PEC 2 REGRESION

Sofía Zdral

17/6/2020

PEC 2 Regresión, modelos y métodos

Ejercicio 1 (50 pt.)

El archivo peru.txt contiene algunas variables posiblemente relacionadas con la presión sanguínea de n = 39 peruanos que se han trasladado de las zonas rurales de gran altitud a las zonas urbanas de menor altitud. Considerar un modelo de regresión múltiple para predecir la presión sistólica Y a partir de las variables:

- X1 = age
- X2 = years in urban area
- X3 = X2/X1 = fraction of life in urban area
- X4 = weight (kg)
- X5 = height (mm)
- X6 = chin skinfold
- X7 = forearm skinfold
- X8 = calf skinfold
- X9 = resting pulse rate

```
#Lo primero que hacemos es cargar nuestros datos
data_peru<- read.delim("C:/Users/Sofia/Downloads/peru.txt")
summary(data_peru) #Y vemos que las variable son las que aparecían en el enunciado. Vemos que tenemos 2 columnas de datos, "Systol" y "Diastol", que aluden a la presión sistólica y diastólica respectivamente.

fraction_urban<-data_peru$Years/data_peru$Age #Creamos la variable que falta, "Fraction of life in urban area" (X3) y la llamaremos fraction_urban y a continuación la añadimos al dataset
data_peru<-as.data.frame(cbind(data_peru, fraction_urban))
head(data_peru) #Vemos que se ha añadido esta última variable
```

(a) Estudiar la posible multicolinealidad de este modelo.

Empezamos creando el modelo de regresión y estudiando la correlación entre variables

```
lm_systol<-
lm(Systol~Age+Years+Weight+Height+Chin+Forearm+Calf+Pulse+fraction_urban, data=data_peru) #Creamos el modelo de regresión con la variable "Systol" como variable respuesta y el resto de variables del conjunto de datos como variables predictoras.
summary(lm_systol) #Aquellas variables predictoras que significativamente
```

```
afectan a la variable respuesta a un nivel de confianza del 95% son: Age,
Years, Weight y la derivada de las dos primeras, fraction_urban.

##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Height + Chin +
##     Forearm + Calf + Pulse + fraction_urban, data = data_peru)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.3442  -6.3972   0.0507   5.7292  14.5257
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     146.81907   48.97096   2.998 0.005526 **
## Age              -1.12144    0.32741  -3.425 0.001855 **
## Years             2.45538    0.81458   3.014 0.005306 **
## Weight            1.41393    0.43097   3.281 0.002697 **
## Height           -0.03464    0.03686  -0.940 0.355194
## Chin             -0.94369    0.74097  -1.274 0.212923
## Forearm          -1.17085    1.19329  -0.981 0.334612
## Calf             -0.15867    0.53716  -0.295 0.769810
## Pulse             0.11455    0.17043   0.672 0.506818
## fraction_urban -115.29395   30.16900  -3.822 0.000648 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.655 on 29 degrees of freedom
## Multiple R-squared:  0.6674, Adjusted R-squared:  0.5641
## F-statistic: 6.465 on 9 and 29 DF,  p-value: 5.241e-05

#Asimismo, el valor de R2 es de 0.67, moderado.

#El siguiente paso es ver las correlaciones entre las variables incluidas
en el modelo
peru_df<-data.frame(data_peru)
cor(peru_df, method=c("pearson","kendall","spearman")) #Si nos fijamos en
los valores de correlación entre las distintas variables, los más altos
(vamos a considerar por encima de 0.6) se dan entre las variables
fraction_urban y Years: 0.938, Calf y Forearm: 0.735, y entre Chin y
Forearm: 0.637.

#Entre las dos primeras es normal que la correlación sea tan alta pues la
primera es una variable derivada de la segunda.
```

Y después miramos los números de condición para ver si hay alguno mayor que 30 y por ende problemas de multicolinealidad.

```
X<-model.matrix(lm_systol)
va<-eigen(t(X) %*% X)$values
sqrt(max(va)/va) #Vemos que sí hay varios números de condición > 30. Por
```

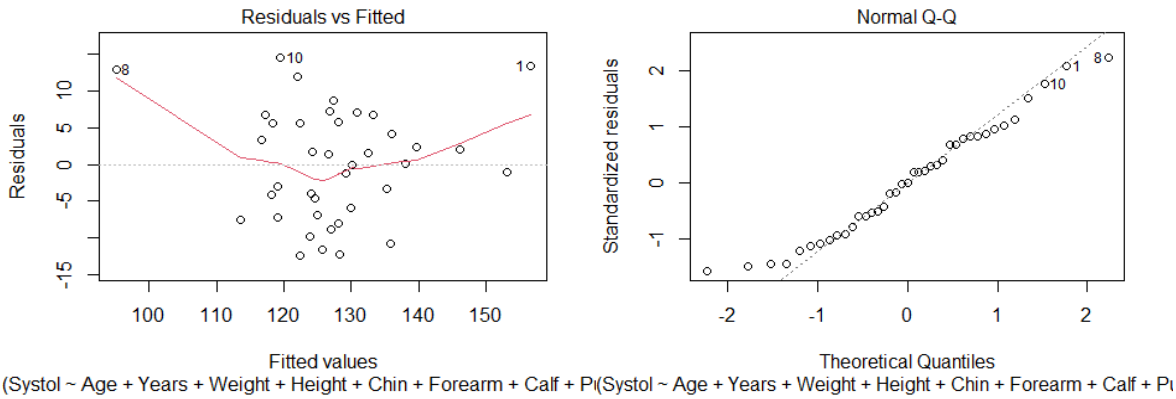
```
tanto, vamos a ver los factores de inflación de la varianza a ver si el
problema de multicolinealidad es grande.

##  [1]     1.0000   127.0065   173.0381   268.2391   292.7051   495.9436
##  [7]   832.5080  1413.8724 34177.3789 56120.3576

library(vctrs)
library(carData)
library(car)

vif(lm_systol) #Vemos que hay factores de inflacción de la varianza que
son bajos como el de Pulse o Heigth pero hay muy altos, el de Years y el
de fraction_urban

##            Age          Years         Weight         Height
Chin
##       3.213372      34.289194       4.747711       1.913991
2.063866
##        Forearm           Calf          Pulse fraction_urban
##       3.802313       2.414602       1.329233      24.387468
```

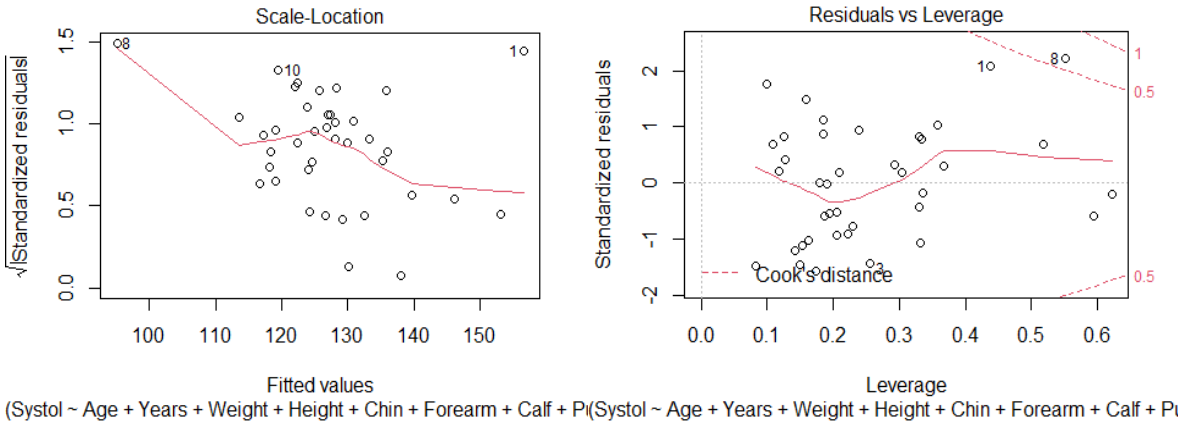
Con todo esto podemos decir que sí existen problemas de multicolinealidad.

(b) Eliminar una única observación de la muestra de forma que el modelo mejore apreciablemente. Razonar la elección.

Para elegir la observación a eliminar primero vamos a representar nuestro modelo

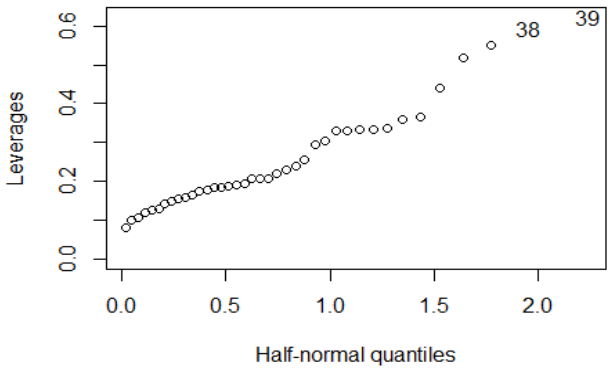
```
plot(lm_systol) #Vemos en los cuatro gráficos que hay dos observaciones
que se alejan mucho del resto de datos, el 1 y el 8. En concreto, la
observación 8, en el gráfico 4 vemos que es el único con una distancia de
Cook por encima de 0.5 de todo el conjunto de datos.
```





En base a lo observado, vamos a realizar algunos estudios adicionales para ver cuál eliminamos finalmente. Así, vamos a estudiar si existen observaciones con un alto Leverage

```
#Lo primero que hacemos es guardar los leverages
hatv.systol <- hatvalues(lm_systol)
head(sort(hatv.systol,decreasing=T))  #Y mostramos las observaciones con mayores leverages

##        39        38         8         5         1         4
## 0.6220632 0.5951000 0.5517533 0.5178689 0.4388073 0.3671580

sum(hatv.systol) #Miramos el sum(leverages), vemos que es 10

#A continuación, realizamos el gráfico "half normal" y pediremos al programa que nos etiquete aquellos valores con leverage más alto.
library(faraway)

peru_leve <- row.names(data_peru)
halfnorm(hatv.systol, labs=peru_leve, ylab="Leverages") #Vemos que hay dos puntos, el 39 y el 38, que se alejan notablemente del resto de los datos.
```

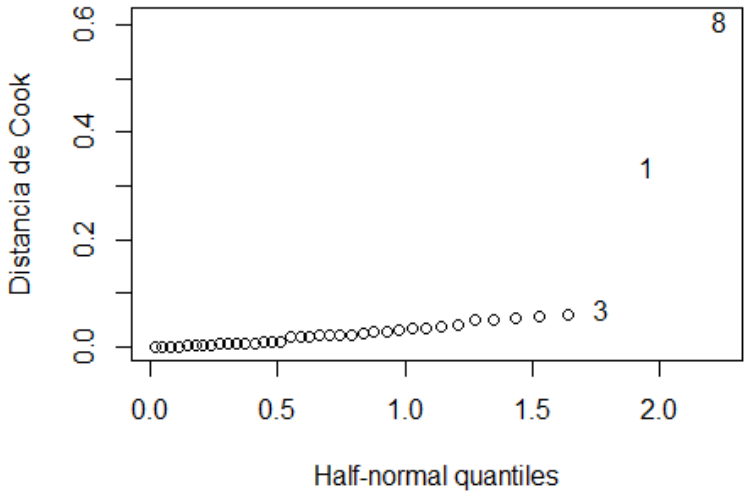


```r
n<-39 #nº observaciones
p<-9 #nº variables predictoras

#Vamos a ver qué observaciones son las que están por encima de la media leverage:
leverage.mean<-p/n
which(hatv.systol > 2*leverage.mean) #Serían la 5,8,38 y 39
```

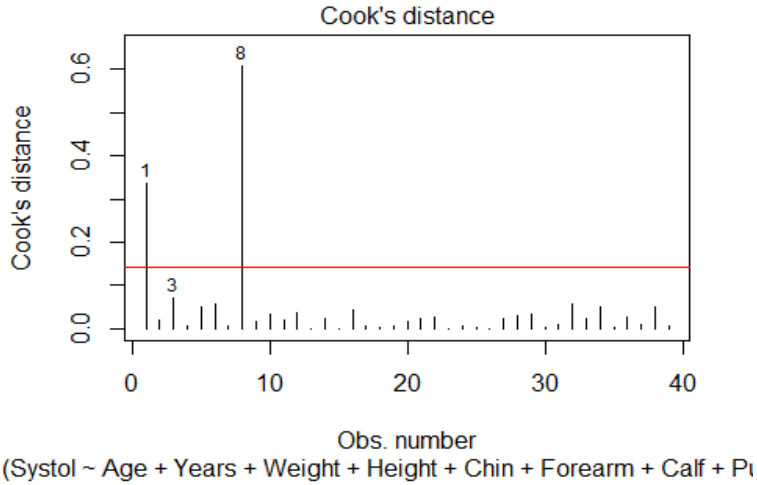
```
##  5  8 38 39
##  5  8 38 39
```

Vamos a retomar la distancia de Cook

```r
#Como vimos en la representación gráfica al inicio del apartado, hay observaciones con distancias de Cook más altas.
cook<-cooks.distance(lm_systol)
halfnorm(cook,nlab=3,ylab="Distancia de Cook")
```




```r
plot(lm_systol, which=4)
abline(h=4/((n-p-2)), col="red") #Destacan la 1 y sobre todo la 8
```



Viendo los resultados obtenidos, tenemos cuatro observaciones candidatas a ser eliminadas debido a la gran influencia que ejercen sobre nuestro modelo: 5,8,38 y 39. En base a que la 8 tiene más distancia de Cook y alto leverage, sería la mejor para eliminar de nuestro modelo. No obstante, vamos a comprobarlo viendo cómo afecta al modelo la eliminación de esta observación versus eliminando las otras tres, cada una por separado.

```
lm_systol_8<-
lm(Systol~Age+Years+Weight+Height+Chin+Forearm+Calf+Pulse+fraction_urban,
data=data_peru[-8,])
summary(lm_systol_8) #Quitando la observación nº 8 hemos mejorado el R2
del modelo a 0.7066

##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Height + Chin +
##     Forearm + Calf + Pulse + fraction_urban, data = data_peru[-8,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1497  -4.1489  -0.2525   5.2688  16.7433
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     166.35727   46.12801   3.606 0.001194 **
## Age              -1.18974    0.30488  -3.902 0.000546 ***
## Years             3.02918    0.79227   3.823 0.000673 ***
## Weight            1.65662    0.41220   4.019 0.000399 ***
## Height           -0.05052    0.03481  -1.451 0.157876
## Chin             -0.94857    0.68696  -1.381 0.178257
## Forearm          -2.38282    1.21649  -1.959 0.060168 .
## Calf              0.38367    0.54705   0.701 0.488874
## Pulse             0.07024    0.15909   0.442 0.662228
## fraction_urban -145.53620   30.68660  -4.743  5.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.025 on 28 degrees of freedom
## Multiple R-squared:  0.7066, Adjusted R-squared:  0.6123
## F-statistic: 7.492 on 9 and 28 DF,  p-value: 1.696e-05

lm_systol_5<-
lm(Systol~Age+Years+Weight+Height+Chin+Forearm+Calf+Pulse+fraction_urban,
data=data_peru[-5,])
summary(lm_systol_5) #Quitando la observación nº 5 hemos mejorado el R2
del modelo a 0.6644

##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Height + Chin +
##     Forearm + Calf + Pulse + fraction_urban, data = data_peru[-5,
```

```
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.7295  -6.4422   0.1116   5.4472  14.6346
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     149.32138   49.57271   3.012  0.00545 **
## Age              -1.09570    0.33269  -3.293  0.00269 **
## Years             2.42121    0.82384   2.939  0.00653 **
## Weight            1.48659    0.44809   3.318  0.00252 **
## Height           -0.03959    0.03792  -1.044  0.30550
## Chin             -0.87208    0.75543  -1.154  0.25809
## Forearm          -1.65676    1.40217  -1.182  0.24732
## Calf             -0.20220    0.54605  -0.370  0.71395
## Pulse             0.14045    0.17625   0.797  0.43221
## fraction_urban -113.61176   30.55590  -3.718  0.00089 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.737 on 28 degrees of freedom
## Multiple R-squared:  0.6644, Adjusted R-squared:  0.5565
## F-statistic: 6.158 on 9 and 28 DF,  p-value: 9.17e-05

lm_systol_38<-
lm(Systol~Age+Years+Weight+Height+Chin+Forearm+Calf+Pulse+fraction_urban,
data=data_peru[-38,])
summary(lm_systol_38) #Quitando la observación nº 38 hemos mejorado el R2
del modelo a 0.6704

##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Height + Chin +
##     Forearm + Calf + Pulse + fraction_urban, data = data_peru[-38,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0084  -5.9756   0.7002   5.5187  14.7876
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     141.71141   50.28240   2.818  0.00876 **
## Age              -1.19496    0.35384  -3.377  0.00217 **
## Years             2.85397    1.06561   2.678  0.01224 *
## Weight            1.29191    0.48251   2.677  0.01227 *
## Height           -0.02529    0.04051  -0.624  0.53752
## Chin             -1.07540    0.78201  -1.375  0.17998
## Forearm          -1.08757    1.21518  -0.895  0.37842
```

```
## Calf             -0.09724    0.55320  -0.176  0.86173
## Pulse             0.10977    0.17257   0.636  0.52988
## fraction_urban -127.58073   36.94707  -3.453  0.00178 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.754 on 28 degrees of freedom
## Multiple R-squared:  0.6704, Adjusted R-squared:  0.5644
## F-statistic: 6.327 on 9 and 28 DF,  p-value: 7.33e-05

lm_systol_39<-
lm(Systol~Age+Years+Weight+Height+Chin+Forearm+Calf+Pulse+fraction_urban,
data=data_peru[-39,])
summary(lm_systol_39) #Quitando la observación nº 39 hemos mejorado el R2
del modelo a 0.6329

##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Height + Chin +
##     Forearm + Calf + Pulse + fraction_urban, data = data_peru[-39,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.5830  -6.7168   0.7005   6.1361  14.6182
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     149.32602   51.42152   2.904 0.007114 **
## Age              -1.12687    0.33413  -3.373 0.002192 **
## Years             2.49444    0.85208   2.927 0.006716 **
## Weight            1.44657    0.46890   3.085 0.004547 **
## Height           -0.03755    0.04034  -0.931 0.359836
## Chin             -0.96430    0.76088  -1.267 0.215469
## Forearm          -1.08829    1.28470  -0.847 0.404113
## Calf             -0.20679    0.59898  -0.345 0.732502
## Pulse             0.12009    0.17562   0.684 0.499715
## fraction_urban -116.74163   31.55940  -3.699 0.000936 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.803 on 28 degrees of freedom
## Multiple R-squared:  0.6329, Adjusted R-squared:  0.515
## F-statistic: 5.365 on 9 and 28 DF,  p-value: 0.0002758

#Queda patente entonces que cuando quitamos la observación 8 es cuando el
valor de R2 de nuestro modelo es más alta, por lo que vamos a quitar esa
observación de nuestros datos.
data_peru_bueno<-data_peru[-8,]
```

(c) Con los 38 datos restantes, hallar el "mejor" modelo consensuado por dos métodos diferentes de selección de variables como, por ejemplo, R2 adj y Cp de Mallows.

Empezamos con el R2 adj como criterio de selección

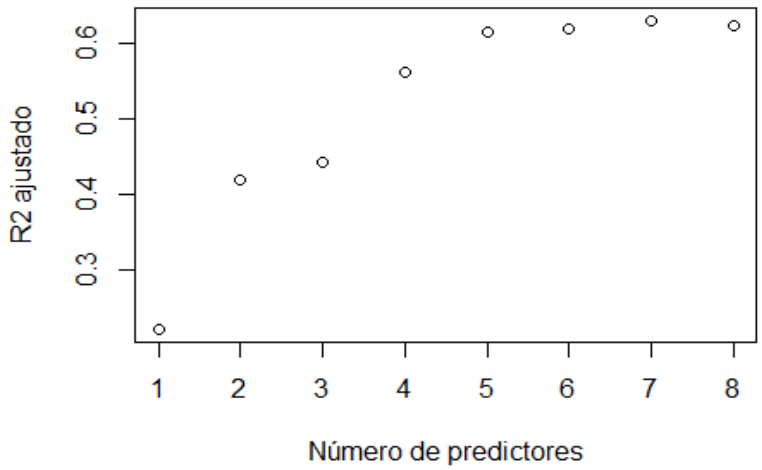
```
library(leaps)
b <-
regsubsets(Systol~Age+Years+Weight+Height+Chin+Forearm+Calf+Pulse+fractio
n_urban, data=data_peru_bueno)
rs <- summary(b)
rs$outmat
```

```
##          Age Years Weight Height Chin Forearm Calf Pulse
fraction_urban
## 1  ( 1 ) " " " "   "*"    " "    " "  " "     " "  " "   " "
## 2  ( 1 ) " " " "   "*"    " "    " "  " "     " "  " "   "*"
## 3  ( 1 ) "*" "*"   " "    " "    " "  " "     " "  " "   "*"
## 4  ( 1 ) "*" "*"   "*"    " "    " "  " "     " "  " "   "*"
## 5  ( 1 ) "*" "*"   "*"    " "    " "  "*"     " "  " "   "*"
## 6  ( 1 ) "*" "*"   "*"    "*"    " "  "*"     " "  " "   "*"
## 7  ( 1 ) "*" "*"   "*"    "*"    "*"  "*"     " "  " "   "*"
## 8  ( 1 ) "*" "*"   "*"    "*"    "*"  "*"     "*"  " "   "*"
```

```
k<-length(rs$rss)
p<-k+1

rs$adjr2
```

```
## [1] 0.2213449 0.4180425 0.4416393 0.5618138 0.6150372 0.6181600
0.6295699
## [8] 0.6230287
```

```
plot(1:k,rs$adjr2, xlab="Número de predictores", ylab="R2 ajustado", axes
= F)
box(); axis(1,at=1:k); axis(2)
```



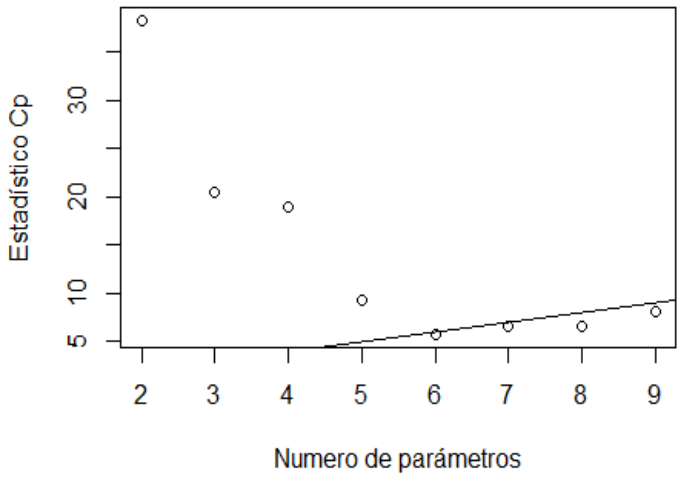
```
# Encontramos el valor óptimo de R2 con 7 predictores, no obstante con cinco predictores es muy parecido.
```

Y probamos ahora con el Cp de Mallows como criterio de selección

```
rs$cp
```

```
## [1] 38.295709 20.532028 18.961941  9.293879  5.771204  6.528675
6.661067
## [8]  8.194940
```

```
plot(2:p,rs$cp, xlab="Numero de parámetros", ylab="Estadístico Cp", axes = F)
box(); axis(1,at=2:p); axis(2)
abline(a=0,b=1) #El mejor valor, el mínimo Cp, se alcanza con 5 predictores (6 parámetros).
```



(i) ¿Cuales son las variables seleccionadas?

```
#Viendo los resultados de R2 y Cp, lo mejor es un modelo con 5 predictores, estos serían:
rs$outmat[5,] #Age, Years, Weight, Forearm y fraction_urban
```

```
##             Age           Years          Weight          Height
Chin
##             "*"             "*"             "*"             " "
" "
##        Forearm            Calf           Pulse fraction_urban
##             "*"             " "             " "             "*"
```

(ii) ¿Cual es el coeficiente de determinación ajustado de este modelo? Compararlo con el del modelo completo.

```
lm_systol_bueno<-
lm(Systol~Age+Years+Weight+Height+Chin+Forearm+Calf+Pulse+fraction_urban,
data=data_peru_bueno)
summary(lm_systol_bueno) #Este sería el modelo completo con todas las
```

```
variables, donde son significativas Age, Years, Weight, Forearm y
fraction_urban, con un p-valor de 0.1 o inferior. El R2 es 0.7066

##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Height + Chin +
##     Forearm + Calf + Pulse + fraction_urban, data = data_peru_bueno)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1497  -4.1489  -0.2525   5.2688  16.7433
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     166.35727   46.12801   3.606 0.001194 **
## Age              -1.18974    0.30488  -3.902 0.000546 ***
## Years             3.02918    0.79227   3.823 0.000673 ***
## Weight            1.65662    0.41220   4.019 0.000399 ***
## Height           -0.05052    0.03481  -1.451 0.157876
## Chin             -0.94857    0.68696  -1.381 0.178257
## Forearm          -2.38282    1.21649  -1.959 0.060168 .
## Calf              0.38367    0.54705   0.701 0.488874
## Pulse             0.07024    0.15909   0.442 0.662228
## fraction_urban -145.53620   30.68660  -4.743  5.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.025 on 28 degrees of freedom
## Multiple R-squared:  0.7066, Adjusted R-squared:  0.6123
## F-statistic: 7.492 on 9 and 28 DF,  p-value: 1.696e-05

lm_systol_bueno_red<-lm(Systol~Age+Years+Weight+Forearm+fraction_urban,
data=data_peru_bueno)
summary(lm_systol_bueno_red) #Y este el modelo con las 5 variables que
elegimos. El R2 es de 0.6671, algo más bajo que el modelo completo pero
el cambio es muy leve.

##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Forearm + fraction_urban,
##     data = data_peru_bueno)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.162  -5.498   0.333   5.539  15.894
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     116.6799    20.0133   5.830 1.78e-06 ***
## Age              -1.2035     0.3024  -3.980 0.000371 ***
## Years             3.2951     0.7724   4.266 0.000165 ***
```

```
## Weight            1.1624     0.2817   4.126 0.000245 ***
## Forearm          -1.7233     0.7307  -2.358 0.024623 *
## fraction_urban -153.2570    30.1142  -5.089 1.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.996 on 32 degrees of freedom
## Multiple R-squared:  0.6671, Adjusted R-squared:  0.615
## F-statistic: 12.82 on 5 and 32 DF,  p-value: 6.991e-07
```

(iii) ¿Se gana en eficiencia con el modelo reducido? Comparar los intervalos de confianza de la estimación del coeficiente de la variable Age.

El siguiente paso sería contrastar si el modelo reducido es intercambiable con el completo. Como vemos, ambos valores de R2 son muy similares, pero habrá que ver con una ANOVA si son equivalentes. La H0 es que todos los parámetros de los predictores que hemos eliminado del modelo son cero.

```
anova(lm_systol_bueno,lm_systol_bueno_red) #Como podemos observar, el p-valor obtenido es de 0.4539, superior a 0.05. La diferencia entre ambos modelos no es significativa. Por simplicidad y comodidad, podemos utilizar el modelo reducido.

## Analysis of Variance Table
##
## Model 1: Systol ~ Age + Years + Weight + Height + Chin + Forearm +
Calf +
##     Pulse + fraction_urban
## Model 2: Systol ~ Age + Years + Weight + Forearm + fraction_urban
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 1803.0
## 2     32 2045.8 -4   -242.84 0.9428 0.4539
```

Y ahora vamos a comparar los I.C. de la estimación del coeficiente de la variable Age.

```
confint(lm_systol_bueno) #Con el modelo completo el I.C. para la variable Age es (-1.8142681, -0.56522175)

##                       2.5 %       97.5 %
## (Intercept)      71.8683281 260.84620905
## Age              -1.8142681  -0.56522175
## Years             1.4062872   4.65207036
## Weight            0.8122712   2.50097753
## Height           -0.1218283   0.02079582
## Chin             -2.3557476   0.45860437
## Forearm          -4.8746857   0.10904896
## Calf             -0.7369032   1.50424248
## Pulse            -0.2556352   0.39611568
## fraction_urban -208.3948497 -82.67754163

confint(lm_systol_bueno_red) #Y con el modelo reducido el I.C. para la variable Age es (-1.8194498  -0.5875311)
```

```
##                        2.5 %      97.5 %
## (Intercept)      75.9141964 157.4455178
## Age              -1.8194498  -0.5875311
## Years             1.7218386   4.8684563
## Weight            0.5885264   1.7361884
## Forearm          -3.2116541  -0.2349512
## fraction_urban -214.5975248 -91.9164499

#Son extremadamente similares.
```

(d) Los investigadores sugieren adoptar el modelo reducido que contenga únicamente las variables significativas (α = 0.1) con el test t en sustitución del modelo completo con las 9 variables explicativas.¿Es ese un buen criterio de selección? Realizar un test adecuado que resuelva su sugerencia. Discutir el resultado en consonancia con los resultados obtenidos en el apartado anterior.

Si nos fijamos en el apartado anterior, cuando elegimos el modelo reducido en base a los criterios de R2 ajustado y Cp de Mallows, nos quedamos solamente con aquellas variables que justo eran significativas a un valor de α = 0.1 o inferior. Con el test de ANOVA vimos que ambos modelos pueden usarse sin que se vea significativamente afectado la bondad del modelo para explicar la presión sistólica en la población estudiada. Por eso, sí sería un buen criterio de selección considerar incluir aquellas variables por encima de 0.1.

(e) Comprobar si hemos solucionado el problema de multicolinealidad en el modelo reducido del apartado anterior.

Vamos a repetir lo mismo que hicimos antes para estudiar si existe multicolinealidad en el modelo reducido.

```
summary(lm_systol_bueno_red) #Recordamos cómo era el modelo reducido

##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Forearm + fraction_urban,
##     data = data_peru_bueno)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.162  -5.498   0.333   5.539  15.894
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     116.6799    20.0133   5.830 1.78e-06 ***
## Age              -1.2035     0.3024  -3.980 0.000371 ***
## Years             3.2951     0.7724   4.266 0.000165 ***
## Weight            1.1624     0.2817   4.126 0.000245 ***
## Forearm          -1.7233     0.7307  -2.358 0.024623 *
## fraction_urban -153.2570    30.1142  -5.089 1.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.996 on 32 degrees of freedom
## Multiple R-squared:  0.6671, Adjusted R-squared:  0.615
## F-statistic: 12.82 on 5 and 32 DF,  p-value: 6.991e-07
```

```
#Estudiamos los números de condición
X2<-model.matrix(lm_systol_bueno_red)
va2<-eigen(t(X2) %*% X2)$values
sqrt(max(va2)/va2)  #Vemos que sí hay tres números de condición > 30. Por ello, vamos a ver los factores de inflación de la varianza a ver si el problema de multicolinealidad es grande.
```

```
## [1]    1.000000    8.240483   14.034478   39.021891  787.431012
1953.077392
```

```
#Factores de inflacción de la varianza
vif(lm_systol_bueno_red) #Vemos que seguimos teniendo valores muy altos como los de Years y fraction_urban.
```

```
##            Age          Years         Weight        Forearm
fraction_urban
##       3.105001      35.118905       2.245658       1.665539
24.738540
```

```
#Aunque algo ha mejorado, seguimos teniendo problemas de multicolinealidad también en el modelo reducido.
```

Como los investigadores no quieren prescindir de más variables, se plantea una regresión Partial Least Squares (PLS). ¿Cuantas componentes se necesitan para minimizar el RMSEP? Calcular los coeficientes de las variables originales, también para β0, que proporciona este método con el número de componentes necesario.

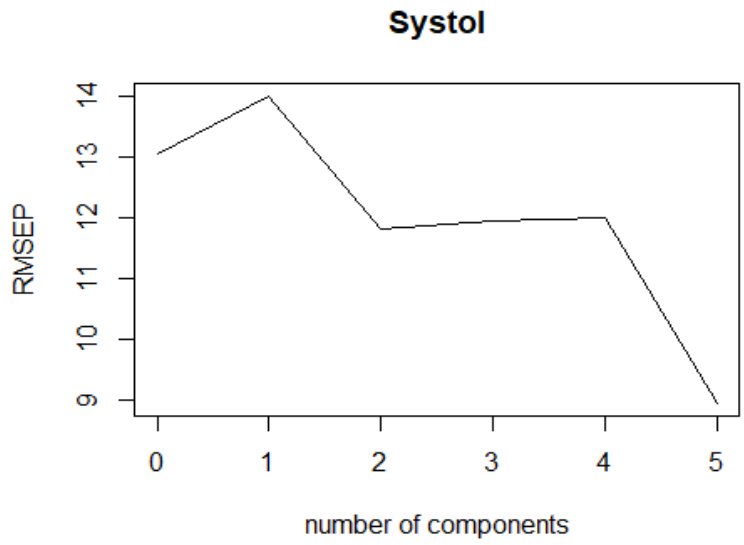
Dado que sigue existiendo el problema de multicolinealidad, vamos a ver cuántos componentes del análisis de componentes principales son necesarios para minimizar el RMSE

```
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
set.seed(222)
pls_peru <- plsr(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = data_peru_bueno, validation = "CV")
peruCV<-RMSEP(pls_peru, estimate="CV")
plot(peruCV)
```




```
which.min(peruCV$val) #Obtenemos que el número mínimo de componentes es de 6, pero realmente serían 5 ya que el primer valor es para el intercept (0).
```

Calculamos los coeficientes

```
pls_peru_5<-plsr(Systol ~ Age + Years + Weight + Forearm + fraction_urban, data = data_peru_bueno, ncomp = 5)
coef(pls_peru_5, intercept = TRUE) #Aquí tenemos los coef. de las variables originales y del intercept (60)
```

```
## , , 5 comps
##
##                      Systol
## (Intercept)     116.679857
## Age              -1.203490
## Years             3.295147
## Weight            1.162357
## Forearm          -1.723303
## fraction_urban -153.256987
```

¿Es adecuado este método de regresión con estas variables? ¿Es útil?

```
summary(lm_systol_bueno_red) #Vemos que los coeficientes obtenidos por PLS arriba respecto al modelo anterior reducido (que incluía las variables con alfa ≤ 0.01) son prácticamente iguales.
```

```
##
## Call:
## lm(formula = Systol ~ Age + Years + Weight + Forearm + fraction_urban,
##     data = data_peru_bueno)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.162  -5.498   0.333   5.539  15.894
```

```
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     116.6799    20.0133   5.830 1.78e-06 ***
## Age              -1.2035     0.3024  -3.980 0.000371 ***
## Years             3.2951     0.7724   4.266 0.000165 ***
## Weight            1.1624     0.2817   4.126 0.000245 ***
## Forearm          -1.7233     0.7307  -2.358 0.024623 *
## fraction_urban -153.2570    30.1142  -5.089 1.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.996 on 32 degrees of freedom
## Multiple R-squared:  0.6671, Adjusted R-squared:  0.615
## F-statistic: 12.82 on 5 and 32 DF,  p-value: 6.991e-07
```

#Además, tenemos el mismo número de variables, no se han reducido, siguen siendo 5. No veo mucha utilidad a este modelo teniendo el anterior, ni mucha mejora al usarlo. También seguimos teniendo el problema de la multicolinealidad.

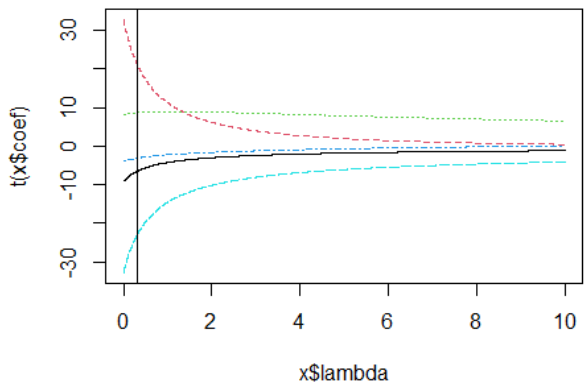
(f) Siguiendo con el modelo reducido, otra posibilidad es utilizar la Ridge Regression. ¿Cuales son los coeficientes obtenidos? Explicar brevemente las ventajas e inconvenientes de este método frente a la selección de variables.

Vamos a utilizar la Ridge Regression y ver las ventajas e inconvenientes

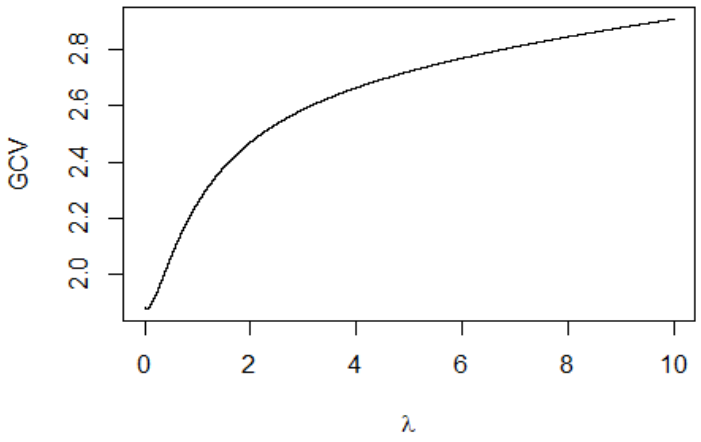
```
library(MASS)
ridge_peru <- lm.ridge(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = data_peru_bueno, lambda = seq(0, 10, 0.0001))
lambda <- which.min(ridge_peru$GCV)
select(ridge_peru)
```

```
## modified HKB estimator is 0.08281436
## modified L-W estimator is 1.778106
## smallest value of GCV  at 0.0331
```

```
plot(ridge_peru)
abline(v=0.3)
```



```
plot(ridge_peru$lambda,ridge_peru$GCV,type="l",xlab=expression(lambda),yl
ab="GCV")
abline(v=lambda,col=2)
```




```
#Vemos que el valor más pequeño de GCV es 0.0331, por lo que
estableceremos λ=0.03 como valor de lambda óptimo.

set.seed(2222)
ridge_peru_003 <- lm.ridge(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = data_peru_bueno, lambda = 0.03)

coef(ridge_peru_003, intercept = TRUE) #Para λ=0.03 estos son los
coeficientes. Vemos que se han reducido levemente respecto al modelo con
las 5 variables que tienen un alfa ≤ 0.01.

##                            Age          Years         Weight
Forearm
##     113.772709      -1.156289       3.132749       1.176816      -
1.674308
## fraction_urban
##    -146.940408
```

ventajas e inconvenientes de este modelo Ridge Regression frente a la selección de variables:

- Ventajas: permite mejorar los errores de predicción al reducir el valor de los coeficientes, tal y como hemos observado arriba. Así, evitamos que sean muy grandes y reducimos el sobreajuste.
- Inconvenientes: no realiza selección de variables, seguimos teniendo los mismo 5 predictores (no elimina las menos influyentes).

Calcular el RMSE de la regresión OLS, PLS (con 5, 4, 3 y 2 componentes) y Ridge (con λ óptima por GCV) para el modelo reducido.

Ahora calculamos el error cuadrático medio (RMSE) de la regresión Ordinary Least Squares (OLS)

```
RSS.ols<-c(crossprod(lm_systol_bueno_red$residuals))
MSE.ols<-RSS.ols/length(lm_systol_bueno_red$residuals)
RMSE.ols<-sqrt(MSE.ols)
RMSE.ols #Sería este valor, 7.337436
```

```
## [1] 7.337436
```

Y a continuación el el error cuadrático medio (RMSE) para los PLS con distintos números de componentes

```
pls_peru_5<-plsr(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = data_peru_bueno, ncomp = 5)
pls_peru_5$coefficients
```

```
## , , 1 comps
##
##                     Systol
## Age             -0.10339839
## Years           -0.16490689
## Weight           1.14904659
## Forearm          0.20717441
## fraction_urban -0.01507042
##
## , , 2 comps
##
##                     Systol
## Age             -0.24195873
## Years           -0.42894308
## Weight           1.29914221
## Forearm          0.22028353
## fraction_urban -0.02477824
##
##
```

```
 , , 3 comps
##
##                     Systol
## Age             -0.08156508
## Years           -0.54508847
## Weight           1.35653895
## Forearm          0.03786032
## fraction_urban -0.04613225
##
## , , 4 comps
##
##                      Systol
## Age             -0.1841884
## Years           -0.5176248
## Weight           1.5122684
## Forearm         -0.6366994
## fraction_urban -0.1211482
##
## , , 5 comps
##
##                       Systol
## Age               -1.203490
## Years              3.295147
## Weight             1.162357
## Forearm           -1.723303
## fraction_urban -153.256987
```

```
RSS.pls5<-c(crossprod(pls_peru_5$residuals))
MSE.pls5<-RSS.pls5/length(pls_peru_5$residuals)
RMSE.pls5<-sqrt(MSE.pls5)
RMSE.pls5 #Para 5 componentes valdría 9.586971
```

```
## [1] 9.586971
```

```
pls_peru_4<-plsr(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = data_peru_bueno, ncomp = 4)
pls_peru_4$coefficients
```

```
## , , 1 comps
##
##                      Systol
## Age             -0.10339839
## Years           -0.16490689
## Weight           1.14904659
## Forearm          0.20717441
## fraction_urban -0.01507042
##
## , , 2 comps
##
##                      Systol
## Age             -0.24195873
## Years           -0.42894308
## Weight           1.29914221
## Forearm          0.22028353
## fraction_urban -0.02477824
##
## , , 3 comps
##
##                      Systol
## Age             -0.08156508
## Years           -0.54508847
## Weight           1.35653895
## Forearm          0.03786032
## fraction_urban -0.04613225
##
## , , 4 comps
##
##                       Systol
## Age             -0.1841884
## Years           -0.5176248
## Weight           1.5122684
## Forearm         -0.6366994
## fraction_urban -0.1211482
```

```
RSS.pls4<-c(crossprod(pls_peru_4$residuals))
MSE.pls4<-RSS.pls4/length(pls_peru_4$residuals)
RMSE.pls4<-sqrt(MSE.pls4)
RMSE.pls4 #Para 4 componentes valdría 10.07115
```

```
## [1] 10.07115
```

```
pls_peru_3<-plsr(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = data_peru_bueno, ncomp = 3)
pls_peru_3$coefficients
```

```
## , , 1 comps
##
##                      Systol
## Age             -0.10339839
## Years           -0.16490689
## Weight           1.14904659
## Forearm          0.20717441
## fraction_urban -0.01507042
##
## , , 2 comps
##
##                      Systol
## Age             -0.24195873
## Years           -0.42894308
## Weight           1.29914221
## Forearm          0.22028353
## fraction_urban -0.02477824
##
## , , 3 comps
##
##                      Systol
## Age             -0.08156508
## Years           -0.54508847
## Weight           1.35653895
## Forearm          0.03786032
## fraction_urban -0.04613225
```

```
RSS.pls3<-c(crossprod(pls_peru_3$residuals))
MSE.pls3<-RSS.pls3/length(pls_peru_3$residuals)
RMSE.pls3<-sqrt(MSE.pls3)
RMSE.pls3 #Para 3 componentes valdría 10.1385
```

```
## [1] 10.1385
```

```
pls_peru_2<-plsr(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = data_peru_bueno, ncomp = 2)
pls_peru_2$coefficients
```

```
## , , 1 comps
##
##                     Systol
## Age            -0.10339839
## Years          -0.16490689
## Weight          1.14904659
## Forearm         0.20717441
## fraction_urban -0.01507042
##
```

```
## , , 2 comps
##
##                     Systol
## Age            -0.24195873
## Years          -0.42894308
## Weight          1.29914221
## Forearm         0.22028353
## fraction_urban -0.02477824
```

```
RSS.pls2<-c(crossprod(pls_peru_2$residuals))
MSE.pls2<-RSS.pls2/length(pls_peru_2$residuals)
RMSE.pls2<-sqrt(MSE.pls2)
RMSE.pls2 #Para 2 componentes valdría 10.22934
```

```
## [1] 10.22934
```

```
#De los 4 el valor más bajo del RMSE es cuando tenemos 5 componentes, y
va aumentando conforme reducimos el nº de componentes
```

Y el RMSE para Ridge

```
library(lmridge)

vif

ridge_1<-lmridge(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = data_peru_bueno, K= 0.03)
res_ridge_1<-residuals.lmridge(ridge_1) #Sacamos los residuos
RSS.rid<-c(crossprod(res_ridge_1))
MSE.rid<-RSS.rid/length(res_ridge_1)
RMSE.rid<-sqrt(MSE.rid)
RMSE.rid #En el caso de la regresión Ridge vemos que el RMSE vale
8.309424
```

```
## [1] 8.309424
```

¿Cual es la valoración con todo lo que sabemos hasta ahora?

Como dicen en esta web, https://acolita.com/que-es-el-error-cuadratico-medio-rmse/, "Cuanto más pequeño es un valor RMSE, más cercanos son los valores predichos y observados". Así, el valor más pequeño de RMSE lo tenemos con el modelo OLS, seguido de la Ridge Regression, y este de el PLS pero con 5 variables. Una vez bajamos de 5 empieza a subir el error RMSE, por lo que dejamos de perder la calidad que habiamos conseguido en el modelo.

(g) Sabemos que el RMSE calculado en un modelo para todos los datos observados es muy optimista. Es mejor un cálculo por validación cruzada.

Con el modelo reducido de los apartados anteriores y para comparar los métodos estudiados (OLS, PLS (con 4 componentes) y Ridge (con λ óptimo por GCV) haremos lo siguiente:

1. Dividiremos los datos aleatoriamente en dos grupos, uno de 8 observaciones (grupo test) y otro del resto (grupo train). Recordemos que el número total de observaciones es ahora de 38.

```
peru_test<-data_peru_bueno[1:8, ]
peru_train<-data_peru_bueno[9:38, ]
```

2. Ajustaremos cada modelo con el grupo train y calcularemos el RMSE con el grupo test.

```
library(Metrics)
set.seed(2222)
#OLS
OLS_train<-lm(Systol ~ Age + Years + Weight + Forearm + fraction_urban,
data = peru_train)
pred_ols<-predict(OLS_train, data = peru_test)
rmse(actual= peru_test$Systol, predicted = pred_ols) #El RMSE obtenido en
el modelo OLS fue de 23.7021

## Warning in actual - predicted: longitud de objeto mayor no es múltiplo
de la
## longitud de uno menor

## [1] 23.70244

#PLS con n = 4 componentes
pls_train_4<-plsr(Systol ~ Age + Years + Weight + Forearm +
fraction_urban, data = peru_train)
pred_pls<-predict(pls_train_4, peru_test)
rmse(actual = peru_test$Systol, predicted = pred_pls) #El RMSE obtenido
en el modelo PLS fue de 18.131

## [1] 18.13126
```

3. Repetiremos los pasos 1 y 2 mil veces.

set.seed(2222)

1000 veces para OLS

```
ols_rep<- rep(NA, 1000)

for (i in 1:1000){

        peru_train_1000<-sample(x = 1:38, 30)

        OLS_train<-lm(Systol ~ Age + Years + Weight + Forearm + fraction_urban,
data=peru_train)

        pred_ols<-predict(OLS_train, data = peru_test) #####ols_rep[i]<-rmse(actual=
        peru_test$Systol, predicted = pred_ols)}
```

1000 veces para PLS

```
pls_rep<- rep(NA, 1000)

for (i in 1:1000){

        peru_train_1000<-sample(x = 1:38, 30)

        PLS_train<-plsr(Systol ~ Age + Years + Weight + Forearm + fraction_urban,
data=peru_train)

        pred_pls<-predict(PLS_train, data = peru_test)

        pls_rep[i]<-rmse(actual= peru_test$Systol, predicted = pred_pls)}
```

Ejercicio 2 (30 pt.)

En el trabajo de Cameron and Pauling[1] se presenta un estudio de los tiempos de supervivencia de 100 pacientes de cáncer terminal a los que se les administró un suplemento de ascorbato de sodio, vitamina C, como parte de su tratamiento rutinario y 1000 controles emparejados, pacientes similares que habían recibido el mismo tratamiento excepto por el ascorbato. El objetivo de la investigación fue determinar si el ascorbato de sodio suplementario prolongaba los tiempos de supervivencia de los pacientes con cáncer humano terminal.

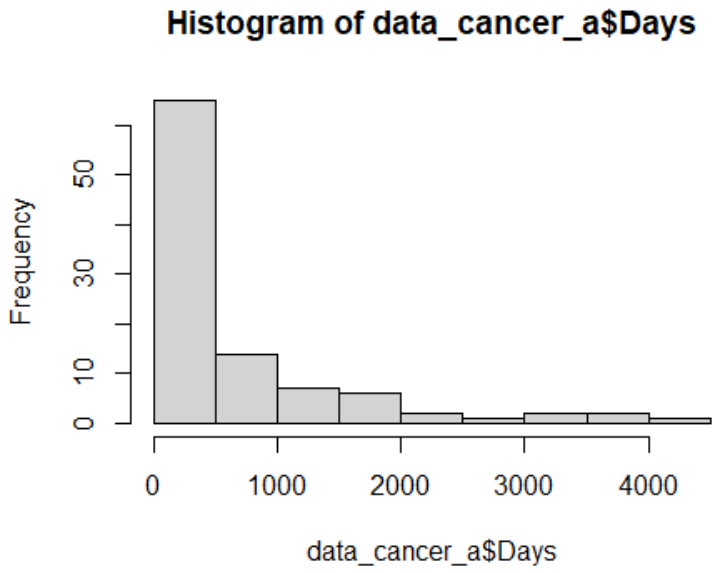
Los datos se hallan en el archivo Table 33.1 de la página https://www2.stat.duke.edu/courses/Spring01/sta114/data/andrews.html. En el archivo descargado observaremos los 100 casos de la Tabla 1 del trabajo de Cameron and Pauling[1]. Las columnas de este archivo, a parte de las tres primeras, son las mismas que en la Tabla 1 del artículo. Falta añadir el tipo de cáncer y eliminar el símbolo + que indica una supervivencia superior al final del periodo de estudio. En la tabla 1 se ven los datos del trabajo de Cameron and Pauling[1] sólo para tres tipos de cáncer: de estómago, de bronquios y de colon. Las variables en esta tabla se corresponden con la Tabla 1 de Cameron and Pauling así: Age = Age, Days = C, Cont. = D.

```
# Lo primero que hacemos es cargar los datos
data_cancer <- read.table("C:/Users/Sofia/Downloads/T33.1", quote="\"",
comment.char="")

library(stringr)
# Eliminamos los + de aquellos datos que presentan este símbolo
data_cancer$V7<-as.numeric(str_remove(data_cancer$V7,"[+]"))
data_cancer$V9<-as.numeric(str_remove(data_cancer$V7,"[+]"))

data_cancer<-data_cancer[1:47,] #Nos quedamos con los primeros 47
pacientes que son los que corresponden a los tipos de cáncer de interés
data_cancer$Tumor_type<-
c(rep("Stomach",13),rep("Bronchus",17),rep("Colon",17)) #Y a continuación
añadimos el tipo de cancer como la variable "tumor type" al igual que
aparece en el estudio original.

names(data_cancer) =
c("1","2","3","ID","Sex","Age","Survival_asc","survival_crtl","Days","Con
t","Tumor_type") #Y finalmente renombramos las columnas de acuerdo al
artículo

summary(data_cancer)
```

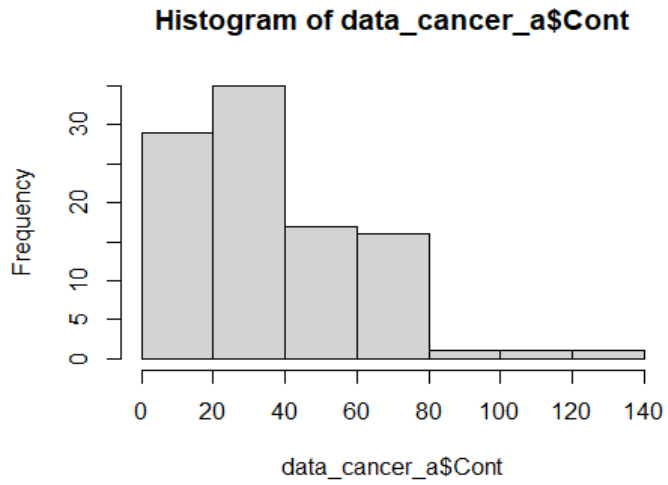
(a) Estudiar la transformación que mejora la distribución de los datos C y los datos D (100 observaciones en cada caso). Se puede utilizar el método de Box-Cox. Una vez transformados, comparar si el tiempo de supervivencia C es superior al de los controles D con todas las observaciones.

Realizamos la transformación logarítmica de los datos

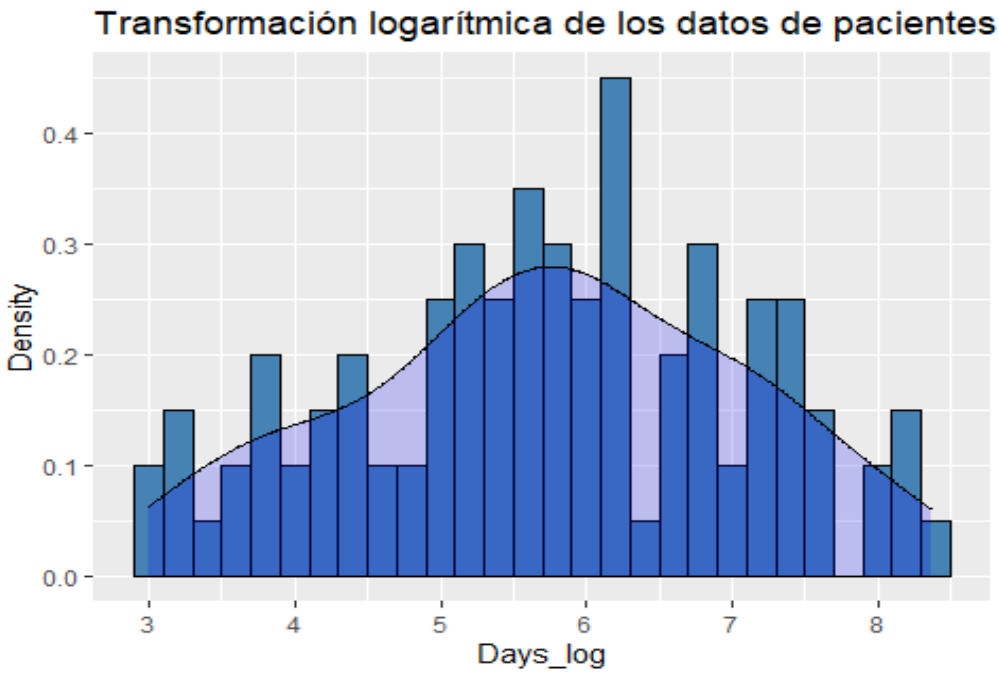
```
#Aunque depuré la tabla al principio, como necesitamos aquí los 100 datos iniciales, cargo los mismos datos pero como un nuevo dataset, "data_cancer_a"
data_cancer_a <- read.table("C:/Users/Sofia/Downloads/T33.1", quote="\"", comment.char="")

#Eliminamos los + de aquellos datos que presentan este símbolo
data_cancer_a$V7<-as.numeric(str_remove(data_cancer_a$V7,"[+]"))
data_cancer_a$V9<-as.numeric(str_remove(data_cancer_a$V7,"[+]"))

names(data_cancer_a) =
c("1","2","3","ID","Sex","Age","Survival_asc","survival_crtl","Days","Cont") #Y enombramos las columnas de acuerdo al artículo

#Antes de transformar los datos, vemos qué apariencia tenían al principio. Para ello representaremos en un histograma los datos de los dos tipos de pacientes

library(ggplot2)
hist(data_cancer_a$Days) #Pacientes con vitamina C
```



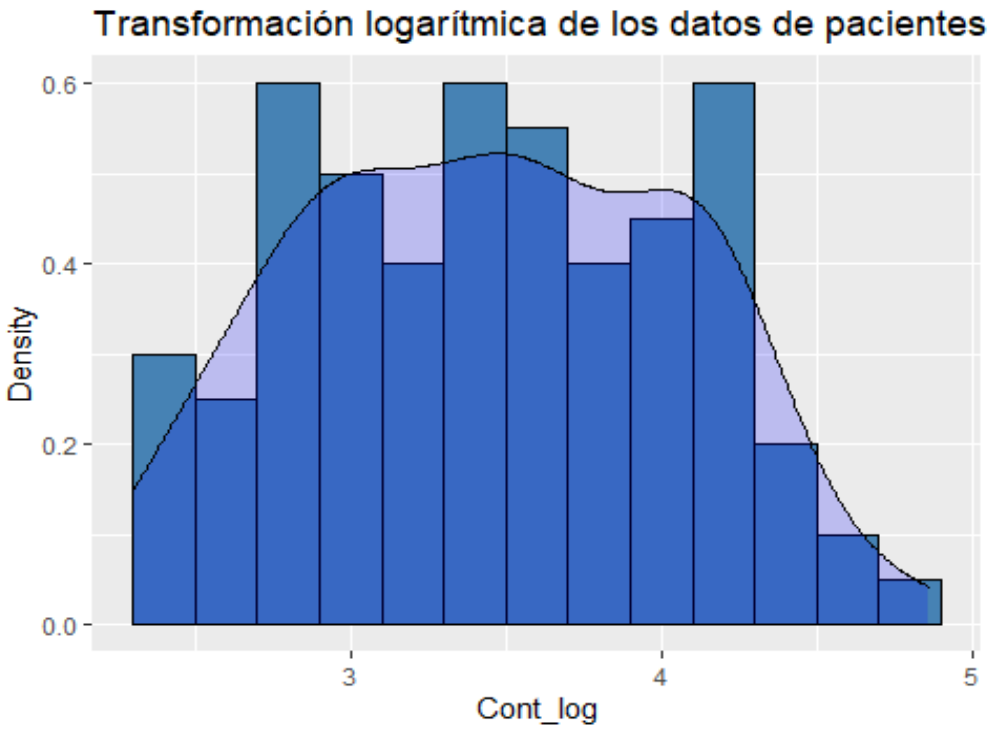

```
hist(data_cancer_a$Cont) #Pacientes Control
```




```
#Vemos que ninguno de los dos sigue una distribución normal.

#Vamos a realizar una transformación logarítmica de los datos, a ver cómo
se distribuyen al aplicar este tipo de transformación
data_cancer_a$Days_log <- log(data_cancer_a$Days)
density <- ggplot(data=data_cancer_a, aes(x=Days_log))
density + geom_histogram(binwidth=0.2, color="black", fill="steelblue",
aes(y=..density..)) + geom_density(stat="density", alpha=I(0.2),
fill="blue") + xlab("Days_log") + ylab("Density") +
ggtitle("Transformación logarítmica de los datos de pacientes tratados
con Vit. C") #Pacientes con vitamina C
```



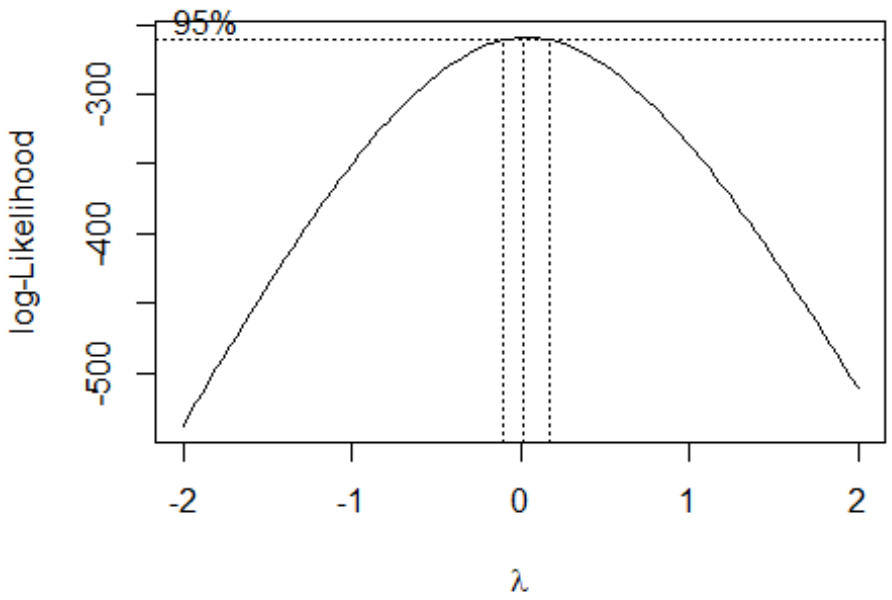
```r
data_cancer_a$Cont_log <- log(data_cancer_a$Cont)
density <- ggplot(data=data_cancer_a, aes(x=Cont_log))
density + geom_histogram(binwidth=0.2, color="black", fill="steelblue",
aes(y=..density..)) + geom_density(stat="density", alpha=I(0.2),
fill="blue") + xlab("Cont_log") + ylab("Density") +
ggtitle("Transformación logarítmica de los datos de pacientes Control")
```



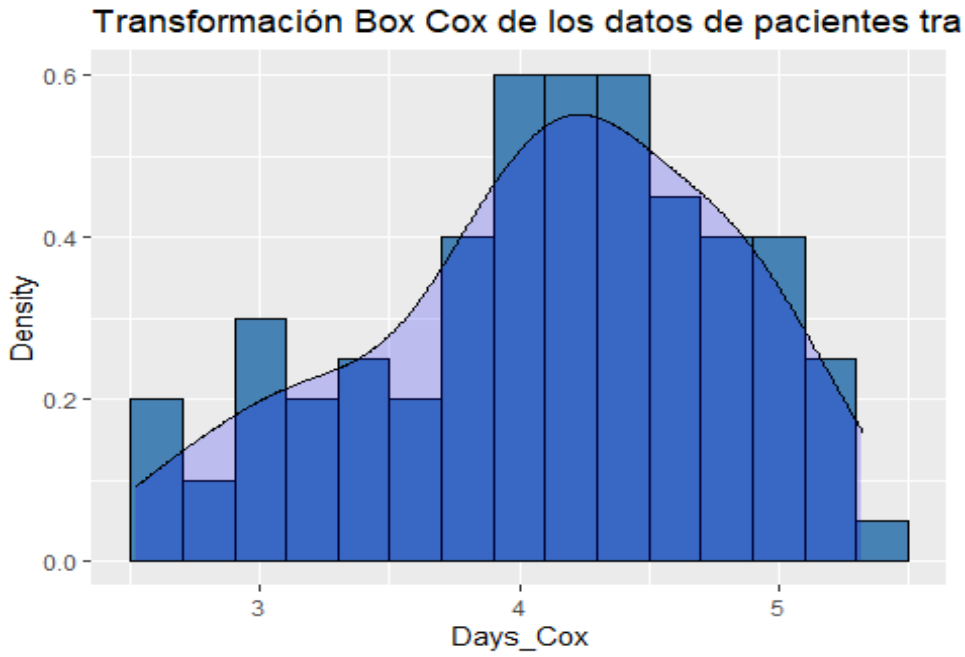

```r
#En el caso de realizar una transformación logarítmica, sí que empiezan a
ajustarse más la distribución de ambos datos a una distribución normal.
```

Y la transformación Box Cox

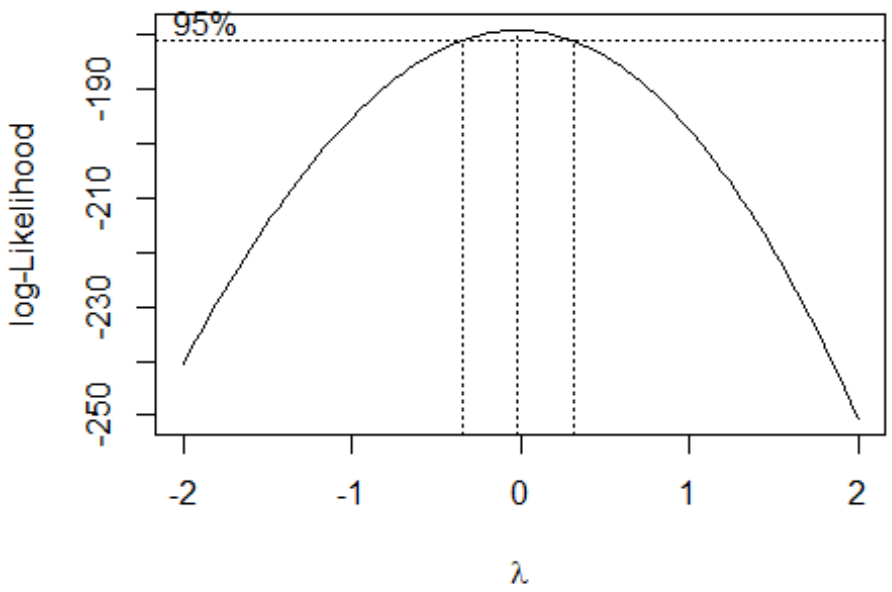
```r
#No obstante, vamos a probar con el método de Box-Cox para ver si mejora
la distribución de los datos. Siguiendo las recomendaciones encontradas
en esta entrada de Minitab (https://support.minitab.com/es-
mx/minitab/19/help-and-how-to/quality-and-process-improvement/quality-
tools/how-to/individual-distribution-identification/perform-the-
analysis/specify-a-box-cox-transformation/), "En la mayoría de los casos,
no se debe usar un valor fuera del rango de -2 y 2", indicaremos en ambos
casos que lambda valga entre esos dos valores.

library(MASS)
library(forecast)

#Primero para los pacientes tratados con Vit. C
boxcox(Days ~ 1, lambda = -2:2, data = data_cancer_a)
```



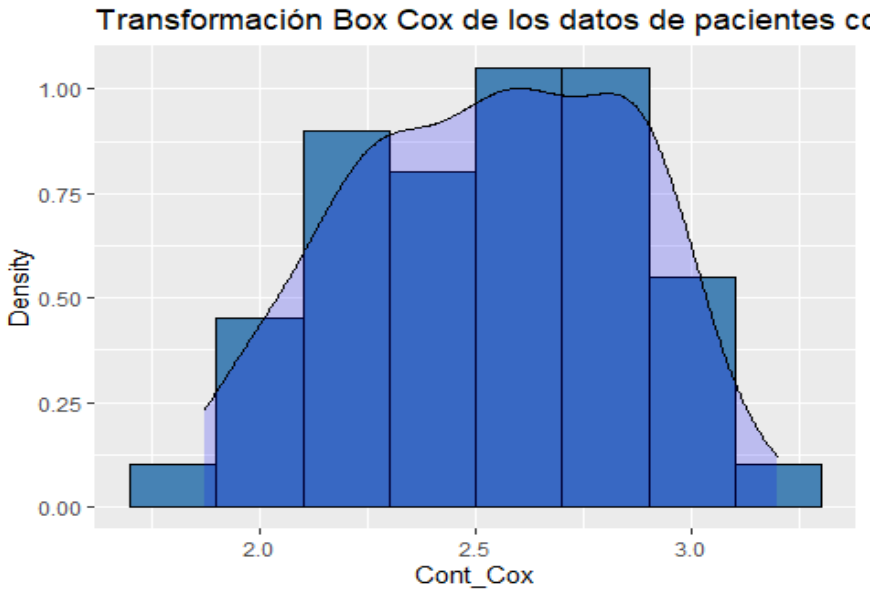

```
lambda_days<- BoxCox.lambda(data_cancer_a$Days)
cox_days<- BoxCox(data_cancer_a$Days, lambda_days)

density <- ggplot(data=data_cancer_a, aes(x=cox_days))
density + geom_histogram(binwidth=0.2, color="black", fill="steelblue",
aes(y=..density..)) + geom_density(stat="density", alpha=I(0.2),
fill="blue") + xlab("Days_Cox") + ylab("Density") +
ggtitle("Transformación Box Cox de los datos de pacientes tratados con
Vit. C") #Pacientes con vitamina C
```



```r
#Y después los control
boxcox(Cont ~ 1, lambda = -2:2, data = data_cancer_a)
```




```r
lambda_cont<- BoxCox.lambda(data_cancer_a$Cont)
cox_cont<- BoxCox(data_cancer_a$Cont, lambda_cont)

density <- ggplot(data=data_cancer_a, aes(x=cox_cont))
density + geom_histogram(binwidth=0.2, color="black", fill="steelblue",
aes(y=..density..)) + geom_density(stat="density", alpha=I(0.2),
fill="blue") + xlab("Cont_Cox") + ylab("Density") +
ggtitle("Transformación Box Cox de los datos de pacientes control")
#Pacientes control
```



Viendo los resultados, nuestros datos se ajustan mejor a una distribución normal tras aplicar una transformación Box Cox que en caso de una transformación logarítmica.

Ahora, vamos a comparar si el tiempo de supervivencia es superior al de los controles. Para ello vamos a realizar un test T de Student. Lo vamos a realizar con los datos transformados.

```
t.test(cox_days,cox_cont) #Vemos con el test T de Student que sí existen diferencias significativas en el tiempo de supervivencia entre ambos grupos a un nivel de confianza del 95%.
```

```
##
##  Welch Two Sample t-test
##
## data:  cox_days and cox_cont
## t = 20.625, df = 138.36, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.433872 1.737941
## sample estimates:
## mean of x mean of y
##  4.117795  2.531888
```

```
t.test(data_cancer_a$Days,data_cancer_a$Cont) #Con lso datos sin transformar también se pueden ver las diferencias.
```

```
##
##  Welch Two Sample t-test
##
## data:  data_cancer_a$Days and data_cancer_a$Cont
## t = 7.1139, df = 99.128, p-value = 1.786e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  462.6482 820.5518
## sample estimates:
## mean of x mean of y
##    679.39     37.79
```

```
mean(data_cancer_a$Days)
```

```
## [1] 679.39
```

```
mean(data_cancer_a$Cont)
```

```
## [1] 37.79
```

```
# Vemos que la media de supervivencia de los pacientes tratados con vitamina C fue de 679.39 días mientras en los pacientes control fue de 37.79 días, siendo significativamente mayor la media de supervivencia (días) en aquellos pacientes que recibieron el tratamiento respecto a los que no.
```

(b) Ahora estamos interesados en comparar la mejora en función del tipo de cáncer. Nos centraremos exclusivamente en los tres tipos de cáncer de la tabla 1 de más arriba y no tendremos en cuenta el sexo. Consideremos la matriz de diseño X correspondiente al modelo yij = μi + Eij con i = 1, 2, 3 donde no hay media común (o término de intercepción). En el libro de regresión aplicada de Rawlings et al. se muestra la tabla 2.

Calcular los elementos de dicha tabla con la matriz de diseño X de este modelo y resolver con ellos el contraste H0 : μ1 = μ2 = μ3 cuando la variable respuesta Y es el logaritmo de la razón entre la supervivencia de los tratados y la supervivencia de sus controles. ¿Cual es la conclusión?. Nota: Habrá que tener en cuenta que en la tabla 2 se supone que el número de réplicas r es el mismo para todos los niveles, cosa que no pasa en este caso.

```
Treated<-data_cancer$Days
Control<-data_cancer$Cont
supervivencia<-log(Treated/Control) #Primero establecemos la
supervivencia, variable Y, como el logaritmo de la razón

lm_type<-lm(supervivencia ~ 0 + Tumor_type,data=data_cancer) #Hacemos el
modelo de regresión sin considerar el intercept, done la variable
respuesta es la variable "supervivencia" y la predictora el tipo de tumor
("Tumor_type").
summary(lm_type) #Como podemos observar, los tres tipos de tumores están
contribuyendo de forma significativa al modelo a un nivel  de alfa =
0.001. Asimismo, nuestro modelo tiene un R2=0.725, bastante bueno.

##
## Call:
## lm(formula = supervivencia ~ 0 + Tumor_type, data = data_cancer)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.88202 -0.78503  0.02611  0.67498  2.77224
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## Tumor_typeBronchus   1.6053     0.2993   5.363 2.88e-06 ***
## Tumor_typeColon      2.3812     0.2993   7.956 4.67e-10 ***
## Tumor_typeStomach    1.6767     0.3423   4.899 1.35e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.234 on 44 degrees of freedom
## Multiple R-squared:  0.7251, Adjusted R-squared:  0.7063
## F-statistic: 38.68 on 3 and 44 DF,  p-value: 2.123e-12

matriz_X<-model.matrix.default(lm_type) #Vemos la matriz de diseño de
nuestro modelo
head(matriz_X)
```

```
##   Tumor_typeBronchus Tumor_typeColon Tumor_typeStomach
## 1                  0               0                 1
## 2                  0               0                 1
## 3                  0               0                 1
## 4                  0               0                 1
## 5                  0               0                 1
## 6                  0               0                 1
```

```
anova(lm_type) #Y aquí hacemos la ANOVA, con H0 : μ1 = μ2 = μ3 y H1: al menos la μ entre dos grupos es distinto. Una vez hecha, el p-valor obtenido fue 2.123e-12 ~ 0.0001 menor de 0.05, por lo que sí podemos decir que exsten diferencias en la supervivencia al menos entre uno de los grupos de tipo de cáncer.
```

```
## Analysis of Variance Table
##
## Response: supervivencia
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## Tumor_type   3 176.753  58.918  38.684 2.123e-12 ***
## Residuals   44  67.015   1.523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) La edad de los pacientes presenta una cierta variabilidad y puede influir en su supervivencia. Añadir a la matriz X del apartado anterior el vector columna con las edades centradas. Utilizar las sumas de cuadrados de los residuos de este modelo y del anterior para contrastar la importancia de ajustar con la edad. ¿Se puede utilizar un test t de Student?

```
edad_cent<-scale(data_cancer$Age, center=TRUE, scale=FALSE)  #Antes escalamos la variable Age
lm_type_age<-lm(supervivencia ~ 0 + Tumor_type + edad_cent, data = data_cancer) #Lo primero que hacemos es un modelo que incluya como variable respuesta la variable Supervivencia, y como variables predictoras el tipo de tumor ("Tumor_type") y Age.
summary(lm_type_age) #Vemos que en esta ocasión, todas las variables tienen una contribución significativa  al modelo (los p-valores son todos < 0.05) menos la variable edad, pues su p-valor es de 0.568 > 0.05. sIN embargo, el valor de R2 sigue siendo bastante bueno, 0.727
```

```
##
## Call:
## lm(formula = supervivencia ~ 0 + Tumor_type + edad_cent, data =
data_cancer)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.01930 -0.86952  0.09928  0.71885  2.64059
##
```

```
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## Tumor_typeBronchus  1.59153    0.30257   5.260 4.30e-06 ***
## Tumor_typeColon     2.38699    0.30179   7.909 6.37e-10 ***
## Tumor_typeStomach   1.68721    0.34540   4.885 1.47e-05 ***
## edad_cent           0.01037    0.01804   0.575    0.568
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.244 on 43 degrees of freedom
## Multiple R-squared:  0.7272, Adjusted R-squared:  0.7018
## F-statistic: 28.65 on 4 and 43 DF,  p-value: 1.237e-11
```

```
matriz_X_age<-model.matrix.default(lm_type_age) #Vemos la matriz de
diseño de nuestro modelo
head(matriz_X_age)
```

```
##   Tumor_typeBronchus Tumor_typeColon Tumor_typeStomach edad_cent
## 1                  0               0                 1 -3.319149
## 2                  0               0                 1  4.680851
## 3                  0               0                 1 -2.319149
## 4                  0               0                 1  1.680851
## 5                  0               0                 1 -1.319149
## 6                  0               0                 1 14.680851
```

```
#No podría utilizarse un test T de Student para estudiar si la edad de
los pacientes influye en la superviencia, ya que la variable tumor_type
es una variable cualitativa con 3 niveles. En vez de usar un test t-
student usaríamos una ANOVA.


anova(lm_type_age) #Y aquí hacemos la ANOVA, donde vemos que sí existen
diferencias en la media de días de la supervivencia según el tipo de
tumor en al menos uno de los grupos pero no según la edad del individuos
(igualdad de medias).
```

```
## Analysis of Variance Table
##
## Response: supervivencia
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Tumor_type  3 176.753  58.918 38.0950 3.558e-12 ***
## edad_cent   1   0.511   0.511  0.3306    0.5683
## Residuals  43  66.504   1.547
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#No obstante,en el caso de querer centrarnos exclusivamente en la
influencia de la edad, sí podríamos realizar una t de student en caso de
que las variables siguieran una distribución normal o un test de U de
Mann Whitney en caso de que no lo fueran.
```

```r
#Primero comprobamos la normalidad de nuestras variables con el test de Shapiro Wilk

shapiro.test(supervivencia) #La variable supervivencia sí presenta una distribución normal (p-valor 0.685 > 0.05)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  supervivencia
## W = 0.98221, p-value = 0.685
```

```r
Age<-data_cancer$Age
shapiro.test(Age) #Pero no la variable Age (p-valor 0.010 < 0.05)
```

```
##  Shapiro-Wilk normality test
## data:  Age
## W = 0.93379, p-value = 0.01044
```

```r
#Como una de ellas no sigue una distribución normal, usamos el test no paramétrico de U de Mann-Whitney:

wilcox.test(supervivencia,Age) #Y nos sale que sí existen diferencias en la supervivencia según la edad del individuo (p-valor 2.2e-16, menor que 0.05)
```

```
## Warning in wilcox.test.default(supervivencia, Age): cannot compute
exact p-value
## with ties
##  Wilcoxon rank sum test with continuity correction
##
## data:  supervivencia and Age
## W = 0, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```r
#Finalmente, si usamos un ANOVA también para comparar el modelo primero que solo tiene en cuenta el tipo del tumor con el otro modelo, el que también incluye la edad, vemos que:
anova(lm_type,lm_type_age) #Como podemos observar, el p-valor obtenido es de 0.5683, superior a 0.05. La diferencia entre ambos modelos no es significativa. Por simplicidad y comodidad, podemos utilizar el modelo que solo tiene en cuenta el tipo de cáncer.
```

```
## Analysis of Variance Table
##
## Model 1: supervivencia ~ 0 + Tumor_type
## Model 2: supervivencia ~ 0 + Tumor_type + edad_cent
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     44 67.015
## 2     43 66.504  1   0.51123 0.3306 0.5683
```

(d) Aunque la regresión de la edad en el modelo anterior pudiera no ser importante, se decidió que cada grupo debería tener su propia regresión sobre la edad para verificar si la edad no es importante en ninguno de los grupos. Modificar adecuadamente la matriz de diseño para acomodar esta nueva situación y completar el test para la hipótesis nula de que la regresión sobre la edad es la misma en los tres grupos de cáncer. ¿Cual es la conclusión?

Hacemos la regresión con la variable supervivencia según la edad pero dividiendo nuestros datos iniciales en los 3 grupos de tipos de cáncer. Así, tenemos una regresión por cada uno de los tipos de cáncer.

```
Stomach<-subset(data_cancer, Tumor_type=="Stomach")
stomach_sup<-log(Stomach$Days/Stomach$Cont)
lm_stomach<-lm(stomach_sup ~ Age, data = Stomach) #Regresión para los
pacientes con cáncer de estómago
summary(lm_stomach) #Vemos que la edad no tiene una contribución
signficativa al modelo de regresión, p-valor de 0.211 > 0.05, y que el
valor de R2 es muy bajo: 0.138
```

```
##
## Call:
## lm(formula = stomach_sup ~ Age, data = Stomach)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9652 -1.1609  0.2033  0.7316  2.0332
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.00938    2.80195  -0.717    0.488
## Age          0.05823    0.04387   1.327    0.211
##
## Residual standard error: 1.34 on 11 degrees of freedom
## Multiple R-squared:  0.138,  Adjusted R-squared:  0.05968
## F-statistic: 1.762 on 1 and 11 DF,  p-value: 0.2113
```

```
Bronchus<-subset(data_cancer, Tumor_type=="Bronchus")
bronchus_sup<-log(Bronchus$Days/Bronchus$Cont)
lm_bronchus<-lm(bronchus_sup ~ Age, data = Bronchus) #Regresión para los
pacientes con cáncer de bronquios
summary(lm_bronchus) #Vemos que la edad no tiene una contribución
signficativa al modelo de regresión, p-valor de 0.527 > 0.05, y que el
valor de R2 es muy bajo: 0.027
```

```
##
## Call:
## lm(formula = bronchus_sup ~ Age, data = Bronchus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5995 -0.5708 -0.1582  0.5821  2.3313
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.67892    1.67818   1.596    0.131
## Age         -0.01635    0.02525  -0.648    0.527
##
## Residual standard error: 1.081 on 15 degrees of freedom
## Multiple R-squared:  0.02721,    Adjusted R-squared:  -0.03765
## F-statistic: 0.4195 on 1 and 15 DF,  p-value: 0.527
```

```
Colon<-subset(data_cancer, Tumor_type=="Colon") #Regresión para los
pacientes con cáncer de colon
colon_sup<-log(Colon$Days/Colon$Cont)
lm_colon<-lm(colon_sup ~ Age, data = Colon)
summary(lm_colon) #Vemos que la edad no tiene una contribución
signficativa al modelo de regresión, p-valor de 0.676 > 0.05, y que el
valor de R2 es muy bajo: 0.012
```

```
##
## Call:
## lm(formula = colon_sup ~ Age, data = Colon)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.04811 -0.79450  0.08827  0.82452  2.23900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.58105    1.90226   0.831    0.419
## Age          0.01255    0.02941   0.427    0.676
##
## Residual standard error: 1.311 on 15 degrees of freedom
## Multiple R-squared:  0.01199,    Adjusted R-squared:  -0.05388
## F-statistic: 0.182 on 1 and 15 DF,  p-value: 0.6757
```

En vista a los resultados obtenidos aquí y anteriormente,para ninguno de los tres tipos de cáncer la edad se muestra como un factor influyente en la supervivencia del paciente de su enfermedad. Sin embargo, en la supervivencia, sí influye el tipo de cáncer que se padezca.

Ejercicio 3 (20 pt.)

El conjunto de datos adjunto diabetes.txt es originario del National Institute of Diabetes and Digestive and Kidney Diseases. El objetivo del conjunto de datos es predecir si un paciente tiene o no diabetes, basándose en ciertas medidas diagnósticas incluidas en el conjunto de datos. Se pusieron varias limitaciones para la selección de estos casos de una base de datos más amplia. En particular, todos los pacientes aquí son mujeres de al menos 21 años de edad de herencia india Pima.

En el archivo diabetes.txt vamos a encontrar las siguientes variables:

- pregnant = Number of times pregnant
- glucose = Plasma glucose concentration (glucose tolerance test)
- pressure = Diastolic blood pressure (mm Hg)
- triceps = Triceps skin fold thickness (mm)
- insulin = 2-Hour serum insulin (mu U/ml)
- mass = Body mass index (weight in kg/(height in m)2)
- pedigree = Diabetes pedigree function
- age = Age (years)
- diabetes = diabetes case (pos/neg)

donde la variable de interés es diabetes.

(a) Ajustar un modelo de regresión logística para predecir la diabetes utilizando todas las otras variables como predictoras. Dar la ecuación del modelo obtenido y clasificar las variables según sean factores protectores o de riesgo para la diabetes.

```
#Lo primero que hacemos es cargar nuestros datos
data_diabetes<- read.csv("C:/Users/Sofia/Downloads/diabetes.txt")
summary(data_diabetes)

head(data_diabetes)
```

Hacemos el modelo de regresión logística para predecir la diabetes

```
data_diabetes$diabetes<-as.factor(data_diabetes$diabetes)
rmod_diabetes<-glm(diabetes ~ pregnant + glucose + pressure + triceps +
insulin + mass + pedigree + age, family = binomial, data = data_diabetes)
summary(rmod_diabetes) #Este es nuestro modelo de regr. logística con la
variable "diabetes" de variable respuesta y el resto del dataset como
predictoras. Vemos que de todas las variables, aquellas que son
significativas a un valor alfa 0.05 serían "glucose", "mass" y
"pedigree". La variable "age" sería significativa también pero cuando
consideramos un nivel de significación del 0.1.

##
## Call:
## glm(formula = diabetes ~ pregnant + glucose + pressure + triceps +
##     insulin + mass + pedigree + age, family = binomial, data =
data_diabetes)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7823  -0.6603  -0.3642   0.6409   2.5612
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.004e+01  1.218e+00  -8.246  < 2e-16 ***
## pregnant     8.216e-02  5.543e-02   1.482  0.13825
## glucose      3.827e-02  5.768e-03   6.635 3.24e-11 ***
## pressure    -1.420e-03  1.183e-02  -0.120  0.90446
## triceps      1.122e-02  1.708e-02   0.657  0.51128
## insulin     -8.253e-04  1.306e-03  -0.632  0.52757
## mass         7.054e-02  2.734e-02   2.580  0.00989 **
## pedigree     1.141e+00  4.274e-01   2.669  0.00760 **
## age          3.395e-02  1.838e-02   1.847  0.06474 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.10  on 391  degrees of freedom
## Residual deviance: 344.02  on 383  degrees of freedom
## AIC: 362.02
##
## Number of Fisher Scoring iterations: 5
```

*La ecuación del modelo respondería a una ecuación del tipo log(p/1-p) = β0 + β*1 …. , si redondeamos a 3 decimales nos acaba quedando algo así:*

*log(p. diabetes/1-p diabetes) = -10.04 + 0.082pregnant + 0.038glucose - 0.001pressure + 0.011triceps - 0.001insulin + 0.007mass + 1.141pedigree + 0.034*age*

Por otro lado, las variables que confieren protección a la hora de desarrollar diabetes serían aquellas con un coeficiente negativo, es decir, "pressure" y "insulin".

Y los factores de riesgo serían el resto: "pregnant", "glucose", "triceps", "mass", "pedigree" y "age".

(b) Calcular el odds ratio de la variable pedigree, así como su intervalo de confianza.

```
exp(coef(rmod_diabetes)["pedigree"]) #Odds ratio: Esto significa que por cada unidad que aumenta el valor de "pedigree", tienes 3.13 veces más de riesgo de desarrollar diabetes
```

```
## pedigree
## 3.129611
```

```
exp(confint(rmod_diabetes, parm = "pedigree")) #I.C. al 95%: (1.378380,7.368273)
```

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

```
##    2.5 %   97.5 %
## 1.378380 7.368273
```

(c) Calcular el odds ratio y la probabilidad de tener diabetes para el individuo de la observación 9.

```
ind9<-data_diabetes[9,]
predict(rmod_diabetes,newdata = data.frame(ind9), type = "response") #La probabilidad de que tenga diabetes es de 0.219 o mejor dicho, del 21,94%
```

```
##         9
## 0.2194284
```

```
odds9<-(0.2194284)/(1 - 0.2194284)
odds9 #El Odds ratio del individuo 9 es de 0.281, < 1, por lo que la asociación es negativa.
```

```
## [1] 0.2811125
```

(d) ¿Como valoras la bondad de ajuste del modelo? Realizar los contrastes o cálculos que se consideren necesarios.

Para mirar la bondad del ajuste nos fijamos en el valor obtenido en el apartado (a) de desviación nula en nuestro modelo, el cual es de 498.10 con 391 g.l. . Al ser un valor alto, la bondad del ajuste es mala.

Vamos a usar el test de Hosmer-Lemeshow para estudiar la bondad de nuestro modelo. Así, H0 sería que el modelo está bien ajustado y H1 que no.

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```
fit.diab<-ifelse(test=rmod_diabetes$fitted.values>0.5, yes=1,no=0)
hoslem.test(rmod_diabetes$y, fit.diab) #El p-valor obtenido en el test de H-L es de 0.294, superior a 0.05, por lo que tenemos suficiente evidencia estadística para aceptar H0 y decir que en nuestro modelo no hay problemas en el ajuste.
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  rmod_diabetes$y, fit.diab
## X-squared = 9.6002, df = 8, p-value = 0.2942
```

Y también podemos calcular el valor de R2

```
library(pscl)
```

```
pR2(rmod_diabetes) #Busco "McFadden" que equivaldría al valor de R2, así sería R2 = 0.309, un valor de bondad de ajuste bajo por este modelo de regresión logística
```

```
## fitting null model for pseudo-r2
```

```
##          llh      llhNull           G2     McFadden         r2ML
r2CU
## -172.0106159 -249.0489027  154.0765735    0.3093300    0.3250067
0.4518042
```

(e) Considerar ahora el modelo reducido con las variables pregnant, glucose, mass, pedigree y age. ¿Es significativa la variable pregnant?

```
rmod_diabetes_red<-glm(diabetes ~ pregnant + glucose + mass + pedigree +
age, family = binomial, data = data_diabetes)
summary(rmod_diabetes_red) #Este es nuestro modelo de regr. logística
reducido. Vemos que de todas las variables predictoras, la variable "age"
sigue siendo signifiativa a un nivel de significación del 0.1. Las
variables "glucose" y "mass" lo son a un nivel de alfa = 0.001 y
"pedigree" al nivel de 0.01.

##
## Call:
## glm(formula = diabetes ~ pregnant + glucose + mass + pedigree +
##     age, family = binomial, data = data_diabetes)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8827  -0.6535  -0.3694   0.6521   2.5814
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.992080   1.086866  -9.193  < 2e-16 ***
## pregnant     0.083953   0.055031   1.526 0.127117
## glucose      0.036458   0.004978   7.324 2.41e-13 ***
## mass         0.078139   0.020605   3.792 0.000149 ***
## pedigree     1.150913   0.424242   2.713 0.006670 **
## age          0.034360   0.017810   1.929 0.053692 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.10  on 391  degrees of freedom
## Residual deviance: 344.89  on 386  degrees of freedom
## AIC: 356.89
##
## Number of Fisher Scoring iterations: 5

#La que no sale significativa en este modelo es la variable "pregnant"
```

Comparamos ahora los 2 modelos usando una ANOVA

```
anova(rmod_diabetes_red,rmod_diabetes,test="Chisq") #Como podemos
observar, el p-valor obtenido es de 0.8639, superior a 0.05. La
diferencia entre ambos modelos no es significativa. Por simplicidad y
comodidad, podemos utilizar el modelo reducido.
```

```
## Analysis of Deviance Table
##
## Model 1: diabetes ~ pregnant + glucose + mass + pedigree + age
## Model 2: diabetes ~ pregnant + glucose + pressure + triceps + insulin
+
##     mass + pedigree + age
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       386     344.89
## 2       383     344.02  3   0.8639   0.8341
```