PRA2

Luis Villazón Esteban, Jose Javier Marti Camarasa

05/01/2021

Enlace GitHub :

[Practica 2] https://github.com/Villaz/indian-liver-patient

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El dataset seleccionado contiene datos referentes a pacientes Indios que sufren de Hígado. El problema que se pretende responder con los datos facilitados es la clasificación de pacientes para saber si en función de sus múltiples atributos es un paciente que sufre de Hígado o no. Para ello el dataset nos ofrece 583 pacientes, de los cuales 416 se encuentran identificados como pacientes que sufren de Hígado y 167 como pacientes que no tienen problemas relacionados con el mismo.

El dataset contiene los siguientes atributos:

- **age**: Edad del paciente, todo aquel paciente cuya edad sea superior a 89 es marcado como 90.
- **gender** Sexo del paciente.
- **tot_bilirubin** Bilirubina Total.
- **direct_bilirubin** Bilirubina en sangre.
- **alkphos** Fosfatasa Alcalina(Niveles altos pueden indicar daño en el hígado).
- **sgpt** Test Alamina aminotransferasa: Test sanguineo para comprobar si hay daño en hígado.
- **sgot** Test Aspartato Aminotransferasa: Test sanguineo para comprobar si hay daño en hígado.
- **tot_proteins** Proteinas totales.
- **albumin** Albumina.
- **ag_ratio** A/G Ratio Albumina y Globulina (Grupo de proteinas solubles en sangre)
- **is_patient** Selector usado para indicar si es paciente de higado. 1 significa si, 2 significa no.

2. Integración y selección de los datos de interés a analizar.

En primer lugar realizamos la carga del dataset en R y transformamos la variable dependiente a tipo factor, transformando el valor 1 a "si_padece" y el valor 2 a "no_padece".

```
ilpd_data <- read.csv("../data/ilpd_data.csv",header = FALSE, col.names = c("edad","sexo","TB","DB","al

ilpd_data <- ilpd_data%>% mutate(
    Padece = as.factor(case_when(
      Padece == "1"  ~ "si_padece",
```

```
      Padece == "2" ~ "no_padece"
          ))
)
```

A continuación mostramos un resumen y una descripción de los valores del dataset.

```
summary(ilpd_data)
```

```
##       edad           sexo           TB               DB
##  Min.   : 4.00   Female:142   Min.   : 0.400   Min.   : 0.100
##  1st Qu.:33.00   Male  :441   1st Qu.: 0.800   1st Qu.: 0.200
##  Median :45.00                Median : 1.000   Median : 0.300
##  Mean   :44.75                Mean   : 3.299   Mean   : 1.486
##  3rd Qu.:58.00                3rd Qu.: 2.600   3rd Qu.: 1.300
##  Max.   :90.00                Max.   :75.000   Max.   :19.700
##     alk_phos         alamine          aspartate            TP
##  Min.   :  63.0   Min.   :  10.00   Min.   :  10.0   Min.   :2.700
##  1st Qu.: 175.5   1st Qu.:  23.00   1st Qu.:  25.0   1st Qu.:5.800
##  Median : 208.0   Median :  35.00   Median :  42.0   Median :6.600
##  Mean   : 290.6   Mean   :  80.71   Mean   : 109.9   Mean   :6.483
##  3rd Qu.: 298.0   3rd Qu.:  60.50   3rd Qu.:  87.0   3rd Qu.:7.200
##  Max.   :2110.0   Max.   :2000.00   Max.   :4929.0   Max.   :9.600
##     albumin           A.G              Padece
##  Min.   :0.900   Min.   :0.3000   no_padece:167
##  1st Qu.:2.600   1st Qu.:0.7000   si_padece:416
##  Median :3.100   Median :0.9200
##  Mean   :3.142   Mean   :0.9443
##  3rd Qu.:3.800   3rd Qu.:1.1000
##  Max.   :5.500   Max.   :2.8000
```

```
str(ilpd_data)
```

```
## 'data.frame':    583 obs. of  11 variables:
##  $ edad     : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ sexo     : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 2 1 1 2 2 ...
##  $ TB       : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ DB       : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
##  $ alk_phos : int  187 699 490 182 195 208 154 202 202 290 ...
##  $ alamine  : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ aspartate: int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP       : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ albumin  : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ A.G      : num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
##  $ Padece   : Factor w/ 2 levels "no_padece","si_padece": 2 2 2 2 2 2 2 2 1 2 ...
```

Podemos comprobar como todos los valores son númericos continuos excepto las varibles **sexo** y **Padece** las cuales son categorícas y se han detectado correctamente.

3. Limpieza de los datos.

3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Según lo observado anteriormente con el uso del método **summary**, no tenemos ninguna variable con datos perdidos. Para tener una visión más clara sobre ello podemos mostrar el número de elementos nulos que existe en cada variable.

```
sort(colMeans(is.na(ilpd_data)), decreasing = TRUE)
```

```
##      edad      sexo        TB        DB  alk_phos   alamine aspartate        TP
##         0         0         0         0         0         0         0         0
##   albumin       A.G    Padece
##         0         0         0
```

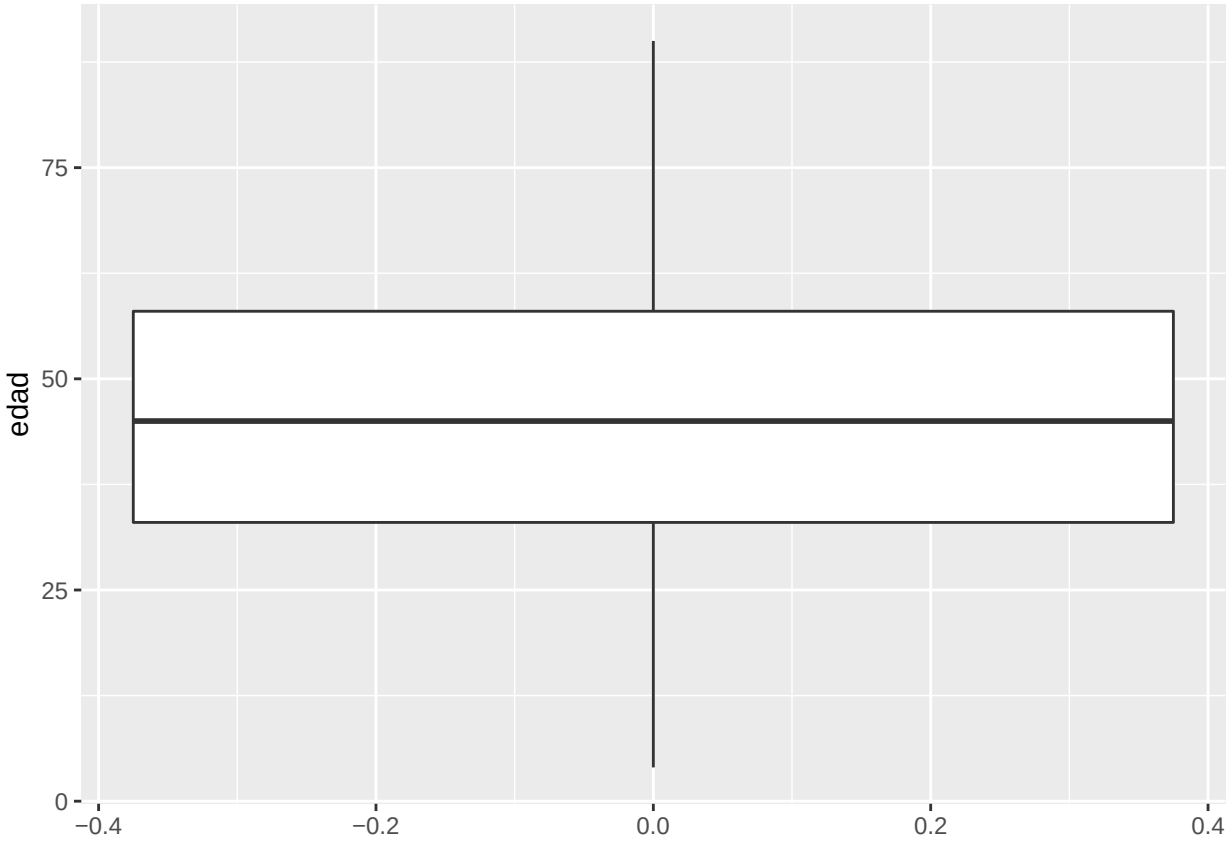
Efectivamente no tenemos nigun variable con datos perdidos. La funcion colMeans,nos muestra qué proporción de datos no disponibles tenemos por columna.

Disponemos por tanto de un data frame formado por 2 variables categóricas y 8 variables exceptuando la variable objetivo, sin valores nulos

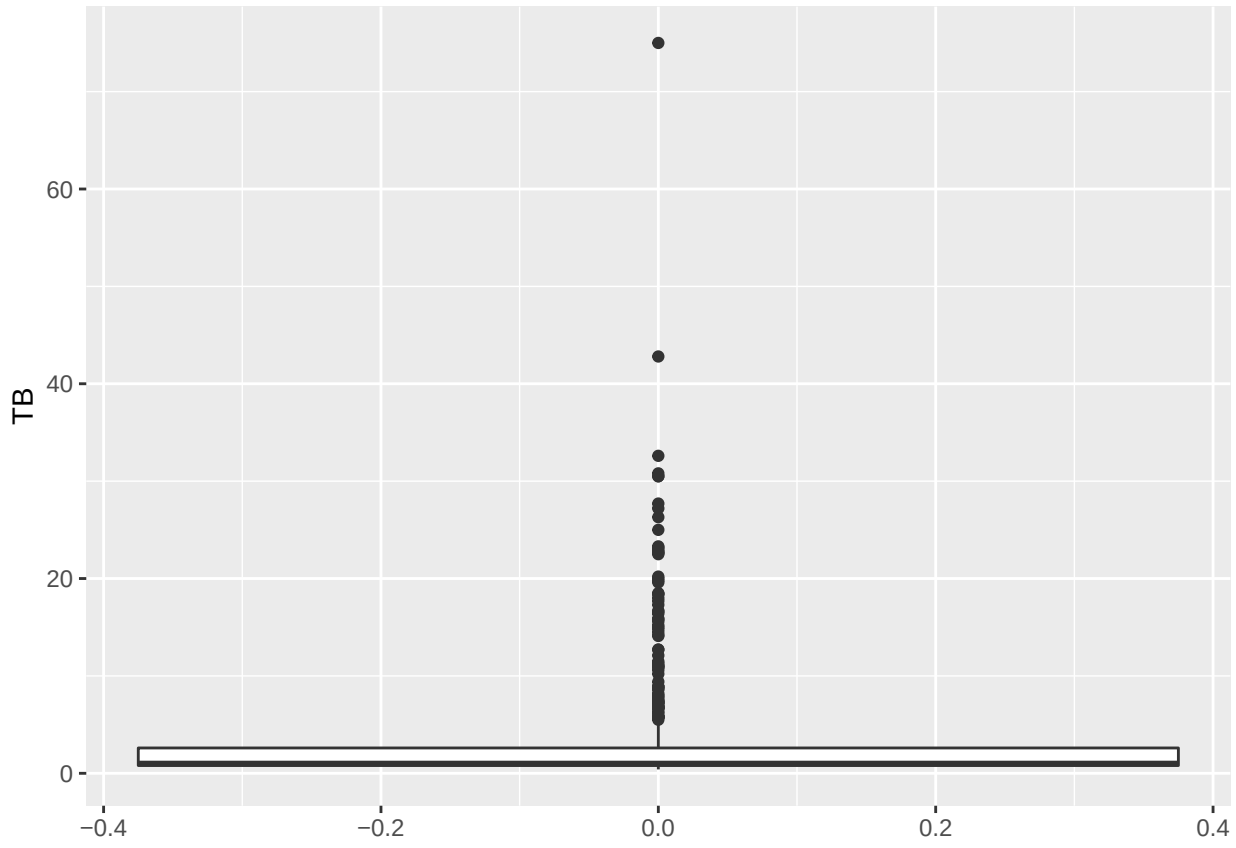
3.2. Identificación y tratamiento de valores extremos.

Para identificar los valores extremos vamos a utilizar diagramas de cajas.

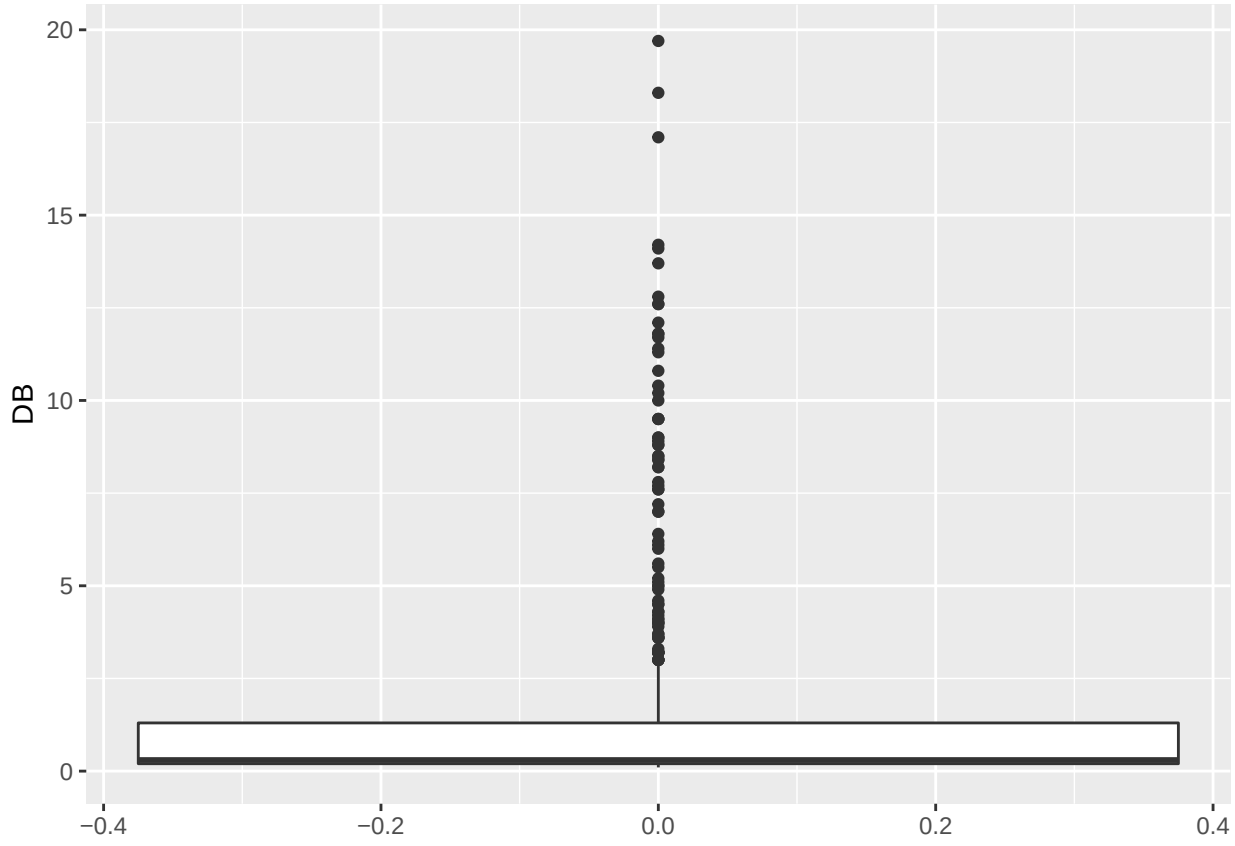
```
library("ggplot2")

ggplot(ilpd_data, aes(y=edad)) + geom_boxplot()
```



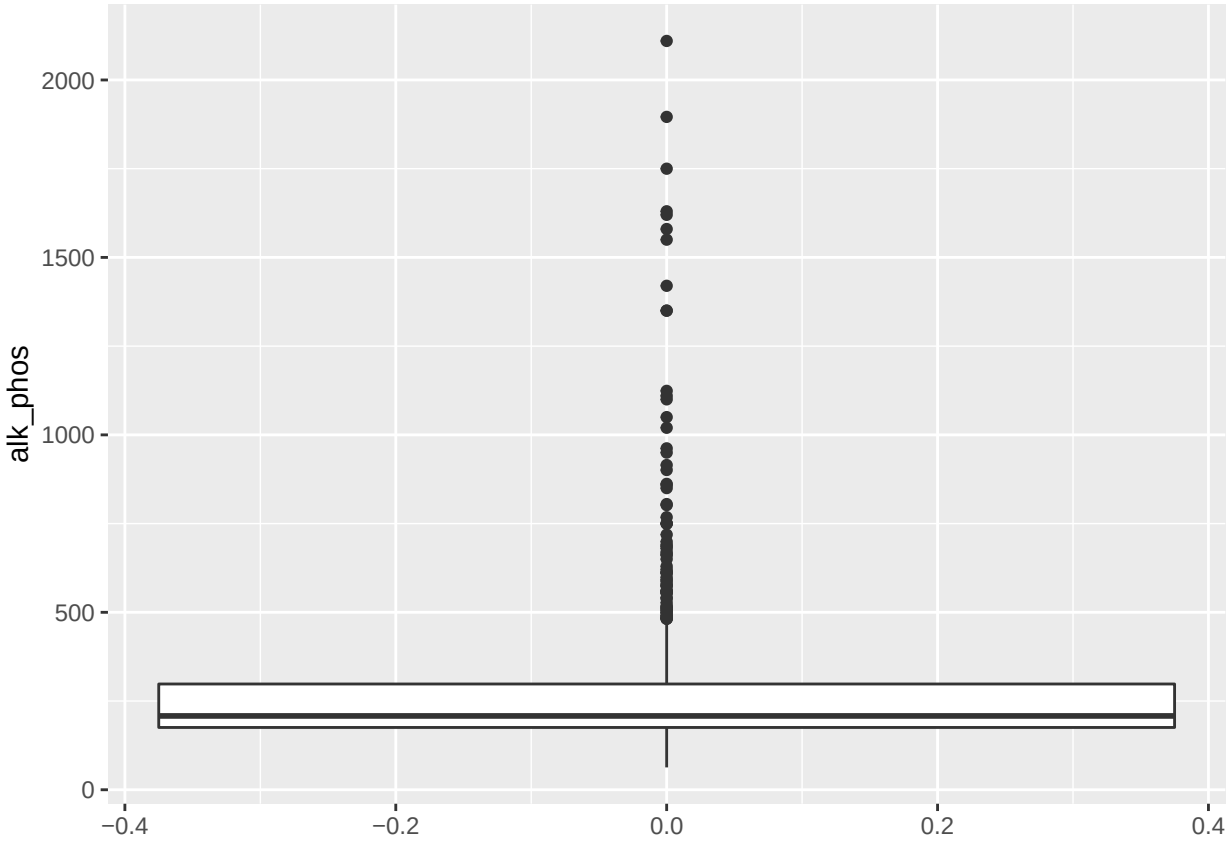

```
ggplot(ilpd_data, aes(y=TB)) + geom_boxplot()
```



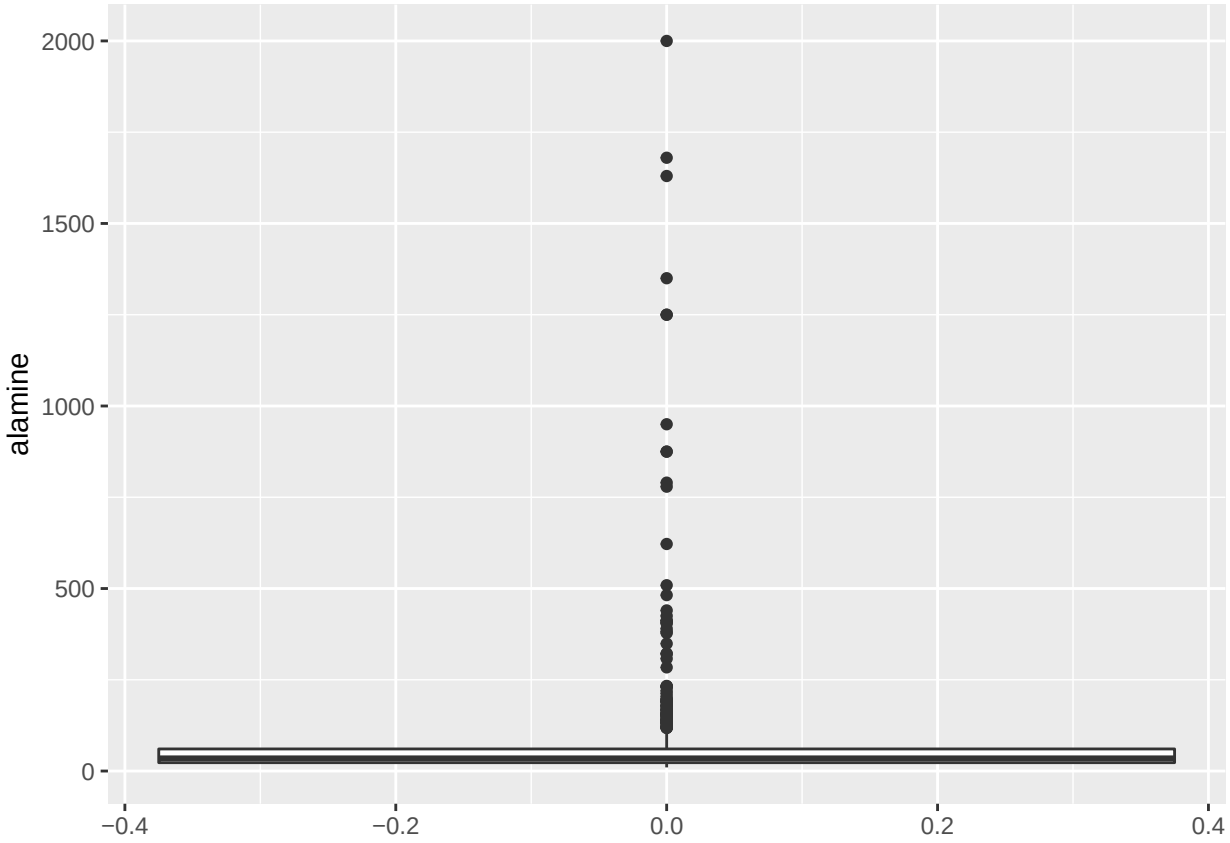

```
ggplot(ilpd_data, aes(y=DB)) + geom_boxplot()
```



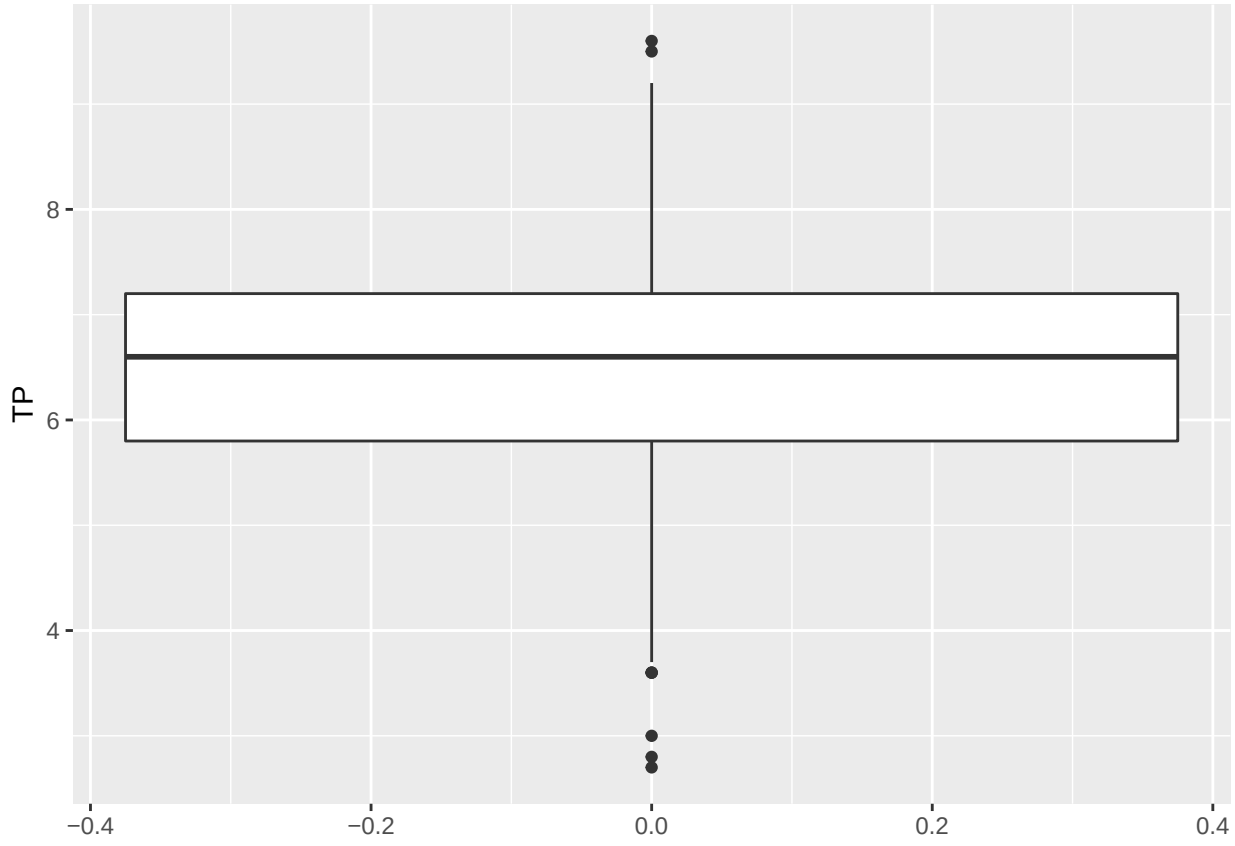

```
ggplot(ilpd_data, aes(y=alk_phos)) + geom_boxplot()
```



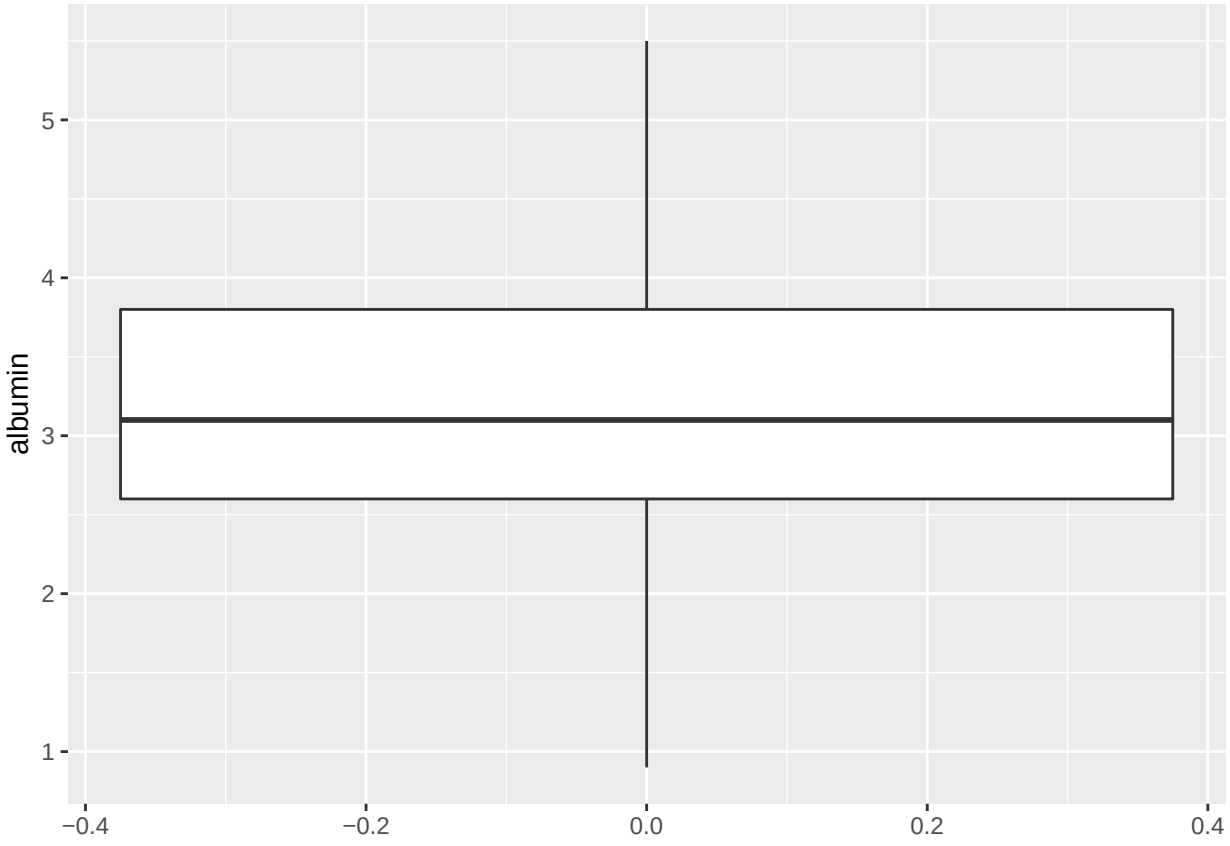

```
ggplot(ilpd_data, aes(y=alamine)) + geom_boxplot()
```



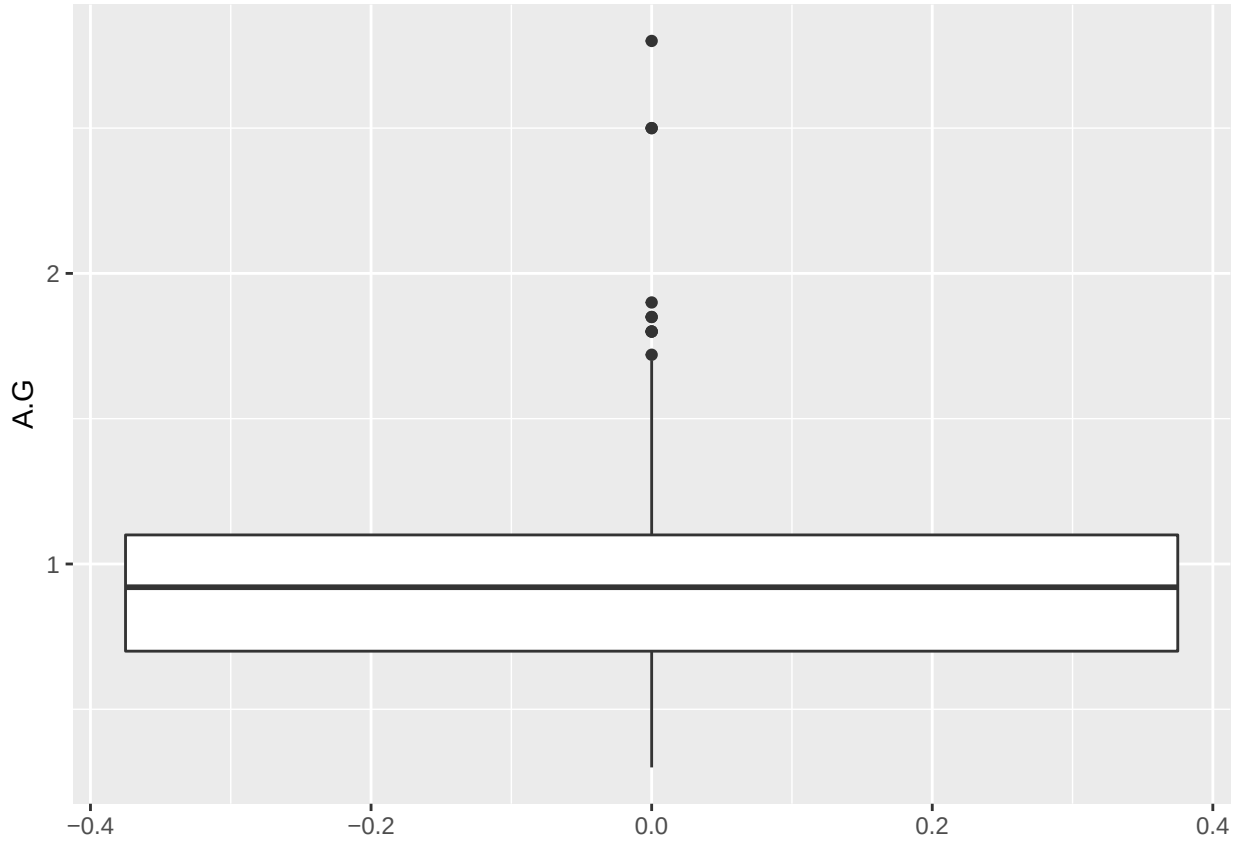

```
ggplot(ilpd_data, aes(y=TP)) + geom_boxplot()
```




```
ggplot(ilpd_data, aes(y=albumin)) + geom_boxplot()
```




```
ggplot(ilpd_data, aes(y=A.G)) + geom_boxplot()
```



Se pueden adivinar posibles outliers o valores extremos. Con un conocimiento del dominio, se podría ver si son susceptibles de quitar o no. Para este análisis los dejaremos por desconocimiento de del dominio.

4. Análisis de los datos.

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

De todos los datos que disponemos, como hemos comentado en el punto 3.1 hay dos variables categóricas que necesitaremos binarizar par poder incluirlas en análsis posteriores como por ejemplo el análisis de correlación.

```
a= model.matrix(~Padece, ilpd_data)
p <- as.data.frame(model.matrix(~Padece, ilpd_data))
sexo <- as.data.frame(model.matrix(~sexo, ilpd_data))
ilpd_data['padece'] <- p$Padecesi_padece
ilpd_data['hombre'] <- sexo$sexoMale

str(ilpd_data)
```

```
## 'data.frame':    583 obs. of  13 variables:
##  $ edad     : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ sexo     : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 2 1 1 2 2 ...
##  $ TB       : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0.7 ...
##  $ DB       : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.2 ...
```

```
##  $ alk_phos : int  187 699 490 182 195 208 154 202 202 290 ...
##  $ alamine  : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ aspartate: int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP       : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8 ...
##  $ albumin  : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1 3.4 ...
##  $ A.G      : num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1 ...
##  $ Padece   : Factor w/ 2 levels "no_padece","si_padece": 2 2 2 2 2 2 2 2 1 2 ...
##  $ padece   : num  1 1 1 1 1 1 1 1 0 1 ...
##  $ hombre   : num  0 1 1 1 1 1 0 0 1 1 ...
```

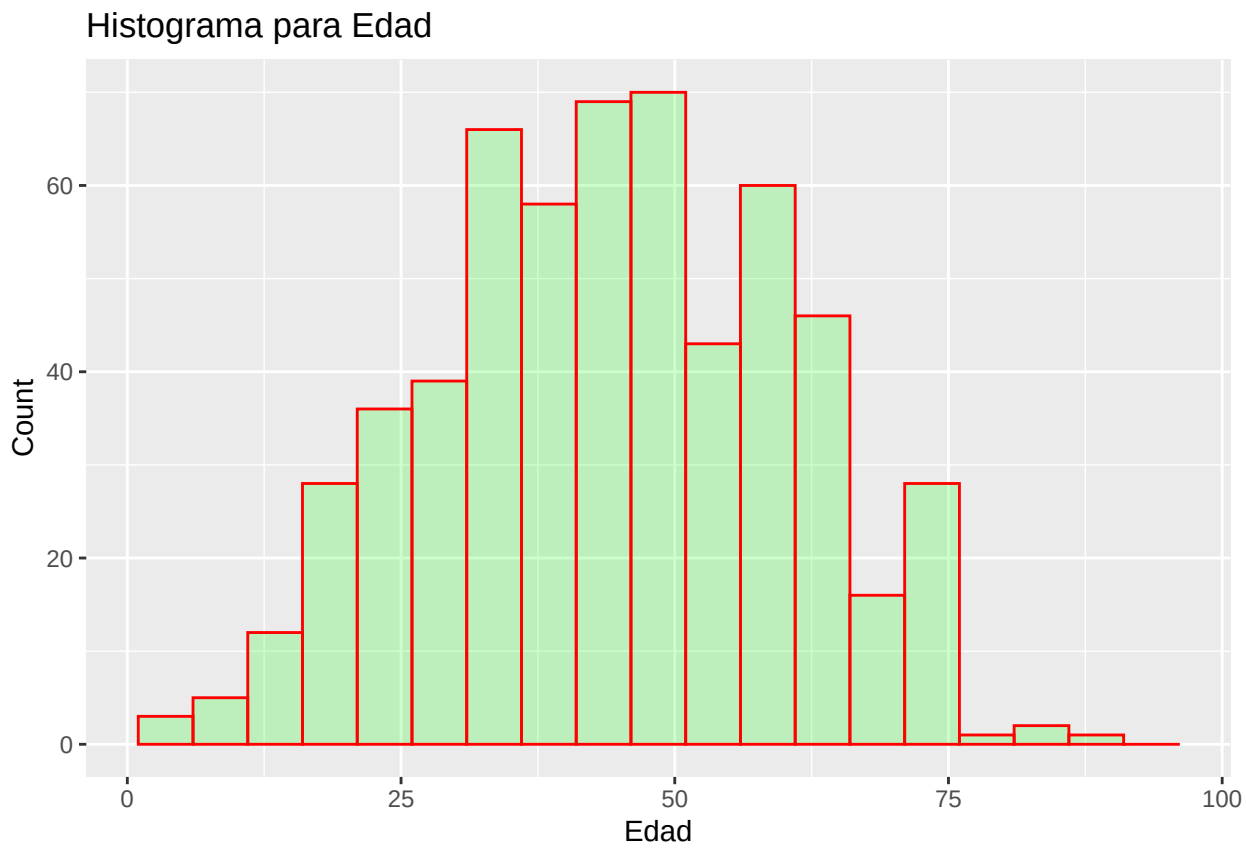
Ahora vemos como hemos obtenido dos nuevas columnas binarizada *padece* y *hombre*

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

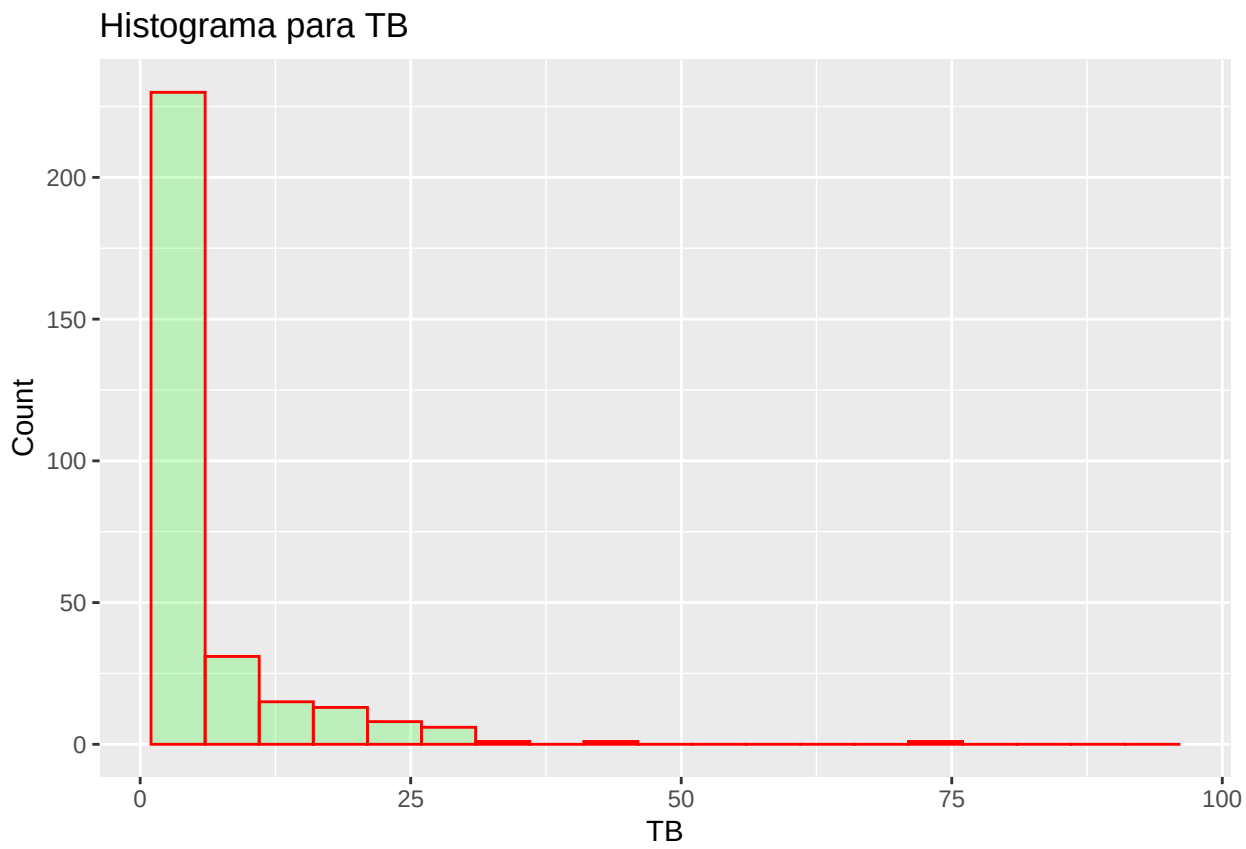
Comprobación de Normalidad

Primeramente vamos a realizar unas inspecciones visuales sobre la normalidad o no de las variables independientes

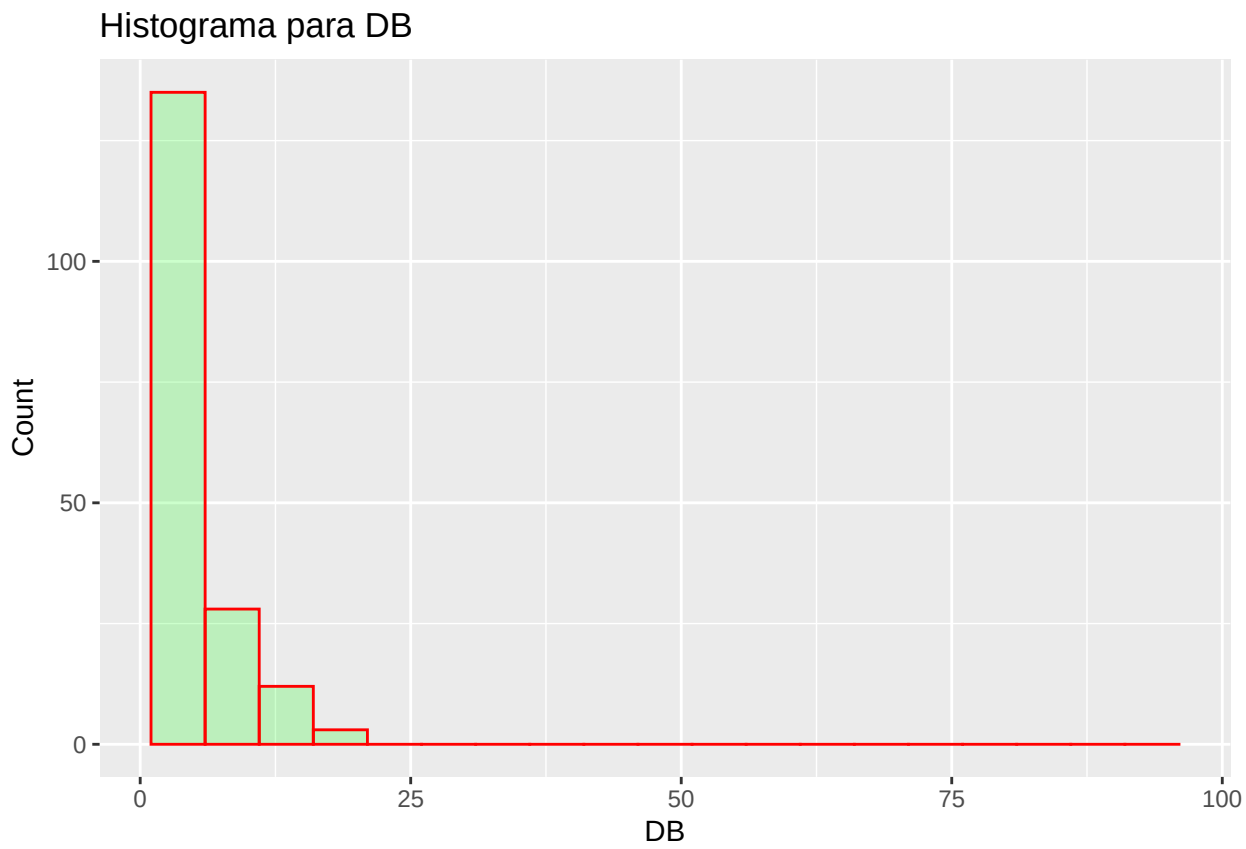
```
par(mfrow=c(4,3))
#Variable Edad
ggplot(data=ilpd_data, aes(x=edad )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para Edad", x="Edad", y="Count")
```



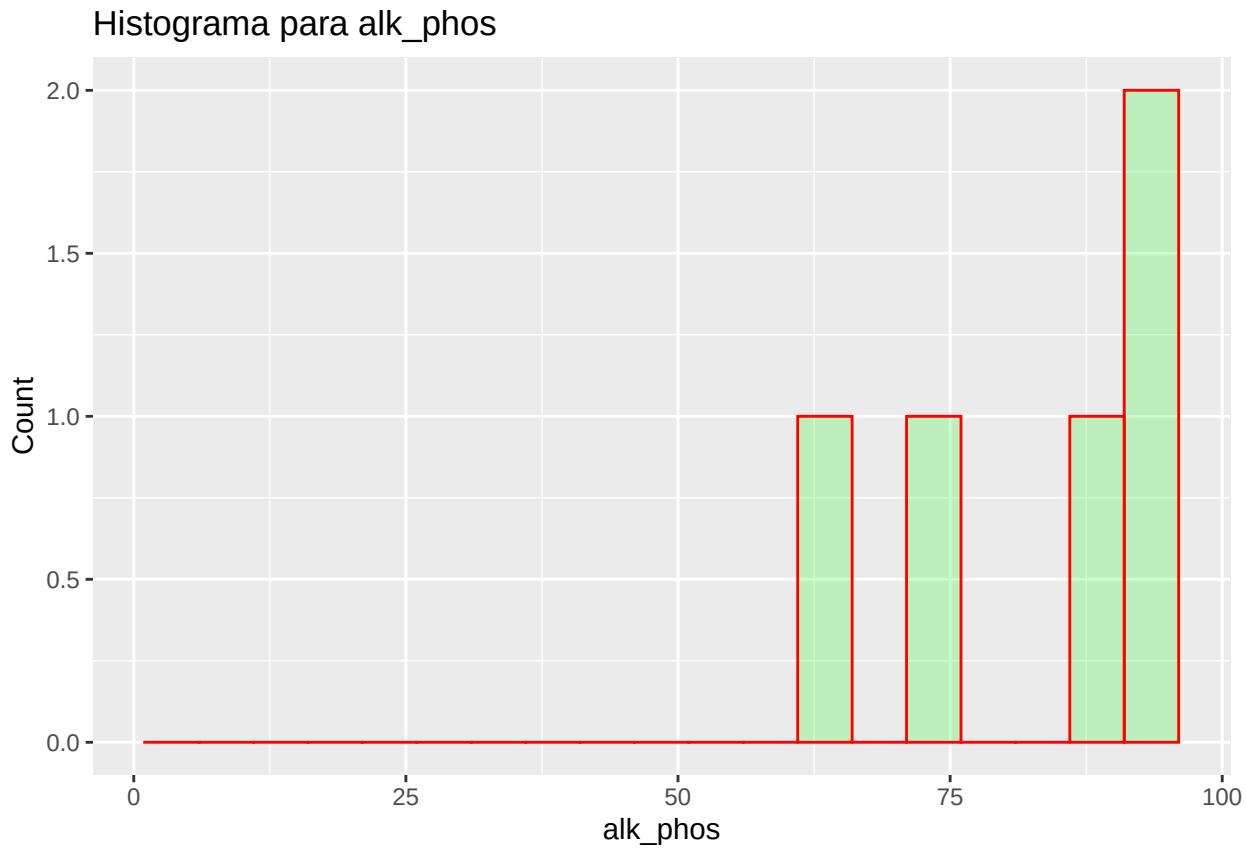

```
#Variable TB
ggplot(data=ilpd_data, aes(x=TB )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para TB", x="TB", y="Count")
```



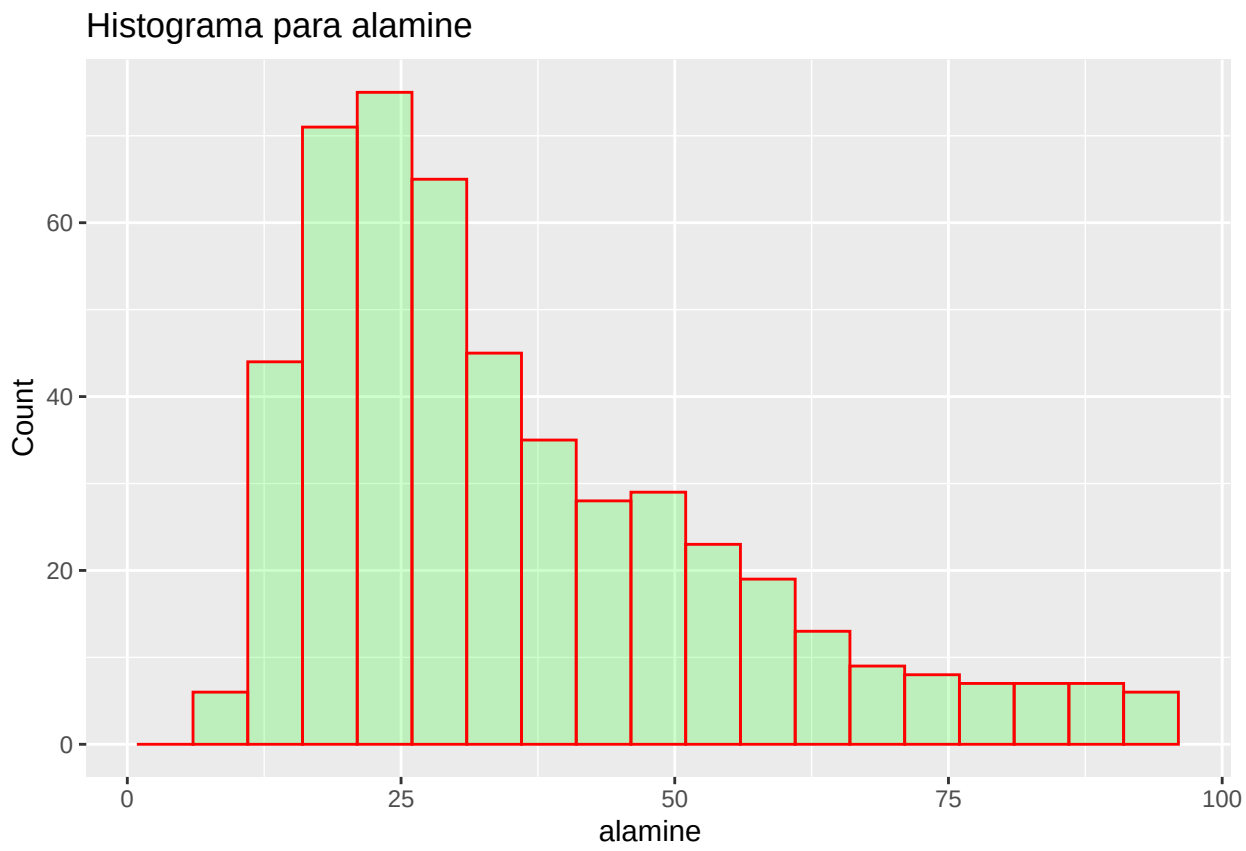

```
#Variable DB
ggplot(data=ilpd_data, aes(x=DB )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para DB", x="DB", y="Count")
```



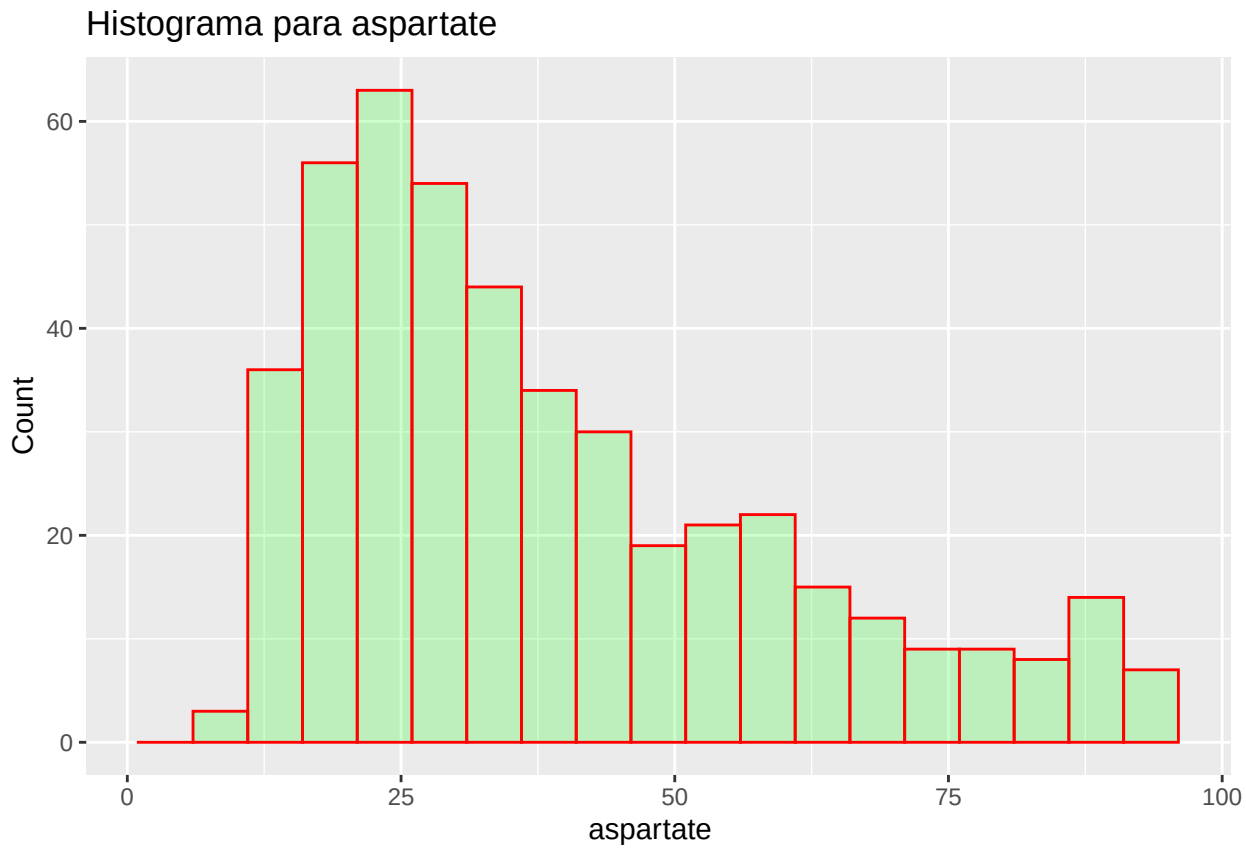

```
#Variable alk_phos
ggplot(data=ilpd_data, aes(x=alk_phos )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para alk_phos", x="alk_phos", y="Count")
```



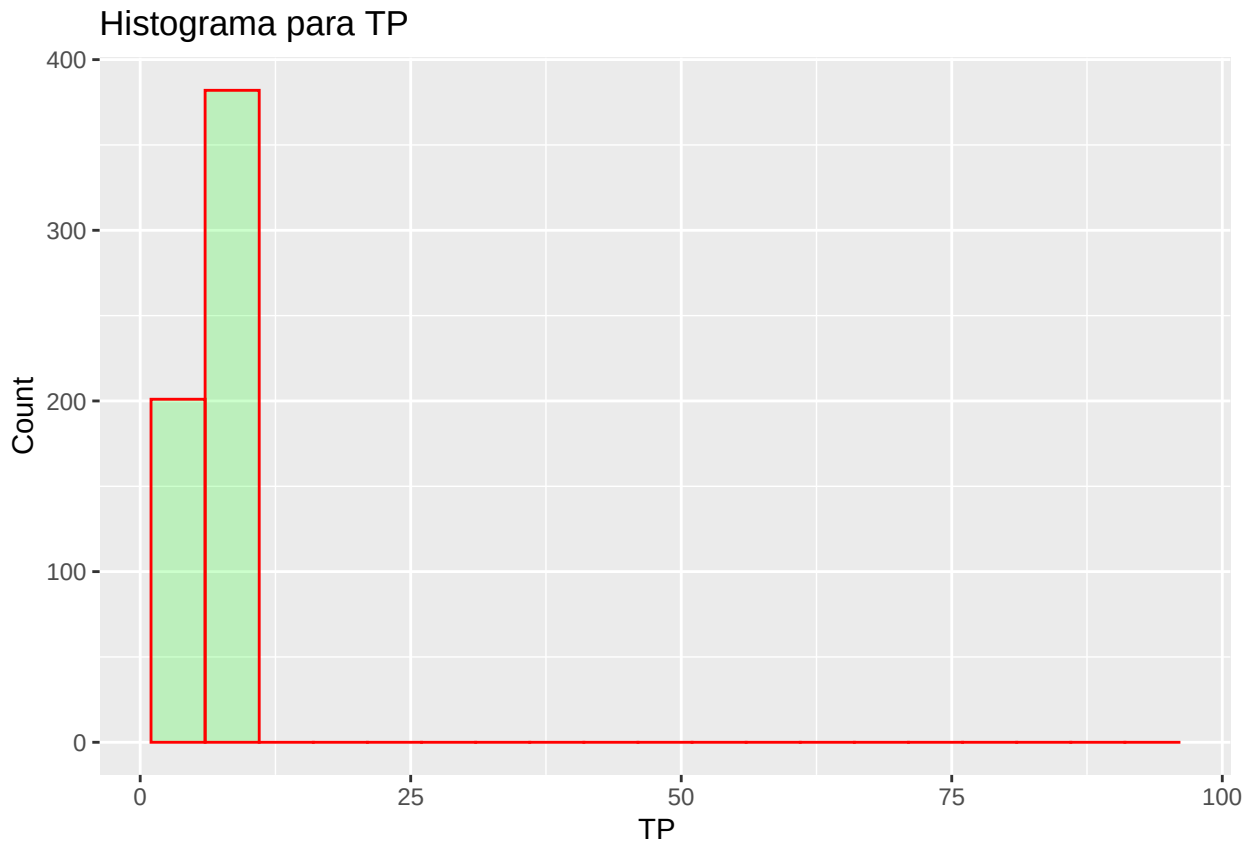

```
#Variable alamine
ggplot(data=ilpd_data, aes(x=alamine   )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para alamine  ", x="alamine  ", y="Count")
```



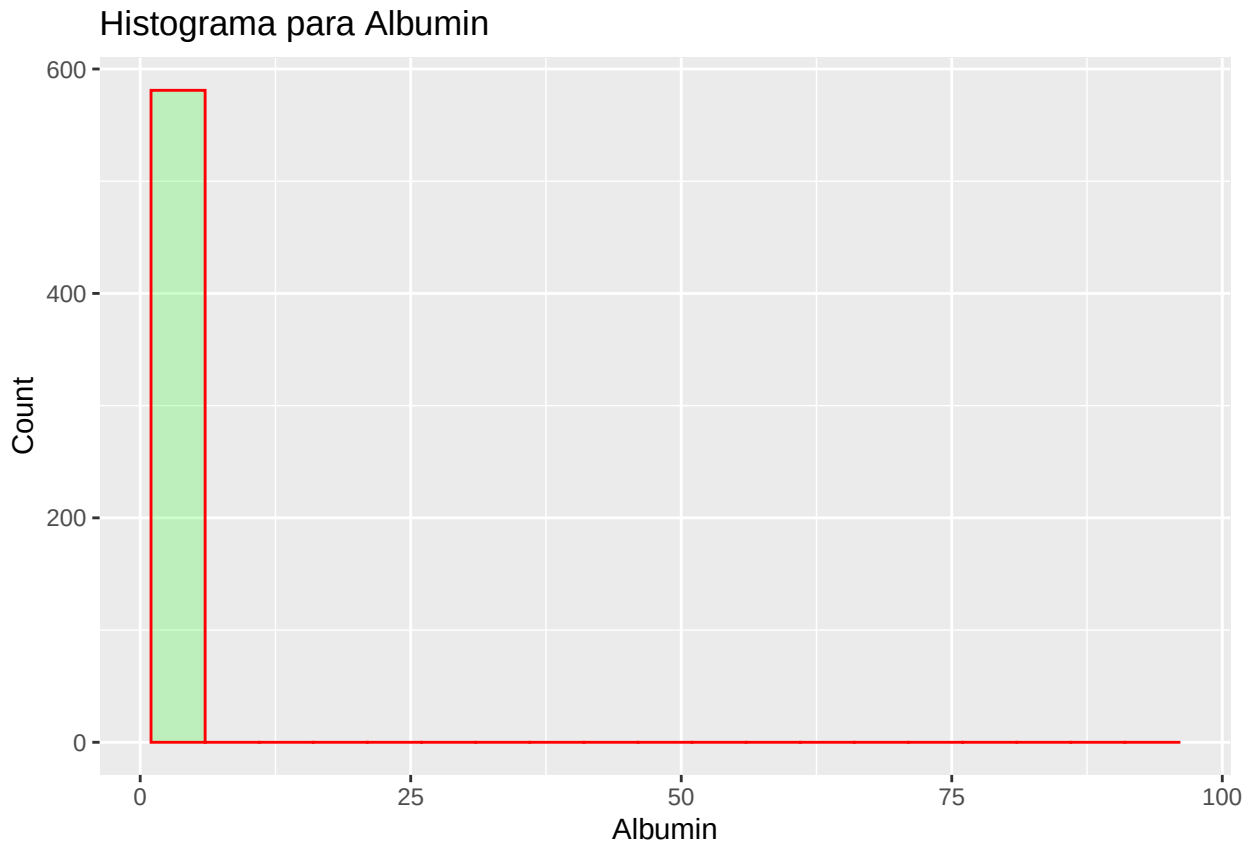

```
#Variable aspartate
ggplot(data=ilpd_data, aes(x=aspartate )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para aspartate  ", x="aspartate  ", y="Count")
```



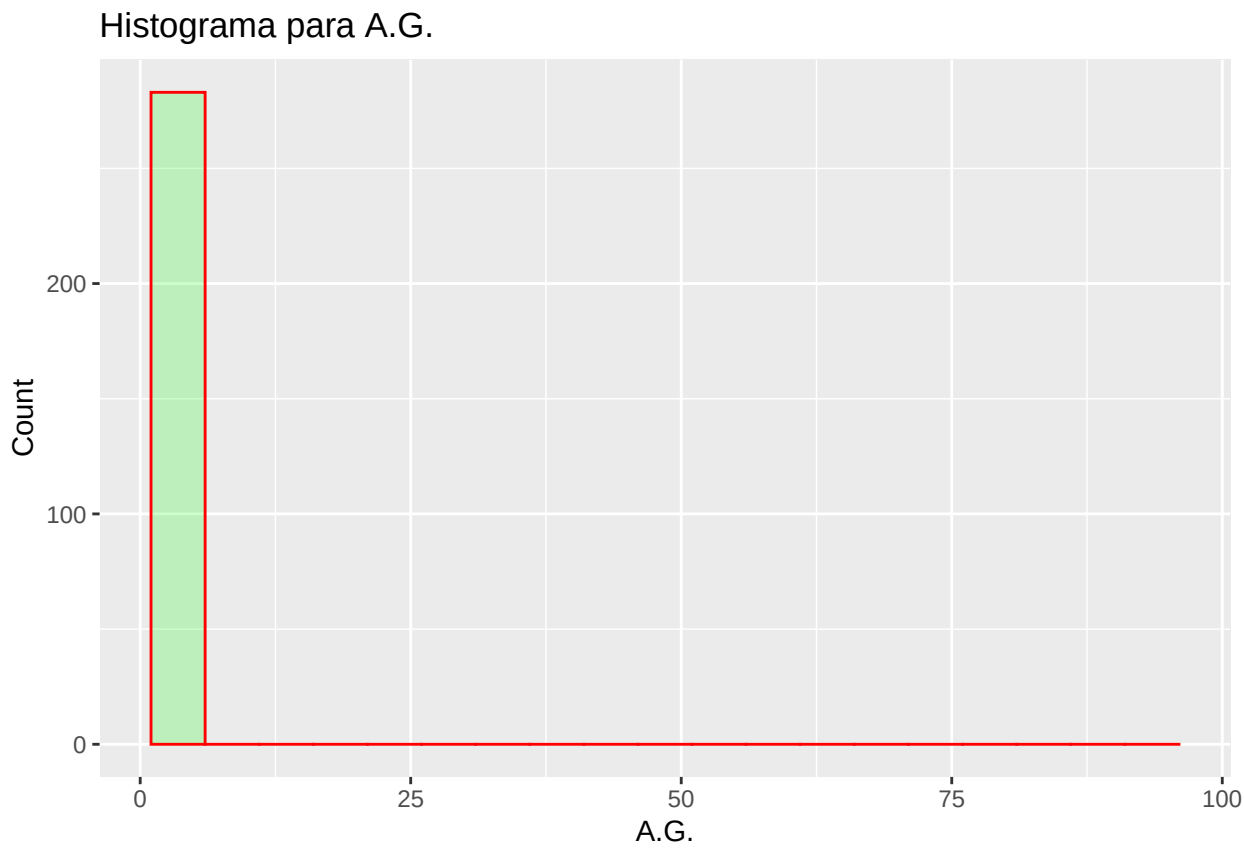

```
#Variable TP
ggplot(data=ilpd_data, aes(x=TP )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para TP  ", x="TP  ", y="Count")
```




```
#Variable albumin
ggplot(data=ilpd_data, aes(x=albumin )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para Albumin  ", x="Albumin", y="Count")
```




```
#Variable A.G.
ggplot(data=ilpd_data, aes(x=A.G )) +
  geom_histogram(aes(y =..count..),
                 breaks=seq(1, 100, by = 5),
                 col="red",
                 fill="green",
                 alpha=.2) +

  labs(title="Histograma para A.G.  ", x="A.G.", y="Count")
```



Ya con las gráficas tenemos una *aproximación visual*, y podemos comprobar que la unica variable que podría seguir una distribución normal es la *Edad*.

Vamos a reforzar esta teoria, aplicando el test de *Anderson Darling* para comprobar la normalidad.

```
library(nortest)
alpha = 0.05
col.names = colnames(ilpd_data)
for (i in 1:ncol(ilpd_data)) {
if (i == 1) cat("Variables que no siguen una distribución normal:\n")
if (is.integer(ilpd_data[,i]) | is.numeric(ilpd_data[,i])) {
p_val = ad.test(ilpd_data[,i])$p.value
if (p_val < alpha) {
cat(col.names[i])
# Format output
if (i < ncol(ilpd_data) - 1) cat(", ")
if (i %% 3 == 0) cat("\n")
}
}
}
```

```
## Variables que no siguen una distribución normal:
## edad, TB,
## DB, alk_phos, alamine,
## aspartate, TP, albumin,
## A.G, padece
## hombre
```

Después de realizar los tests, comprobamos que ninguna variable siguen la Distribucion Normal. Muchas veces las gráficas nos *engañan* a la vista y es necesario realizar pruebas mas profundas

Comprobación de Homocedasticidad

Seguidamente, pasamos a estudiar la homogeneidad de varianzas mediante la aplicación de un test de *Fligner-Killeen*, ya que las variables no siguen una distribución normal. Si hubieran seguido una distribucion se habria podido utilizar el *Test de Levene*.

En este caso, estudiaremos esta homogeneidad en cuanto a los grupos de variables contra la variable *Padece*. En el siguiente test, la *hipótesis nula* consiste en que ambas varianzas son iguales.

```
#Variable Edad
fligner.test(edad ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  edad by Padece
## Fligner-Killeen:med chi-squared = 2.4883, df = 1, p-value = 0.1147
```

```
#Variable TB
fligner.test(TB  ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  TB by Padece
## Fligner-Killeen:med chi-squared = 123.8, df = 1, p-value < 2.2e-16
```

```
#Variable DB
fligner.test( DB  ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  DB by Padece
## Fligner-Killeen:med chi-squared = 120.68, df = 1, p-value < 2.2e-16
```

```
#Variable Alk-Phos
fligner.test(alk_phos  ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  alk_phos by Padece
## Fligner-Killeen:med chi-squared = 46.527, df = 1, p-value = 9.035e-12
```

```
#Variable Alamine
fligner.test(alamine  ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  alamine by Padece
## Fligner-Killeen:med chi-squared = 61.52, df = 1, p-value = 4.382e-15
```

```
#Variable Aspartate
fligner.test(aspartate  ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  aspartate by Padece
## Fligner-Killeen:med chi-squared = 79.531, df = 1, p-value < 2.2e-16
```

```
#Variable Tp
fligner.test(TP  ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  TP by Padece
## Fligner-Killeen:med chi-squared = 0.00028959, df = 1, p-value = 0.9864
```

```
#Variable albumine
fligner.test(albumin  ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  albumin by Padece
## Fligner-Killeen:med chi-squared = 0.24248, df = 1, p-value = 0.6224
```

```
#Variable A.G.
fligner.test(A.G  ~ Padece , data = ilpd_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  A.G by Padece
## Fligner-Killeen:med chi-squared = 2.0607, df = 1, p-value = 0.1511
```

Después de estudiar todas las variables, vemos que de todas, solo par las variables* Edad,TP,Albumin y A.G *, se acepta la Hipótesis Nula de que las varianzas de ambas muestras son iguales.

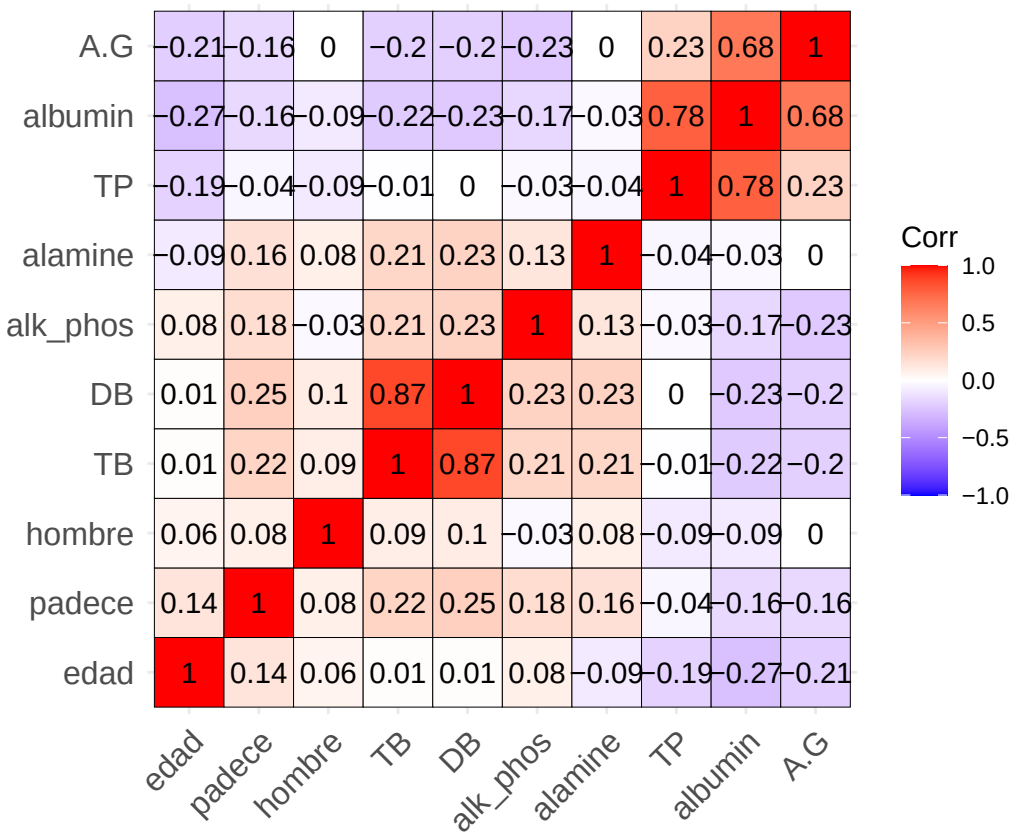
4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Correlación

Dado que queremos conocer si un paciente padece o no de Hígado, vamos a comprobar cual es la correlación de la variable dependiente **Padece** con cada una de las variables independientes existentes en el Dataset.

```
corr <- cor(ilpd_data[, c('edad', 'padece', 'hombre', 'TB', 'DB', 'alk_phos', 'alamine', 'TP', 'albumin
```

```
ggcorrplot(corr, outline.col = "black", lab=TRUE)
```



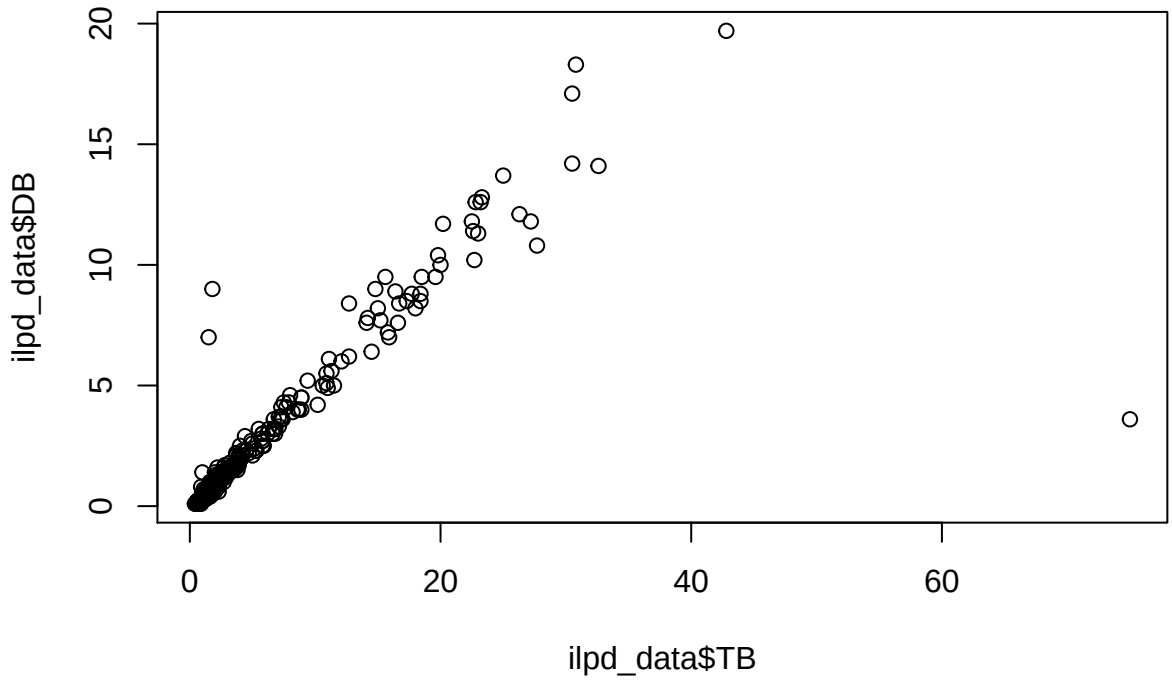
Si nos fijamos en la correlación de las variables existentes con la variable dependiente de padecer enfermedad, podemos observar como no existe una fuerte correlación entre esta y el resto de las variables. Fijandonos en los valores de correlación podemos extraer la siguiente información:

- La edad influye muy levemente en la posibilidad de padecer, a medida que aumenta la edad, existe un ligero aumento en la posibilidad de padecer de Hígado.
- El ser hombre o mujer no tiene prácticamente correlación con la variable, ocurre lo mismo con la variable TP.
- Las variables TB y DB son las que tienen una mayor correlación directa con padecer, a medida que aumentan estas variables puede ser posible que aumenten los casos de paceder de Hígado.
- Por el contrario las variables de albumina y AG tienen una correlación inversa, a medida que estas tienen un valor mayor, el número de casos que padecen de Hígado disminuyen.

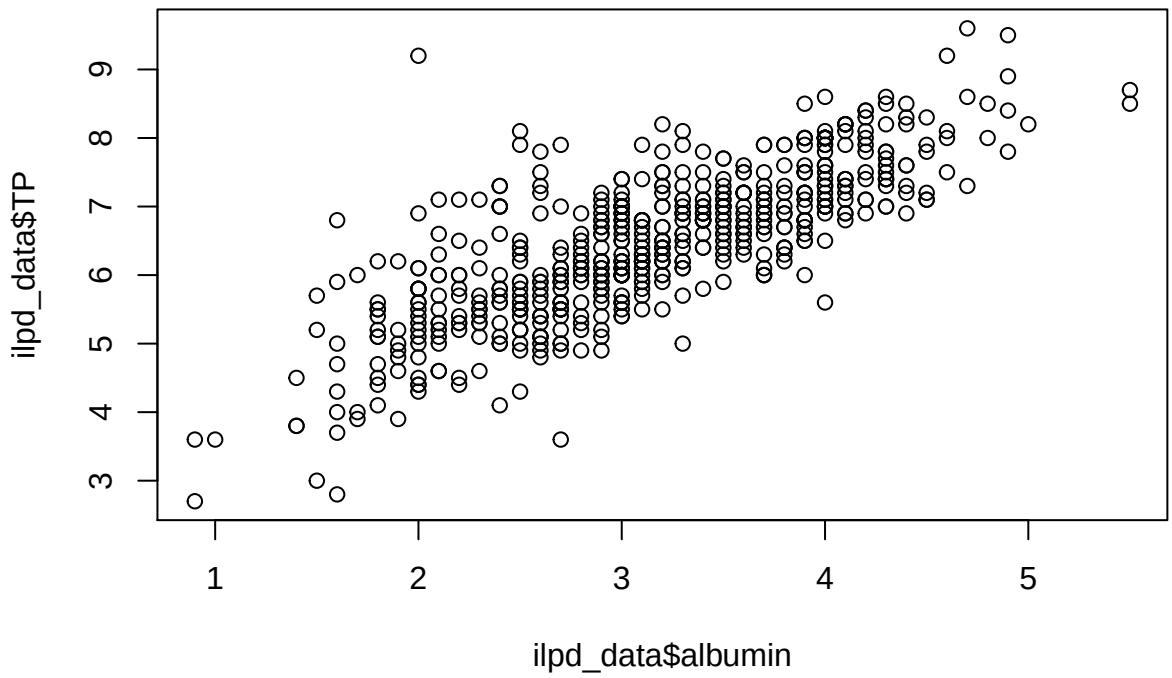
Adicionalmente podemos observar como las variables TB y DB se encuentran fuertemente relacionadas entre si, así como las variables de albumina con TP y albumina con AG.

Al mostrar un gráfico de coordenadas, se puede ver como en estas relaciones, existe una tendencia de los valores a mantenerse en la diagonal, lo que nos hace indicar que existe una relación entre las mismas.

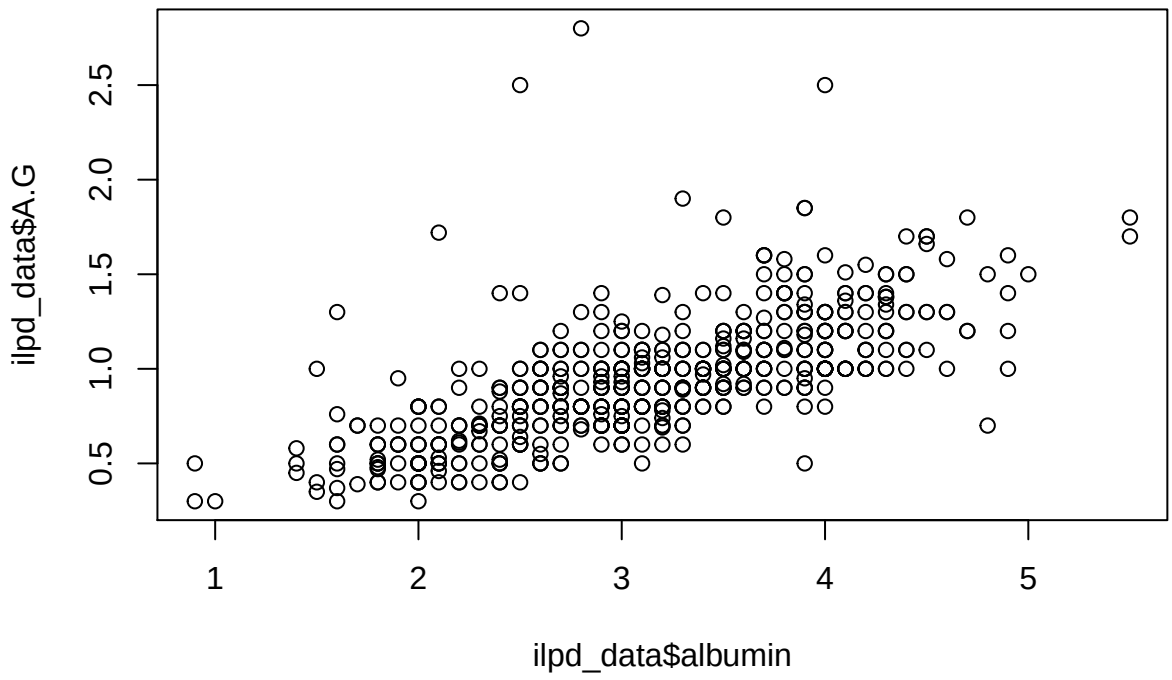
```
plot(ilpd_data$TB, ilpd_data$DB)
```




```
plot(ilpd_data$albumin, ilpd_data$TP)
```




```
plot(ilpd_data$albumin, ilpd_data$A.G)
```



Contraste de hipótesis.

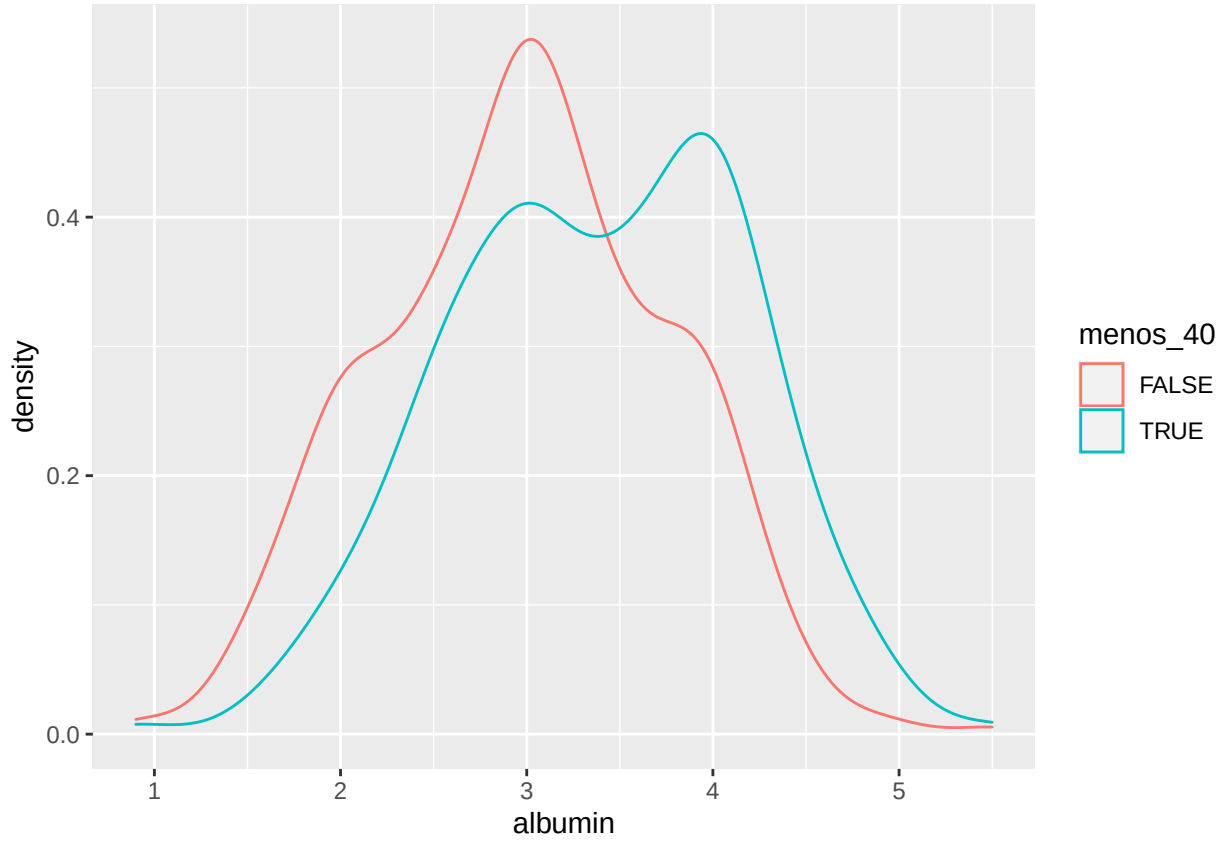
En este apartado vamos a responder a la pregunta de si las personas de menos de 40 años tienen la alumina superior a las personas de más de 40 años.

Para ello en primer lugar escribimos la hipótesis nula y alternativa.

$$H_0 : \mu_{menos40} = \mu_{mas40} H_1 : \mu_{menos40} > \mu_{mas40}$$

Podemos comprobar gráficamente como se distribuye la albumina según el intervalo indicado. A primera vista puede parece que a niveles más altos de albumina se tiene menos edad.

```
ilpd_data['menos_40'] <- ilpd_data$edad < 40
ggplot(ilpd_data,aes(x=albumin,col=menos_40)) + geom_density()
```



Para conocer que test debemos aplicar en primer lugar debemos comprobar si la variable **alumina** sigue una distribución normal, dado que el número de elementos existentes en la muestra es de 583, por el teorema del límite central podemos suponer que así es.

Además hemos de comprobar si la varianza de ambas medidas es diferente

```
var.test( ilpd_data$albumin[ilpd_data$edad < 40], ilpd_data$albumin[ilpd_data$edad >= 40] )
```

```
##
##  F test to compare two variances
##
## data:  ilpd_data$albumin[ilpd_data$edad < 40] and ilpd_data$albumin[ilpd_data$edad >= 40]
## F = 1.0494, num df = 224, denom df = 357, p-value = 0.6819
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8308766 1.3341126
## sample estimates:
## ratio of variances
##           1.049429
```

Observando el p-value se observa como es superior a 0.05, por lo tanto descartamos igualdad de varianzas en las dos poblaciones.

En consecuencia, aplicamos un test de dos muestras independientes sobre la media con varianza desconocida y diferente. Es un test unilateral por la derecha.

```
t.test( ilpd_data$albumin[ilpd_data$edad < 40], ilpd_data$albumin[ilpd_data$edad >= 40], var.equal=FALSE
```

```
##
##  Welch Two Sample t-test
##
## data:  ilpd_data$albumin[ilpd_data$edad < 40] and ilpd_data$albumin[ilpd_data$edad >= 40]
## t = 6.0244, df = 467.45, p-value = 1.721e-09
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2890839       Inf
## sample estimates:
## mean of x mean of y
##  3.386222  2.988268
```

El valor de p es 1.721e-09, el cual es muy inferior a 0.05, por lo tanto se rechaza la hipótesis nula y podemos concluir que el nivel de albumina es superior en los pacientes que tienen menos de 40 años.

Método Regresión

En este apartado vamos a realizar un modelo de regresión para comprobar si una persona en base a sus atributos tiene posibilidades de sufrir afección de hígado. Para realizar el modelo vamos a realizar una aproximación creciente, es decir, dado que el número de variables independientes o factores es comedido podemos ir añadiendo factores nuevos al modelo hasta alcanzar aquel que nos ofrezca un mejor resultado. Para medir cual de los modelos se adapta mejor a los datos utilizaremos el valor AIC, el cual a menor valor mejor se adapta el modelo.

```
glm(Padece ~ edad, data=ilpd_data, family=binomial(link=logit))$aic
```

```
## [1] 691.2874
```

```
glm(Padece ~ edad + TB, data=ilpd_data, family=binomial(link=logit))$aic
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## [1] 624.9866
```

```
glm(Padece ~ edad + TB + DB, data=ilpd_data, family=binomial(link=logit))$aic
```

```
## [1] 623.4061
```

```
glm(Padece ~ edad + TB + DB + alk_phos, data=ilpd_data, family=binomial(link=logit))$aic
```

```
## [1] 614.718
```

```
glm(Padece ~ edad + TB + DB + alk_phos + alamine, data=ilpd_data, family=binomial(link=logit))$aic
```

```
## [1] 594.3286
```

```
glm(Padece ~ edad + TB + DB + alk_phos + alamine + aspartate , data=ilpd_data, family=binomial(link=log
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## [1] 595.0804
```

```
glm(Padece ~ edad + TB + DB + alk_phos + alamine + aspartate + TP, data=ilpd_data, family=binomial(link=
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## [1] 596.8309
```

```
glm(Padece ~ edad + TB + DB + alk_phos + alamine + aspartate + TP + albumin, data=ilpd_data, family=bin
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## [1] 593.6008
```

```
glm(Padece ~ edad + TB + DB + alk_phos + alamine + aspartate + TP + albumin + A.G, data=ilpd_data, famil
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## [1] 593.694
```

Como se puede observar el modelo que ofrece mejores resultados es aquel que tiene los factores **edad**,**TB**,**DB**, **alk_phos**, **alamine**, **aspartate**, **TP** y **albumin**. A continuación vamos a calcular un resumen del modelo para comprobar cuales son los coeficientes y el p-value de cada uno de los factores.

```
model.logit <- glm(Padece ~ edad + TB + DB + alk_phos + alamine + aspartate + TP + albumin + A.G, data=
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model.logit)
```

```
##
## Call:
## glm(formula = Padece ~ edad + TB + DB + alk_phos + alamine +
##     aspartate + TP + albumin + A.G, family = binomial(link = logit),
##     data = ilpd_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1725  -1.0690   0.3830   0.9073   1.6568
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.0360877  1.1579713  -2.622  0.00874 **
## edad         0.0187431  0.0063740   2.941  0.00328 **
```

```
## TB           0.0111605  0.0870451   0.128  0.89798
## DB           0.4500660  0.2402127   1.874  0.06098 .
## alk_phos     0.0012465  0.0008022   1.554  0.12023
## alamine      0.0114436  0.0050188   2.280  0.02260 *
## aspartate    0.0027682  0.0031401   0.882  0.37802
## TP           0.7438741  0.3217803   2.312  0.02079 *
## albumin     -1.3188692  0.6138121  -2.149  0.03166 *
## A.G          1.2224420  0.9191210   1.330  0.18351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 698.37  on 582  degrees of freedom
## Residual deviance: 573.69  on 573  degrees of freedom
## AIC: 593.69
##
## Number of Fisher Scoring iterations: 7
```

Observamos como los factores **TB**, **alk_pho**, **aspartate** y **A.G** tienen un p-value mayor a 0.05, por lo tanto son atributos no significativos y pueden ser eliminados del modelo.

```
model.logit <- glm(Padece ~ edad + DB + alamine + TP + albumin, data=ilpd_data, family=binomial(link=log
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model.logit)
```

```
##
## Call:
## glm(formula = Padece ~ edad + DB + alamine + TP + albumin, family = binomial(link = logit),
##     data = ilpd_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0639  -1.0801   0.4079   0.9187   1.5142
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.682471   0.772928  -2.177  0.02950 *
## edad         0.018523   0.006361   2.912  0.00359 **
## DB           0.547026   0.173892   3.146  0.00166 **
## alamine      0.015842   0.003922   4.040 5.35e-05 ***
## TP           0.432552   0.175239   2.468  0.01357 *
## albumin     -0.666590   0.249315  -2.674  0.00750 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 698.37  on 582  degrees of freedom
## Residual deviance: 579.46  on 577  degrees of freedom
```

```
## AIC: 591.46
##
## Number of Fisher Scoring iterations: 7
```

Una vez eliminados los factores no significativos se observa como el valor de **AIC** ha disminuido a 591.46 por lo que lo podemos considerar como el mejor modelo que podemos obtener.

Para comprobar el comportamiento del modelo vamos a relizar un test de bondad de ajuste. Debido a que las variables explicativas son continuas, vamos a utilizar el **Test de Hosmer-Lemeshow**.

En este test se comparan los valores previstos por el modelo con los valores obtenidos, siendo la hipotesis nula la no existencia de diferencias entre los valores observados y los previstos.

```
actual=as.data.frame(to.dummy(ilpd_data$Padece,"si_padece"))
actual=actual$si_padece.si_padece
hoslem.test(actual, fitted(model.logit))
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  actual, fitted(model.logit)
## X-squared = 4.4292, df = 8, p-value = 0.8165
```

Según el valor obtenido el p-value es de 0.81 el cual es superior a 0.05 por lo tanto no se rechaza la hipótesis nula y podemos asegurar con un 95% que los valores previstos se asemejan a los valores obtenidos.

5. Representación de los resultados a partir de tablas y gráficas.

Tablas y gráficas modelo regresión.

Odd-Ratio

En este punto vamos a calcular cual es el odd-ratio de cada uno de los factores que utilizamos en el modelo de regresión para calcular si una persona puede sufrir afecciones de hígado.

```
exp(coefficients(model.logit))
```

```
## (Intercept)        edad          DB     alamine          TP     albumin
##   0.1859140   1.0186960   1.7281055   1.0159686   1.5411860   0.5134564
```

Según los datos obtenidos podemos deducir que: - La edad afecta a la afección positivamente, a mayor edad la probabilidad de sufrir de hígado aumenta. En este caso, por cada unidad que aumenta la edad, la probabilidad aumenta en un 1.8%. - El DB afecta a la afección positivamente, por cada unidad aumentada de DB la probabilidad aumenta en un 72%. - La alamina afecta a la afección positivamente, por cada unidad aumentada de alamina la probabilidad aumenta en un 1.5%. - La TP afecta a la afección positivamente, por cada unidad aumentada de TP la probabilidad aumenta en un 54%. - La albumina afecta a la afección negativamente, por cada unidad aumentada de albumina la probabilidad disminuye en un 51%.

Matriz de confusión

```
predicciones <- factor(ifelse(test = model.logit$fitted.values > 0.5, yes="si_padece", no="no_padece"))
caret::confusionMatrix(ilpd_data$Padece, predicciones, positive="si_padece")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  no_padece si_padece
##   no_padece        42       125
##   si_padece        33       383
##
##                Accuracy : 0.729
##                  95% CI : (0.6909, 0.7647)
##     No Information Rate : 0.8714
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2062
##
##  Mcnemar's Test P-Value : 4.501e-13
##
##             Sensitivity : 0.7539
##             Specificity : 0.5600
##          Pos Pred Value : 0.9207
##          Neg Pred Value : 0.2515
##              Prevalence : 0.8714
##          Detection Rate : 0.6569
##    Detection Prevalence : 0.7136
##       Balanced Accuracy : 0.6570
##
##        'Positive' Class : si_padece
##
```

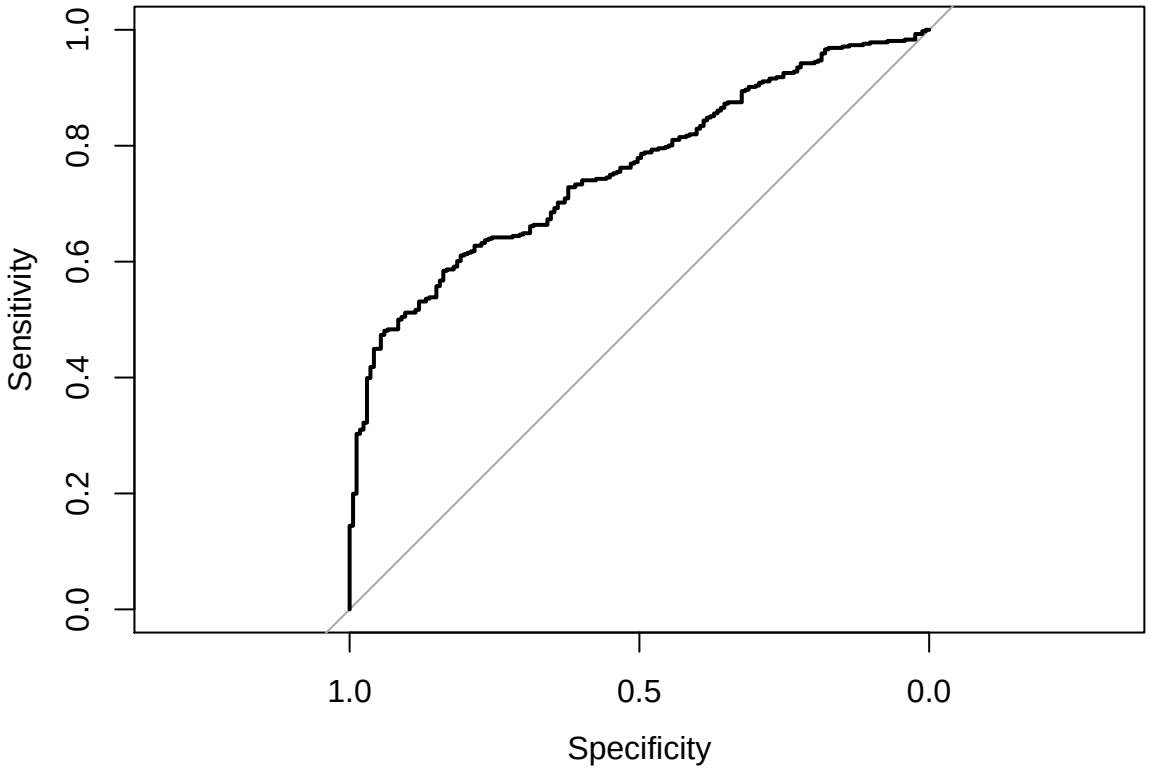
Al observar la matriz de confusión observamos como la exactitud del modelo se encuentra en un 72.9%, Esto quiere decir que en un 72.9% el modelo acierta en su predicción. También podemos observar la sensibilidad del modelo, el número de veces que el modelo indica que si_padece y el valor real es si_padece_, el cual es de un 75%, por lo que el modelo es capaz de encontrar la mayoría de los casos en los cuales un paciente si sufre de afección de higado. Al observar la especificidad podemos ver como su valor es de un 56%, este valor indica el número de veces que el modelo indica que no se padece una afección cuando en verdad no se padece. En este caso el modelo predice el valor correcto únicamente en el 56% de los casos, dando un alto porcentaje de falsos negativos, lo cual a la hora de dar un diagnóstico es preocupante, ya que un 44% de los resultados negativos no serían realmente negativos sino que el paciente si estaría sufriendo de higado.

```
p <- predict(model.logit, ilpd_data[, c("edad","DB","alamine","TP","albumin" )], type="response")
r <- roc(ilpd_data$Padece, p, data=ilpd_data)
```

```
## Setting levels: control = no_padece, case = si_padece
```

```
## Setting direction: controls < cases
```

```
plot(r)
```




```
auc(r)
```

```
## Area under the curve: 0.7583
```

Dado que el area debajo de la curva es de un 0.75, podemos decir que el modelo discrimina de manera adecuada los datos, pero se encuentra lejos del valor 0.8, a partir del cual se consideraría como una discriminación excelente.

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?