```

Por otro lado, en la siguiente imagen aparecen los VIF de las variables en el modelo 1 sin eliminar los outliers.

```
> vif(modelo.posible1) #Algunos muy elevados
     lcavol     lweight        lbph     gleason         svi lweight:svi    lbph:svi
   1.762792    1.540284    1.676779    1.576084  380.647901  134.924628    1.505025
    svi:lcp gleason:svi   svi:pgg45
   3.536061  225.179034    6.875441
```

Observamos que hay algunas variables con un VIF muy alto (>10), lo cual causará graves problemas en la estimación de los correspondientes β_j.

La eliminación de los outliers no mejora en ningún caso este problema.

Diagnóstico: Al tener graves problemas de colinealidad, decidimos **descartar** el modelo 1.

4.3.2. Modelo 2

Primero estudiamos la colinealidad del modelo 2, comparando si mejora al quitar las observaciones señaladas anteriormente.

En ambos casos tenemos un número de condición alrededor de 23.4 y 23.5, lo cual indica que a priori no hay problemas de colinealidad.

La siguiente imagen muestra los índices de condición del modelo 2 sin eliminar ningún outlier. Observamos que solo hay un índice de condición superior a 1000, el resto no da problemas .

```
[1] 1
[1] 7.050797
[1] 9.541752
[1] 27.96063
[1] 61.6995
[1] 156.4254
[1] 1311.576
```

Si no consideramos los outliers, se obtendrán índices muy parecidos.

Por otro lado, en esta tabla encontramos los VIF de las variables en el modelo 2 sin eliminar los outliers; y eliminando los outliers, respectivamente.

```
> vif(modelo.posible2)
    lcavol    lweight       lbph lcavol:svi   lbph:svi    svi:lcp
  1.576775   1.346216   1.618799   4.565295   1.294677   4.046693
> vif(modelo.definitivo2) #Parecidos en ambos casos no superan la cifra de 5
    lcavol    lweight       lbph lcavol:svi   lbph:svi    svi:lcp
  1.549030   1.345572   1.601649   4.509073   1.298714   3.979398
```

Observamos que en ambos casos los VIF son muy parecidos y ninguno supera el valor 5. Por tanto, no tendremos problemas de colinealidad en la estimación y la inferencia sobre los correspondientes β_j.

Al no presentar graves problemas de colinealidad seguimos estudiando las hipótesis de regresión lineal mútiple.

Empezamos comparando los errores en los β's de ambos casos:

```
Call:
lm(formula = lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi +
    lcp:svi, data = prostate)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5140 -0.4355 -0.0334  0.4078  1.5806

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.36110    0.60191   0.600  0.55007
lcavol       0.52879    0.07613   6.946  5.7e-10 ***
lweight      0.33679    0.16695   2.017  0.04665 *
lbph         0.14312    0.06267   2.284  0.02474 *
lcavol:svi   0.44284    0.13881   3.190  0.00196 **
lbph:svi    -0.22106    0.13098  -1.688  0.09493 .
svi:lcp     -0.26606    0.17596  -1.512  0.13402
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

s: 0.7002 on 90 degrees of freedom
Multiple R-squared: 0.6551,
Adjusted R-squared: 0.6321
F-statistic: 28.49 on 6 and 90 DF,  p-value: < 2.2e-16
```

Summary modelo 2 con outliers

```
Call:
lm(formula = lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi +
    lcp:svi, data = PRS2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.34825 -0.42024  0.00848  0.40249  1.53944

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.34858    0.58485   0.596  0.55268
lcavol       0.52614    0.07398   7.112 2.76e-10 ***
lweight      0.34176    0.16223   2.107  0.03796 *
lbph         0.14224    0.06089   2.336  0.02174 *
lcavol:svi   0.37786    0.13732   2.752  0.00718 **
lbph:svi    -0.31008    0.13209  -2.348  0.02111 *
svi:lcp     -0.23094    0.17154  -1.346  0.18163
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

s: 0.6803 on 89 degrees of freedom
Multiple R-squared: 0.6534,
Adjusted R-squared:  0.63
F-statistic: 27.96 on 6 and 89 DF,  p-value: < 2.2e-16
```

Summary modelo 2 sin outliers

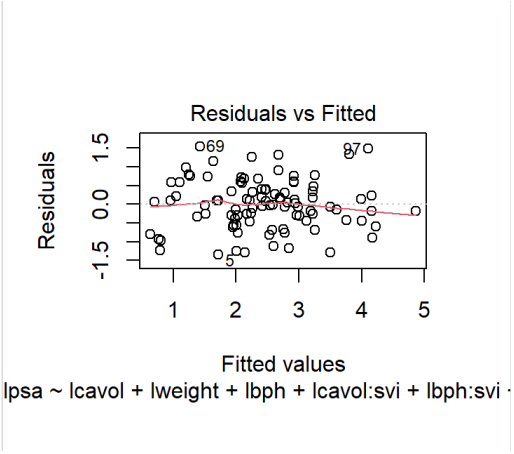
Observamos que en ambos tenemos p-valores y errores estándar muy parecidos y el mismo R-cuadrado ajustado.

Ahora vamos a realizar el *bptest* en el modelo 2 sin eliminar outliers; y eliminando outliers, respectivamente, para observar si hay varianza constante.

```
> bptest(modelo.posible2)

        studentized Breusch-Pagan test

data:  modelo.posible2
BP = 12.931, df = 6, p-value = 0.04415

> bptest(modelo.definitivo2)

        studentized Breusch-Pagan test

data:  modelo.definitivo2
BP = 8.3184, df = 6, p-value = 0.2157
```

Vemos que mejora bastante el modelo 2 eliminando los outliers con respecto a no eliminarlos. Por tanto, podemos suponer varianza constante pues el p-valor ahora es 0.2157 y la hipótesis nula es la suposición de varianza constante.

También podemos ver la homocedasticidad gráficamente:

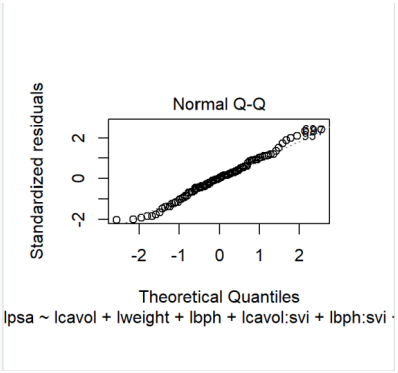


Por último realizaremos el *shapiro test* en el modelo 2 sin eliminar outliers, y eliminando outliers, respectivamente, para observar la normalidad en los residuos de nuestros modelos.

```
> shapiro.test(resid(modelo.posible2))

        Shapiro-Wilk normality test

data:  resid(modelo.posible2)
W = 0.99047, p-value = 0.7215

> shapiro.test(resid(modelo.definitivo2)) #Da pvalor 0.52 que sigue
 siendo indicador de acpetar normalidad en los residuos

        Shapiro-Wilk normality test

data:  resid(modelo.definitivo2)
W = 0.9878, p-value = 0.5235
```

Observamos que la normalidad del modelo 2 eliminando los outliers empeora un poco al tener un p-valor menor. Sin embargo, en ambos casos podemos suponer normalidad en los residuos pues el p-valor ahora es 0.5235, que sigue indicando aceptar la normalidad en los residuos.

También podemos comprobar la normalidad de los errores con el siguiente gráfico



<u>Diagnóstico</u>: A pesar de tener el último índice de condición elevado, como su número de condición es muy bajo y los VIF's no superan el 5, y se verifican el resto de las hipótesis de modelo de regresión lineal múltiple, consideramos el modelo 2 eliminando las outliers.

4.3.3. Modelo 3

Vamos a estudiar la colinealidad del modelo 3, con y sin outliers.

El número de condición en ambos casos es alrededor de 27 por lo que no indica problemas de colinealidad a priori.

Esta imagen muestra los índices de condición del modelo 3 sin eliminar ningún outlier.

```
+ print(indice.condicion3[j])}
[1] 1
[1] 13.82034
[1] 143.8354
[1] 997.3925
```

De todos los modelos inspeccionados es el que tiene el máximo índice de condición más pequeño. Los índices de condición sin considerar los outliers son muy similares.

Por otro lado, en esta tabla encontramos los VIF de las variables en el modelo 3 sin eliminar los outliers; y eliminando los outliers, respectivamente. Observamos que en ambos casos los VIF son muy parecidos y ninguno supera el valor 1.5. Por tanto, no tendremos problemas de colinealidad en la estimación y la inferencia sobre los correspondientes β_j.

```
> vif(modelo.posible3)
  lcavol  lweight      svi
1.447048 1.039188 1.409189
> vif(modelo.definitivo3)
  lcavol  lweight      svi
1.403679 1.035127 1.367226
```

Al no presentar graves problemas de colinealidad seguimos estudiando las hipótesis de regresión lineal mútiple.

Empezamos comparando los errores en los β's de ambos casos:

```
Call:
lm(formula = lpsa ~ lcavol + lweight + svi, data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.72964 -0.45764  0.02812  0.46403  1.57013

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.26809    0.54350  -0.493  0.62298
lcavol       0.55164    0.07467   7.388  6.3e-11 ***
lweight      0.50854    0.15017   3.386  0.00104 **
svi          0.66616    0.20978   3.176  0.00203 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

s: 0.7168 on 93 degrees of freedom
Multiple R-squared: 0.6264,
Adjusted R-squared: 0.6144
F-statistic: 51.99 on 3 and 93 DF,  p-value: < 2.2e-16
```

Summary modelo 3 con outliers

```
Call:
lm(formula = lpsa ~ lcavol + lweight + svi, data = PRS3)

Residuals:
     Min       1Q   Median       3Q      Max
-1.51700 -0.44932  0.02491  0.45068  1.42934

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.35369    0.51826  -0.682 0.496687
lcavol       0.54617    0.07107   7.685 1.71e-11 ***
lweight      0.53370    0.14319   3.727 0.000336 ***
svi          0.68000    0.20451   3.325 0.001275 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

s: 0.6819 on 91 degrees of freedom
Multiple R-squared: 0.6438,
Adjusted R-squared: 0.632
F-statistic: 54.82 on 3 and 91 DF,  p-value: < 2.2e-16
```

Summary modelo 3 sin outliers

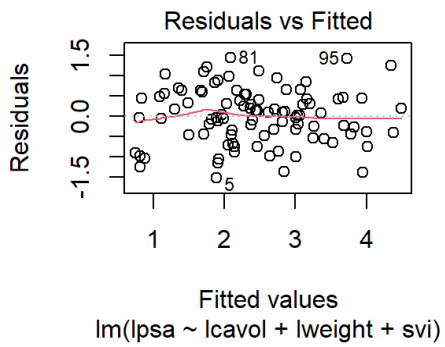
Observamos que en el modelo 3 sin outliers tenemos p-valores y errores estándar algo más pequeños en comparación al modelo 3 con outliers.

Ahora vamos a realizar el *bptest* en el modelo 3 sin eliminar outliers; y eliminando outliers, respectivamente, para observar si hay varianza constante.

```
> bptest(modelo.posible3)

        studentized Breusch-Pagan test

data:  modelo.posible3
BP = 4.2572, df = 3, p-value = 0.235

> bptest(modelo.definitivo3)#pvalor 0.34 (mejora)

        studentized Breusch-Pagan test

data:  modelo.definitivo3
BP = 3.3559, df = 3, p-value = 0.3399
```

Vemos que mejora el modelo 3 eliminando los outliers con respecto a no eliminarlos. Por tanto, podemos suponer varianza constante pues el p-valor ahora es 0.3399.

Comprobamos homocedasticidad con el siguiente gráfico:

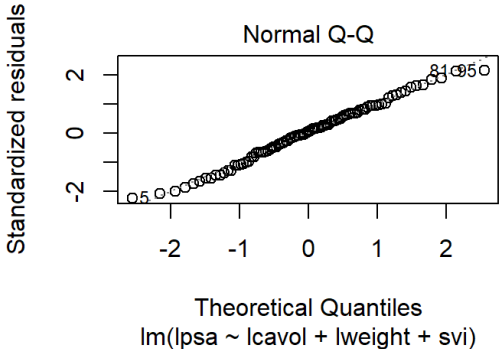


Por último realizaremos el shapiro test en el modelo 3 sin eliminar outliers, y eliminando outliers, respectivamente, para observar la normalidad en los residuos de nuestros modelos.

```
> shapiro.test(resid(modelo.posible3))

        Shapiro-Wilk normality test

data:  resid(modelo.posible3)
W = 0.99338, p-value = 0.9182

> shapiro.test(resid(modelo.definitivo3))

        Shapiro-Wilk normality test

data:  resid(modelo.definitivo3)
W = 0.9908, p-value = 0.7591
```

Aunque el p-valor sea inferior sigue siendo lo suficientemente alto (0.7591) para aceptar normalidad de los resisuos.

Con la imagen siguiente comprobamos lo aceptado con el test anterior:



Diagnóstico: A pesar de tener el último índice de condición elevado (973), el resto son mucho más pequeños y los VIF's no superan el 1.5, por lo que consideraremos el modelo 3 eliminando las outliers.

5.Construcción de modelos (sin las interacciones de la variable cualitativa)

Antes de todo, construimos el modelo completo con la cualitativa y sin interacciones: **lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi.** Para obtener una información inicial hacemos el summary y el ANOVA de este modelo.

SUMMARY

```
Call:
lm(formula = lpsa ~ lcavol + lweight + age + lbph + lcp + gleason +
    pgg45 + svi, data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.7331 -0.3713 -0.0170  0.4141  1.6381

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.669337   1.296387   0.516  0.60693
lcavol       0.587022   0.087920   6.677 2.11e-09 ***
lweight      0.454467   0.170012   2.673  0.00896 **
age         -0.019637   0.011173  -1.758  0.08229 .
lbph         0.107054   0.058449   1.832  0.07040 .
lcp         -0.105474   0.091013  -1.159  0.24964
gleason      0.045142   0.157465   0.287  0.77503
pgg45        0.004525   0.004421   1.024  0.30886
svi          0.766157   0.244309   3.136  0.00233 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

s: 0.7084 on 88 degrees of freedom
Multiple R-squared: 0.6548,
Adjusted R-squared: 0.6234
F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```

ANOVA

```
Analysis of Variance Table

Response: lpsa
          Df Sum Sq Mean Sq  F value    Pr(>F)
lcavol     1 69.003  69.003 137.4962 < 2.2e-16 ***
lweight    1  5.949   5.949  11.8531 0.0008832 ***
age        1  0.420   0.420   0.8369 0.3627958
lbph       1  1.069   1.069   2.1302 0.1479839
lcp        1  0.833   0.833   1.6599 0.2009901
gleason    1  0.516   0.516   1.0284 0.3133153
pgg45      1  1.030   1.030   2.0515 0.1555983
svi        1  4.936   4.936   9.8346 0.0023287 **
Residuals 88 44.163   0.502
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Debido a que algunas covariables tienen p-valores muy elevados, deberemos construir modelos que se ajusten mejor a las hipótesis de regresión lineal múltiple.

5.1. Procedimientos basados en pruebas.

Siguiendo la misma idea que considerando las interacciones, para cada train distinto obtenemos:

->Train 1: Después de realizar nuestras iteraciones llegamos al siguiente modelo: **lpsa ~ lcavol + lweight + svi**

->Train 2: Después de realizar nuestras iteraciones llegamos al siguiente modelo: **lpsa ~ lcavol + lweight + pgg45 + svi**

->Train 3: Después de realizar nuestras iteraciones llegamos al siguiente modelo: **lpsa ~ lcavol + lweight + lcp + pgg45 + svi**

Nos quedamos con aquel que mejor predice su correspondiente parte de test:

lpsa ~ lcavol + lweight + svi

Observación: Este modelo es el mismo que se obtiene con el tercer método de la construcción de modelos con las interacciones de la variable cualitativa (**svi)**, (modelo 3).

5.2. Procedimientos basados en criterios.

5.2.1. Criterio Cp de Mallow: Siguiendo la misma idea que considerando las interacciones, para cada train distinto obtenemos:

->Train 1: El modelo obtenido es: **lpsa ~ lcavol + lweight + lcp + pgg45 + svi.**

->Train 2: El modelo obtenido es: **lpsa ~ lcavol + age + lbph + svi.**

->Train 3: El modelo obtenido es: **lpsa ~ lcavol + age + lbph + gleason + svi.**

Nos quedamos con el que mejor predice su parte de test:

lpsa ~ lcavol + lweight + lcp + pgg45 + svi

5.2.2. Método de entrenamiento y test con regsubsets: Siguiendo la misma idea que con las interacciones de la cualitativa obtenemos el modelo:

lpsa ~ lcavol + lweight + lbph + svi

5.3. Modelos finales a considerar

->Modelo 4: Modelo obtenido por el criterio Cp de Mallow**:**
lpsa ~ lcavol + lweight + lcp + pgg45 + svi

->Modelo 5: Modelo obtenido por el método de entrenamiento y test con regsubset:
lpsa ~ lcavol + lweight + lbph + svi

6.Diagnóstico de modelos (sin las interacciones de la variable cualitativa)

6.1. Diagnóstico inicial

Realizamos el contraste de hipótesis para ver si podemos suponer la eliminación de las variables excluidas en cada modelo. Para cada uno obtenemos los siguientes p-valores:

->Modelo 4: p-valor = 0.1636.

->Modelo 5: p-valor = 0.3355.

Podemos suponer la eliminación de las correspondientes covariables con hasta un grado de significación del 0.15 .

Ahora realizaremos un diagnóstico inicial de cada modelo realizando un summary de cada uno y fijándonos en los p-valores de las distintas variables.

->Modelo 4: El mayor p-valor es 0.36788 por lo que tendríamos una variable no significativa que podría dar problemas.

```
Call:
lm(formula = lpsa ~ lcavol + lweight + lcp + pgg45 + svi, data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.65037 -0.42498  0.03406  0.41354  1.65549

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.422604   0.556675  -0.759  0.44972
lcavol       0.566150   0.086414   6.552 3.34e-09 ***
lweight      0.509887   0.150458   3.389  0.00104 **
lcp         -0.082411   0.091068  -0.905  0.36788
pgg45        0.004510   0.003353   1.345  0.18191
svi          0.690457   0.242436   2.848  0.00544 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 0.7171 on 91 degrees of freedom
Multiple R-squared: 0.6342,
Adjusted R-squared: 0.6141
F-statistic: 31.56 on 5 and 91 DF,  p-value: < 2.2e-16
```

->Modelo 5: El mayor p-valor es 0.11213. Las variables son más significativas con respecto al modelo anterior.

```
Call:
lm(formula = lpsa ~ lcavol + lweight + lbph + svi, data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.82653 -0.42270  0.04362  0.47041  1.48530

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.14554    0.59747   0.244  0.80809
lcavol       0.54960    0.07406   7.422 5.64e-11 ***
lweight      0.39088    0.16600   2.355  0.02067 *
lbph         0.09009    0.05617   1.604  0.11213
svi          0.71174    0.20996   3.390  0.00103 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 0.7108 on 92 degrees of freedom
Multiple R-squared: 0.6366,
Adjusted R-squared: 0.6208
F-statistic: 40.29 on 4 and 92 DF,  p-value: < 2.2e-16
```

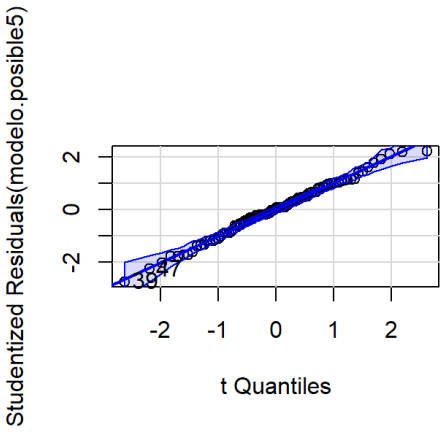
6.2. Outliers.

6.2.1. Outliers modelo 5

Outliers respecto a Y: Empezamos tomando las dos observaciones con mayor residuo estandarizado en valor absoluto 39 (2.64), 47 (2.21) y 69 (2.18).

Aplicando el método de Bonferroni con $\alpha = 0.20$, se obtiene que ninguna de las observaciones puede considerarse outlier.

Por otro lado, el qqplot de los residuos estudentizados nos da dos observaciones inusuales a considerar: 39 y 47.



Outliers respecto las covariables: Empezamos viendo que 32, 69 y 89 son tres observaciones con alto leverage superando el valor crítico aunque, no tienen porqué ser influyentes.

Distancia de Cook: Las tres observaciones con mayor distancia de Cook son 32, 69 y 39. Sin embargo, ninguna de nuestras observaciones será influyente pues $Di < F_{o.5;p,n-p}$ con p=5 y n= 97

DFFITSi: Realizando ahora la prueba de DFFITSi, tomaremos otras siete observaciones: 1, 32, 39, 47, 69, 95 y 96.

```
> which(abs(dffitsmodel)>dffitscv)#1, 32, 39, 47, 69, 95, 96
 1 32 39 47 69 95 96
 1 32 39 47 69 95 96
>
```

DFBETAS: Obtenemos las observaciones 32, 38 y 69.

Observaciones a eliminar: Tras varias pruebas decidimos eliminar la observación 39 del modelo 5 con la finalidad de mejorar las hipótesis de regresión lineal múltiple.

Nota: No hemos estudiado los outliers del modelo 4 puesto que nos va a presentar grandes problemas de colinealidad, los cuales no mejoraran significativamente quitando outliers.

6.3. Estudio de las hipótesis de regresión lineal múltiple

6.3.1. Modelo 4

Antes de todo, vamos a estudiar la colinealidad en el modelo 4. El número de condición es 458.62 aproximadamente lo que indica problemas de colinealidad. A continuación, hallamos los índices de condición:

```
+ print(indice.condicion4[j])} #Indices de condicion muy elevados
[1] 1
[1] 149.5665
[1] 756.0389
[1] 3198.852
[1] 15878.16
[1] 86761.83
```

Observamos que hay índices de condición muy elevados, algunos superando los valores entre 500 y 1000. Por tanto, habrá graves problemas de colinealidad.

Para un análisis más detallado hallamos los VIF:

```
> vif(modelo.posible4)
  lcavol  lweight      lcp    pgg45      svi
1.936743 1.042444 3.027291 1.669510 1.880792
```

Vemos que todas las variables tienen un VIF que no supera la cifra de 3.

Diagnóstico: Al tener graves problemas de colinealidad con el número de condición y los índices de condición, decidimos **descartar** el modelo 4.

6.3.2. Modelo 5

Comenzamos estudiando la colinealidad del modelo 5. Su número de condición es 35.10 (aceptable). Si quitamos los outliers especificados anteriormente no cambia notoriamente.

La siguiente imagen muestra los índices de condición del modelo 5 sin eliminar ningún outlier.

```
> for (j in 1:p){indice.condicion5[j]<-lambda.max5/eigenB5[j]
+ print(indice.condicion5[j])}
[1] 1
[1] 8.050326
[1] 13.95455
[1] 145.9434
[1] 1232.476
```

Observamos que el quinto índice es muy elevado, con un valor superior a 1000. Sin embargo, el resto de índices de condición son bajos.

Se puede comprobar que en el modelo 5 eliminando los outliers encontramos índices de condición muy similares, y podremos sacar conclusiones análogas.

Por otro lado, en esta tabla encontramos los VIF de las variables en el modelo 5 sin eliminar los outliers; y eliminando los outliers, respectivamente

```
> vif(modelo.posible5)#Muy buenos, se mueven entre 1 y 2
  lcavol  lweight     lbph      svi
1.447473 1.291345 1.261576 1.435481
> vif(modelo.definitivo5)#Muy parecidos
  lcavol  lweight     lbph      svi
1.428627 1.281752 1.260954 1.422033
```

Observamos que en ambos casos los VIF son muy parecidos y ninguno supera el valor 1.5. Por tanto, no tendremos problemas en la estimación de los correspondientes β_j.

A pesar de tener un índice de condición elevado, como el resto de índices era bajo junto con el número de condición y los VIF, decidimos seguir con el diagnóstico de las hipótesis del modelo de regresión lineal múltiple.

Empezamos comparando los errores estándar de los β's y los p-valores:

```
Call:
lm(formula = lpsa ~ lcavol + lweight + lbph + svi, data = prostate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.82653 -0.42270  0.04362  0.47041  1.48530

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.14554    0.59747   0.244  0.80809
lcavol       0.54960    0.07406   7.422 5.64e-11 ***
lweight      0.39088    0.16600   2.355  0.02067 *
lbph         0.09009    0.05617   1.604  0.11213
svi          0.71174    0.20996   3.390  0.00103 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

s: 0.7108 on 92 degrees of freedom
Multiple R-squared: 0.6366,
Adjusted R-squared: 0.6208
F-statistic: 40.29 on 4 and 92 DF,  p-value: < 2.2e-16
```

Summary modelo 5 con outliers

```
Call:
lm(formula = lpsa ~ lcavol + lweight + lbph + svi, data = PRS5)

Residuals:
     Min       1Q   Median       3Q      Max
-1.63996 -0.40596  0.03938  0.48964  1.47587

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.10608    0.57755   0.184 0.854678
lcavol       0.54934    0.07156   7.676 1.78e-11 ***
lweight      0.40124    0.16046   2.501 0.014191 *
lbph         0.10313    0.05448   1.893 0.061551 .
svi          0.80719    0.20586   3.921 0.000171 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

s: 0.6869 on 91 degrees of freedom
Multiple R-squared: 0.6641,
Adjusted R-squared: 0.6494
F-statistic: 44.99 on 4 and 91 DF,  p-value: < 2.2e-16
```

Summary modelo 5 sin outliers

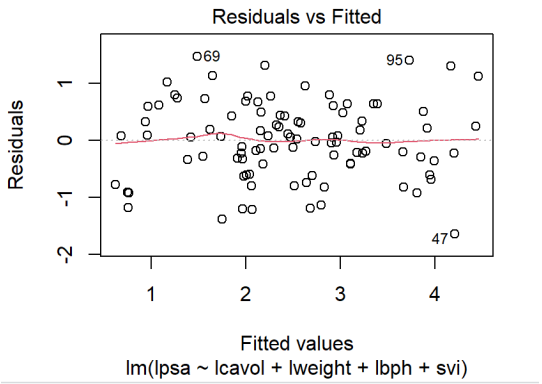
Observamos que en el modelo 5 eliminando outliers tenemos p-valores más pequeños y un R-cuadrado ajustado mayor, en comparación con el modelo 5 sin eliminar outliers.

Ahora vamos a realizar el *bptest* en el modelo 5 sin eliminar outliers; y eliminando outliers, respectivamente, para observar si hay varianza constante.

```
> bptest(modelo.posible5)

        studentized Breusch-Pagan test

data:  modelo.posible5
BP = 5.193, df = 4, p-value = 0.2681

> bptest(modelo.definitivo5)#pvalor 0.43 (mejora)

        studentized Breusch-Pagan test

data:  modelo.definitivo5
BP = 3.8123, df = 4, p-value = 0.432
```

Vemos que mejora bastante el modelo 5 eliminando los outliers con respecto a no eliminarlos al tener un p-valor mayor. Al valer 0.432 podemos suponer varianza constante.

También, lo podemos corroborar con el siguiente gráfico:

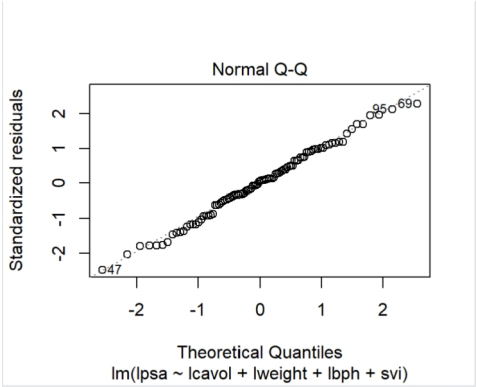


Por último, realizaremos el *shafiro test* en el modelo 5 sin eliminar outliers, y eliminando outliers, respectivamente, para observar la normalidad en los residuos de nuestros modelos.

```
> shapiro.test(resid(modelo.posible5))

        Shapiro-Wilk normality test

data:  resid(modelo.posible5)
W = 0.99133, p-value = 0.787

> shapiro.test(resid(modelo.definitivo5))#0.82, mejora

        Shapiro-Wilk normality test

data:  resid(modelo.definitivo5)
W = 0.99175, p-value = 0.8227
```

Vemos que mejora el modelo 5 eliminando los outliers con respecto a no eliminarlos al obtener un p-valor superior. Al valer 0.8227 aceptaremos la hipótesis de normalidad en los residuos.

Esto último lo podemos comprobar con la siguiente imagen:



Diagnóstico: A pesar de tener un índice de condición elevado para j= 5, el resto de j's son bajos, al igual que su número de condición y los VIF's, que no superan el 1.5, y se verifican el resto de las hipótesis de modelo de regresión lineal múltiple. Por tanto, consideramos el modelo 5 eliminando las outliers.

7.Elección del modelo

Los modelos candidatos a ser el modelo final son el 2, el 3 y el 5 sin considerar sus respectivos outliers señalados anteriormente. Por lo tanto, como criterio, vamos a fijarnos en el error de validación cruzada de cada uno de los modelos.

Para el modelo 2 el error es 0.5065, para el 3 es 0.4949 y para el 5 es 0.5070. Observamos que prácticamente todos tienen el mismo error de validación cruzada, estrictamente, es el 3 el que presenta menor error.

Por otro lado, como el modelo 2 no mejora la predicción respecto al resto y al tener más variables y VIF más elevados decidimos descartarlo.

El 3 y el 5 poseen un R-cuadrado ajustado muy parecido, los errores estándar de los β son pequeños y ambos cumplen las hipótesis de regresión lineal múltiple a la perfección.

8.Conclusión

Debido a que el modelo 3 posee menos covariables que el 5 proponemos el modelo 3 (sin considerar los outiers marcados antes) como modelo final:

lpsa ~ lcavol + lweight + svi

A continuación mostramos el *summary* y el *ANOVA*:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.35369    0.51826  -0.682 0.496687
lcavol       0.54617    0.07107   7.685 1.71e-11 ***
lweight      0.53370    0.14319   3.727 0.000336 ***
svi          0.68000    0.20451   3.325 0.001275 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 0.6819 on 91 degrees of freedom
Multiple R-squared: 0.6438,
Adjusted R-squared: 0.632
F-statistic: 54.82 on 3 and 91 DF,  p-value: < 2.2e-16
```

```
Response: lpsa
          Df Sum Sq Mean Sq F value    Pr(>F)
lcavol     1 64.934  64.934 139.657 < 2.2e-16 ***
lweight    1  6.388   6.388  13.739 0.0003605 ***
svi        1  5.141   5.141  11.056 0.0012753 **
Residuals 91 42.311   0.465
```

Para finalizar, mostramos los intervalos de confianza simultáneos de los β´s por el método de Bonferroni con grado de significación 0.10 :

```
                5%       95%
LCAVOL  0.3925903 0.6997416
LWEIGHT 0.2242757 0.8431144
SVI     0.2380966 1.1219069
```

9.Anexo

Observación: Como la variable **lpsa** toma valores negativos no hemos podido probar a transformar la variable respuesta mediante *boxCox*.

Observación: Al estar los valores de la variable **lpsa** ordenados de menor a mayor, el estadístico de Durbin Watson (véase la fórmula) nos daba un p-valor pequeño. Sin embargo, al hacer una permutación de los mismos, desaparecía dicho problema.

Script de R:

```
library(MASS)
library(car)
library(faraway)
library(leaps)
library(mixlm)
library(PASWR)
library(ggplot2)
library(lmtest)
library(psych)



################################################################################
################################################################################
################################################################################
############################################################
#
# GRUPO 13
# CONJUNTO DE DATOS<-prostate
# VARIABLE RESPUESTA<-lpsa
#
# BREVE DESCRIPCION DE LAS VARIABLES:
# Se ha realizado un estudio a 97 hombres que van a recibir una prostatectomía radical.
# lcavol <- variable continua volumen del cancer (aplicado el log)
# lweight <- variable continua peso de la prostata (aplicado el log)
# age <- variable discreta la edad
# lbph <- variable continua cantidad de hiperplasia prostatica benigna, es decir, agrandamiento de la glandula prostatica (aplicado el log)
# svi <- variable CATEGORICA invasion de vesicula seminal (0's o 1's)
# lcp <- variable continua penetracion capsular mide lo que ha alcanzado la pared exterior de la prostata el cancer (aplicado el log)
```

gleason <- variable discreta puntaje de gleason, va de 2 a 10, es la suma de los dos grados mas comunes en el tejido canceroso (grados se puntuan de 1 al 5) cuanto mas alto las celulas cancerigenas son mas peligrosas

Para ver informacion de gleason -> https://www.cancer.org/es/tratamiento/como-comprender-su-diagnostico/pruebas/como-comprender-su-informe-de-patologia/patologia-de-la-prostata/patologia-del-cancer-de-prostata.html

pgg45 <- variable continua es el porcentaje de tejido cancerigeno que tiene grado de Gleason 4~5

lpsa <- variable continua indica la concentracion de PSA en la sangre, la PSA es una proteina producida por celulas normales y celulas malignas de la glandula prostaticas. La concentracion de PSA en sangre es frecuente en hombres con cancer de propstata (aplicado el log)

#

##
##
##
##

#INSPECCION DE LAS VARIABLES

typeof(prostate$lcavol)

typeof(prostate$lweight)

typeof(prostate$age)

typeof(prostate$lbph)

typeof(prostate$svi) #Son enteros, 0 indica no svi y 1 indica svi

typeof(prostate$lcp)

typeof(prostate$gleason)

typeof(prostate$pgg45)

typeof(prostate$lpsa)

summary(prostate) #Nos da informacion util sobre cada variable de primeras

multi.hist(prostate$lcavol)

multi.hist(prostate$lweight)

multi.hist(prostate$age)

multi.hist(prostate$lbph) #la mayoria de los datos se concentran en -1.3

multi.hist(prostate$svi)

```
multi.hist(prostate$lcp) #la moyoria se concentra en el valor minimo como lbph

multi.hist(prostate$gleason)

multi.hist(prostate$pgg45)

multi.hist(prostate$lpsa)


Y<-prostate$lpsa

x8<-factor(prostate$svi, labels=c("NO", "SI")) #Lo ponemos como factor

plot(Y~x8) #No svi indica menor lpsa que si svi en general lo que indica que puede ser interesante el considerar la cualitativa

scatterplotMatrix(x=prostate)

correlacion<-cor(prostate[-5])#Nos fijamos en aquellos que esten correlacionados pues si estuviesen los dos podriamos tener problemas de colinealidad

correlacion #lcavol esta muy correlacionado con lpsa lo que indica que sera interesante considerarlo en nuestro modelo




#####################################################

##     CON INTERACCION  DE LA CUALITATIVA       ##

#####################################################


#CONSTRUIMOS EL MODELO CON LA CUALITATIVA Y LAS INTERACCIONES

modelo.completo<-lm(lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi + lcavol:svi + lweight:svi + age:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi, data = prostate)

summary(modelo.completo)

anova(modelo.completo)


#SEPARACION TRAIN Y TEST

train<-sample(c(TRUE,FALSE), size=nrow(prostate), replace=TRUE, prob=c(0.70,0.30))

train

test<-(!train)
```

```
prop.table(table(train))
```

#CONSTRUCCION DE MODELO CON INTERACCION METODO BACKWARD FIJANDONOS EN LOS PVALORES (LO HACEMOS PARA 3 CONJUNTOS DISTINTOS DE TRAIN) (DESPUES HALLAREMOS EL ERROR DE PREDICCION DE CADA MODELO OBTENIDO CON SU TEST ASOCIADO)

#Cada iteracion actualizo train y hago el backward (a mano) con alfa = 0.20

```
modelo.completo_train<-lm(lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi + lcavol:svi + lweight:svi + age:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi, data = prostate[train,])

summary(modelo.completo_train)


modelo.update1<-update(modelo.completo_train,.~.-lbph:svi)

summary(modelo.update1)


modelo.update2<-update(modelo.update1,.~.-gleason)

summary(modelo.update2)


modelo.update3<-update(modelo.update2,.~.-lcp)

summary(modelo.update3)


modelo.update4<-update(modelo.update3,.~.-svi)

summary(modelo.update4)


modelo.update5<-update(modelo.update4,.~.-svi:gleason)

summary(modelo.update5)


modelo.update6<-update(modelo.update5,.~.-pgg45)

summary(modelo.update6)


modelo.update7<-update(modelo.update6,.~.-age)

summary(modelo.update7)
```

```
modelo.update8<-update(modelo.update7,.~.-lweight:svi)

summary(modelo.update8)


modelo.update9<-update(modelo.update8,.~.-lweight)

summary(modelo.update9)

#El esquema del backward que aparece corresponde al tercer conjunto train que cogimos


#NOTA: Estos son los modelos que obtuvimos para cada conjunto train distinto, al no usar semilla no tiene sentido volver ejecutar esto pues no tenemos los conjuntos train originales

#train 1 -> lpsa ~ lcavol + lweight + lbph + gleason + svi + lweight:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi

pred_1<-predict(modelo.update6, newdata = prostate[test,])#pongo modelo.update6 puesto que es el que obtuve con la primera iteracion pero no tiene por que ser el que te de ahora pues train no es el mismo

error_pred_1<-sum((prostate$lpsa[test]-pred_1)**2)/length(prostate$lpsa[test])


#train 2 -> lpsa ~ lcavol + lbph + pgg45 + svi + lbph:svi + pgg45:svi

pred_2<-predict(modelo.update9, newdata = prostate[test,])

error_pred_2<-sum((prostate$lpsa[test]-pred_2)**2)/length(prostate$lpsa[test])


#train 3 -> lpsa ~ lcavol + lweight + age + lbph + lcavol:svi + lweight:svi + age:svi + svi:lcp + svi:pgg45

pred_3<-predict(modelo.update6, newdata = prostate[test,])

error_pred_3<-sum((prostate$lpsa[test]-pred_3)**2)/length(prostate$lpsa[test])


#NOS QUEDAMOS CON EL QUE TIENE MENOR ERROR DE PREDICCION

which.min(c(error_pred_1, error_pred_2, error_pred_3))

#Obtenemos asi el modelo: lpsa ~ lcavol + lweight + lbph + gleason + svi + lweight:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi




#CONSTRUCCION DE MODELO EN BASE A CRITERIO Cp (MISMA IDEA QUE CON EL BACKWARD)
```

#NOTA: Estos son los modelos que obtuvimos para cada conjunto train distinto, al no usar semilla no tiene sentido volver ejecutar esto pues no tenemos los conjuntos train originales

```
models <-regsubsets(lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi + lcavol:svi + lweight:svi + age:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi, data = prostate[train,])

summary(models)

plot(models,scale="Cp")



#train 1 -> lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi

mod1<-lm(lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi, data = prostate[train,])

pred_1_Cp<-predict(mod1, newdata = prostate[test,])

error_pred_1_Cp<-sum((prostate$lpsa[test]-pred_1_Cp)**2)/length(prostate$lpsa[test])


#train 2 -> lpsa ~ lcavol + lweight + pgg45 + svi + lweight:svi + age:svi + lcp:svi

mod2<-lm(lpsa ~ lcavol + lweight + pgg45 + svi + lweight:svi + age:svi + lcp:svi, data = prostate[train,])

pred_2_Cp<-predict(mod2, newdata = prostate[test,])

error_pred_2_Cp<-sum((prostate$lpsa[test]-pred_2_Cp)**2)/length(prostate$lpsa[test])


#train 3 -> lpsa ~ lcavol + age + lbph + lcp + pgg45 + svi + lbph:svi + gleason:svi

mod3<-lm(lpsa ~ lcavol + age + lbph + lcp + pgg45 + svi + lbph:svi + gleason:svi, data = prostate[train,])

pred_3_Cp<-predict(mod3, newdata = prostate[test,])

error_pred_3_Cp<-sum((prostate$lpsa[test]-pred_3_Cp)**2)/length(prostate$lpsa[test])


which.min(c(error_pred_1_Cp, error_pred_2_Cp, error_pred_3_Cp))

#Obtenemos asi el modelo: lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi



#CONSTRUCCION DE MODELO CON ENTRENAMIENTO Y TEST CON REGSUBSETS Y k=n

predict.regsubsets<-function(object, newdata, id,...){  #Funcion auxiliar
```

```
 form<-as.formula(object$call[[2]])
 mat<-model.matrix(form, newdata)
 coefi<-coef(object, id=id)
 xvar<-names(coefi)
 mat[,xvar]%*%coefi
}

n<-nrow(prostate)
k<-n #Número de grupos que voy a hacer pues dejo solo 1 fuera
folds<-sample(x=1:k, size = n, replace=FALSE)
folds
cv.errors<-matrix(NA, k, 8, dimnames=list(NULL, paste(1:8)))#Matriz para albergar los errores
for (j in 1:k){
 best.fit<-regsubsets(lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi +
lcavol:svi + lweight:svi + age:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi,
data=prostate[folds!=j,],  method = "exhaustive")#cogemos datos del train que son todos
menos j
 for (i in 1:8){
   pred<-predict.regsubsets(best.fit, newdata = prostate[folds==j,], id=i)
   cv.errors[j,i]<-mean((prostate$lpsa[folds==j]-pred)^2)
 }
}
cv.errors
mean.cv.errors<-apply(cv.errors, 2, mean)
mean.cv.errors
coef(best.fit, which.min(mean.cv.errors))
#Obtenemos lpsa ~ lcavol + lweight + svi (el mismo que con BIC con todos los datos) El
regsubsets con method = "exhaustive" y tambien coincide con method = "backward" y method
= "forward"



#MODELOS A CONSIDERAR
```

```
modelo.posible1<-lm(lpsa ~ lcavol + lweight + lbph + gleason + svi + lweight:svi + lbph:svi + lcp:svi + gleason:svi + pgg45:svi, data = prostate) #Backward

modelo.posible2<-lm(lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi, data = prostate) #Cp

modelo.posible3<-lm(lpsa ~ lcavol + lweight + svi, data = prostate) #predict.regsubsets (tambien BIC si cogemos data=prostate)


#COMPROBAMOS CON CONTRASTE DE HIPOTESIS LA ELIMINACION DE LAS CORRESPONDIENTES VARIABLES

anova(modelo.completo, modelo.posible1) #pvalor 0.4375

anova(modelo.completo, modelo.posible2) #pvalor 0.4027

anova(modelo.completo, modelo.posible3) #pvalor 0.174



#VAMOS A ESTUDIAR LOS OUTLIERS DE CADA MODELO POSIBLE Y HAREMOS EL DIAGNOSTICO DE CADA UNO DE ELLOS

##########################################################################################


#DETECCION DE OBSERVACIONES RARAS E INFLUYENTES Y DIAGNOSTICO DE modelo.posible1


#Comenzamos viendo outliers para comparar si al quitarlas mejoran las hipotesis que debe cumplir un modelo de regresion lineal multiple

#OUTLIERS RESPECTO A Y

sort(abs(rstandard(modelo.posible1)), decreasing = TRUE)[1:3] #Cojo las tres observaciones con mayor residuo estandarizado en valor absoluto

#Obtenemos las observaciones 97 (2.65), 96 (2.63) y 39 (2.36)

#Ahora segun el metodo de Bonferroni alfa = 0.20

BCV<-qt(1-0.20/(2*length(prostate[,1])), length(prostate[,1])-12)

BCV

sum(abs(rstudent(modelo.posible1))>BCV) #Segun el metodo de bonferroni ningun residuo puede considerarse outlier
```

```
outlierTest(modelo.posible1)

qqPlot(modelo.posible1) #Nos da un qqplot de los residuos studentizados. Obtenemos el 96 y 97 como observaciones inusuales


#OUTLIERS RESPECTO A LAS COVARIABLES


hii<-hatvalues(modelo.posible1)#Elementos diag

sort(hii,decreasing=TRUE)

hcv<-2*11/97

sum(hii>hcv)

which(hii>hcv)#47,74,89,92 las mayores

#(No todas tienen por que ser influyentes)


dcooks<-cooks.distance(modelo.posible1)

sort(dcooks,decreasing=TRUE)#97,96,95,39

f_teo<-qf(0.5,11,86)

which(dcooks>f_teo)


dffitscv<-2*sqrt(11/97)

dffitsmodel<-dffits(modelo.posible1)

sort(abs(dffitsmodel),decreasing=TRUE)

sum(abs(dffitsmodel)>dffitscv) #10 observaciones

which(abs(dffitsmodel)>dffitscv)#97,96,95,39,32 (las mayores)


dfbetacv<-2/sqrt(97)

dfbetamodel<-dfbeta(modelo.posible1)

head(dfbetamodel)

sum(abs(dfbetamodel[,1])>dfbetacv) #7 VALORES, como siempre, luego cogeremos los mas altos, no todos

sum(abs(dfbetamodel[,2])>dfbetacv)

sum(abs(dfbetamodel[,3])>dfbetacv)

sum(abs(dfbetamodel[,4])>dfbetacv)
```

```
sum(abs(dfbetamodel[,5])>dfbetacv)

sum(abs(dfbetamodel[,6])>dfbetacv) #23 VALORES

sum(abs(dfbetamodel[,7])>dfbetacv) #4 VALORES

sum(abs(dfbetamodel[,8])>dfbetacv)

sum(abs(dfbetamodel[,9])>dfbetacv)

sum(abs(dfbetamodel[,10])>dfbetacv)#1 VALOR

sum(abs(dfbetamodel[,11])>dfbetacv)


sort(abs(dfbetamodel[,1]),decreasing = TRUE)

which(abs(dfbetamodel[,1])>dfbetacv)#32,69


sort(abs(dfbetamodel[,6]),decreasing = TRUE)

which(abs(dfbetamodel[,6])>dfbetacv)#96, 47


sort(abs(dfbetamodel[,7]),decreasing = TRUE)

which(abs(dfbetamodel[,7])>dfbetacv)#95, 89


sort(abs(dfbetamodel[,10]),decreasing = TRUE)

which(abs(dfbetamodel[,10])>dfbetacv)#96


#Tras varias pruebas, decidimos eliminar las siguientes:

obs.out1<-c(97)

PRS1<-prostate[-obs.out1,]

modelo.definitivo1<-lm(lpsa ~ lcavol + lweight + lbph + gleason + svi + lweight:svi + lbph:svi +
lcp:svi + gleason:svi + pgg45:svi,data=PRS1)


#COLINEALIDAD

kappa(modelo.posible1)

kappa(modelo.definitivo1) #COLINEALIDAD MUY GRAVE EN AMBOS CASOS

X1<-model.matrix(modelo.posible1)

p<-ncol(X1)
```

```
eigenB1<-eigen(t(X1)%*%X1,only.values = TRUE)$values

lambda.max1<-max(eigenB1)

lambda.min1<-min(eigenB1)

indice.condicion1<-0

for (j in 1:p){indice.condicion1[j]<-lambda.max1/eigenB1[j]

print(indice.condicion1[j])} #muy elevados en general

vif(modelo.posible1) #Algunos muy elevados

#Se comprueba que esto tampoco mejora con modelo.definitivo1

#AL SER MUY ELEVADA LA COLINEALIDAD DECIDIMOS DESCARTAR EL MODELO:
modelo.posible1



################################################################################
###


#DETECCION DE OBSERVACIONES RARAS E INFLUYENTES Y DIAGNOSTICO DE modelo.posible2


#Comenzamos viendo outliers para comparar si al quitarlas mejoran las hipotesis que debe
cumplir un modelo de regresion lineal multiple

#OUTLIERS RESPECTO A Y

sort(abs(rstandard(modelo.posible2)), decreasing = TRUE)[1:3] #Cojo las tres observaciones
con mayor residuo estandarizado en valor absoluto

#Obtenemos las observaciones 69 (2.36), 96 (2.44) y 39 (2.30)

#Ahora segun el metodo de Bonferroni alfa = 0.20

BCV<-qt(1-0.20/(2*length(prostate[,1])), length(prostate[,1])-8)

BCV

sum(abs(rstudent(modelo.posible2))>BCV) #Segun el metodo de bonferroni ningun residuo
puede considerarse outlier

outlierTest(modelo.posible2)

qqPlot(modelo.posible2) #Nos da un qqplot de los residuos studentizados. Obtenemos el 96 y
69 como observaciones inusuales


#OUTLIERS RESPECTO A LAS COVARIABLES
```

```
hii<-hatvalues(modelo.posible2)#Elementos diag
sort(hii,decreasing=TRUE)
hcv<-2*7/97
sum(hii>hcv) #Obtenemos 6 observaciones
which(hii>hcv)#47,32,92 las mayores de las seis
#(No todas tienen por que ser influyentes)

dcooks<-cooks.distance(modelo.posible2)
sort(dcooks,decreasing=TRUE)#96,32
f_teo<-qf(0.5,7,90)
which(dcooks>f_teo) #Ninguna

dffitscv<-2*sqrt(7/97)
dffitsmodel<-dffits(modelo.posible2)
sort(abs(dffitsmodel),decreasing=TRUE)
sum(abs(dffitsmodel)>dffitscv) #8 observaciones
which(abs(dffitsmodel)>dffitscv)#96,32,69,39,95 las mayores

dfbetacv<-2/sqrt(97)
dfbetamodel<-dfbeta(modelo.posible2)
head(dfbetamodel)
sum(abs(dfbetamodel[,1])>dfbetacv) #3 observaciones
sum(abs(dfbetamodel[,2])>dfbetacv)
sum(abs(dfbetamodel[,3])>dfbetacv)
sum(abs(dfbetamodel[,4])>dfbetacv)
sum(abs(dfbetamodel[,5])>dfbetacv)
sum(abs(dfbetamodel[,6])>dfbetacv)
sum(abs(dfbetamodel[,7])>dfbetacv)

sort(abs(dfbetamodel[,1]),decreasing = TRUE)
```

```
which(abs(dfbetamodel[,1])>dfbetacv)#32,69,38


#Tras varias pruebas, decidimos eliminar las siguientes:

obs.out2<-c(96)

PRS2<-prostate[-obs.out2,]

modelo.definitivo2<-lm(lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi,data=PRS2)


#COMPARACION y DIAGNOSTICO

kappa(modelo.posible2)

kappa(modelo.definitivo2) #numeros de condicion alrededor de 23.4 lo que NO indica colinealidad moderada apriori

X2<-model.matrix(modelo.definitivo2)

p<-ncol(X2)

eigenB2<-eigen(t(X2)%*%X2,only.values = TRUE)$values

lambda.max2<-max(eigenB2)

lambda.min2<-min(eigenB2)

indice.condicion2<-0

for (j in 1:p){indice.condicion2[j]<-lambda.max2/eigenB2[j]

print(indice.condicion2[j])} #para j=7 indice de condicion superior a 1000 -> Colinealidad muy grave

#Si consideras modelo.definitivo2 los indices son muy similares

vif(modelo.posible2)

vif(modelo.definitivo2) #Parecidos en ambos casos no superan la cifra de 5


summary(modelo.definitivo2)#pvalores muy parecidos y mismo R2ajustado

plot(modelo.definitivo2,which=c(1,2))

bptest(modelo.definitivo2) #Mejora bastante, ahora podemos suponer varianza constante pues el pvalor ahora es 0.2157

shapiro.test(resid(modelo.definitivo2)) #Da pvalor 0.52 que sigue siendo indicador de acpetar normalidad en los residuos
```

```
summary(modelo.posible2)

plot(modelo.posible2,which=c(1,2))

bptest(modelo.posible2)

shapiro.test(resid(modelo.posible2))


#A PESAR DE TENER UN INDICE DE CONDICION ELEVADO, COMO SU NUMERO DE CONDICION ES MUY BAJO Y LOS VIF´S NO SUPERAN EL 5 Y VERIFICA EL RESTO DE HIPOTESIS DEL MODELO DE REGRESION LINEAL MULTIPLE CONSIDERAMOS EL MODELO: modelo.definitivo2




################################################################################################


#DETECCION DE OBSERVACIONES RARAS E INFLUYENTES Y DIAGNOSTICO DE modelo.posible3


#Comenzamos viendo outliers para comparar si al quitarlas mejoran las hipotesis que debe cumplir un modelo de regresion lineal multiple

#OUTLIERS RESPECTO A Y

sort(abs(rstandard(modelo.posible3)), decreasing = TRUE)[1:3] #Cojo las tres observaciones con mayor residuo estandarizado en valor absoluto

#Obtenemos las observaciones 5 (2.13), 96 (2.24) y 39 (2.47)

#Ahora segun el metodo de Bonferroni alfa = 0.20

BCV<-qt(1-0.20/(2*length(prostate[,1])), length(prostate[,1])-5)

BCV

sum(abs(rstudent(modelo.posible3))>BCV) #Segun el metodo de bonferroni ningun residuo puede considerarse outlier

outlierTest(modelo.posible3)

qqPlot(modelo.posible3) #Nos da un qqplot de los residuos studentizados. Obtenemos el 96 y 39 como observaciones inusuales


#OUTLIERS RESPECTO A LAS COVARIABLES


hii<-hatvalues(modelo.posible3)#Elementos diag
```

```
sort(hii,decreasing=TRUE)
hcv<-2*4/97
sum(hii>hcv)
which(hii>hcv)#32,89 la mayor
#(No todas tienen por que ser influyentes)

dcooks<-cooks.distance(modelo.posible3)
sort(dcooks,decreasing=TRUE)#32
f_teo<-qf(0.5,4,93)
which(dcooks>f_teo)

dffitscv<-2*sqrt(4/97)
dffitsmodel<-dffits(modelo.posible3)
sort(abs(dffitsmodel),decreasing=TRUE)
sum(abs(dffitsmodel)>dffitscv)
which(abs(dffitsmodel)>dffitscv)#32, 39 (las mayores)

dfbetacv<-2/sqrt(97)
dfbetamodel<-dfbeta(modelo.posible3)
head(dfbetamodel)
sum(abs(dfbetamodel[,1])>dfbetacv) #2 observaciones
sum(abs(dfbetamodel[,2])>dfbetacv)
sum(abs(dfbetamodel[,3])>dfbetacv)
sum(abs(dfbetamodel[,4])>dfbetacv)

sort(abs(dfbetamodel[,1]),decreasing = TRUE)
which(abs(dfbetamodel[,1])>dfbetacv)#32,38

#Tras varias pruebas, decidimos eliminar las siguientes:
obs.out3<-c(39,96)
PRS3<-prostate[-obs.out3,]
```

```
modelo.definitivo3<-lm(lpsa ~ lcavol + lweight + svi,data=PRS3)


#COMPARACION y DIAGNOSTICO

kappa(modelo.posible3) #No cambia practicamente

kappa(modelo.definitivo3) #numeros de condicion alrededor de 27 lo que NO indica colinealidad moderada apriori

X3<-model.matrix(modelo.posible3)

p<-ncol(X3)

eigenB3<-eigen(t(X3)%*%X3,only.values = TRUE)$values

lambda.max3<-max(eigenB3)

lambda.min3<-min(eigenB3)

indice.condicion3<-0

for (j in 1:p){indice.condicion3[j]<-lambda.max3/eigenB3[j]

print(indice.condicion3[j])} #para j=4 indice de condicion 997 no superior a 1000. De todos los modelos inspeccionados es el que tiene un indice de condicion mayor mas pequeño

#El modelo definitivo (con la observacion quitada) baja el indice de condicion mayor a 973. Para el resto de j´s son bajos

vif(modelo.posible3)

vif(modelo.definitivo3) #Parecidos en ambos casos no superan la cifra de 1.5 mucho mejor que el modelo.definitivo2


summary(modelo.definitivo3)#pvalores mas pequeños

plot(modelo.definitivo3,which=c(1,2)) #Se ajusta a una normal los errores mucho mejor que el modelo.definitivo2

bptest(modelo.definitivo3)#pvalor 0.32 (mejora)

shapiro.test(resid(modelo.definitivo3))#pvalor 0.76 (da pvalor menor pero sigue siendo bastante aceptable)


summary(modelo.posible3)

plot(modelo.posible3,which=c(1,2))

bptest(modelo.posible3)

shapiro.test(resid(modelo.posible3))
```

#AUNQUE TENIAMOS UN INDICE DE CONDICION ELEVADO (973), EL RESTO ERAN MUCHO MAS PEQUEÑOS Y LOS VIF´S NO SUPERAN EL 1.5 POR LO QUE SI VAMOS A CONSIDERAR EL MODELO: modelo.definitivo3

```
##################################################
##     SIN INTERACCION  DE LA CUALITATIVA       ##
##################################################

#CONSTRUIMOS EL MODELO CON LA CUALITATIVA Y SIN LAS INTERACCIONES

modelo.completo<-lm(lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi, data = prostate)

summary(modelo.completo)

anova(modelo.completo)


#Nombramos las variables


Y<-prostate$lpsa

x1<-prostate$lcavol

x2<-prostate$lweight

x3<-prostate$age

x4<-prostate$lbph

x5<-prostate$lcp

x6<-prostate$gleason

x7<-prostate$pgg45


#CALCULO BETAS MODELO COMPLETO (a mano)


X<- cbind(rep(1,length(x1)),x1,x2,x3,x4,x5,x6,x7,prostate$svi)

Y<- as.matrix(Y,nrow=length(x1))

betahat<-solve(t(X)%*%X)%*%t(X)%*%Y
```

```
beta0hat <-betahat[1,1]

beta1hat <-betahat[2,1]

beta2hat <-betahat[3,1]

beta3hat <-betahat[4,1]

beta4hat <-betahat[5,1]

beta5hat <-betahat[6,1]

beta6hat <-betahat[7,1]

beta7hat <-betahat[8,1]

beta8hat <-betahat[9,1]

betahat<-
c(beta0hat,beta1hat,beta2hat,beta3hat,beta4hat,beta5hat,beta6hat,beta7hat,beta8hat)

betahat



#SEPARACION TRAIN Y TEST

train<-sample(c(TRUE,FALSE), size=nrow(prostate), replace=TRUE, prob=c(0.70,0.30))

train

test<-(!train)

prop.table(table(train))


#CONSTRUCCION DE MODELO SIN INTERACCION METODO BACKWARD FIJANDONOS EN LOS
PVALORES (LO HACEMOS PARA 3 CONJUNTOS DISTINTOS DE TRAIN)

#Cada iteracion actualizo train y hago el backward (a mano) con alfa = 0.20


modelo.completo_train<-lm(lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi,
data = prostate[train,])

summary(modelo.completo_train)


modelo.update1<-update(modelo.completo_train,.~.-age)

summary(modelo.update1)


modelo.update2<-update(modelo.update1,.~.-lbph)
```

```
summary(modelo.update2)

modelo.update3<-update(modelo.update2,.~.-gleason)
summary(modelo.update3)

modelo.update4<-update(modelo.update3,.~.-lcp)
summary(modelo.update4)

modelo.update5<-update(modelo.update4,.~.-pgg45)
summary(modelo.update5)

#El esquema del backward que aparece corresponde al tercer conjunto train que cogimos
#NOTA: Estos son los modelos que obtuvimos para cada conjunto train distinto, al no usar semilla no tiene sentido volver ejecutar esto pues no tenemos los conjuntos train originales
#train 1 -> lpsa ~ lcavol + lweight + svi
pred_1_sin<-predict(modelo.update5, newdata = prostate[test,])
error_pred_1_sin<-sum((prostate$lpsa[test]-pred_1_sin)**2)/length(prostate$lpsa[test])

#train 2 -> lpsa ~ lcavol + lweight + pgg45 + svi
pred_2_sin<-predict(modelo.update4, newdata = prostate[test,])
error_pred_2_sin<-sum((prostate$lpsa[test]-pred_2_sin)**2)/length(prostate$lpsa[test])

#train 3 -> lpsa ~ lcavol + lweight + lcp + pgg45 + svi
pred_3_sin<-predict(modelo.update3, newdata = prostate[test,])
error_pred_3_sin<-sum((prostate$lpsa[test]-pred_3_sin)**2)/length(prostate$lpsa[test])

#COGEMOS EL DE MENOR ERROR PREDICTIVO
which.min(c(error_pred_1_sin, error_pred_2_sin, error_pred_3_sin))
#Obtenemos asi el modelo: lpsa ~ lcavol + lweight + svi que es el ya obtenido con preddict.regsubsets con las interacciones (modelo.posible3)
```

```
#CONSTRUCCION DE MODELO EN BASE A CRITERIOS Cp

#NOTA: al no usar semilla no tiene sentido volver ejecutar esto pues no tenemos los conjuntos train originales

models <-regsubsets(lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi, data = prostate[train,])

summary(models)

plot(models,scale="Cp")


#train 1 -> lpsa ~ lcavol + lweight + lcp + pgg45 + svi

mod1_sin<-lm(lpsa ~ lcavol + lweight + lcp + pgg45 + svi, data = prostate[train,])

pred_1_Cp_sin<-predict(mod1_sin, newdata = prostate[test,])

error_pred_1_Cp_sin<-sum((prostate$lpsa[test]-pred_1_Cp_sin)**2)/
length(prostate$lpsa[test])


#train 2 -> lpsa ~ lcavol + age + lbph + svi

mod2_sin<-lm(lpsa ~ lcavol + age + lbph + svi, data = prostate[train,])

pred_2_Cp_sin<-predict(mod2_sin, newdata = prostate[test,])

error_pred_2_Cp_sin<-sum((prostate$lpsa[test]-pred_2_Cp_sin)**2)/
length(prostate$lpsa[test])


#train 3 -> lpsa ~ lcavol + age + lbph + gleason + svi

mod3_sin<-lm(lpsa ~ lcavol + age + lbph + gleason + svi, data = prostate[train,])

pred_3_Cp_sin<-predict(mod3_sin, newdata = prostate[test,])

error_pred_3_Cp_sin<-sum((prostate$lpsa[test]-pred_3_Cp_sin)**2)/
length(prostate$lpsa[test])


which.min(c(error_pred_1_Cp_sin, error_pred_2_Cp_sin, error_pred_3_Cp_sin))

#Obtenemos asi el modelo: lpsa ~ lcavol + lweight + lcp + pgg45 + svi




#CONSTRUCCION DE MODELO CON ENTRENAMIENTO Y TEST CON REGSUBSETS Y k=n
```

```
predict.regsubsets<-function(object, newdata, id,...){  #Funcion auxiliar
 form<-as.formula(object$call[[2]])
 mat<-model.matrix(form, newdata)
 coefi<-coef(object, id=id)
 xvar<-names(coefi)
 mat[,xvar]%*%coefi
}

n<-nrow(prostate)
k<-n #Número de grupos que voy a hacer pues dejo solo 1 fuera
folds<-sample(x=1:k, size = n, replace=FALSE)
folds
cv.errors<-matrix(NA, k, 8, dimnames=list(NULL, paste(1:8)))#Matriz para albergar los errores
for (j in 1:k){
 best.fit<-regsubsets(lpsa ~ lcavol + lweight + age + lbph + lcp + gleason + pgg45 + svi,
data=prostate[folds!=j,],  method = "exhaustive")
 for (i in 1:8){
   pred<-predict.regsubsets(best.fit, newdata = prostate[folds==j,], id=i)
   cv.errors[j,i]<-mean((prostate$lpsa[folds==j]-pred)^2)
 }
}
cv.errors
mean.cv.errors<-apply(cv.errors, 2, mean) #1 para aplicarlo por filas, 2 para aplicarlo por
columnas
mean.cv.errors
coef(best.fit, which.min(mean.cv.errors))
#Obtenemos lpsa ~ lcavol + lweight + lbph + svi

#MODELOS A CONSIDERAR
modelo.posible4<-lm(lpsa ~ lcavol + lweight + lcp + pgg45 + svi, data = prostate)
modelo.posible5<-lm(lpsa ~ lcavol + lweight + lbph + svi, data = prostate)
```

```
#CONTRASTE DE HIPOTESIS ELIMINACION DE LAS VARIABLES A NO CONSIDERAR

anova(modelo.completo, modelo.posible4) #pvalor 0.16

anova(modelo.completo, modelo.posible5) #pvalolr 0.335

################################################################################
#####



#DETECCION DE OBSERVACIONES RARAS E INFLUYENTES Y DIAGNOSTICO DE modelo.posible5

#OUTLIERS EN Y -> #39, 47

sort(abs(rstandard(modelo.posible5)),decreasing=TRUE)[1:3] #39, 47, 69

BCV<-qt(1-0.20/(2*length(prostate[,1])), length(prostate[,1])-6)

BCV

sum(abs(rstudent(modelo.posible5))>BCV) #Segun el metodo de bonferroni ningun residuo
puede considerarse outlier

outlierTest(modelo.posible5)

qqPlot(modelo.posible5) #39, 47


#OUTLIERS EN COVARIABLES

hii<-hatvalues(modelo.posible5)

hcv<-2*5/97

sum(hii>hcv) #3 observaciones

which(hii>hcv)#32,69,89

#(No todas tienen por que ser influyentes)


dcooks<-cooks.distance(modelo.posible5)

sort(dcooks,decreasing=TRUE)

f_teo<-qf(0.5,5,92)

which(dcooks>f_teo) #Ninguno


dffitscv<-2*sqrt(5/97)
```

```
dffitsmodel<-dffits(modelo.posible5)
sort(abs(dffitsmodel),decreasing=TRUE)
sum(abs(dffitsmodel)>dffitscv) #7 observaciones
which(abs(dffitsmodel)>dffitscv)#1, 32, 39, 47, 69, 95, 96

dfbetacv<-2/sqrt(97)
dfbetamodel<-dfbeta(modelo.posible5)
head(dfbetamodel)
sum(abs(dfbetamodel[,1])>dfbetacv) #3
sum(abs(dfbetamodel[,2])>dfbetacv) #Ninguno
sum(abs(dfbetamodel[,3])>dfbetacv) #Ninguno
sum(abs(dfbetamodel[,4])>dfbetacv) #Ninguno
sum(abs(dfbetamodel[,5])>dfbetacv) #Ninguno

sort(abs(dfbetamodel[,1]),decreasing = TRUE)
which(abs(dfbetamodel[,1])>dfbetacv)#32,38,69

outlierTest(modelo.posible5)
influencePlot(modelo.posible5)#32,39,47,69,89
influence.measures(modelo.posible5)

#Tras varias pruebas, decidimos eliminar las siguientes:

obs.out<-c(39)
PRS5<-prostate[-obs.out,]
modelo.definitivo5<-lm(lpsa~lcavol+lweight+lbph+svi,data=PRS5)

#Comparamos si ha mejorado
#colinealidad
correlacion5<-cor(prostate[-c(3,5,6,7,8)])
correlacion5#No muy excesivas entre las covariables
```

```
X5<-model.matrix(modelo.posible5)
eigenB5<-eigen(t(X5)%*%X5,only.values = TRUE)$values
lambda.max5<-max(eigenB5)
lambda.min5<-min(eigenB5)
numero.condicion5<-sqrt(lambda.max5/lambda.min5)
numero.condicion5#dependencia moderada (35.00 aprox)
t(X5)%*%(X5)
det(t(X5)%*%(X5))

#Indice condicion
indice.condicion5<-0
p<-ncol(X5)
for (j in 1:p){indice.condicion5[j]<-lambda.max5/eigenB5[j]
print(indice.condicion5[j])}#Bastante buenos salvo para j=5 con un valor de 1232.47
#No cambia mucho los indices de condicion con respecto a modelo.definitivo5
vif(modelo.posible5)#Muy buenos, se mueven entre 1 y 2
vif(modelo.definitivo5)#Muy parecidos

summary(modelo.posible5)
summary(modelo.definitivo5)#Mejor R2 ajustado y pvalores mas pequeños
shapiro.test(resid(modelo.posible5))
shapiro.test(resid(modelo.definitivo5))#0.82, mejora
plot(modelo.posible5,which=c(1,2))
plot(modelo.definitivo5,which=c(1,2)) #Cumple muy bien homocedasticidad
bptest(modelo.posible5)
bptest(modelo.definitivo5)#pvalor 0.43 (mejora)

#AL CUMPLIR LAS HIPOTESIS DE LA REGRESION LINEAL MULTIPLE A PESAR DE TENER UN INDICE DE CONDICION ELEVADO CONSIDERAMOS EL MODELO: modelo.definitivo5 YA QUE SU NUMERO DE CONDICION ERA BAJO JUNTO A LOS VIF'S
```

```
##############################################################################
################################
```

#DETECCION DE OBSERVACIONES RARAS E INFLUYENTES Y DIAGNOSTICO DE modelo.posible4

#Al estudiar la colinealidad primero nos dimos cuenta directamente que teniamos que descartar el modelo

#colinealidad

#Correlacion y varianzas

correlacion4<-cor(prostate[-c(3,4,7)]) #correlaciones mayores que 0.6 con lcp~pgg45 y lcp~lcavol puede dar problemas de colinealidad

correlacion4

#Numero de condicion

kappa(modelo.posible4) #458.62 indica colinealidad muy grave

#Indice condicion

X4<-model.matrix(modelo.posible4)

eigenB4<-eigen(t(X4)%*%X4,only.values = TRUE)$values

lambda.max4<-max(eigenB4)

lambda.min4<-min(eigenB4)

indice.condicion4<-0

p<-ncol(X4)

for (j in 1:p){indice.condicion4[j]<-lambda.max4/eigenB4[j]

print(indice.condicion4[j])} #Indices de condicion muy elevados

#VIF

vif(modelo.posible4)

#NO VAMOS A CONSIDERAR EL MODELO modelo.posible4 POR LA PRESENCIA DE COLINEALIDAD MUY GRAVE

```
##############################################################################
#####################################

#MODELOS DEFINITIVOS


modelo.definitivo2

modelo.definitivo3

modelo.definitivo5


#NOTA: NO PODEMOS USAR boxCox en nuestros modelos puesto que para usarlo, las entradas de la variable respuesta deben ser todas positivas y esto no ocurre.

#NOTA: INTENTAMOS HACER ALGUNA TRANSFORMACION EN LAS COVARIABLES DE LOS MODELOS DEFINITIVOS PARA REDUCIR LA COLINEALIDAD PERO NO LLEGAMOS A NADA, AUN ASI, LOS VIF´S EN GENERAL SON BASTANTE PEQUEÑOS Y NO VA A DAR PROBLEMA EN LA ESTIMACION DE LOS B´s


#Vamos a hallar el CV_n de modelo.definitivo2 (validacion cruzada)

n<-nrow(PRS2)

k<-n #Número de grupos que voy a hacer pues dejo solo 1 fuera

folds<-sample(x=1:k, size = n, replace=FALSE)

folds

errors_2<-matrix(NA, k, 1)#Matriz para albergar los errores

for (j in 1:k){

  modelo2<-lm(lpsa ~ lcavol + lweight + lbph + lcavol:svi + lbph:svi + lcp:svi, data=PRS2[folds!=j,])

  pred<-predict(modelo2, newdata = PRS2[folds==j,])

  errors_2[j]<-(PRS2$lpsa[folds==j]-pred)^2

}

errors_2

CV_n_2 #0.5065042
```

```
#Vamos a hallar el CV_n de modelo.definitivo3 (validacion cruzada)
n<-nrow(PRS3)
k<-n #Número de grupos que voy a hacer pues dejo solo 1 fuera
folds<-sample(x=1:k, size = n, replace=FALSE)
folds
errors_3<-matrix(NA, k, 1)#Matriz para albergar los errores
for (j in 1:k){
  modelo3<-lm(lpsa ~ lcavol + lweight + svi, data=PRS3[folds!=j,])
  pred<-predict(modelo3, newdata = PRS3[folds==j,])
  errors_3[j]<-(PRS3$lpsa[folds==j]-pred)^2
}
errors_3
CV_n_3<-mean(errors_3)
CV_n_3 #0.4949762

#Vamos a hallar el CV_n de modelo.definitivo5 (validacion cruzada)
n<-nrow(PRS5)
k<-n #Número de grupos que voy a hacer pues dejo solo 1 fuera
folds<-sample(x=1:k, size = n, replace=FALSE)
folds
errors_5<-matrix(NA, k, 1)#Matriz para albergar los errores
for (j in 1:k){
  modelo5<-lm(lpsa~lcavol+lweight+lbph+svi, data=PRS5[folds!=j,])
  pred<-predict(modelo5, newdata = PRS5[folds==j,])
  errors_5[j]<-(PRS5$lpsa[folds==j]-pred)^2
}
errors_5
CV_n_5<-mean(errors_5)
CV_n_5 #0.5070473
```

#CONCLUIMOS QUE PRACTICAMENTE TODOS TIENEN EL MISMO ERROR DE VALIDACION CRUZADA, ESTRICTAMENTE modelo.definitivo3 ES EL QUE MENOR ERROR DE PREDICCION TIENE

vif(modelo.definitivo2) #Como modelo.definitivo2 no mejora la prediccion notablemente y tiene vifs mas elevados y tiene mas variables descartamos dicho modelo

vif(modelo.definitivo3)

vif(modelo.definitivo5)

summary(modelo.definitivo3)

summary(modelo.definitivo5) #R2 ajustados muy parecidos. Errores estandar de los betas ligeramente superiores en el modelo.definitivo5

plot(modelo.definitivo3, which = c(1,2))

plot(modelo.definitivo5, which = c(1,2)) #Ambos cumplen muy bien las hipotesis de regresion lineal multiple

##

CONCLUSION

##

#COMO MODELO FINAL PROPONEMOS EL MODELO: modelo.definitivo3 <- lm(lpsa ~ lcavol + lweight + svi, data = PRS3) ya que tiene menos variables que el modelo.definitivo5 y este ultimo no aportaba ninguna ventaja con respecto al modelo.definitivo3

summary(modelo.definitivo3)#R2 ajustado 0.632 y errores estandar de los betas

anova(modelo.definitivo3)

#BETAS MODELO FINAL

summary(modelo.definitivo3)$coef[,1]

#Intervalos de estimacion betas para un coeficiente de confianza de familia del

#90 % por los metodos de Scheffe y de Bonferroni

```
#Intervalo de confianza simultaneo por Bonferroni

alpha<-0.1
summary(modelo.definitivo3)$coef
b<-summary(modelo.definitivo3)$coef[2:4,1]
s.b<-summary(modelo.definitivo3)$coef[2:4,2]
g<-3 #Numero de parametros
n<-nrow(PRS3)
p<-4
t_teo<-qt(1-alpha/(2*g),n-p)
BomSimCI<-matrix(c(b-t_teo*s.b,b+t_teo*s.b),ncol=2)
conf<-c("5%","95%")
bnam<-c("LCAVOL","LWEIGHT","SVI")
dimnames(BomSimCI)<-list(bnam,conf)
BomSimCI

#Intervalo de confianza simultaneo por schefee

Q<-p-1
f_teo<-qf(0.9,Q,n-p)
schSimCI<-matrix(c(b-sqrt(Q*f_teo)*s.b,b+sqrt(Q*f_teo)*s.b),ncol=2)
conf<-c("5%","95%")
dimnames(schSimCI)<-list(bnam,conf)
schSimCI
```