Práctica 2. Tipologia de Datos. Análisis y Visualización

Nuria Garcia y Alicia Perdices

03/06/2020

Contents

PRÁCTICA 2:

0.-Librerias necesarias

Abrimos las librerías necesarias para el buen funcionamiento de la práctica.

```
library(readr)
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 3.6.3
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 3.6.3
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(ggplot2)
library("gridExtra")
```

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

El dataset objeto de estudio se ha obtenido a partir del siguiente enlace https://www.kaggle.com/mansoordaku/ckdisease en la web de Kaggle.Consta de 400 registros y 25 variables (25+Id) con información sobre pacientes del hospital Apollo en Tamil Nadu, estado del sur de la India, tanto con enfermedad renal crónica como no (determinados por el campo class).

La enfermedad renal crónica es un problema de salud global que crece constantemente, según nos indica el World Kidney Day (https://www.worldkidneyday.org/), actualmente afecta a 850 millones de personas, es decir, 1 de cada 10 adultos padecen esta enfermedad. El estudio de estos datos a través de la minería de datos

puede ayudar a la detección precoz de dicha enfermedad (evitando una gravedad en el paciente), a realizar un mejor diagnóstico ayudando así a una mejor implementación de los tratamientos, y en los gastos sanitarios (evitando la cronificación se reducen los gastos derivados de un empeoramiento de la enfermedad)*.

Por tanto, este conjunto de datos nos va a servir para predecir qué pacientes tienen más probabilidades de desarrollar una insuficiencia renal crónica a partir de parámetros como la edad, recuento de células en sangre (bioquímica básica) y ciertos aspectos como la presencia o no de diabetes, hipertensión, afecciones cardíacas etc.

El dataset se compone de los siguientes parámetros:

- *Id*: (Numerical):Identificador
- *Age* (Numerical): age in years
- *bp* (Numerical): Blood presure(bp in mm/Hg)
- *sp*: Specific Gravity(nominal).sg - (1.005,1.010,1.015,1.020,1.025)
- *al*: Albumin(nominal):al - (0,1,2,3,4,5)
- *su*: Sugar(nominal):su - (0,1,2,3,4,5)
- *rbc*: Red Blood Cells(nominal):rbc - (normal,abnormal)
- *pc*: Pus Cell (nominal):pc - (normal,abnormal)
- *pcc*: Pus Cell clumps(nominal):pcc - (present,notpresent)
- *ba*: Bacteria(nominal):ba - (present,notpresent)
- *bgr*: Blood Glucose Random(numerical):bgr in mgs/dl
- *bu*: Blood Urea(numerical):bu in mgs/dl
- *sc*: Serum Creatinine(numerical):sc in mgs/dl
- *sod*: Sodium(numerical):sod in mEq/L
- *pot*: Potassium(numerical):pot in mEq/L
- *hemo*: Hemoglobin(numerical:hemo in gms
- *pcv*: Packed Cell Volume(numerical)
- *wc*: White Blood Cell Count(numerical):wc in cells/cumm
- *rc*: Red Blood Cell Count(numerical):rc in millions/cmm
- *htn*: Hypertension(nominal):htn - (yes,no)
- *dm*: Diabetes Mellitus(nominal):dm - (yes,no)
- *cad*: Coronary Artery Disease(nominal):cad - (yes,no)
- *appet*: Appetite(nominal):appet - (good,poor)
- *pe*: Pedal Edema(nominal):pe - (yes,no)
- *ane*: Anemia(nominal):ane - (yes,no)
- *classification*: Class (nominal):class - (ckd,notckd)
- Para más información del impacto del análisis de los datos en este tipo de datos recomendamos el artículo 'Informatics in Medicine Unlocked' (https://www.sciencedirect.com/science/article/pii/S2352914818302387)

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

Realizamos carga de los datos para su análisis (utilizamos el fichero kidney_disease_clean_2.csv que se ha obtenido en Kidey_disease_cleaning_2.Rmd después de la limpieza del fichero original):

```
datos<-read.csv("data/kidney_disease_clean_2.csv",sep=",",header = TRUE)
head(datos)
```

```
##   id age bp    sg al su    rbc       pc        pcc         ba bgr bu  sc sod
## 1  0  48 80 1.020  1  0 normal   normal notpresent notpresent 121 36 1.2 140
## 2  1   7 50 1.020  4  0 normal   normal notpresent notpresent  92 18 0.8 141
## 3  2  62 80 1.010  2  3 normal   normal notpresent notpresent 423 53 1.8 134
## 4  3  48 70 1.005  4  0 normal abnormal    present notpresent 117 56 3.8 111
## 5  4  51 80 1.010  2  0 normal   normal notpresent notpresent 106 26 1.4 141
## 6  5  60 90 1.015  3  0 normal   normal notpresent notpresent  74 25 1.1 142
##   pot hemo pcv   wc  rc htn  dm cad appet  pe ane classification
## 1 5.0 15.4  44 7800 5.2 yes yes  no  good  no  no            ckd
## 2 4.1 11.3  38 6000 4.7  no  no  no  good  no  no            ckd
## 3 4.3  9.6  31 7500 4.1  no yes  no  poor  no yes            ckd
## 4 2.5 11.2  32 6700 3.9 yes  no  no  poor yes yes            ckd
## 5 4.2 11.6  35 7300 4.6  no  no  no  good  no  no            ckd
## 6 3.2 12.2  39 7800 4.4 yes yes  no  good yes  no            ckd
```

Veamos la estructura de los datos cargados:

```
str(datos)
```

```
## 'data.frame':    400 obs. of  26 variables:
##  $ id            : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ age           : int  48 7 62 48 51 60 68 24 52 53 ...
##  $ bp            : int  80 50 80 70 80 90 70 80 100 90 ...
##  $ sg            : num  1.02 1.02 1.01 1 1.01 ...
##  $ al            : num  1 4 2 4 2 3 0 2 3 2 ...
##  $ su            : num  0 0 3 0 0 0 0 4 0 0 ...
##  $ rbc           : Factor w/ 2 levels "abnormal","normal": 2 2 2 2 2 2 2 2 2 1 ...
##  $ pc            : Factor w/ 2 levels "abnormal","normal": 2 2 2 1 2 2 2 1 1 1 ...
##  $ pcc           : Factor w/ 2 levels "notpresent","present": 1 1 1 2 1 1 1 1 2 2 ...
##  $ ba            : Factor w/ 2 levels "notpresent","present": 1 1 1 1 1 1 1 1 1 1 ...
##  $ bgr           : int  121 92 423 117 106 74 100 410 138 70 ...
##  $ bu            : num  36 18 53 56 26 25 54 31 60 107 ...
##  $ sc            : num  1.2 0.8 1.8 3.8 1.4 1.1 1.8 1.1 1.9 7.2 ...
##  $ sod           : int  140 141 134 111 141 142 104 142 136 114 ...
##  $ pot           : num  5 4.1 4.3 2.5 4.2 3.2 4 4.4 4.9 3.7 ...
##  $ hemo          : num  15.4 11.3 9.6 11.2 11.6 12.2 12.4 12.4 10.8 9.5 ...
##  $ pcv           : int  44 38 31 32 35 39 36 44 33 29 ...
##  $ wc            : int  7800 6000 7500 6700 7300 7800 6000 6900 9600 12100 ...
##  $ rc            : num  5.2 4.7 4.1 3.9 4.6 4.4 3.9 5 4 3.7 ...
##  $ htn           : Factor w/ 2 levels "no","yes": 2 1 1 2 1 2 1 1 2 2 ...
##  $ dm            : Factor w/ 2 levels "no","yes": 2 1 2 1 1 2 1 2 2 2 ...
##  $ cad           : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ appet         : Factor w/ 2 levels "good","poor": 1 1 2 2 1 1 1 1 1 2 ...
##  $ pe            : Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 2 1 1 ...
##  $ ane           : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 1 1 2 2 ...
##  $ classification: Factor w/ 2 levels "ckd","notckd": 1 1 1 1 1 1 1 1 1 1 ...
```

Como ya se ha comentado partimos de 400 registros donde tenemos: 250 registros de pacientes con enfermedad renal crónica y 150 registros de pacientes que no presentan enfermedad renal.

```
table(datos$classification)
```

```
##
##    ckd notckd
##    250    150
```

Para poder deducir qué factores son los más determinantes a la hora de desarrollar una enfermedad como la insuficiencia renal crónica, contaremos con todos los atributos del dataset una vez completada la fase de limpieza correspondiente.Tan solo prescindiremos del campo Id que es simplemente identificativo del registro y no nos aporta información al estudio.

3.-Limpieza de datos

3.1-¿Los datos contienen ceros o elementos vacios?¿Como gestionarías cada uno de estos casos?

Todos los atributos del dataset, contienen elementos vacíos. La cantidad de valores nulos identificados en los campos es el siguiente:

- *Id*: No presenta valores nulos ni vacíos.
- *Age*: 9 valores nulos.
- *bp*: 12 valores nulos.
- *sp*: 47 valores vacíos.
- *al*: 46 valores vacíos.
- *su*: 49 valores vacíos.
- *rbc*: 152 valores vacíos.
- *pc*: 65 valores vacíos.
- *pcc*: 4 valores vacíos.
- *ba*: 4 valores vacíos.
- *bgr*: 44 valores nulos.
- *bu*: 19 valores nulos.
- *sc*: 17 valores nulos.
- *sod*: 87 valores nulos.
- *pot*: 88 valores nulos.
- *hemo*: 52 valores nulos.
- *pcv*: 71 valores nulos.
- *wc*: 106 valores nulos.

- *rc*: 131 valores nulos.
- *htn*: 2 valores vacíos.
- *dm*: 2 valores vacíos y 5 valores mal identificados.
- *cad*: 2 valores vacíos y 2 valores mal identificados.
- *appet*: 1 valor vacío.
- *pe*: 1 valor vacío.
- *ane*: 1 valor vacío.
- *classification*: 2 valores mal identificados.
- En primer lugar, se ha transformado los valores de todas las columnas al tipo de datos correcto, además de corregir las categorías establecidas en las variables cualitativas.
- Los elementos vacíos se han gestionado de dos formas
  - **Método 1**.Para las variables categóricas, se han sustituido por la clase más frecuente. Y para las variables numéricas, por la media del atributo en cuestión. Al final del proceso, hemos obtenido el archivo **kidney_disease_clean_1.csv.(Kidey_disease_cleaning.Rmd)**
  - **Método 2**.Para las variables categóricas, se han sustituido por la clase más frecuente al igual que en el método anterior. Pero en este caso, para los elemenos vacíos de las variables numéricas, se ha utilizado el algoritmo KNN, de los vecinos más cercanos, para imputar estos valores.Al final del proceso. Además se ha realizado estudio de valores atípicos. Como resultado se ha obtenido el archivo **kidney_disease_clean_2.csv. (Kidey_disease_cleaning_2.Rmd)**

3.2-Identificación y tratamiento de valores extremos

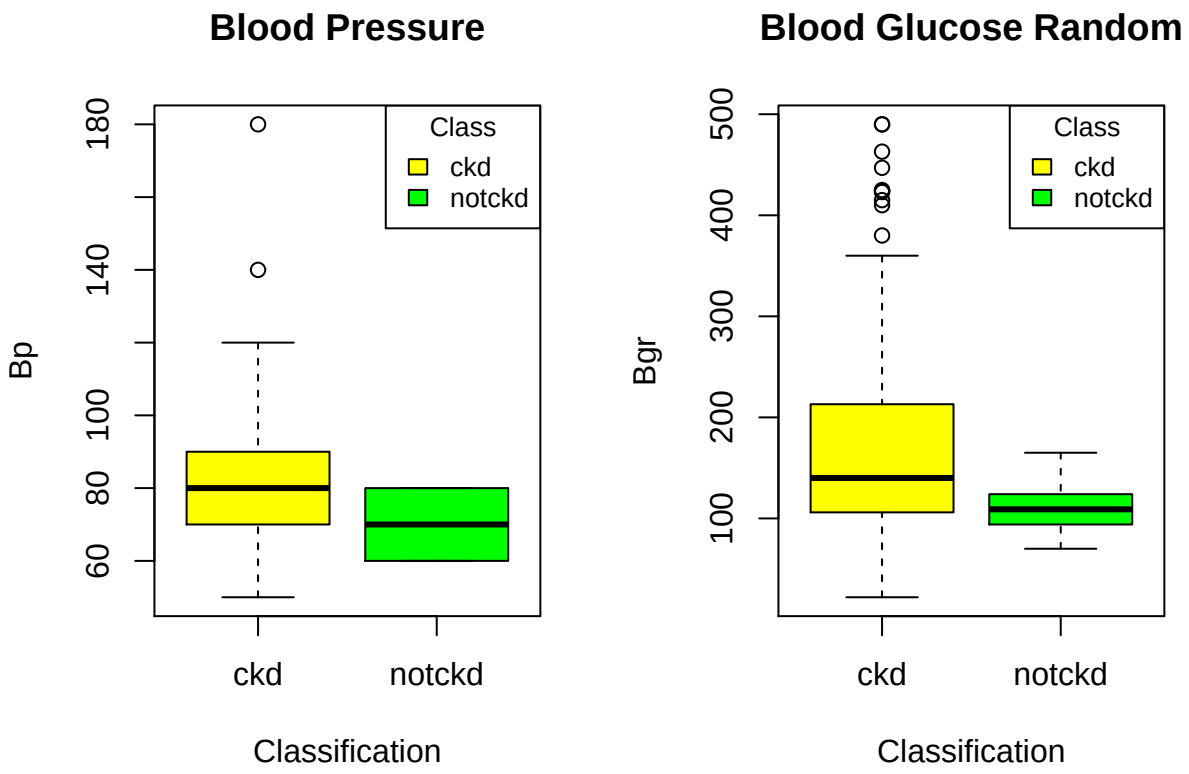
Los valores extremos se han identificado mediante diagramas de cajas.Se han tratado en el apartado de limpieza reemplazando solo aquellos no compatibles con la realidad de la enfermedad. El resto de outliers es importante conservarlos, ya que pueden condicionar que una enfermedad renal crónica sea diagnosticada correctamente o no.

Realizaremos un breve analisis de los diagramas de cajas, visualizando las características de los pacientes que tienen diagnosticado la enfermedad crónica (etiquetados como ckd) y los que no la tienen (etiquetados como notckd). Recordamos que en **Kidey_disease_cleaning_2.Rmd** se ha realizado un previo estudio donde algunos valores extremos han sido tratados y substituidos por la imputación utilizada en dicho 'cleaning'.

Nota: la distinción de color solo indica la dos clases: ckd (pacientes con enfermedad crónica) serán amarillos y notckd (pacientes sin enfermedad) serán verdes.

DIAGRAMAS DE CAJAS

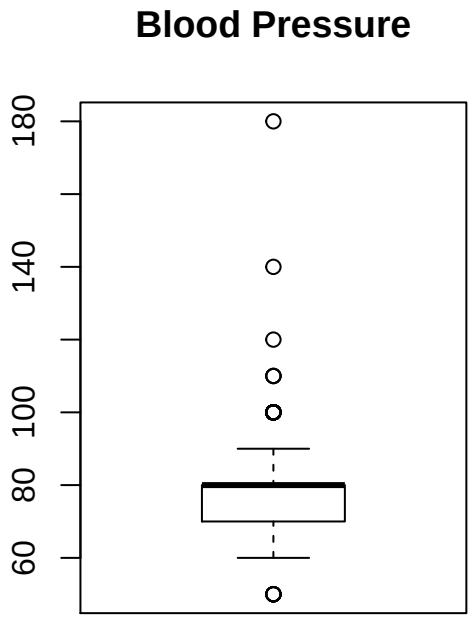
```
par(mfrow=c(1,2))
boxplot(datos$bp ~ datos$classification,col = c("yellow", "green"),
                          xlab="Classification", ylab="Bp", main='Blood Pressure',)
legend("topright", legend=c("ckd", "notckd"), c("yellow", "green"), title="Class", fill=c("yellow", "gr

boxplot(datos$bgr ~ datos$classification,col = c("yellow", "green"),
                        xlab="Classification", ylab="Bgr",main='Blood Glucose Random')
legend("topright", legend=c("ckd", "notckd"), col=c("yellow", "green"), title="Class", fill=c("yellow",
```

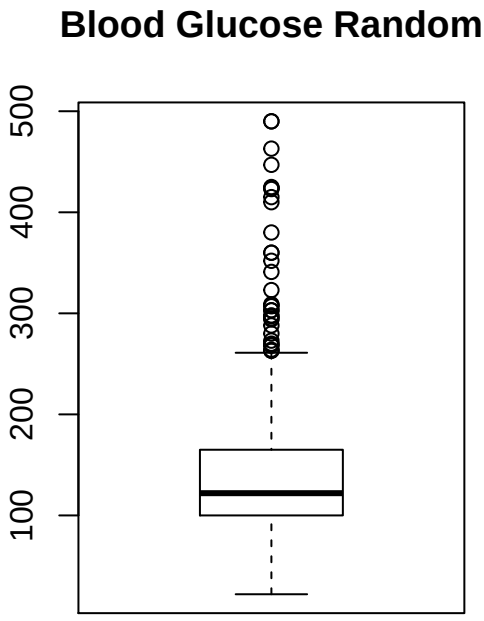


Observamos que los pacientes diagnosticados tienen valores más altos que los pacientes sin la enfermedad. Además se obtienen valores fuera de los normal. Tanto en presión sanguínea como en glucosa, la media está por encima de los valores normales.

Los valores extremos encontrados en las varibles, considerándolas en su globalidad, son:

```
par(mfrow=c(1,2))
boxplot(datos$bp,main='Blood Pressure')
boxplot(datos$bgr,main='Blood Glucose Random')
```






```
values_bp<-boxplot.stats(datos$bp)$out
cat("Valores extremos de Blood Pressure :", toString(values_bp), "\n")
```

```
## Valores extremos de Blood Pressure : 50, 100, 100, 100, 100, 100, 100, 100, 100, 100, 110, 100, 100,
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_bp, dnn = "Count values_bp:")
```

```
## Count values_bp:
##  50 100 110 120 140 180
##   6  26   3   1   1   1
```

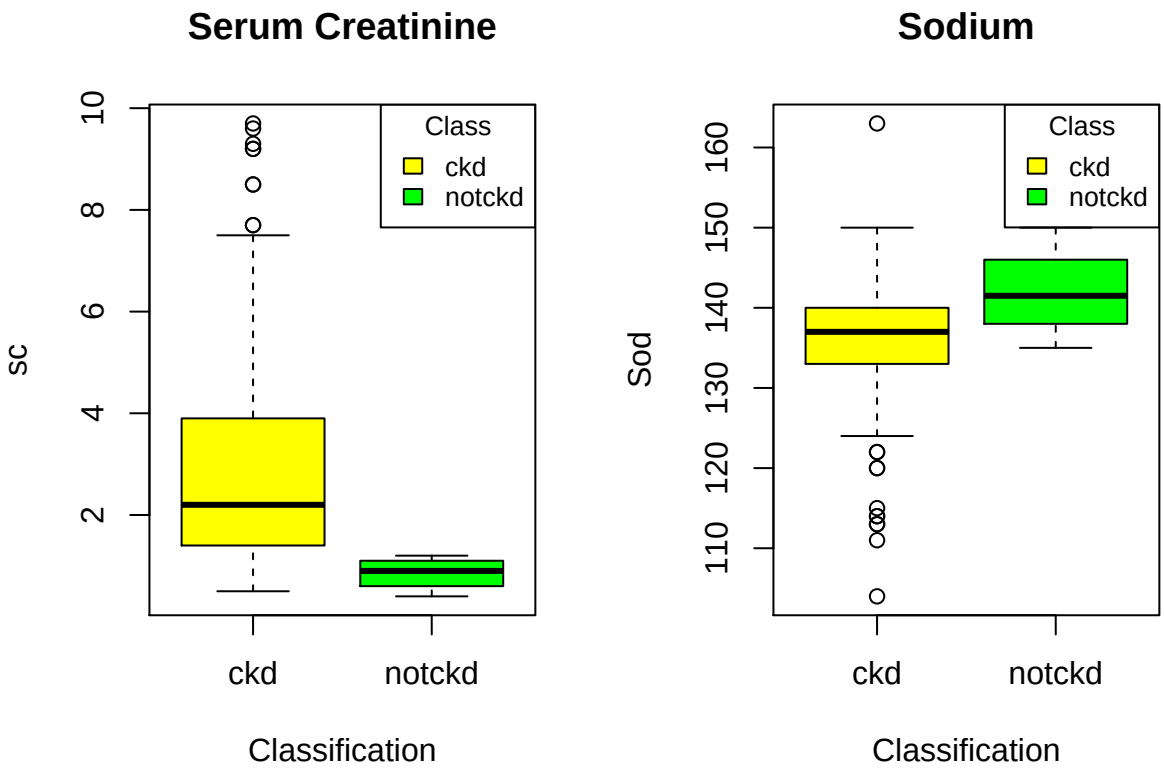
```
values_bgr<-boxplot.stats(datos$bgr)$out
cat("Valores extremos de Blood Glucose Random :",toString(values_bgr), "\n")
```

```
## Valores extremos de Blood Glucose Random : 423, 410, 490, 380, 263, 263, 264, 270, 425, 360, 360, 415
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_bgr, dnn = "Count values_bgr:")
```

```
## Count values_bgr:
## 263 264 268 269 270 273 280 288 294 295 297 298 303 307 308 309 323 341 352 360
##   2   1   1   1   1   1   1   1   1   1   1   1   2   1   1   1   1   1   1   2
## 380 410 415 423 424 425 447 463 490
##   1   1   1   1   2   1   1   1   2
```

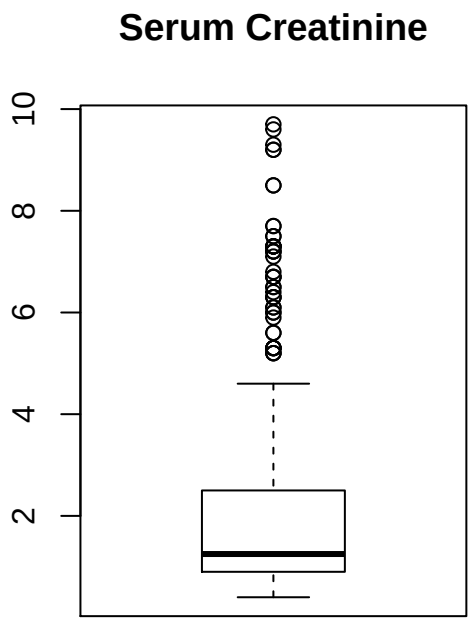
```
par(mfrow=c(1,2))
boxplot(datos$sc ~ datos$classification,col = c("yellow", "green"),
                            xlab="Classification", ylab="sc", main='Serum Creatinine',)
legend("topright", legend=c("ckd", "notckd"), c("yellow", "green"), title="Class", fill=c("yellow", "gr

boxplot(datos$sod ~ datos$classification,col = c("yellow", "green"),
                       xlab="Classification", ylab="Sod",main='Sodium')
legend("topright", legend=c("ckd", "notckd"), col=c("yellow", "green"), title="Class", fill=c("yellow",
```

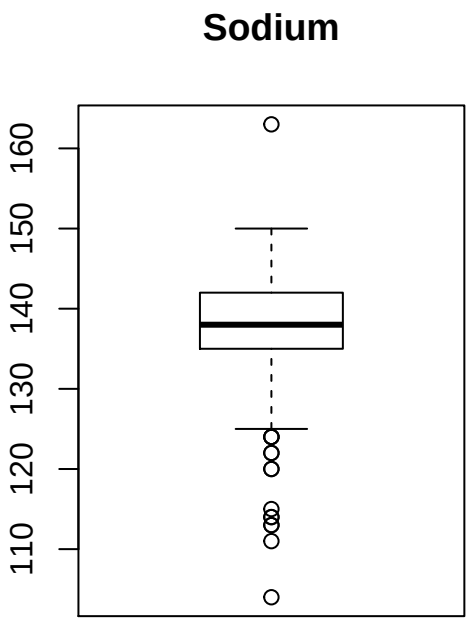


Los valores de Creatinina son mucho más altos en los pacientes con enfermedad. El Sodio mantienen, en general, una media más baja que pacientes diagnosticados.También vemos algún valor alterado superior.

Los valores extremos encontrados en las varibles, considerándolas en su globalidad, son:

```
par(mfrow=c(1,2))
boxplot(datos$sc,main='Serum Creatinine')
boxplot(datos$sod,main='Sodium')
```






```
values_sc<-boxplot.stats(datos$sc)$out
cat("Valores extremos de Serum Creatinine :",toString(values_sc), "\n")
```

```
## Valores extremos de Serum Creatinine : 7.2, 9.6, 5.2, 7.7, 7.3, 5.2, 7.7, 6.3, 5.9, 9.7, 7.3, 6.4, 6
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_sc, dnn = "Count values_sc:")
```

```
## Count values_sc:
## 5.2 5.3 5.6 5.9   6 6.1 6.3 6.4 6.5 6.7 6.8 7.1 7.2 7.3 7.5 7.7 8.5 9.2 9.3 9.6
##   3   5   2   1   2   2   3   1   2   3   1   1   3   5   2   2   2   2   1   1
## 9.7
##   1
```

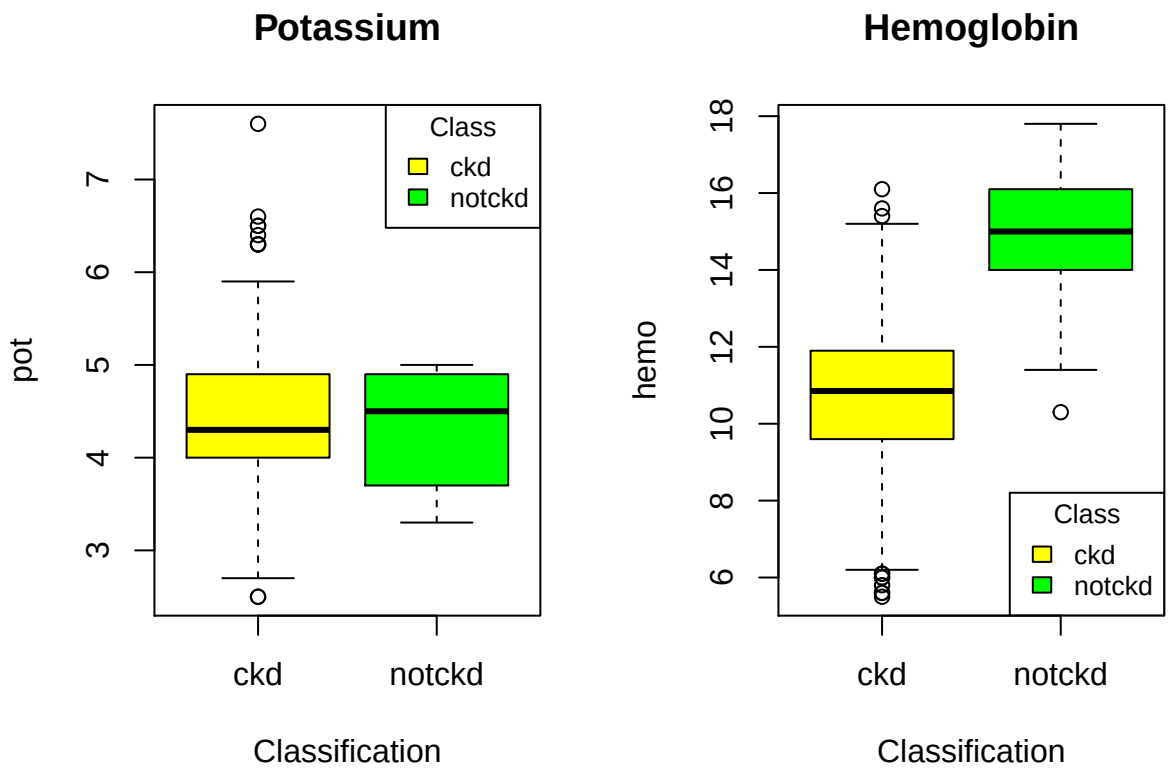
```
values_sod<-boxplot.stats(datos$sod)$out
cat("Valores extremos de Sodium :",toString(values_sod), "\n")
```

```
## Valores extremos de Sodium : 111, 104, 114, 163, 122, 124, 115, 113, 113, 122, 124, 114, 120, 120, 1
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_sod, dnn = "Count values_sod:")
```

```
## Count values_sod:
## 104 111 113 114 115 120 122 124 163
##   1   1   2   2   1   2   2   3   1
```

```r
par(mfrow=c(1,2))
boxplot(datos$pot ~ datos$classification,col = c("yellow", "green"),
                          xlab="Classification", ylab="pot", main='Potassium',)

legend("topright", legend=c("ckd", "notckd"), c("yellow", "green"),
       title="Class", fill=c("yellow", "green"), cex=0.8)

boxplot(datos$hemo ~ datos$classification,col = c("yellow", "green"),
                          xlab="Classification", ylab="hemo", main='Hemoglobin')

legend("bottomright", legend=c("ckd", "notckd"), col=c("yellow", "green"),
       title="Class", fill=c("yellow", "green"), cex=0.8)
```

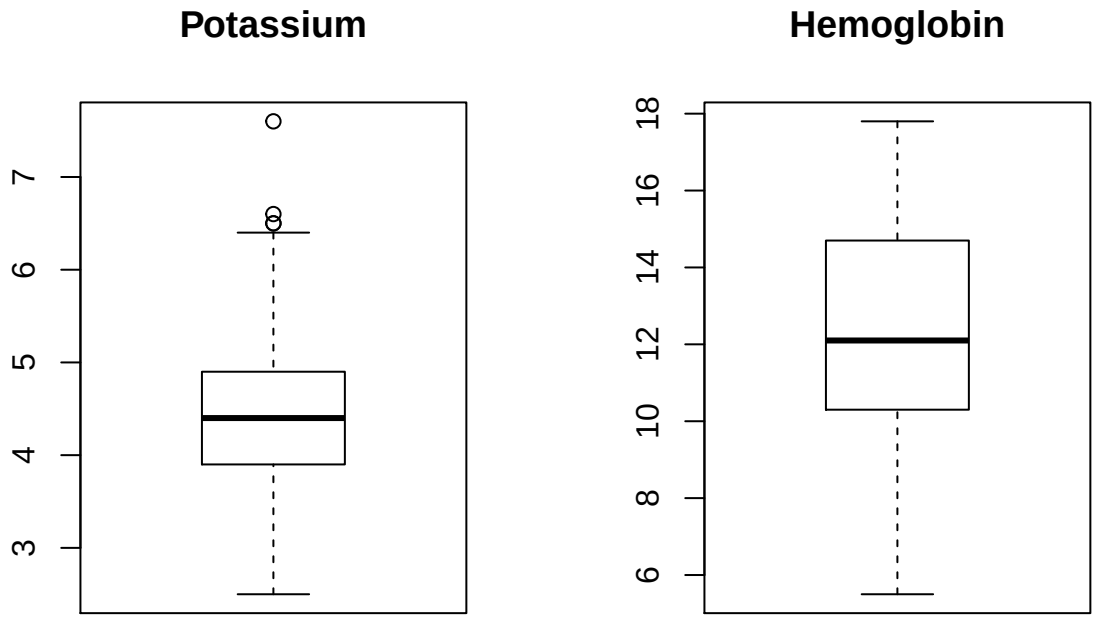


Vemos que el postasio en media se mantiene bastante cercana a la media de pacientes sanos. Se pueden observar valores outliers en pacientes enfermos. Podemos pensar que el potasio puede ser un indicador donde no quede muy claro si el paciente está sano o no, a no ser de tener un indicador muy alterado.

La Hemoglobina se encuentra más baja que en pacientes sanos. Tenemos algunos valores alterados que, estudiandolos, estarían dentro de los parámetros normales. Igual que algún valor en pacientes sanos que podría dar indicios de enfermedad. Seguramente, estos pacientes no serán determinados por la hemoglobina pero sí por otros parámetros.

Los valores extremos encontradoss en las varibles (mirándolas en su globalidad) son:

```r
par(mfrow=c(1,2))
boxplot(datos$pot,main='Potassium')
boxplot(datos$hemo,main='Hemoglobin')
```




```
values_pot<-boxplot.stats(datos$pot)$out
cat("Valores extremos de Potassium :",toString(values_pot), "\n")
```

```
## Valores extremos de Potassium : 6.6, 7.6, 6.5, 6.5
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_pot, dnn = "Count values_pot:")
```

```
## Count values_pot:
## 6.5 6.6 7.6
##   2   1   1
```

```
values_hemo<-boxplot.stats(datos$hemo)$out
cat("Valores extremos de Hemoglobin :",toString(values_hemo), "\n")
```

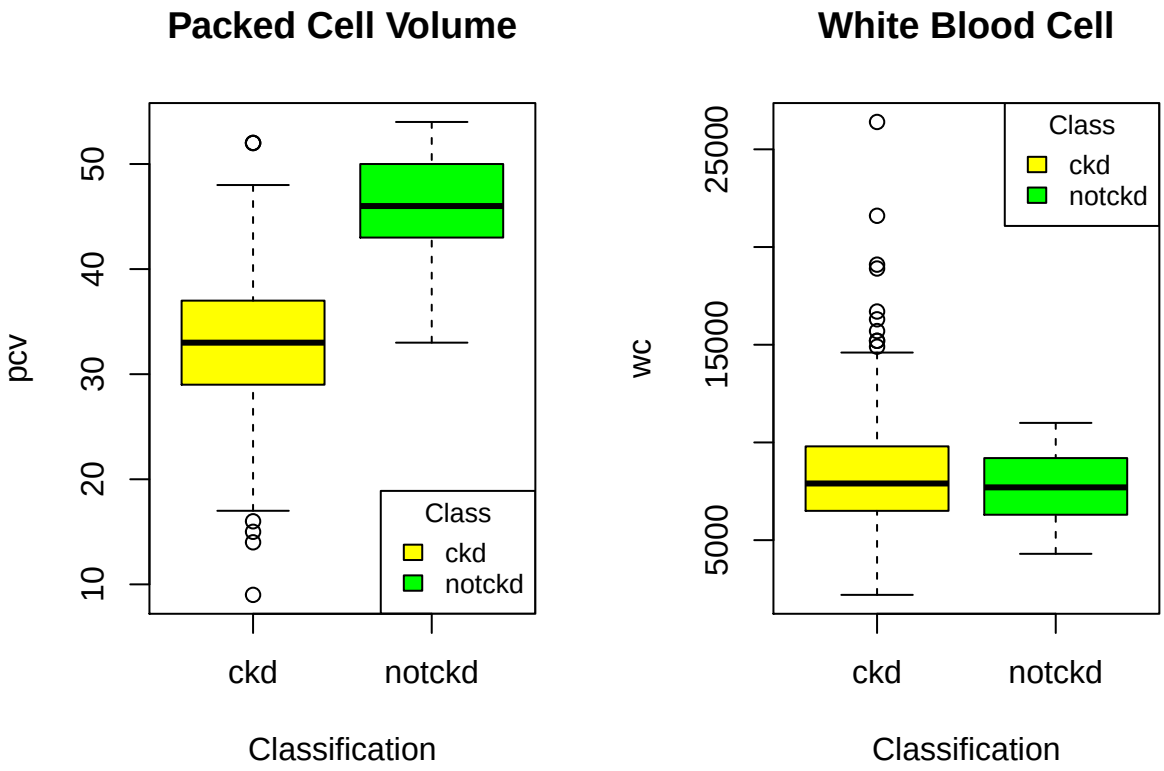
```
## Valores extremos de Hemoglobin :
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_hemo, dnn = "Count values_hemo:")
```

```
## < table of extent 0 >
```

En la Hemoglobina vemos que no tenemos valores extremos si miramos la variable en conjunto.

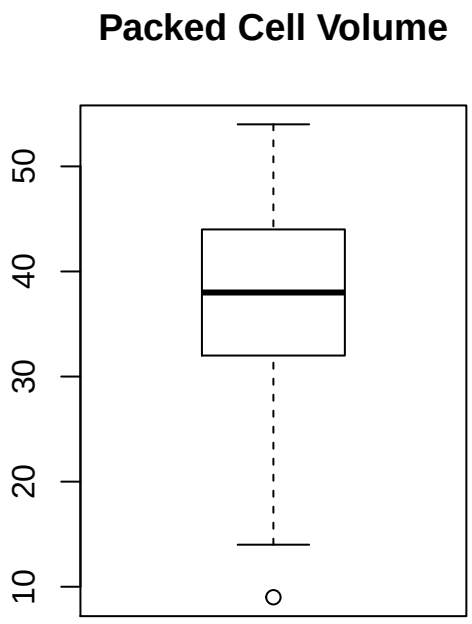
```
par(mfrow=c(1,2))
boxplot(datos$pcv ~ datos$classification,col = c("yellow", "green"),
                          xlab="Classification", ylab="pcv", main='Packed Cell Volume',)

legend("bottomright", legend=c("ckd", "notckd"), c("yellow", "green"),
       title="Class", fill=c("yellow", "green"), cex=0.8)

boxplot(datos$wc ~ datos$classification,col = c("yellow", "green"),
                          xlab="Classification", ylab="wc", main='White Blood Cell')

legend("topright", legend=c("ckd", "notckd"), col=c("yellow", "green"),
       title="Class", fill=c("yellow", "green"), cex=0.8)
```

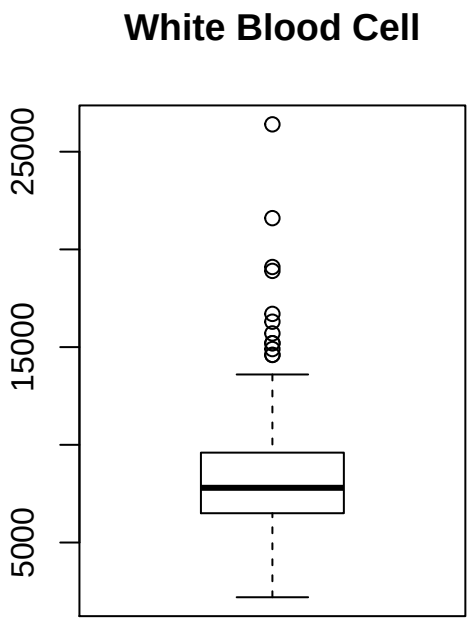


El volumen de células es muy inferior en pacientes crónicos que en pacientes no crónicos. Respecto a los glóbulos blancos, la media de pacientes crónicos es muy parecida a la media de pacientes no crónicos. Los valores outliers son característicos de los pacientes crónicos. Por tanto, si no tienen valores muy significativos, será difícil determinar si són crónicos o no. Dependeran también de otros parámetros para acabar de decidir.

Los valores extremos encontradoss en las varibles (mirándolas en su globalidad) son:

```
par(mfrow=c(1,2))
boxplot(datos$pcv,main='Packed Cell Volume')
boxplot(datos$wc,main='White Blood Cell')
```






```
values_pcv<-boxplot.stats(datos$pcv)$out
cat("Valores extremos de Packed Cell Volume :",toString(values_pcv), "\n")
```

```
## Valores extremos de Packed Cell Volume : 9
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_pcv, dnn = "Count values_pcv:")
```

```
## Count values_pcv:
## 9
## 1
```

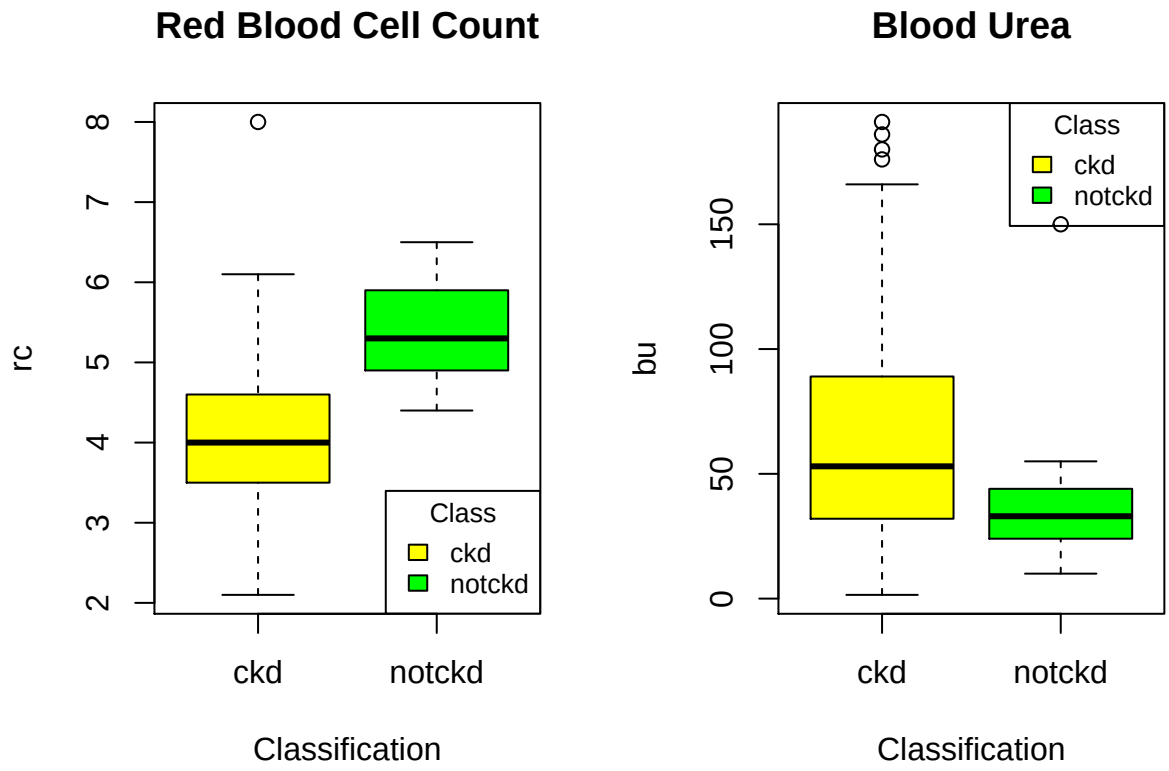
```
values_wc<-boxplot.stats(datos$wc)$out
cat("Valores extremos de White Blood Cell :",toString(values_wc), "\n")
```

```
## Valores extremos de White Blood Cell : 18900, 21600, 14600, 14900, 15200, 16300, 15200, 14600, 19100
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_wc, dnn = "Count values_wc:")
```

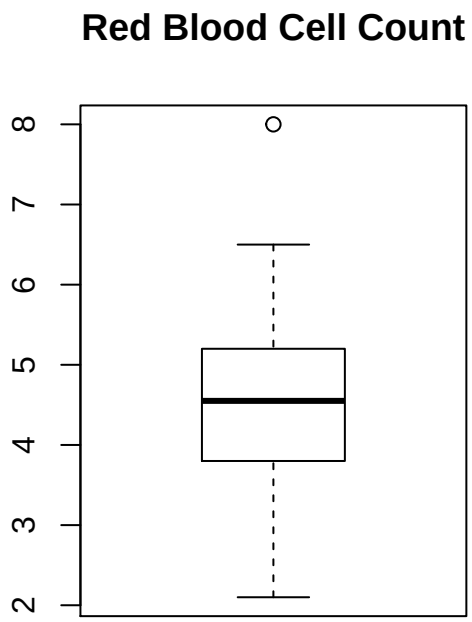
```
## Count values_wc:
## 14600 14900 15200 15700 16300 16700 18900 19100 21600 26400
##     2     1     2     1     1     1     1     1     1     1
```

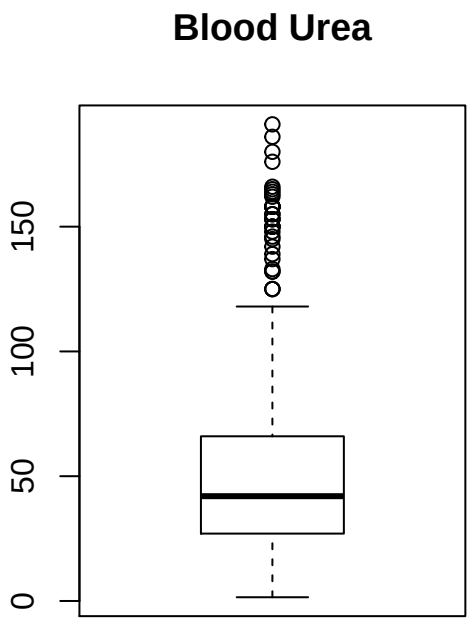
```r
par(mfrow=c(1,2))
boxplot(datos$rc ~ datos$classification,col = c("yellow", "green"),
                          xlab="Classification", ylab="rc", main='Red Blood Cell Count',)

legend("bottomright", legend=c("ckd", "notckd"), c("yellow", "green"),
       title="Class", fill=c("yellow", "green"), cex=0.8)

boxplot(datos$bu ~ datos$classification,col = c("yellow", "green"),
                          xlab="Classification", ylab="bu", main='Blood Urea')

legend("topright", legend=c("ckd", "notckd"), col=c("yellow", "green"),
       title="Class", fill=c("yellow", "green"), cex=0.8)
```



Vemos que los valores en pacientes crónicos de glóbulos rojos es inferior y la urea en sangre es más alta.

```r
par(mfrow=c(1,2))
boxplot(datos$rc,main='Red Blood Cell Count')
boxplot(datos$bu,main='Blood Urea')
```






```
values_rc<-boxplot.stats(datos$rc)$out
cat("Valores extremos de Red Blood Cell Count :",toString(values_rc), "\n")
```

```
## Valores extremos de Red Blood Cell Count : 8
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_rc, dnn = "Count values_rc:")
```

```
## Count values_rc:
## 8
## 1
```

```
values_bu<-boxplot.stats(datos$bu)$out
cat("Valores extremos de Blood Urea :",toString(values_bu), "\n")
```

```
## Valores extremos de Blood Urea : 162, 148, 180, 163, 155, 153, 158, 164, 155, 142, 153, 139, 186, 125
```

```
#contabilizamos cuantos tenemos de cada uno:
table(values_bu, dnn = "Count values_bu:")
```

```
## Count values_bu:
## 125 132 133 137 139 142 145 146 148 150 153 155 158 162 163 164 165 166 176 180
##   3   2   1   1   1   1   1   1   1   4   3   2   6   1   1   1   1   1   1   1
## 186 191
##   1   1
```

Como podemos observar en todos los campos tenemos valores extremos. Como se ha indicado anteriormente, no prescindiremos de ellos ya que pueden aportar información al estudio. En el proceso de limpieza ya se han eliminado los valores que se consideraban fuera del rango o improbables posiblemente debido a una mala introducción de ese dato.

```
summary(datos)
```

```
##        id               age              bp               sg
##  Min.   :  0.00   Min.   : 2.00   Min.   : 50.00   Min.   :1.005
##  1st Qu.: 99.75   1st Qu.:42.00   1st Qu.: 70.00   1st Qu.:1.015
##  Median :199.50   Median :54.00   Median : 80.00   Median :1.020
##  Mean   :199.50   Mean   :51.47   Mean   : 76.45   Mean   :1.018
##  3rd Qu.:299.25   3rd Qu.:64.00   3rd Qu.: 80.00   3rd Qu.:1.020
##  Max.   :399.00   Max.   :90.00   Max.   :180.00   Max.   :1.025
##        al            su              rbc             pc              pcc
##  Min.   :0.0   Min.   :0.000   abnormal: 47   abnormal: 76   notpresent:358
##  1st Qu.:0.0   1st Qu.:0.000   normal  :353   normal  :324   present   : 42
##  Median :0.0   Median :0.000
##  Mean   :0.9   Mean   :0.395
##  3rd Qu.:2.0   3rd Qu.:0.000
##  Max.   :5.0   Max.   :5.000
##          ba           bgr            bu               sc
##  notpresent:378   Min.   : 22   Min.   :  1.50   Min.   :0.400
##  present   : 22   1st Qu.:100   1st Qu.: 27.00   1st Qu.:0.900
##                   Median :122   Median : 42.00   Median :1.250
##                   Mean   :147   Mean   : 54.16   Mean   :2.152
##                   3rd Qu.:165   3rd Qu.: 66.00   3rd Qu.:2.500
##                   Max.   :490   Max.   :191.00   Max.   :9.700
##       sod             pot             hemo            pcv
##  Min.   :104.0   Min.   :2.500   Min.   : 5.50   Min.   : 9.00
##  1st Qu.:135.0   1st Qu.:3.900   1st Qu.:10.30   1st Qu.:32.00
##  Median :138.0   Median :4.400   Median :12.10   Median :38.00
##  Mean   :138.1   Mean   :4.392   Mean   :12.31   Mean   :37.86
##  3rd Qu.:142.0   3rd Qu.:4.900   3rd Qu.:14.70   3rd Qu.:44.00
##  Max.   :163.0   Max.   :7.600   Max.   :17.80   Max.   :54.00
##       wc              rc          htn        dm        cad        appet
##  Min.   : 2200   Min.   :2.100   no :253   no :263   no :366   good:318
##  1st Qu.: 6500   1st Qu.:3.800   yes:147   yes:137   yes: 34   poor: 82
##  Median : 7800   Median :4.550
##  Mean   : 8202   Mean   :4.531
##  3rd Qu.: 9600   3rd Qu.:5.200
##  Max.   :26400   Max.   :8.000
##    pe       ane      classification
##  no :324   no :340   ckd   :250
##  yes: 76   yes: 60   notckd:150
##
##
##
##
```

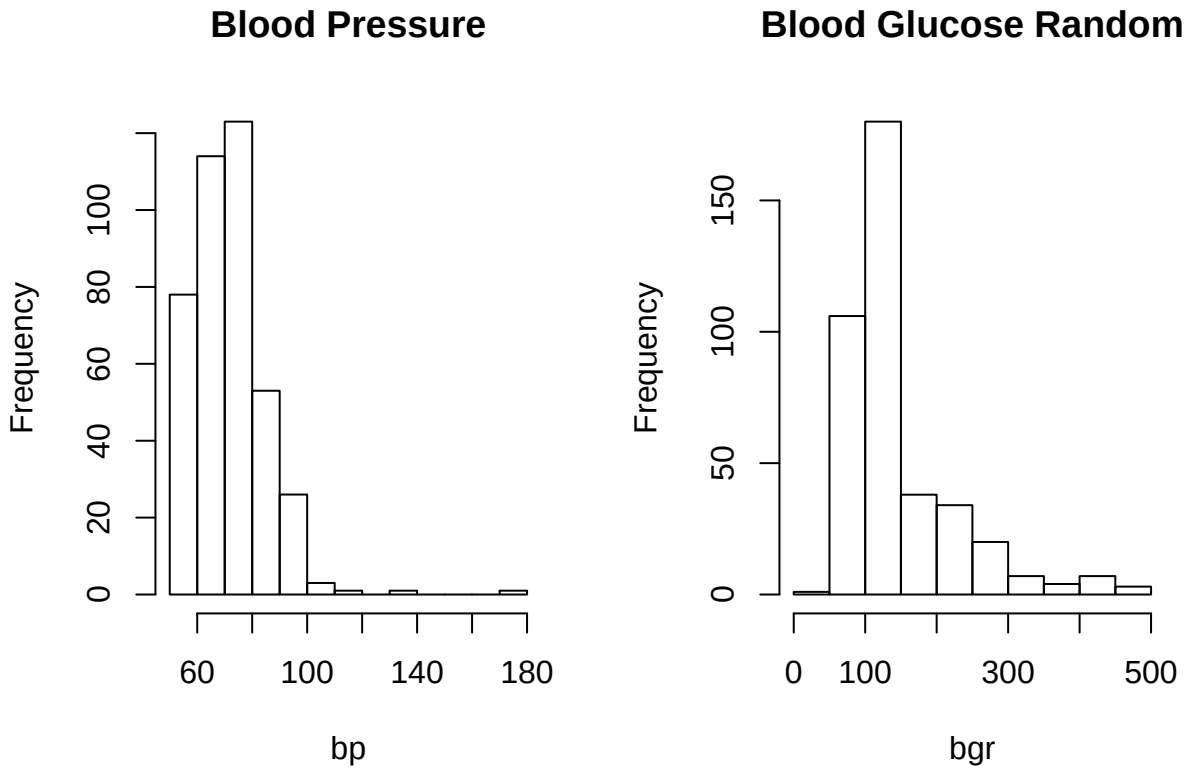
Tras el estudio de los diagramas de cajas, observamos, en parámetros generales que:

- La mayoría de los pacientes tienen una tensión arterial normal.

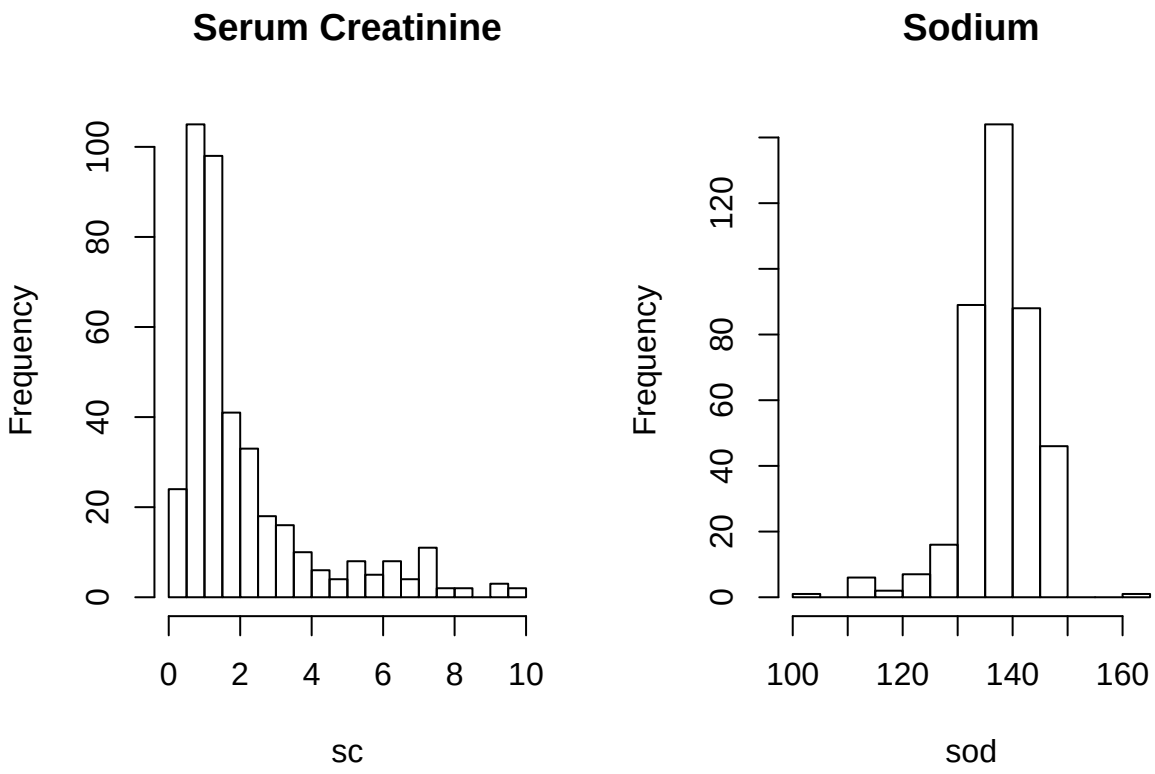
- Aproximadamente el 50% de los registros cuentan con valores de glucosa entre 100-150 unidades.
- Como media los iones de sodio y potasio están en rango.
- La creatinina muestra en varios registros valores fuera de rango, teniendo 2.152 unidades de media.
- Un tercio de los pacientes presentan, hipertensión y diabetes.
- Un 80% de los pacientes, tiene buen apetito y menos del 10% tiene una efermedad cardio-vascular de base.

HISTOGRAMAS

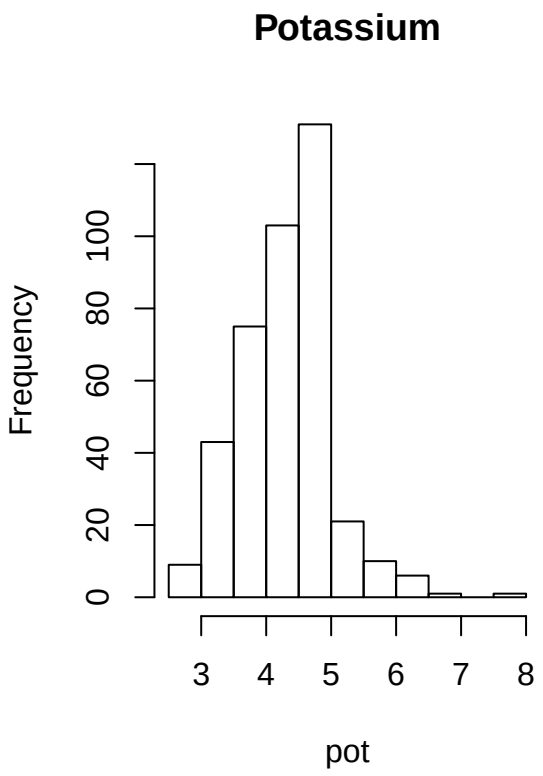
```
#HISTOGRAMAS
#=========================
par(mfrow=c(1,2))
hist(datos$bp,breaks=15,main='Blood Pressure', xlab="bp")
hist(datos$bgr,breaks=15,main='Blood Glucose Random', xlab="bgr")
```

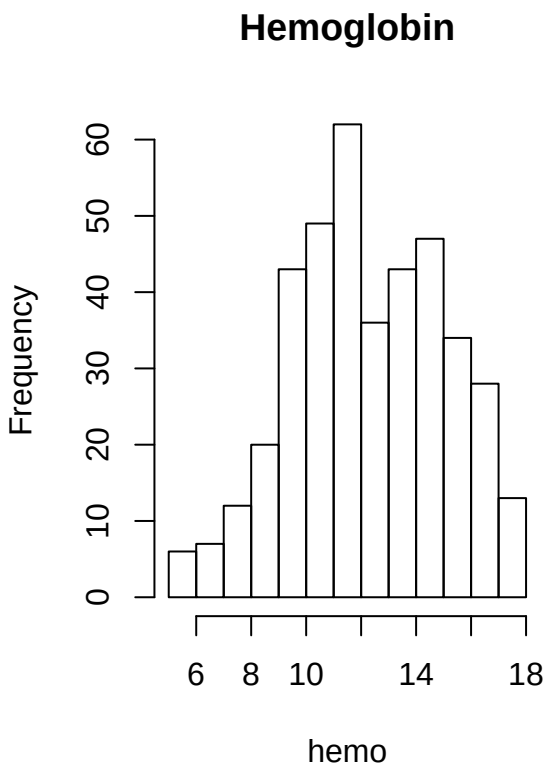



```
par(mfrow=c(1,2))
hist(datos$sc,breaks=15,main='Serum Creatinine', xlab="sc")
hist(datos$sod,breaks=15,main='Sodium', xlab="sod")
```

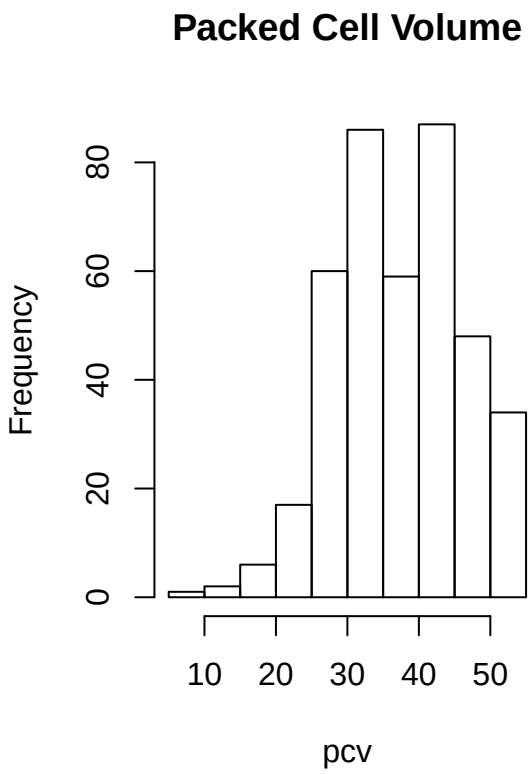


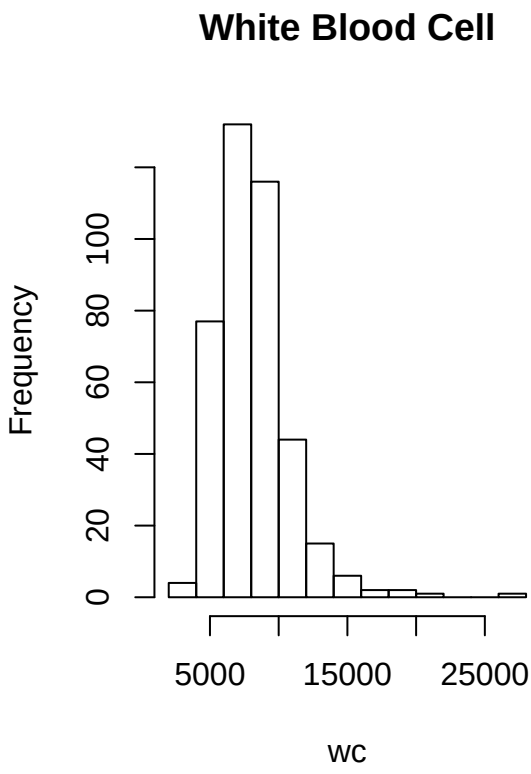

```
par(mfrow=c(1,2))
hist(datos$pot,breaks=15,main='Potassium', xlab="pot")
hist(datos$hemo,breaks=15,main='Hemoglobin', xlab="hemo")
```



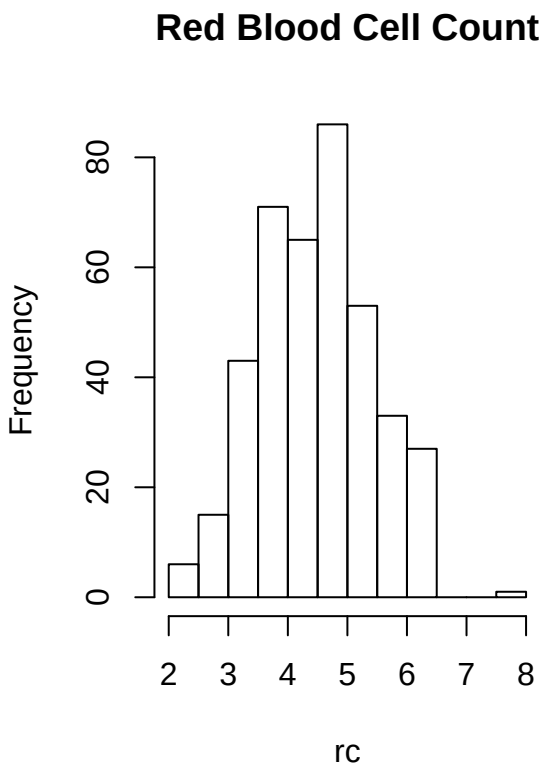


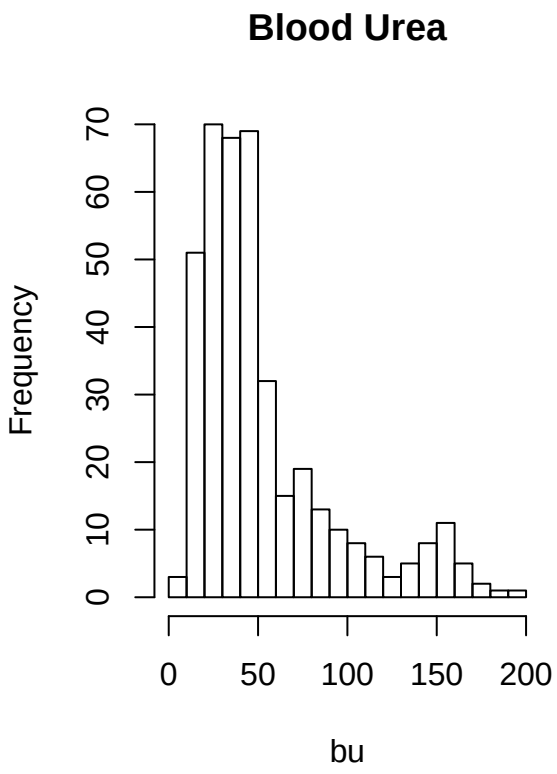

```
par(mfrow=c(1,2))
hist(datos$pcv,breaks=15,main='Packed Cell Volume', xlab="pcv")
hist(datos$wc,breaks=15,main='White Blood Cell', xlab="wc")
```






```
par(mfrow=c(1,2))
hist(datos$rc,breaks=15,main='Red Blood Cell Count', xlab="rc")
hist(datos$bu,breaks=15,main='Blood Urea', xlab="bu")
```

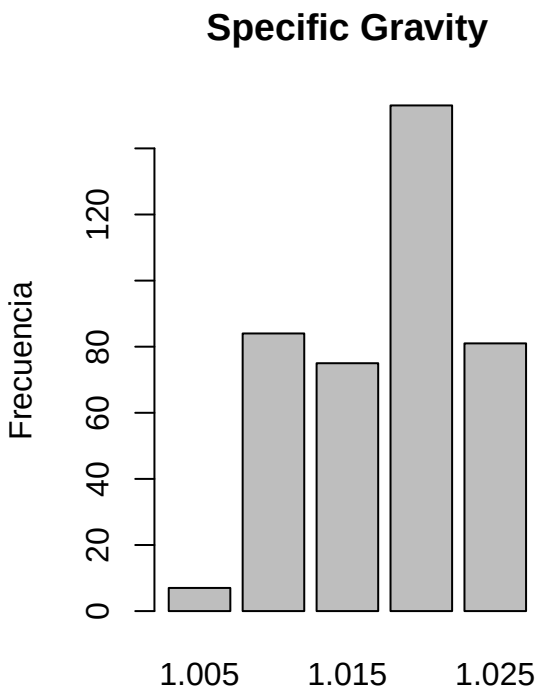


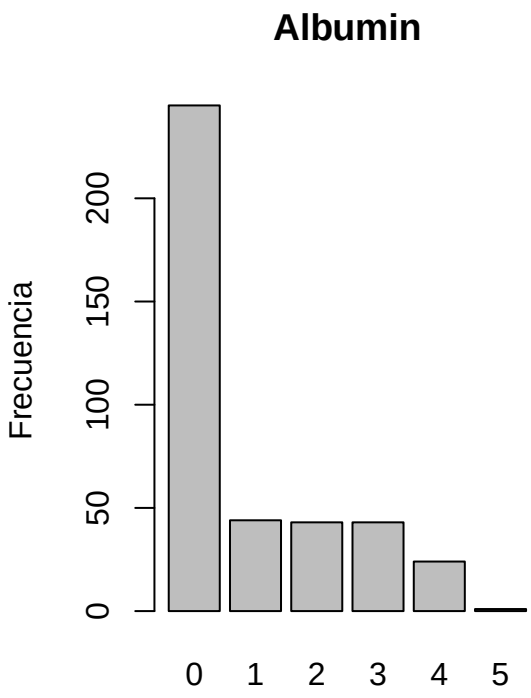


Comprobamos cómo se distribuyen los valores de los distintos atributos, y observamos que, solo parte de los registros, tienen valores en un rango normal. El estudio sobre ellos y sus combinaciones nos va a posibilitar clasificar a un paciente con insuficiencia renal crónica.

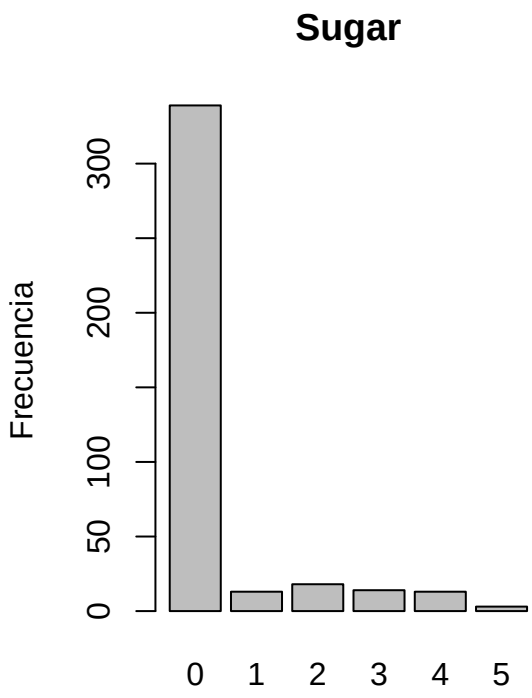
DIAGRAMAS DE BARRAS. vARIABLES CATEGÓRICAS

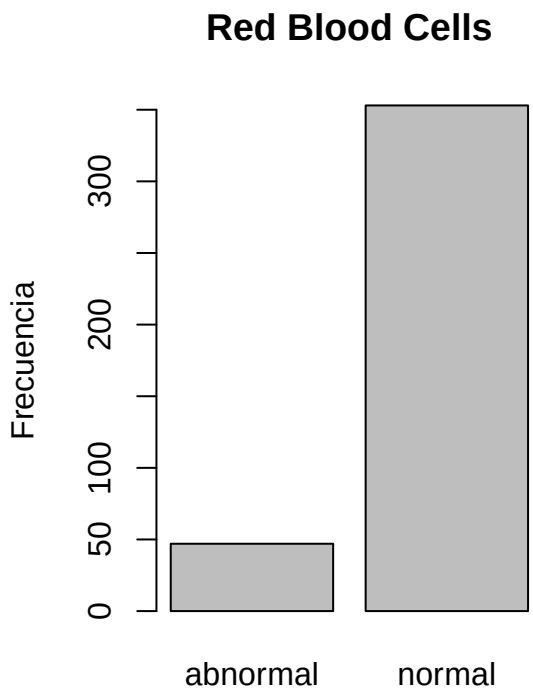
```
par(mfrow=c(1,2))
#Specific Gravity
###################

datos$sg<-as.factor(datos$sg)
plot(datos$sg, main="Specific Gravity", ylab="Frecuencia", col=c("gray"))
#Albumin
#############
datos$al<-as.factor(datos$al)
plot(datos$al, main="Albumin", ylab="Frecuencia", col=c("gray"))
```



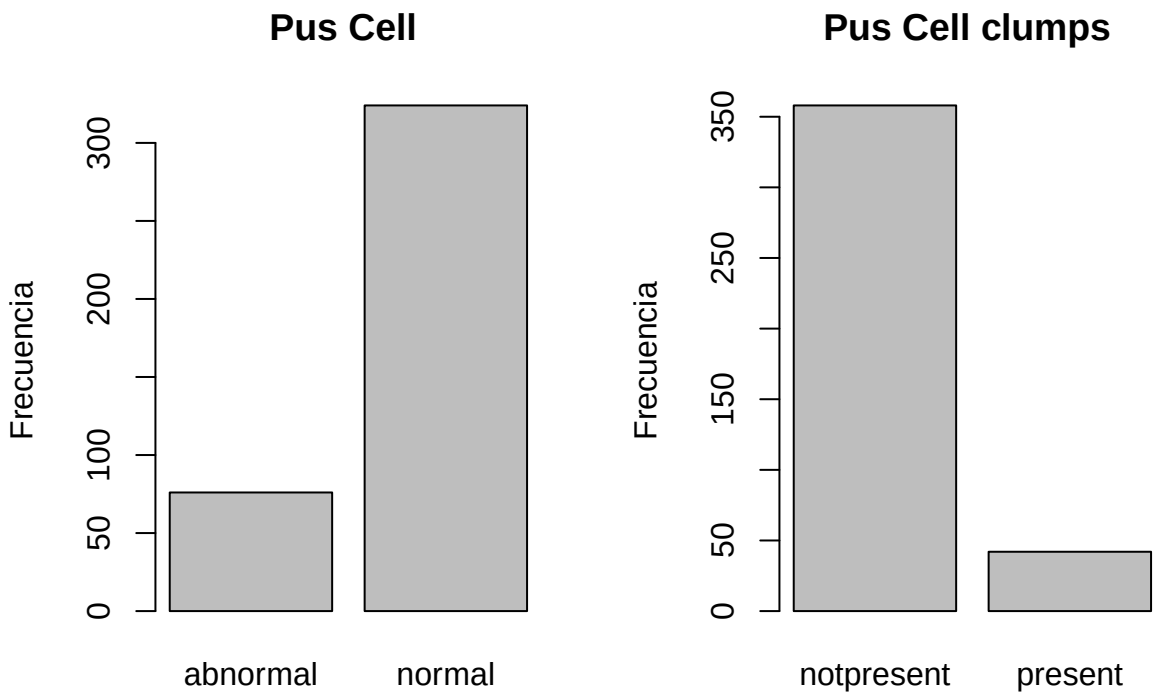



```
par(mfrow=c(1,2))
#Sugar
########

datos$su<-as.factor(datos$su)
plot(datos$su, main="Sugar", ylab="Frecuencia", col=c("gray"))

#Red Blood Cells
################

datos$rbc<-as.factor(datos$rbc)
plot(datos$rbc, main="Red Blood Cells", ylab="Frecuencia", col=c("gray"))
```

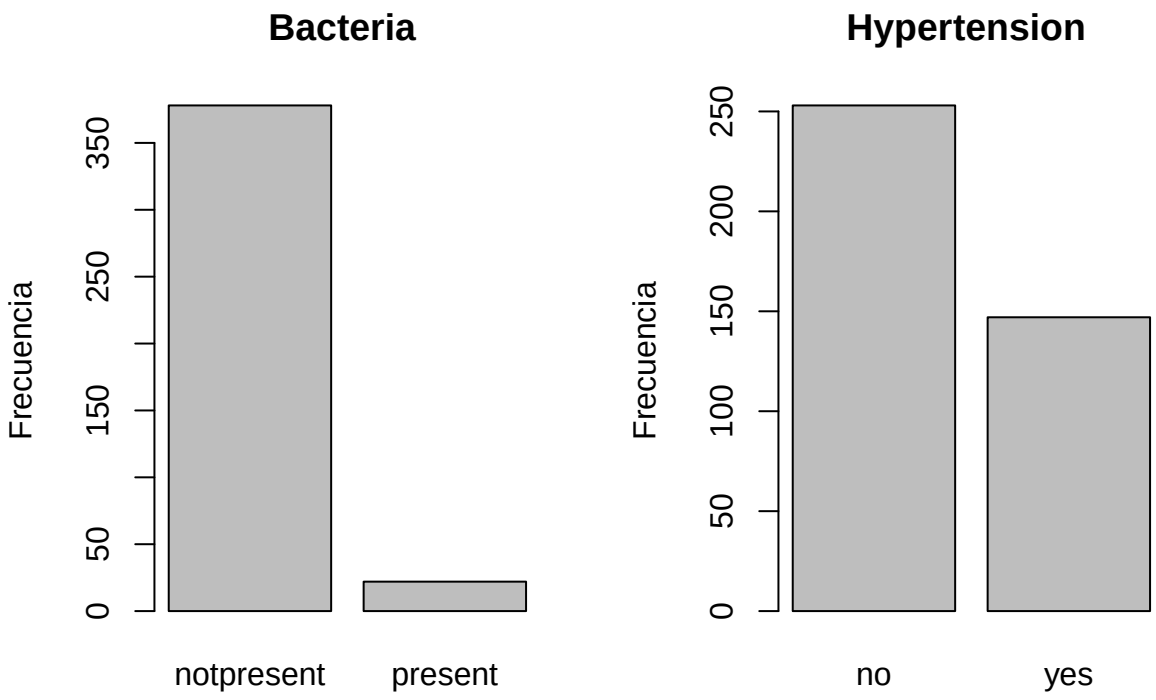




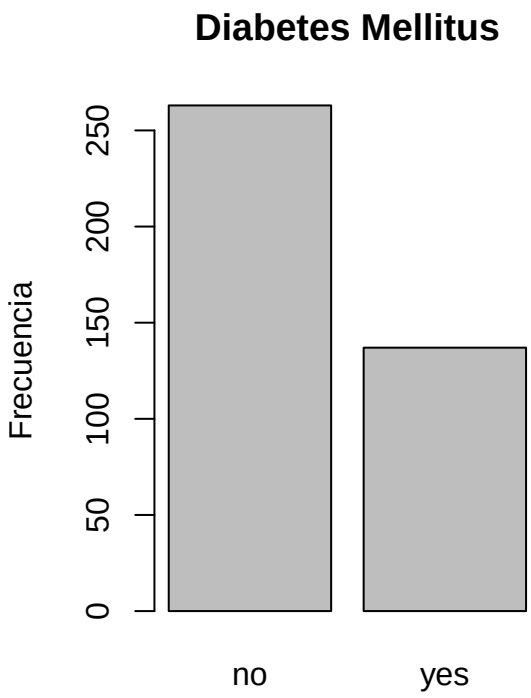

```
par(mfrow=c(1,2))
#Pus Cell
#########
plot(datos$pc, main="Pus Cell ", ylab="Frecuencia", col=c("gray"))

#Pus Cell clumps
################
plot(datos$pcc, main="Pus Cell clumps ", ylab="Frecuencia", col=c("gray"))
```

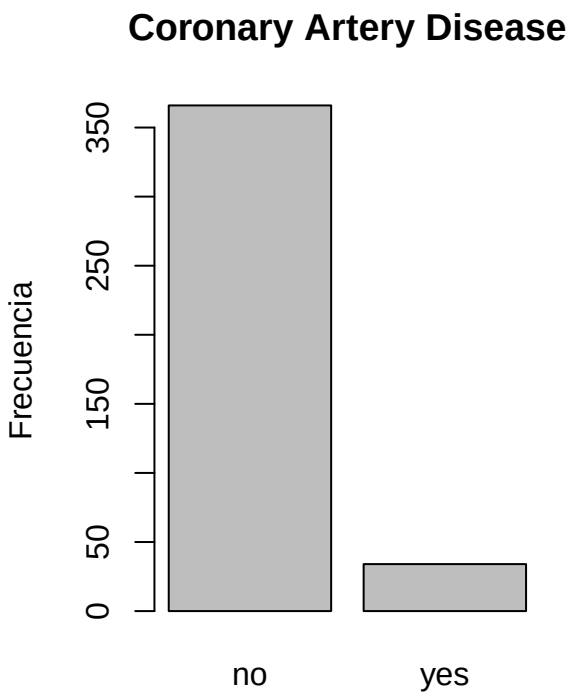



```
par(mfrow=c(1,2))
#Bacteria
#########
plot(datos$ba, main="Bacteria ", ylab="Frecuencia", col=c("gray"))

#Hypertension
################
plot(datos$htn, main="Hypertension ", ylab="Frecuencia", col=c("gray"))
```

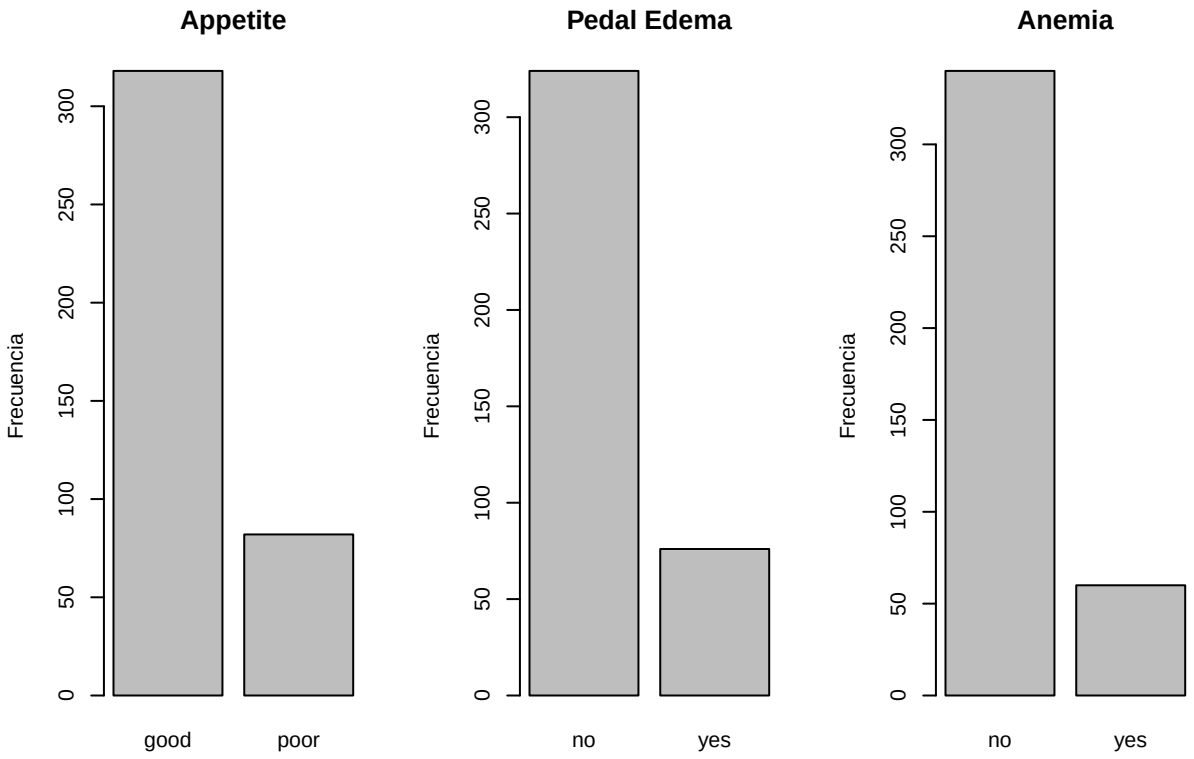



```
par(mfrow=c(1,2))
#Diabetes Mellitus
#########
plot(datos$dm, main="Diabetes Mellitus", ylab="Frecuencia", col=c("gray"))

#Coronary Artery Disease
########################
plot(datos$cad, main="Coronary Artery Disease", ylab="Frecuencia", col=c("gray"))
```






```
par(mfrow=c(1,3))
#.Appetite
#########
plot(datos$appet, main="Appetite", ylab="Frecuencia", col=c("gray"))

#Pedal Edema
############
plot(datos$pe, main="Pedal Edema", ylab="Frecuencia", col=c("gray"))

#Anemia
############
plot(datos$ane, main="Anemia", ylab="Frecuencia", col=c("gray"))
```



Del análisis de los gráficos en relación a las variables cualitativas, observamos:

- Existe un número moderado de pacientes con diabetes.
- Existe un número moderado de pacientes con hipertensión.
- La mayoría de los pacientes no presentan enfermedades coronarias ni anemia.
- Gran parte de los pacientes aseguran tener buen apetito
- Los glóbulos rojos son normales
- No hay presencia de pus ni de bacterias, mayoritariamente
- Mayoritariamente los pacientes no presentan ni anemia ni edema.

4.-Análisis de datos

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

De entre todos los atributos recogidos en el dataset, los que a priori mejor describen una posible insuficiencia renal crónica es la **creatinina**, valores de la bioquímica básica, como puedan ser rango de iones **sodio-potasio**, y **hematocrito (glóbulos rojos)**. Todos estos valores combinados con los valores que muestran si el paciente tiene **diabetes** e **hipertensión** nos ayudarán a categorizar este tipo de insuficiencia.

Es decir, sabiendo que la creatinina es un valor/factor importante en el diagnóstico de la IRC, intentaremos buscar si existen relaciones o dependencias con los valores típicos de cualquier bioquímica habitual, para poder establecer patrones que sirvan para la detección precoz de la enfermedad.

Estudiaremos principalmente como se distribuyen estas variables para luego profundizar en el estudio y sus relaciones.

- Estadística inferencial de la creatinina para conocer mejor este factor determinante.
- Estudio sobre diferencia de medias (creatinina y diabetes). Intentaremos averiguar si la diabetes tiene un papel importante en los valores de creatinina de los pacientes.
- Procederíamos de igual modo con la hipertensión regulada también por el riñon.
- Estudio sobre la anemia que es una variable que puede ayudarnos a diagnosticar de forma incipiente la IRC ya que es el riñon, quien regula a través de la eritropoyetina la estimulación de eritrocitos o glóbulos rojos. Estimaremos en este caso modelos de regresión para profundizar en el estudio de la anemia y descubrir qué variables puedan estar más fuertemente relacionadas.

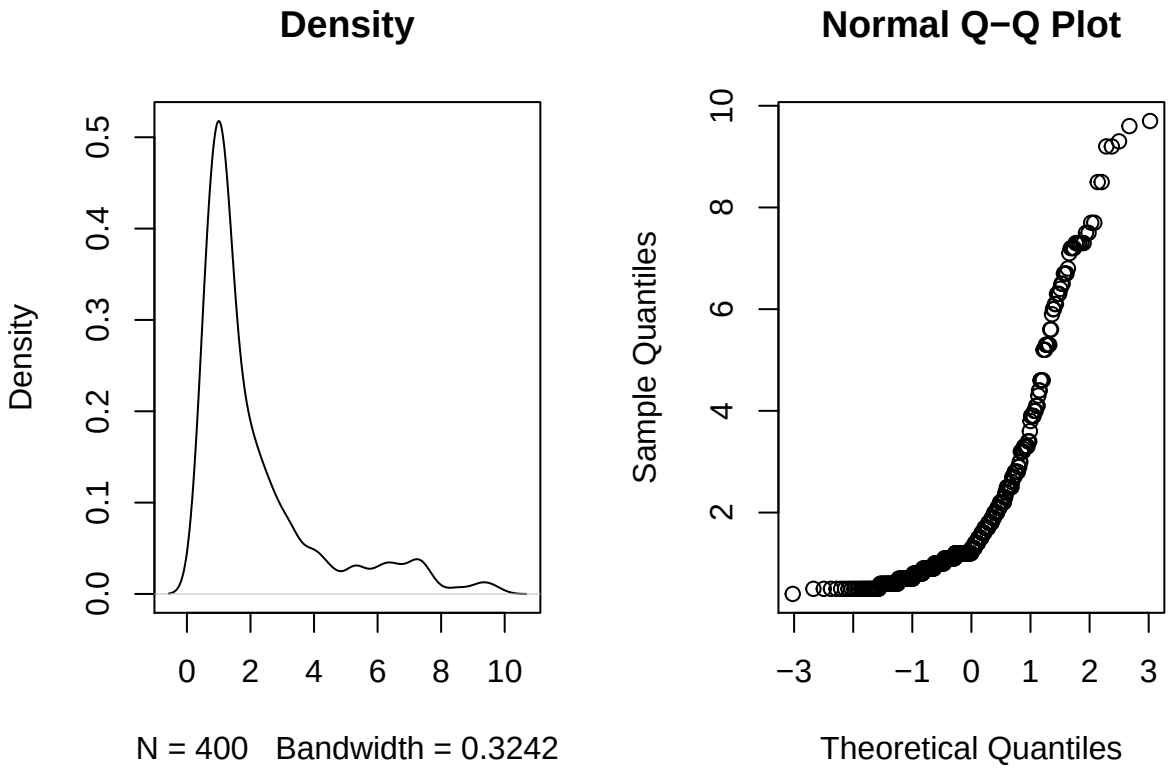
4.2-Comprobación de la normalidad y homogeneidad de la varianza.

Comprobemos si las variables Serum Creatinine (sc), Sodium (sod), Potassium (pot),Red Blood Cell Count(rc) se distribuyen normalmente.

SERUM CREATININE (sc)

Para comprobar si la variable 'sc' sigue una distribución normal, utilizaremos un test gráfico qqnorm, que dará como resultado una gráfica en la que si los puntos resultantes siguen una línea recta ascendente podremos asumir normalidad.

```
par(mfrow=c(1,2))
plot(density(datos$sc),main="Density")
qqnorm(datos$sc)
```



Parece seguir una línea recta en un rango muy corto en la parte central del gráfico, por lo que para poder asumir normalidad aplicaremos el test **Shapiro-Wilk Normality**. En este test se plantea como hipótesis nula que la muestra proviene de una población normalmente distribuida. Se rechazará la hipótesis nula si W que es el estadístico del test, es demasiado pequeño (oscila entre 0 y 1). Por otro lado si p-valor es menor que , se rechazará la hipótesis nula, y los datos no seguirían una distribución normal.

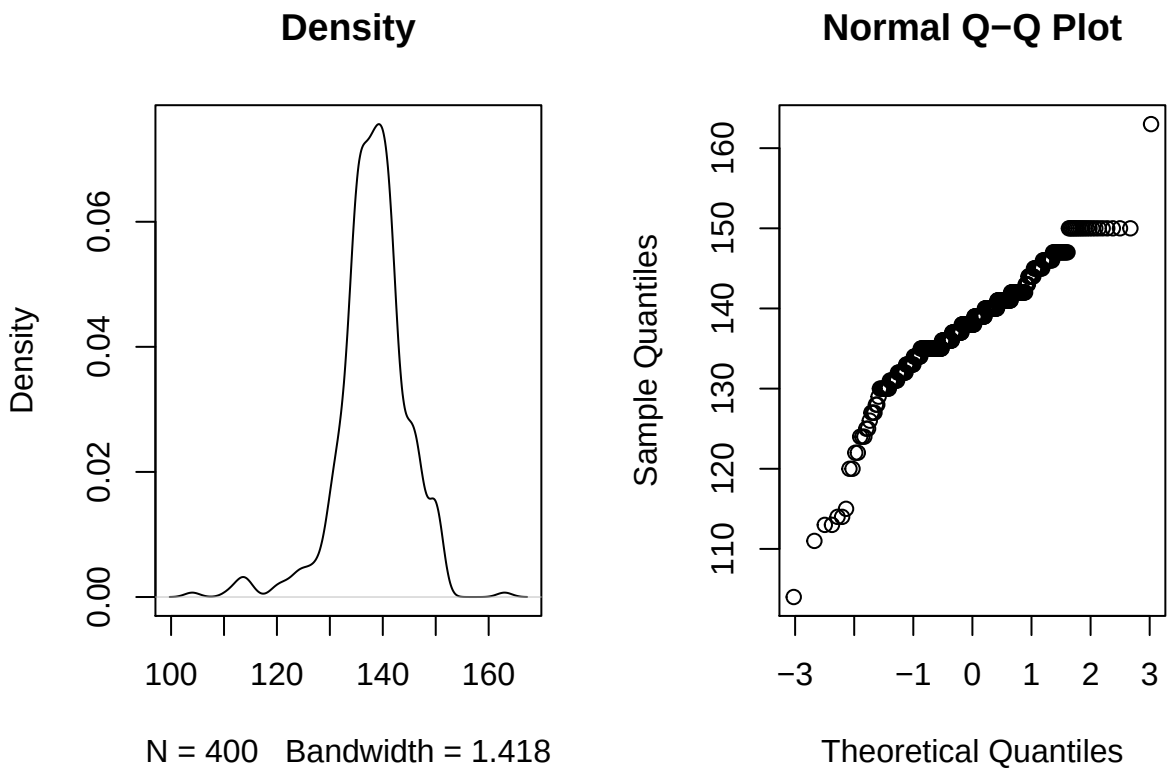
```
shapiro.test(datos$sc)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos$sc
## W = 0.75245, p-value < 2.2e-16
```

Como p-valor es menor que 0.05, no podemos asumir normalidad.

SODIUM (sod)

```
par(mfrow=c(1,2))
plot(density(datos$sod),main="Density")
qqnorm(datos$sod)
```



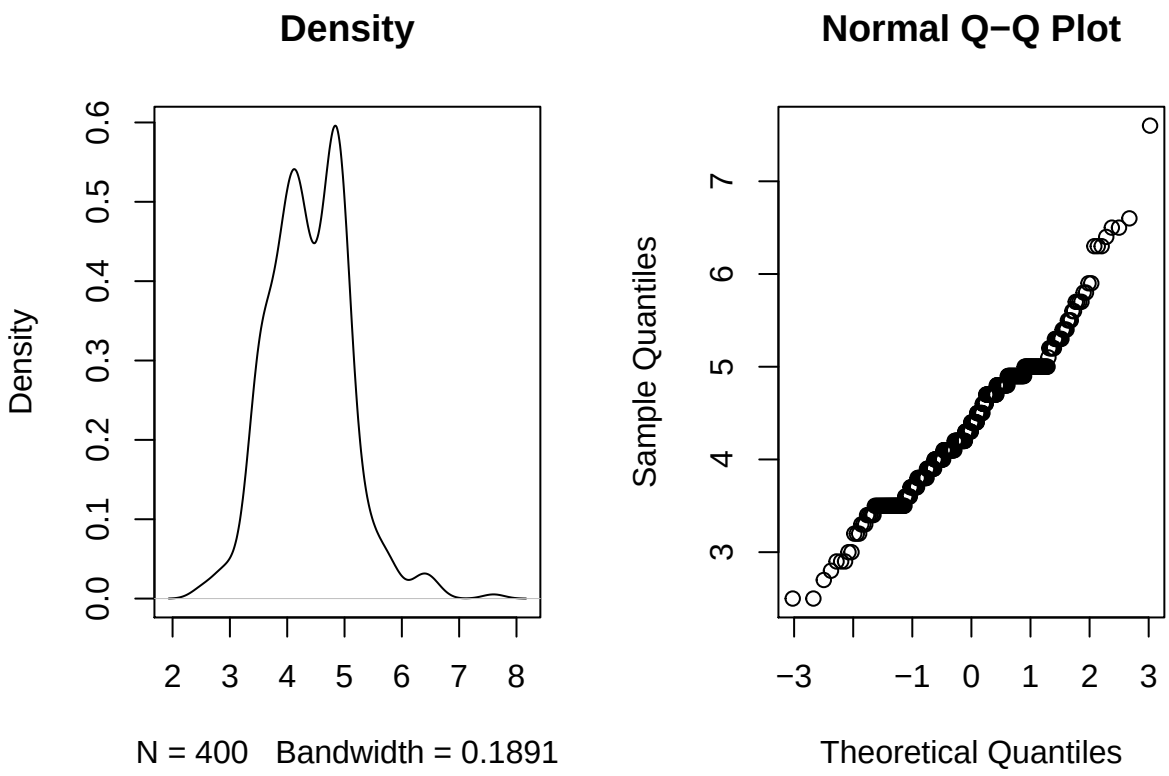
```r
shapiro.test(datos$sod)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos$sod
## W = 0.92558, p-value = 3.351e-13
```

Como p-valor es menor que 0.05, no podemos asumir normalidad.

POTASSIUM (pot)

```r
par(mfrow=c(1,2))
plot(density(datos$pot),main="Density")
qqnorm(datos$pot)
```



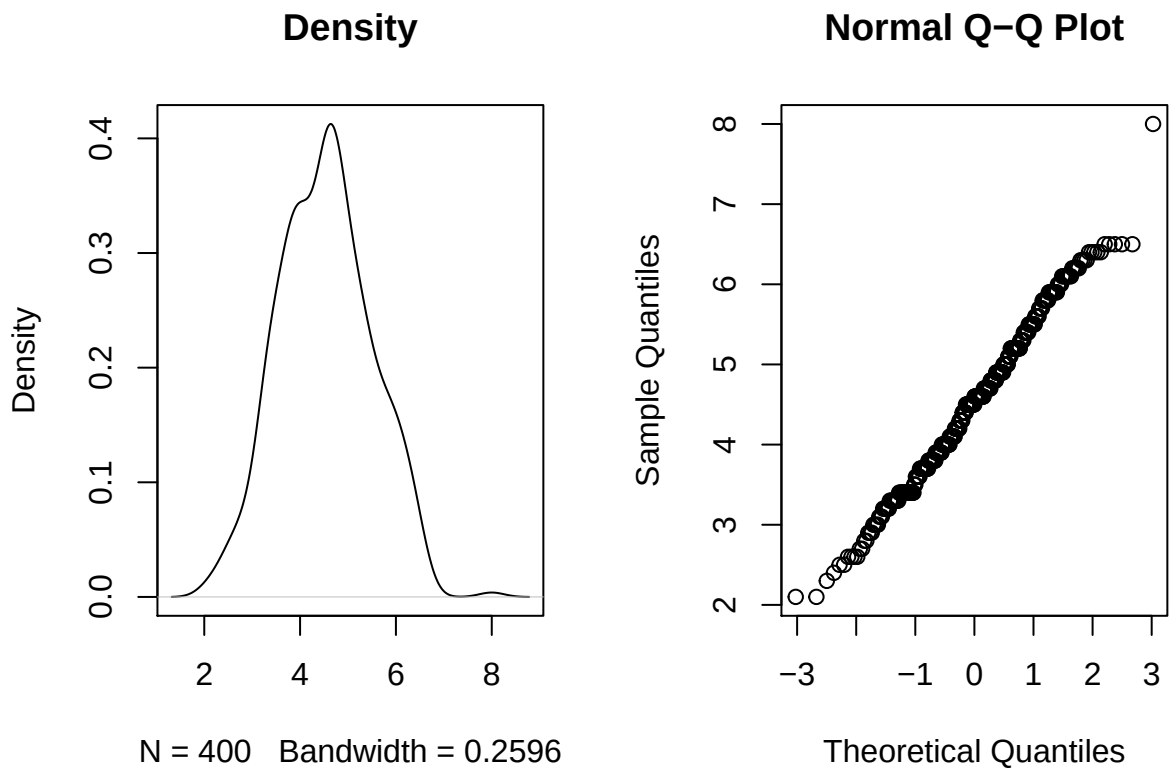

```r
shapiro.test(datos$pot)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos$pot
## W = 0.97361, p-value = 1.187e-06
```

Como p-valor es menor que 0.05, no podemos asumir normalidad.

RED BLOOD CELL Count(rc)

```
par(mfrow=c(1,2))
plot(density(datos$rc),main="Density")
qqnorm(datos$rc)
```




```
shapiro.test(datos$rc)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  datos$rc
## W = 0.99225, p-value = 0.0357
```

Como p-valor es menor que 0.05, no podemos asumir normalidad.

Por tanto, ninguna de las 3 variables estudiadas nos ha dado normalidad en los datos según el test de Saphiro-Wilk. De todas maneras, que nos haya salido esto no quiere decir que no puedan ser normalizables.Por el teorema del límite central, al tener más de 30 elementos (n > 30) en las observaciones, sea cual sea la distribución de la variable de interés, la distribución de la media muestral será aproximadamente una normal. Así, podemos aproximar a la normalidad y asumir que tenemos normalidad.

Ahora vamos a comprobar la homocedasticidad en los datos, es decir, la igualdad de las varianzas. Aplicamos el test de Fligner-Killeen entre la hemoglobina y la hipertensión:

```
fligner.test(hemo ~ htn, data = datos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  hemo by htn
## Fligner-Killeen:med chi-squared = 7.8375, df = 1, p-value = 0.005117
```

Obtenemos un p-value < 0.05, por tanto, se rechaza la hipótesis nula de homocedasticidad y se concluye que la variable hemo presenta varianzas estadísticamente diferentes grupos que tenemos en htn (es decir, pacientes hipertensos y pacientes no hipertensos).

4.3-Aplicación de pruebas estadísticas para comparar grupos de datos.Aplicad tres métodos de análisis diferentes.

ESTADÍSTICA INFERENCIAL

Teniendo en cuenta que un intervalo de confianza para un cierto parámetro con un nivel de confianza C%, es un intervalo calculado a partir de una muestra de manera que el procedimiento de cálculo garantiza que el C% de las muestras dan lugar a un intervalo que contiene el valor real del parámetro, procedemos a calcular el intervalo de confianza al 95% de la variable sc (serum creatinine) de los pacientes.

- Calculamos la media de la Creatinina
- La desviación típica
- Fijamos un nivel de confianza (1-α)=95%
- Calculamos el error estándar:

$$s_x = \frac{s}{\sqrt{n}}$$

- Calculamos el valor crítico que es el punto $t_{\alpha/2,n-1}$ tal que:

$$P(t_{n-1} >= t_{\alpha/2,n-1}) = \alpha/2$$

- Calculamos el margen de error:

$$t_{\alpha/2,n-1}\frac{s}{\sqrt{n}}$$

- Siendo el intervalo de confianza:

$$(\overline{x} - t_{\alpha/2,n-1}\frac{s}{\sqrt{n}}, \overline{x} + t_{\alpha/2,n-1}\frac{s}{\sqrt{n}})$$

```r
n<-nrow(datos)
me<-mean(datos$sc)
me
```

```
## [1] 2.151625
```

```r
s<- sd(datos$sc)
s
```

```
## [1] 1.95296
```

```r
t<-qt(0.025,399)
t
```

```
## [1] -1.965927
```

```r
es<-s/sqrt(n)
es
```

```
## [1] 0.09764798
```

```r
i1=me-(t*es)
i2=me+(t*es)
i1
```

```
## [1] 2.343594
```

```r
i2
```

```
## [1] 1.959656
```

El intervalo de confianza sería **(1.9596,2.3435)**, siendo la media muestral de **2.1516**.

Verificamos con el t-test que hemos realizado bien los cálculos:

```r
tt<-t.test( datos$sc, conf.level=0.95 )
tt
```

```
##
##  One Sample t-test
##
## data:  datos$sc
## t = 22.035, df = 399, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1.959656 2.343594
## sample estimates:
## mean of x
##  2.151625
```

Si repetimos en un número elevado de muestras el mismo procedimiento, el 95% de los intervalos encontrados contendrán el valor real de la media de la creatinina.

CONTRASTE DE HIPOTESIS PARA LA DIFERENCIA DE MEDIAS

Intentaremos dar respuesta a los siguientes interrogantes: - ¿Podemos aceptar que los pacientes que tienen diabetes tienen la creatinina más alta que aquellos que no han desarrollado esta enfermedad?(Nivel de confianza del 95%). Aplicaremos un test de t Student. Veamos primero si podemos asumir igualdad de varianzas.

```
sc_condm<- datos$sc[datos$dm=='yes']
sc_sindm<- datos$sc[datos$dm=='no']
var.test(sc_condm,sc_sindm)
```

```
##
##  F test to compare two variances
##
## data:  sc_condm and sc_sindm
## F = 1.705, num df = 136, denom df = 262, p-value = 0.0002491
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.279575 2.303482
## sample estimates:
## ratio of variances
##           1.704978
```

Observamos un p-valor <0.05 por lo que rechazaríamos la hipotesis nula que implica que no podemos considerar varianzas iguales.

Por lo tanto tendremos que usar un test no paramétrico, el conocido cmo test U de Mann-Whitney que aplicado sobre muestras independientes nos permite comparar sus medias:

La hipótesis nula, h_0, recoge el hecho que queremos someter a prueba. Y la hipótesis alternativa, h_1, es la que se ofrece como alternativa a la nula.

- $h_0 : \mu_1 - \mu_2 = 0$
- $h_1 : \mu_1 - \mu_2 > 0$

```
wilcox.test(datos$sc~datos$dm,alt="greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  datos$sc by datos$dm
## W = 6704, p-value = 1
## alternative hypothesis: true location shift is greater than 0
```

Como p-valor es mayor que 0.05 aceptamos la hipótesis nula, y por tanto no habría diferencias entre las medias de las creatininas de la población con diabetes en relación con aquellos pacientes que no han desarrollado la enfermedad.

REGRESIÓN LINEAL

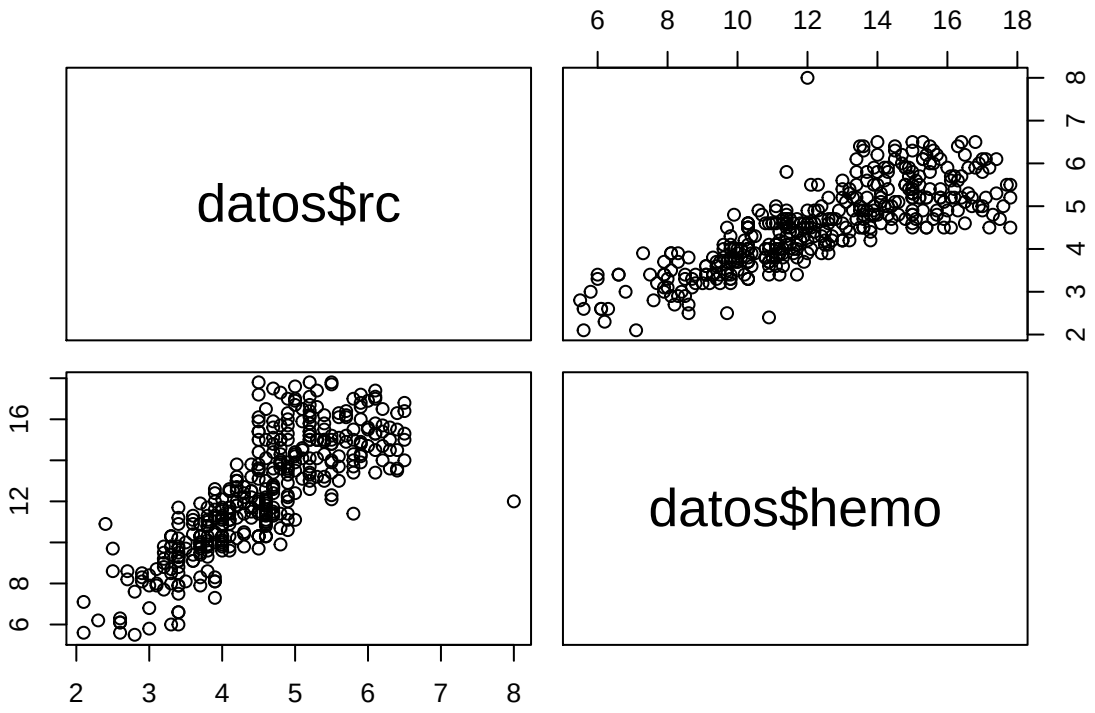
Estudiaremos la relación entre distintas variables y la identificación de factores de riesgo asociados al desarrollo de una Insuficiencia Renal Crónica.

*** GLOBULOS ROJOS~HEMOGLOBINA Y EN DIFERENTES NIVELES DE ALBÚMINA**

Vamos a estimar por mínimos cuadrados ordinarios un modelo lineal que explique la variable que indica el número de glóbulos rojos en función de la hemoglobina.

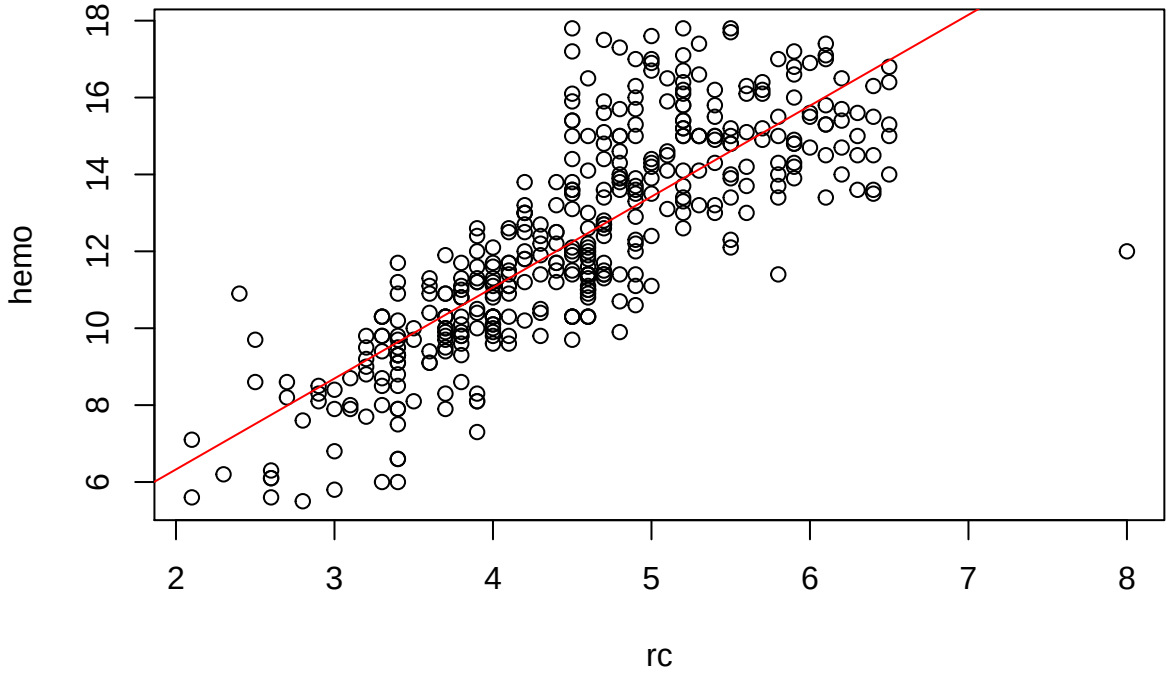
Representamos una matriz de diagramas de dispersión para comprobarlo gráficamente.

```
pairs(datos$rc~datos$hemo)
```



Estimamos el modelo:

```
#Estimación del modelo

model<- lm(hemo~rc, data=datos)

plot(datos$rc,datos$hemo,xlab="rc", ylab="hemo")
abline(model,col="red")
```




```
summary(model)
```

```
##
## Call:
## lm(formula = hemo ~ rc, data = datos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5183 -1.0172 -0.1574  0.9546  5.5593
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.59800    0.40049    3.99 7.86e-05 ***
## rc           2.36504    0.08649   27.34  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.651 on 398 degrees of freedom
## Multiple R-squared:  0.6526, Adjusted R-squared:  0.6517
## F-statistic: 747.7 on 1 and 398 DF,  p-value: < 2.2e-16
```

A la vista de los resultados, existe una relación lineal positiva moderada, entre ambas variables. Se observa que el coeficiente de determinación ajustado es: 0.6517 . Es decir, el **modelo explica el 65.17%** de la variabilidad de la variable hemoglobina.

La relación calculada anteriormente varía dependiendo de la nutrición de cada paciente.Haremos el mismo estudio dividiendo la muestra en 6 partes,1 para cada nivel de albúmina.(Cuanto más se aproxime a 5 el nivel de albúmina presentado, la nutrición del paciente será mejor).

```
#Nivel 0: La Albumin está a 0
#########
al_0 <- which(datos$al==0 )
data_0=datos[al_0,]
dim(data_0)
```

```
## [1] 245  26
```

Nivel 0: Obtenemos 245 registros.

```
#Nivel 1:  La Albumin está a 1
#########
al_1 <- which(datos$al==1 )
data_1=datos[al_1,]
dim(data_1)
```

```
## [1] 44 26
```

Nivel 1: Obtenemos 44 registros.

```
#Nivel 2: La Albumin está a 2
#########
al_2 <- which(datos$al==2 )
data_2=datos[al_2,]
dim(data_2)
```

```
## [1] 43 26
```

Nivel 2: Obtenemos 43 registros.

```
#Nivel 3: La Albumin está a 3
#########
al_3 <- which(datos$al==3 )
data_3=datos[al_3,]
dim(data_3)
```

```
## [1] 43 26
```

Nivel 3: Obtenemos 43 registros.

```
#Nivel 4: La Albumin está a 4
#########
al_4 <- which(datos$al==4 )
data_4=datos[al_4,]
dim(data_4)
```

```
## [1] 24 26
```

Nivel 4: Obtenemos 24 registros.

```r
#Nivel 5: La Albumin está a 5
#########
al_5 <- which(datos$al==5 )
data_5=datos[al_5,]
dim(data_5)
```

```
## [1]  1 26
```

Nivel 5: Obtenemos 1 registro.

Estimamos un modelo para cada subconjunto:

Modelo Nivel 0

```r
#Nivel 0
model_0<- lm(hemo~rc, data=data_0)
summary(model_0)
```

```
##
## Call:
## lm(formula = hemo ~ rc, data = data_0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1201 -1.2198 -0.0952  1.0315  5.2059
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.4733     0.6051   4.088 5.93e-05 ***
## rc            2.2491     0.1223  18.384  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.748 on 243 degrees of freedom
## Multiple R-squared:  0.5817, Adjusted R-squared:   0.58
## F-statistic:   338 on 1 and 243 DF,  p-value: < 2.2e-16
```

Modelo Nivel 1

```r
#Nivel 1
model_1<- lm(hemo~rc, data=data_1)
summary(model_1)
```

```
##
## Call:
## lm(formula = hemo ~ rc, data = data_1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3397 -0.9212  0.1151  0.7909  2.6247
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8773     1.3213   0.664     0.51
## rc            2.4519     0.3215   7.625 1.88e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.302 on 42 degrees of freedom
## Multiple R-squared:  0.5806, Adjusted R-squared:  0.5706
## F-statistic: 58.15 on 1 and 42 DF,  p-value: 1.878e-09
```

Modelo Nivel 2

```
#Nivel 2
model_2<- lm(hemo~rc, data=data_2)
summary(model_2)
```

```
##
## Call:
## lm(formula = hemo ~ rc, data = data_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.55856 -0.67627  0.06393  0.63713  2.13713
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.5978     1.1259   1.419    0.163
## rc            2.2536     0.2768   8.142 4.24e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.168 on 41 degrees of freedom
## Multiple R-squared:  0.6178, Adjusted R-squared:  0.6085
## F-statistic: 66.29 on 1 and 41 DF,  p-value: 4.239e-10
```

Modelo Nivel 3

```
#Nivel 3
model_3<- lm(hemo~rc, data=data_3)
summary(model_3)
```

```
##
## Call:
## lm(formula = hemo ~ rc, data = data_3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.46430 -0.64668  0.02761  0.70999  2.03570
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.4486     0.8500   2.881  0.00628 **
```

```
## rc            2.0809     0.2147   9.691 3.66e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.094 on 41 degrees of freedom
## Multiple R-squared:  0.6961, Adjusted R-squared:  0.6887
## F-statistic: 93.91 on 1 and 41 DF,  p-value: 3.659e-12
```

Modelo Nivel 4

```
#Nivel 4
model_4<- lm(hemo~rc, data=data_4)
summary(model_4)
```

```
##
## Call:
## lm(formula = hemo ~ rc, data = data_4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3260 -0.6466  0.2776  0.6131  3.9547
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.7237     1.2228   3.863 0.000842 ***
## rc            1.3253     0.2844   4.660 0.000120 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.544 on 22 degrees of freedom
## Multiple R-squared:  0.4967, Adjusted R-squared:  0.4738
## F-statistic: 21.71 on 1 and 22 DF,  p-value: 0.0001205
```

Modelo Nivel 5

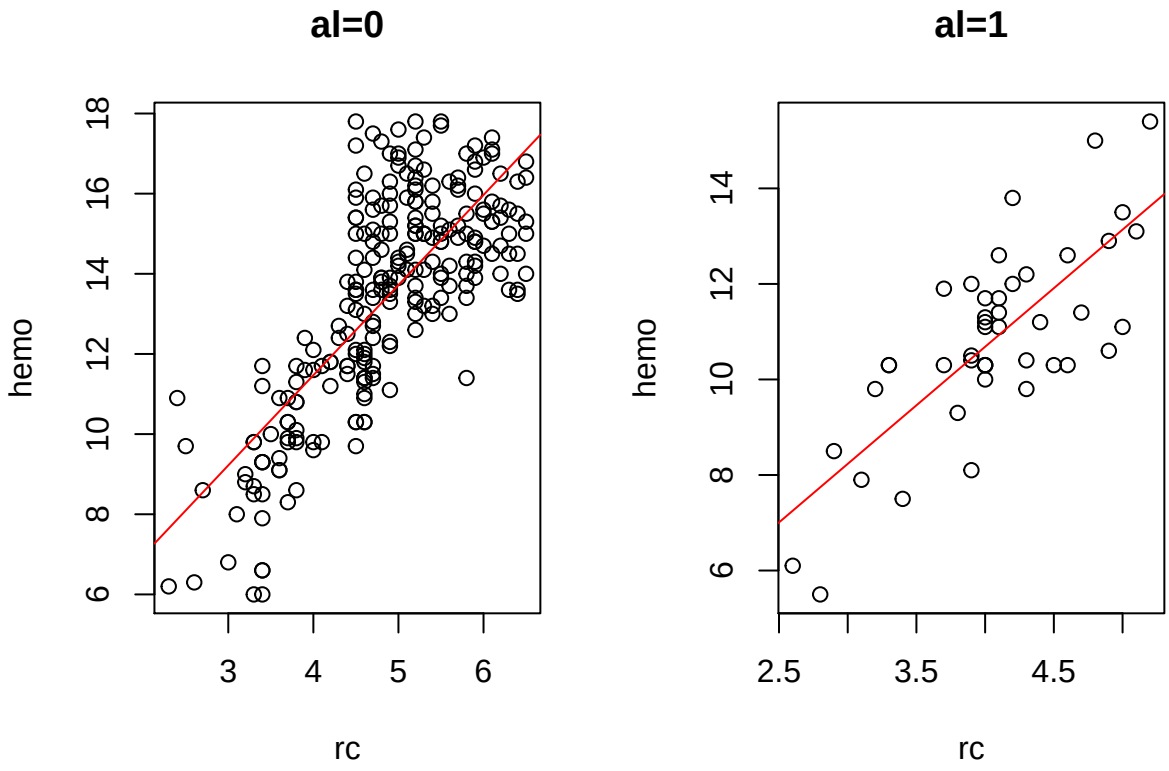
```
#Nivel 5
model_5<- lm(hemo~rc, data=data_5)
summary(model_5)
```

```
##
## Call:
## lm(formula = hemo ~ rc, data = data_5)
##
## Residuals:
## ALL 1 residuals are 0: no residual degrees of freedom!
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8         NA      NA       NA
## rc                NA         NA      NA       NA
##
## Residual standard error: NaN on 0 degrees of freedom
```

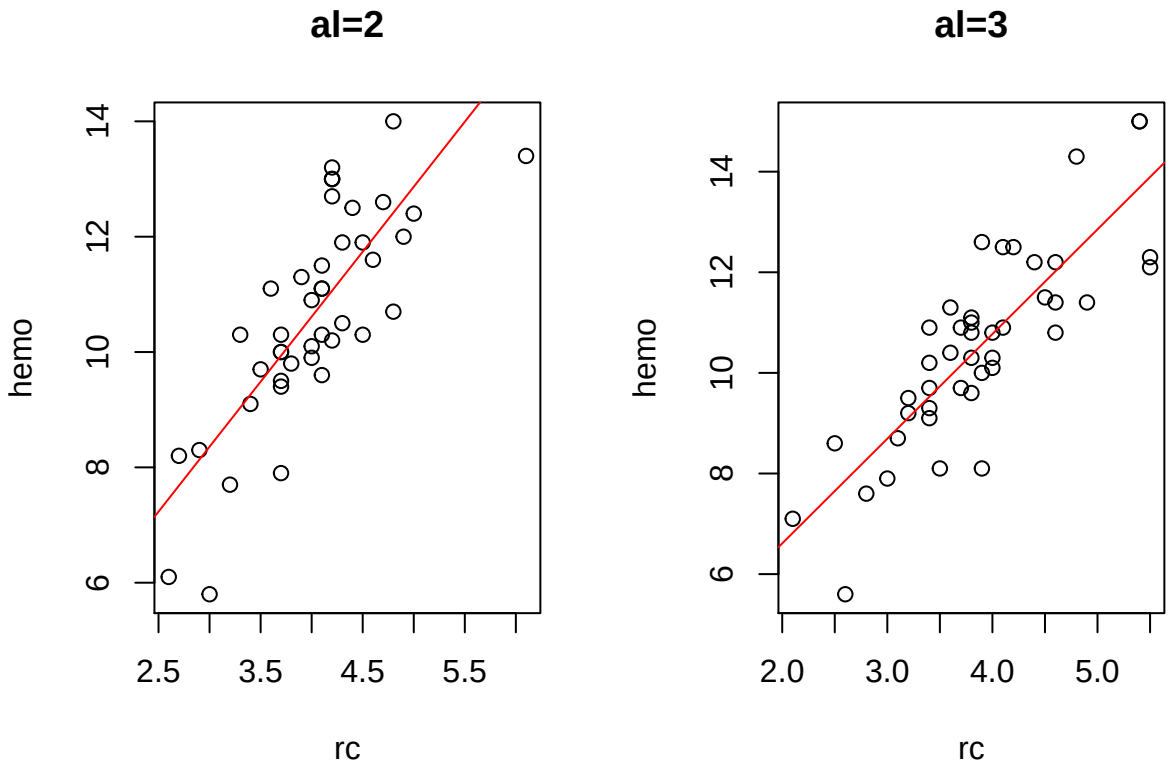
A la vista de los resultados, podemos concluir que la presencia o no de desnutrición influye ya que los coeficientes de determinación varían aunque se observan valores en torno al 60% en los tres primeros casos (excepto en nivel 4 que obtenemos un 47.38% y en nivel 5 donde no obtenemos nada pues solo tenemos un punto). Se puede observar que la relación lineal en presencia de desnutrición (menor cantidad de albúmina) mejora.

Veámos la relación lineal con los diagramas:

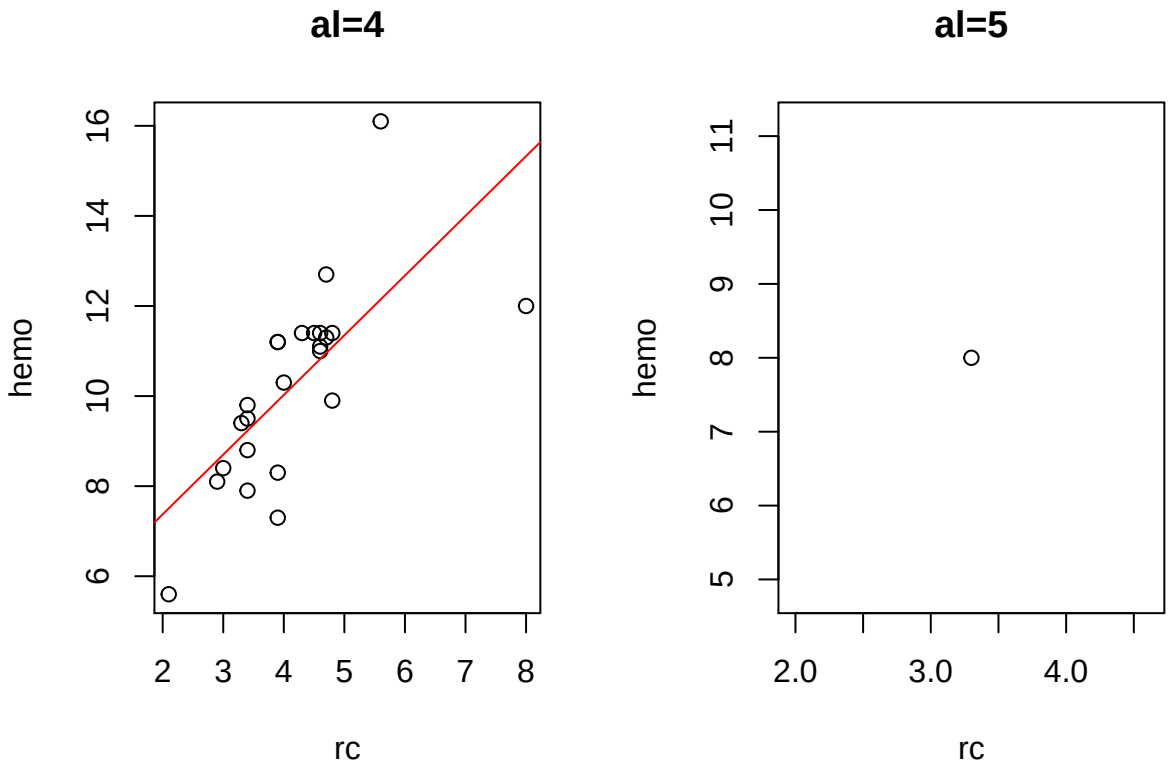
```
par(mfrow=c(1,2))
plot(data_0$rc,data_0$hemo,xlab="rc", ylab="hemo",main = "al=0")
abline(model_0,col = "red")

plot(data_1$rc,data_1$hemo,xlab="rc", ylab="hemo",main = "al=1")
abline(model_1,col = "red")
```




```
par(mfrow=c(1,2))
plot(data_2$rc,data_2$hemo,xlab="rc", ylab="hemo",main = "al=2")
abline(model_2,col = "red")

plot(data_3$rc,data_3$hemo,xlab="rc", ylab="hemo",main = "al=3")
abline(model_3,col = "red")
```




```
par(mfrow=c(1,2))
plot(data_4$rc,data_4$hemo,xlab="rc", ylab="hemo",main = "al=4")
abline(model_4,col = "red")

plot(data_5$rc,data_5$hemo,xlab="rc", ylab="hemo",main = "al=5")
```



*** GLOBULOS ROJOS~HEMOGLOBINA EN PACIENTES NORMOTENSIONADOS O HIPERTENSOS** Vamos a estudiar ahora la relación de los glóbulos rojos y la hemoglobina en función de si el paciente presenta hipertensión o no. Para ello, creados dos grupos: pacientes sin hipertensión (h_no) y pacientes con hipertensión (h_yes):

```
#Nivel yes: Pacientes con hipertensión
#########
h_yes <- which(datos$htn=="yes")
data_hy=datos[h_yes,]
dim(data_hy)
```

```
## [1] 147  26
```

```
#Nivel no; Pacientes sin hipertensión
#########
h_no <- which(datos$htn=="no")
data_hn=datos[h_no,]
dim(data_hn)
```

```
## [1] 253  26
```

Obtenemos 147 pacientes con hipertensión y 253 pacientes sin hipertensión. Ahora vamos a estudiar en cada caso la linealidad entre globulos rojos y la hemoglobina, ¿la hipertensión afecta? Veámoslo:

Modelo hemo~rc para hipertensos

```r
model_yes<- lm(hemo~rc, data=data_hy)
summary(model_yes)
```

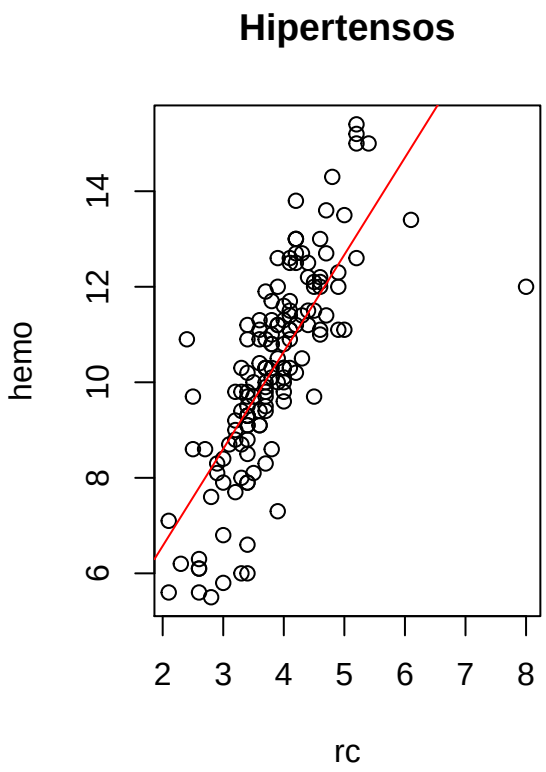
```
##
## Call:
## lm(formula = hemo ~ rc, data = data_hy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7686 -0.6334  0.0650  0.7747  3.5138
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.5080     0.5677   4.418 1.94e-05 ***
## rc            2.0326     0.1454  13.976  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.361 on 145 degrees of freedom
## Multiple R-squared:  0.5739, Adjusted R-squared:  0.571
## F-statistic: 195.3 on 1 and 145 DF,  p-value: < 2.2e-16
```

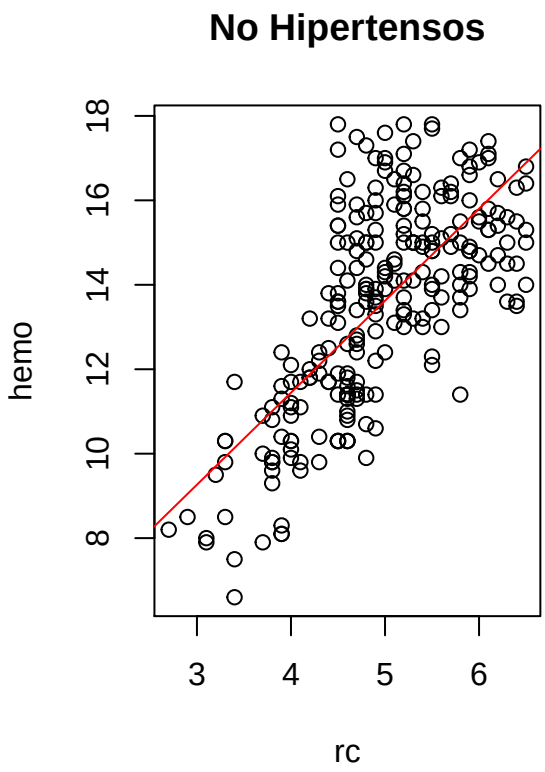
Modelo hemo~rc para No hipertensos

```r
model_no<- lm(hemo~rc, data=data_hn)
summary(model_no)
```

```
##
## Call:
## lm(formula = hemo ~ rc, data = data_hn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.965 -1.312 -0.169  1.081  5.267
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.7308     0.6899   3.958 9.83e-05 ***
## rc            2.1783     0.1379  15.802  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.753 on 251 degrees of freedom
## Multiple R-squared:  0.4987, Adjusted R-squared:  0.4967
## F-statistic: 249.7 on 1 and 251 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(1,2))
plot(data_hy$rc,data_hy$hemo,xlab="rc", ylab="hemo",main = "Hipertensos")
abline(model_yes,col = "red")

plot(data_hn$rc,data_hn$hemo,xlab="rc", ylab="hemo",main = "No Hipertensos")
abline(model_no,col = "red")
```





A la vista de los resultados, podemos concluir que la presencia o no de hipertensión influye, ya que los coeficientes de determinación varían aunque, se observan valores en torno al 40-50% en ambos casos. Se puede observar que la relación lineal en presencia de hipertensión es mayor.

*** GLOBULOS ROJOS~CREATININA EN PACIENTES NORMOTENSIONADOS O HIPERTENSOS** Hacemos lo mismo que en el apartado anterior pero entre globulos rojos y creatinina. Los grupos de estudios son los mismos, pacientes hipertensos y no hipertensos

```
#Nivel yes: Pacientes con hipertensión
#########
h_yes <- which(datos$htn=="yes")
data_hy=datos[h_yes,]
dim(data_hy)
```

```
## [1] 147  26
```

```
#Nivel no; Pacientes sin hipertensión
#########
h_no <- which(datos$htn=="no")
data_hn=datos[h_no,]
dim(data_hn)
```

```
## [1] 253  26
```

Obtenemos 147 pacientes con hipertensión y 253 pacientes sin hipertensión. Ahora veámos como afecta la hipertensión a la relación entre glóbulos rojos y creatinina:

Modelo hemo~rc para hipertensos

```
model_yes_1<- lm(rc~sc, data=data_hy)
summary(model_yes_1)
```

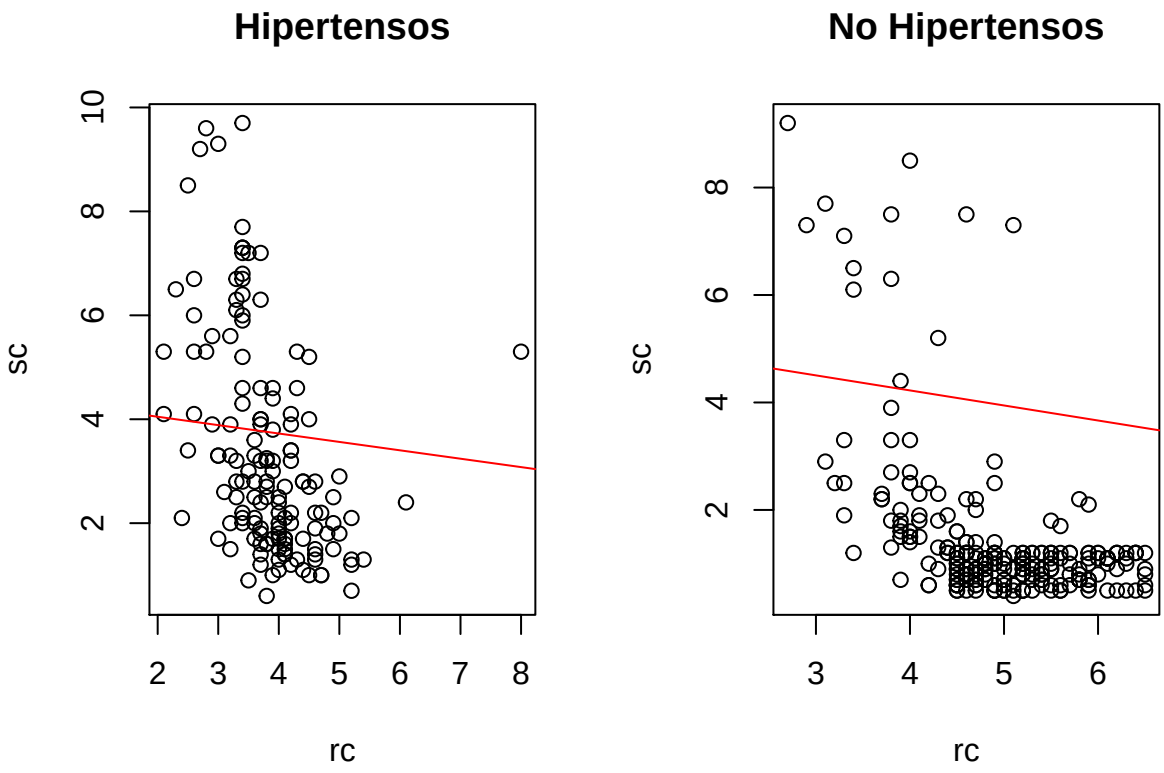
```
##
## Call:
## lm(formula = rc ~ sc, data = data_hy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6323 -0.3839 -0.0254  0.3242  4.4847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.37163    0.10876  40.197  < 2e-16 ***
## sc          -0.16157    0.02735  -5.907 2.38e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6978 on 145 degrees of freedom
## Multiple R-squared:  0.194,  Adjusted R-squared:  0.1884
## F-statistic: 34.89 on 1 and 145 DF,  p-value: 2.376e-08
```

Modelo hemo~rc para No hipertensos

```
model_no_1<- lm(rc~sc, data=data_hn)
summary(model_no_1)
```

```
##
## Call:
## lm(formula = rc ~ sc, data = data_hn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6079 -0.5079 -0.1079  0.4921  1.7993
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.34376    0.06169  86.625   <2e-16 ***
## sc          -0.27987    0.03029  -9.241   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6932 on 251 degrees of freedom
## Multiple R-squared:  0.2538, Adjusted R-squared:  0.2509
## F-statistic: 85.39 on 1 and 251 DF,  p-value: < 2.2e-16
```

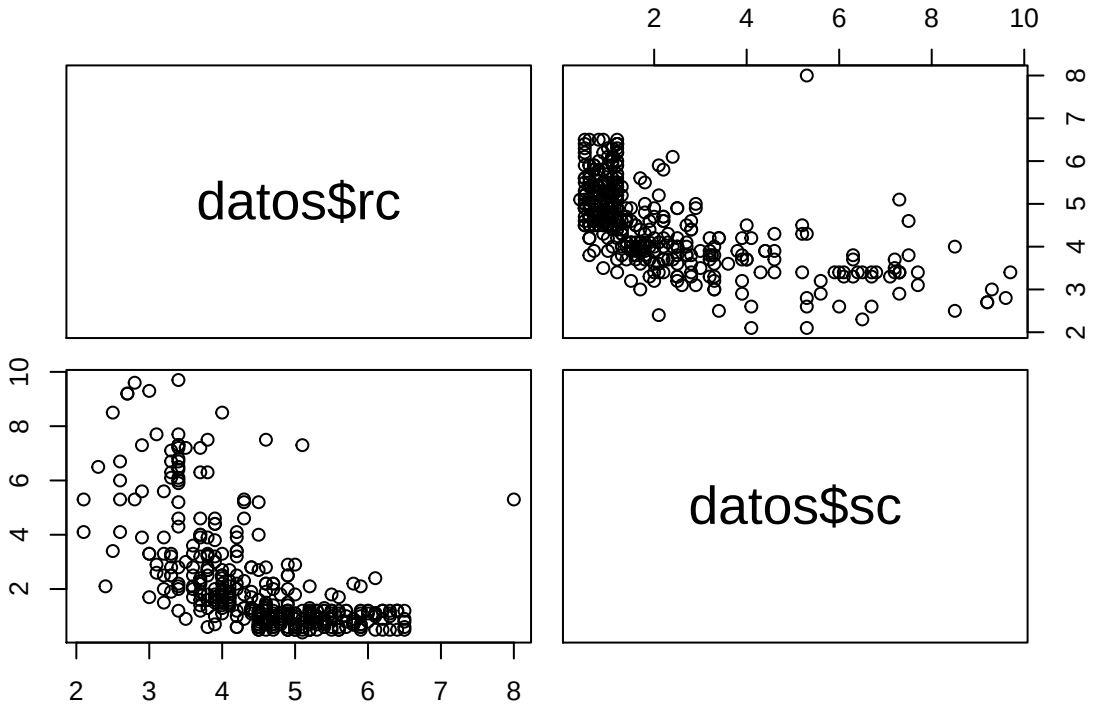
```
par(mfrow=c(1,2))
plot(data_hy$rc,data_hy$sc,xlab="rc", ylab="sc",main = "Hipertensos")
abline(model_yes_1,col = "red")

plot(data_hn$rc,data_hn$sc,xlab="rc", ylab="sc",main = "No Hipertensos")
abline(model_no_1,col = "red")
```



Observamos que no existe una relación lineal entre los globulos rojos y la creatinina de un paciente.

Mirando la dispersión en su globalidad (sin realizar grupos) realmente podemos observar que no existía relación:

```
pairs(datos$rc~datos$sc)
```

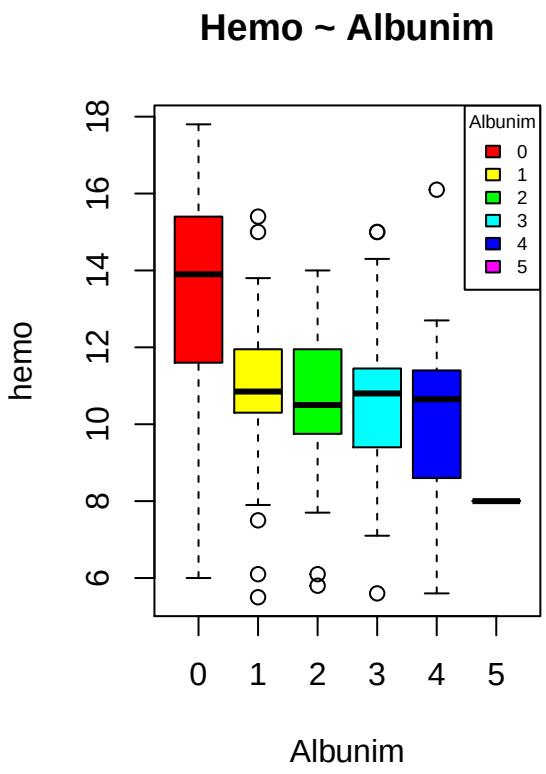


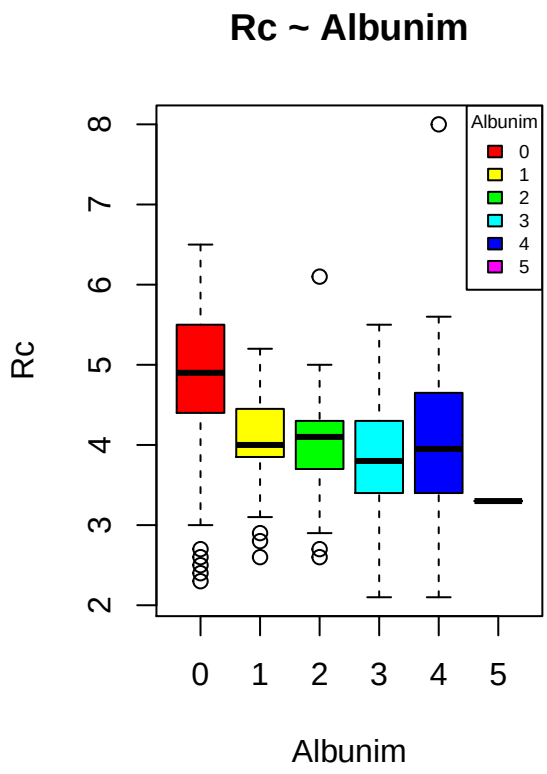
5.-Representación de los resultados mediante gráficas y tablas.

De entre las representaciones posibles, implementamos los diagramas de cajas de las variables que hemos comparado.

* HEMATOCRITO Y HEMOGLOBINA EN DIFERENTES ESTADOS DE NUTRICIÓN

```
par(mfrow=c(1,2))
boxplot(datos$hemo ~ datos$al,col = rainbow(length(unique(datos$al))),
                          xlab="Albunim", ylab="hemo", main='Hemo ~ Albunim',)

legend("topright", legend=c(0, 1, 2, 3, 4, 5),
       title="Albunim", fill=rainbow(length(unique(datos$al))), cex=0.6, horiz=FALSE)

boxplot(datos$rc ~ datos$al,col = rainbow(length(unique(datos$al))),
                          xlab="Albunim", ylab="Rc", main='Rc ~ Albunim',)

legend("topright", legend=c(0, 1, 2, 3, 4, 5),
       title="Albunim", fill=rainbow(length(unique(datos$al))), cex=0.6, horiz=FALSE)
```

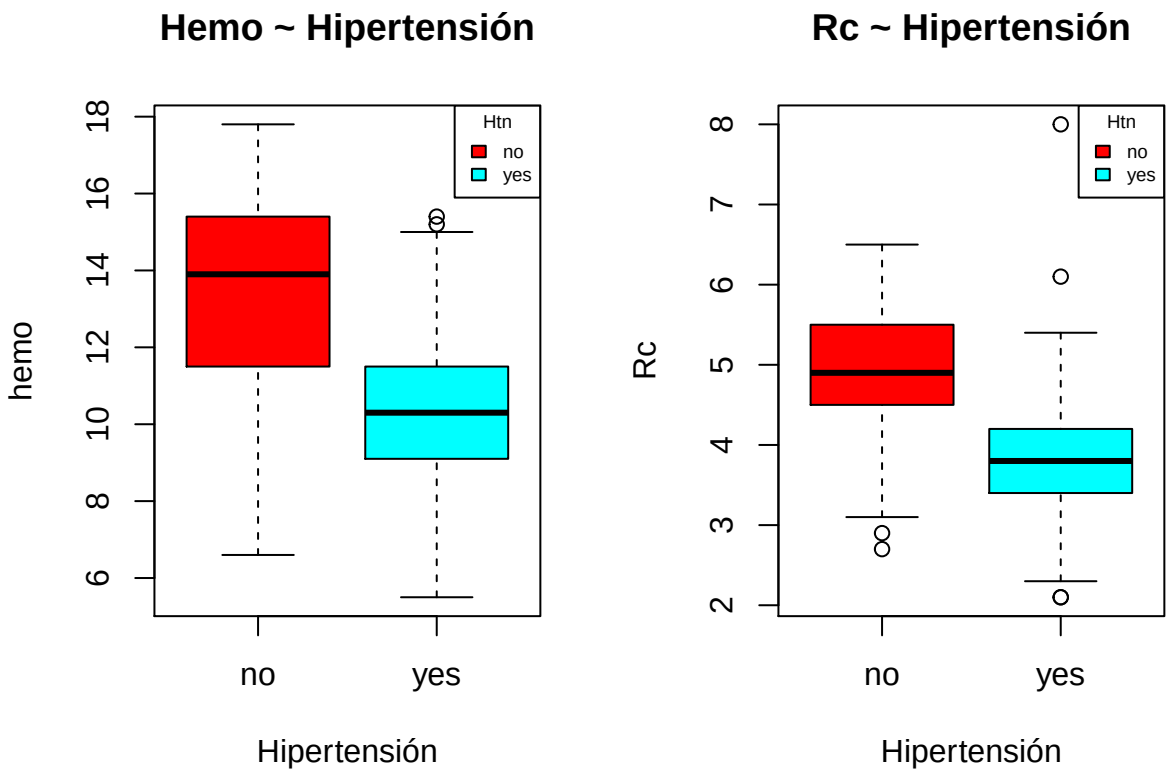




Podemos comprobar las variaciones del hematocrito y el número de glóbulos rojos en relación a los diferentes estados de nutrición del paciente.

* HEMATOCRITO Y HEMOGLOBINA EN PACIENTES HIPERTENSOS Y AQUELLOS QUE PRESENTAN UNA TENSIÓN NORMAL

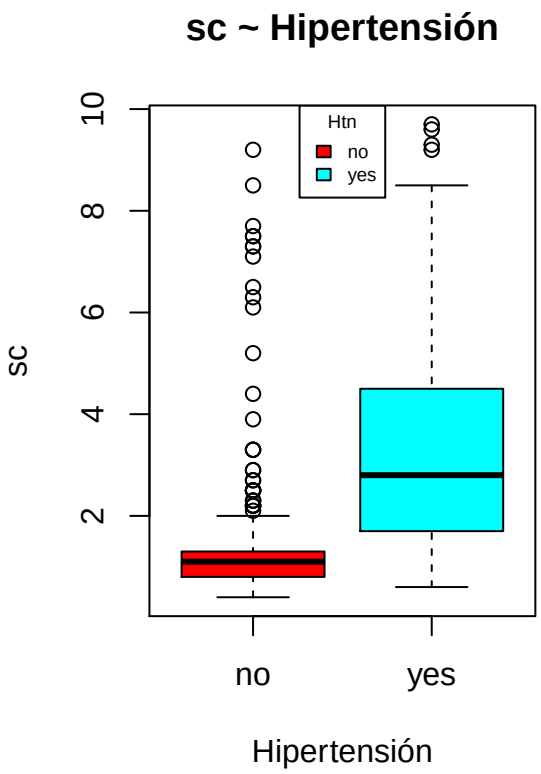
```
par(mfrow=c(1,2))
boxplot(datos$hemo ~ datos$htn,col = rainbow(length(unique(datos$htn))),
                          xlab="Hipertensión", ylab="hemo", main='Hemo ~ Hipertensión',)

legend("topright", legend=c("no", "yes"),
       title="Htn", fill=rainbow(length(unique(datos$htn))), cex=0.6, horiz=FALSE)

boxplot(datos$rc ~ datos$htn,col = rainbow(length(unique(datos$htn))),
                          xlab="Hipertensión", ylab="Rc", main='Rc ~ Hipertensión',)

legend("topright", legend=c("no", "yes"),
       title="Htn", fill=rainbow(length(unique(datos$htn))), cex=0.6, horiz=FALSE)
```

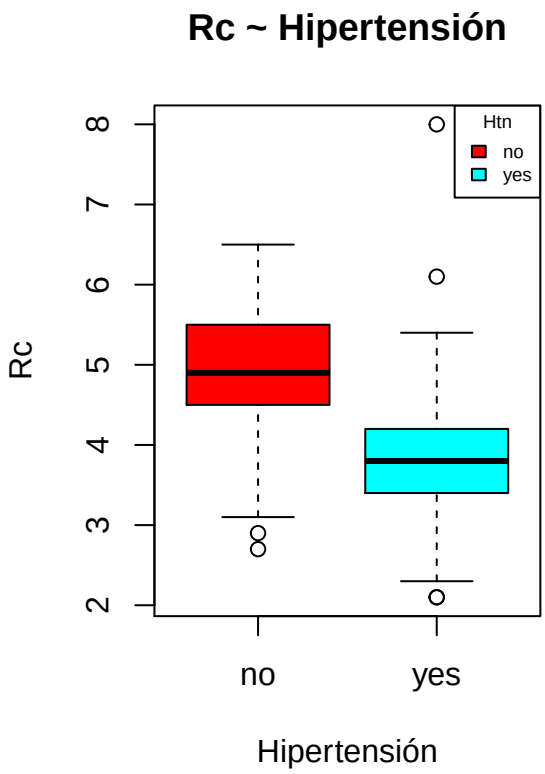


En general los pacientes hipertensos presentan niveles de hemoglobina y hematocrito peores.

* HEMATOCRITO Y CREATININA EN PACIENTES HIPERTENSOS Y AQUELLOS QUE PRESENTAN UNA TENSIÓN NORMAL

```
par(mfrow=c(1,2))
boxplot(datos$sc ~ datos$htn,col = rainbow(length(unique(datos$htn))),
                          xlab="Hipertensión", ylab="sc", main='sc ~ Hipertensión',)

legend("top", legend=c("no", "yes"),
       title="Htn", fill=rainbow(length(unique(datos$htn))), cex=0.6, horiz=FALSE)

boxplot(datos$rc ~ datos$htn,col = rainbow(length(unique(datos$htn))),
                          xlab="Hipertensión", ylab="Rc", main='Rc ~ Hipertensión',)

legend("topright", legend=c("no", "yes"),
       title="Htn", fill=rainbow(length(unique(datos$htn))), cex=0.6, horiz=FALSE)
```





Observamos niveles más altos de creatinina en pacientes con hipertensión y como media mayor número de globulos rojos en pacientes normotensionados.

Hemos realizado varios estudios pero vamos a comprobar si tenemos alguna relación entre las variables cuantitativas. Para ello creamos un dataset con solo las variables Cuantitativas:

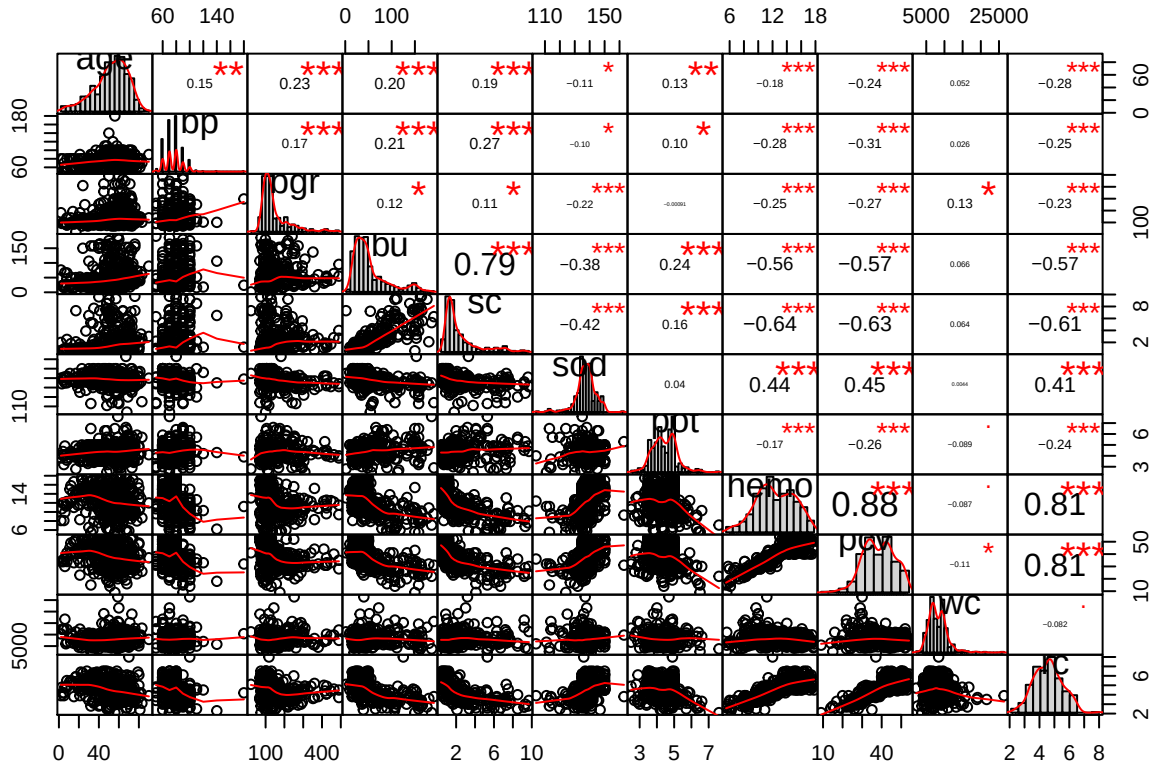
```
names<-colnames(datos)
idQuantitative <- which(names=="age"| names=="bp" | names=="bgr"| names=="bu" | names=="sc" | names=="s

data_Quantitive <- datos[,idQuantitative]
head(data_Quantitive)
```

```
##   age bp bgr bu  sc sod pot hemo pcv   wc  rc
## 1  48 80 121 36 1.2 140 5.0 15.4  44 7800 5.2
## 2   7 50  92 18 0.8 141 4.1 11.3  38 6000 4.7
## 3  62 80 423 53 1.8 134 4.3  9.6  31 7500 4.1
## 4  48 70 117 56 3.8 111 2.5 11.2  32 6700 3.9
## 5  51 80 106 26 1.4 141 4.2 11.6  35 7300 4.6
## 6  60 90  74 25 1.1 142 3.2 12.2  39 7800 4.4
```

* CORRELACIONES

Ahora vamos a visualizar la correlación existente entre las varibles para ver si existe alguna relación entre ellas:

```r
#Vemos la correlación a partir de un gráfico
chart.Correlation(data_Quantitive)
```



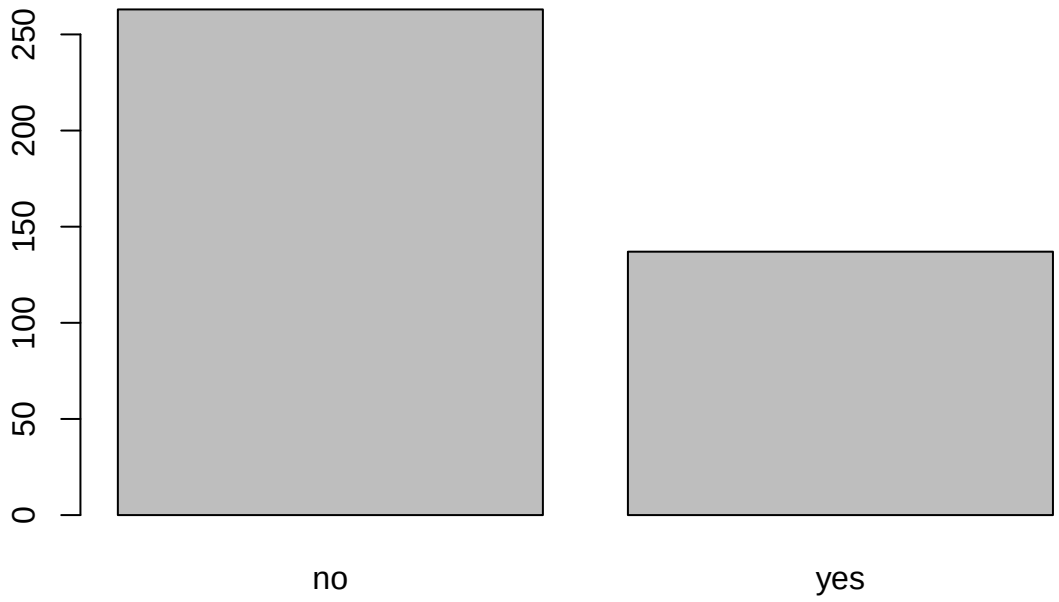
Mirando el diagrama vemos que hemo y pcv (es decir, hemoglobina y porcentaje de células) están fuertemente correlacionadas con un valor de correlación de 0.88. Vemos que las dos variables (hemo y pcv) tienen la misma correlación con la variable rc (hematocrito), que tiene valor 0.81. Por otro lado, otra correlación que destaca es la relación entre sc (creatinina) y bu (urea en sangre) correlacionadas con un 0.79. Hay otras correlaciones, aunque no tan superiores puede ser la hemoglobina con la creatinina y la urea, o, estas últimas con los hematocritos.

* DIABÉTICOS

Como afecta la diabétes

Vista la relación entre las variables y vista la relación con la hipertensión, veámos qué pasa con la diabetes.

```r
#registos de diabéticos que tenemos en el dataset
barplot(table(datos$dm))
```

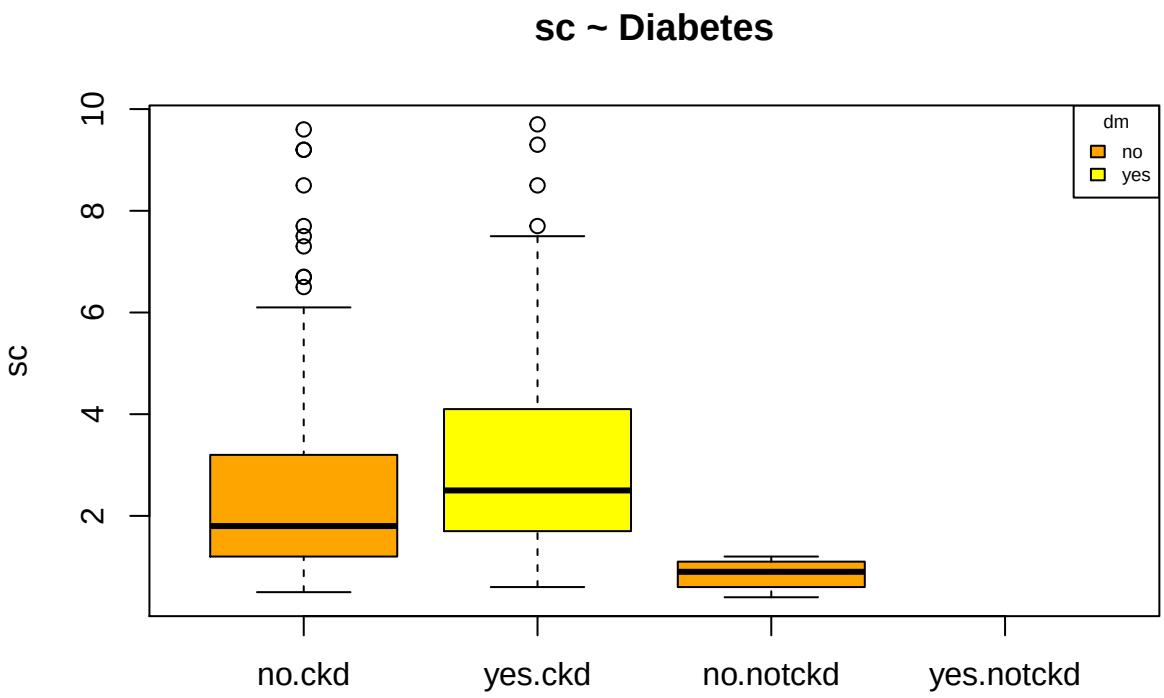



```
table(datos$dm)
```

```
##
##  no yes
## 263 137
```

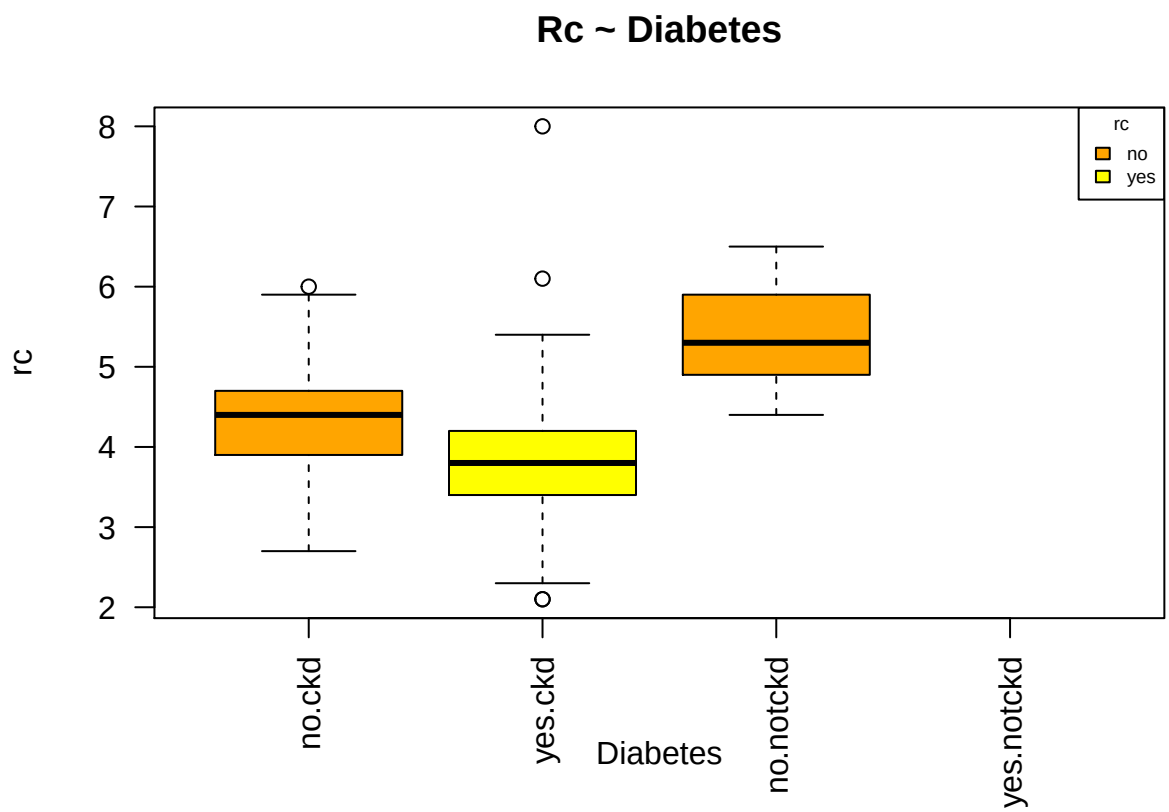
En la muestra tenemos menos diabéticos que no diabéticos. Tenemos 263 registros no diabéticos frente a 137 sí diabéticos. Casi tenemos el doble de no que de sí.

```
#Creatinina en diabéticos
boxplot(datos$sc ~ datos$dm + datos$classification,col = c("orange", "yellow"),
                          xlab="", ylab="sc", main='sc ~ Diabetes')

legend("topright", legend=c("no", "yes"),
       title="dm", fill=c("orange", "yellow"), cex=0.6, horiz=FALSE)
```



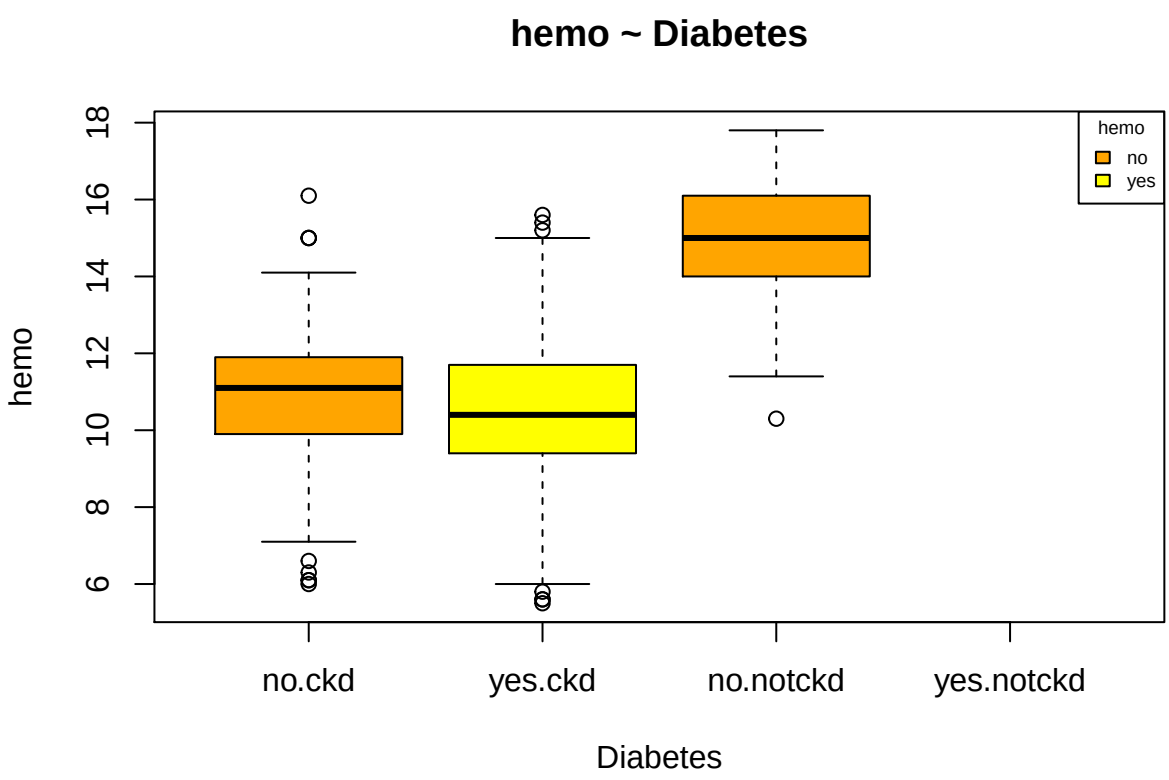
Vemos que los pacientes con la enfermedad crónica pueden ser diabéticos o no. Lo que no ocurre es que, pacientes que no tienen enfermedad en el riñon sean diabéticos. Entonces, en nuestros datos, si eres diabético eres un paciente con enfermedad crónica en el riñón. Observamos que los pacientes enfermos, al sufrir diabetes, tienen la creatinina alterada y algo más superiores que los pacientes no enfermos. De hecho solo por ser pacientes enfermos, ya tiene más alto la creatinina (lo podemos ver si comparamos no.ckd y no.notckd).

```
#Hematocritos en Diabéticos
boxplot(datos$rc ~ datos$htn + datos$classification,col = c("orange", "yellow"),
                          xlab="Diabetes", ylab="rc", main='Rc ~ Diabetes', las = 2)

legend("topright", legend=c("no", "yes"),
       title="rc", fill=c("orange", "yellow"), cex=0.6, horiz=FALSE)
```



Se puede observar que los pacientes enfermos tienen los hematocritos más bajos que los pacientes sin enfermedad. De hecho, si encima es diabético aún los pueden tener más bajo.

```
#Hemoglobina en Diabéticos
boxplot(datos$hemo ~ datos$dm + datos$classification, col = c("orange", "yellow"),
                          xlab="Diabetes", ylab="hemo", main='hemo ~ Diabetes',)

legend("topright", legend=c("no", "yes"),
       title="hemo", fill=c("orange", "yellow"), cex=0.6, horiz=FALSE)
```



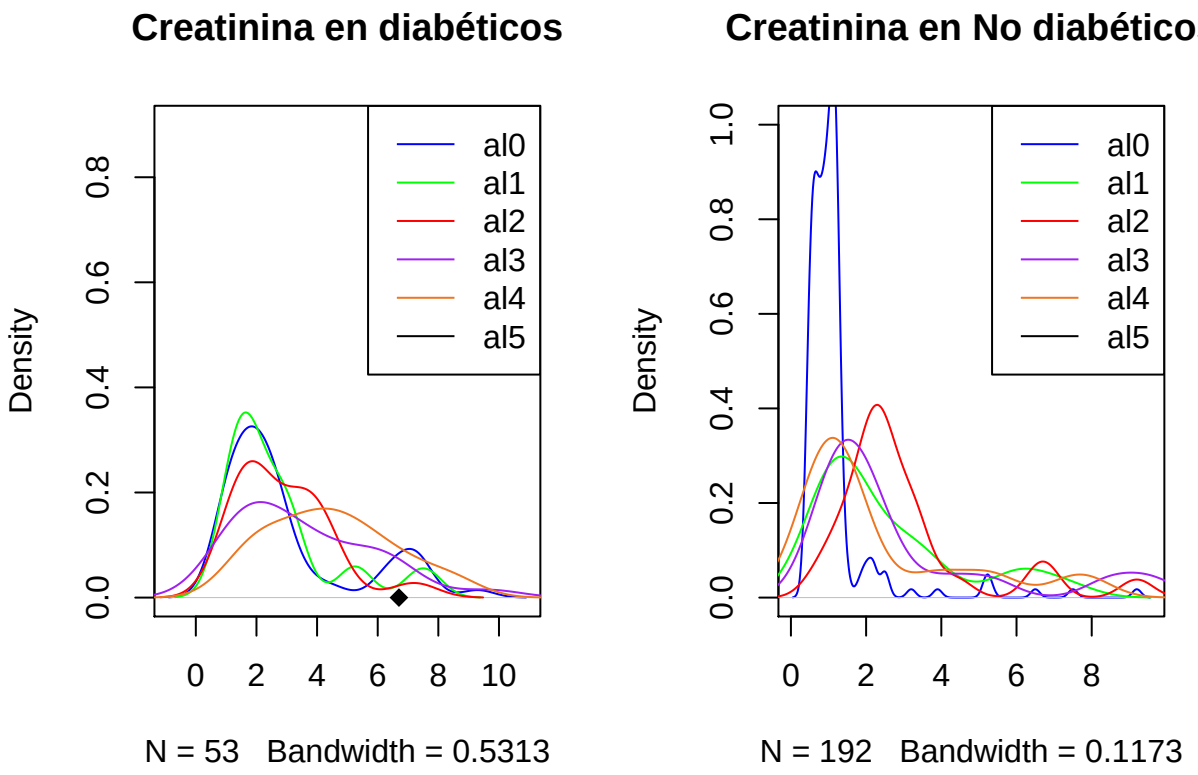
En este caso vemos que la hemoglobina tiene parámetros muy parecidos entre los pacientes con enfermedad crónica, independientemente de si es diabético o no. Sí tenemos diferencia entre los pacientes no diagnosticados y los diagnosticados, los primeros tiene la hemoglobina más alta que los segundos.

Por supuesto la diabetes es un factor de riesgo para la enfermedad crónica. Independientemente vemos que se puede generar la enfermedad sin ser diabético (ya que los parámetros de diabéticos y no diabéticos en pacientes con enfermedad renal se asemejan mucho).

Un parámetro importante en la detección de la enfermedad es la Albúmina veámos cómo se mueve para pacientes diabéticos. Utilizaremos los grupos generados anteriormente: data_0, data_1, data_2, data_3, data_4 y data_5 para localizare el nivel de albúminia (ya que recordemos que tenemos 6 categorías según el nivel de albúminia):

```
par(mfrow=c(1,2))
#Diagrama densidad en los diferentes niveles de albúminia y siendo diabéticos
plot(density(data_0[data_0$dm=="yes","sc"]), col = "blue", ylim = c(0, 0.9), main ="Creatinina en diabé
lines(density(data_1[data_1$dm=="yes","sc"]), col = "green")
lines(density(data_2[data_2$dm=="yes","sc"]), col = "red")
lines(density(data_3[data_3$dm=="yes","sc"]), col = "purple")
lines(density(data_4[data_4$dm=="yes","sc"]), col = "chocolate2")
points(data_5[data_5$dm=="yes","sc"], 0, col="black", cex=1.3, pch=18)
legend("topright", c("al0", "al1", "al2", "al3", "al4", "al5"),
       lty = 1, col = c("blue", "green", "red", "purple", "chocolate2", "black"))

#Diagrama densidad en los diferentes niveles de albúminia y siendo NO diabéticos
plot(density(data_0[data_0$dm=="no","sc"]), col = "blue", ylim = c(0, 1), main ="Creatinina en No diabé
lines(density(data_1[data_1$dm=="no","sc"]), col = "green")
```

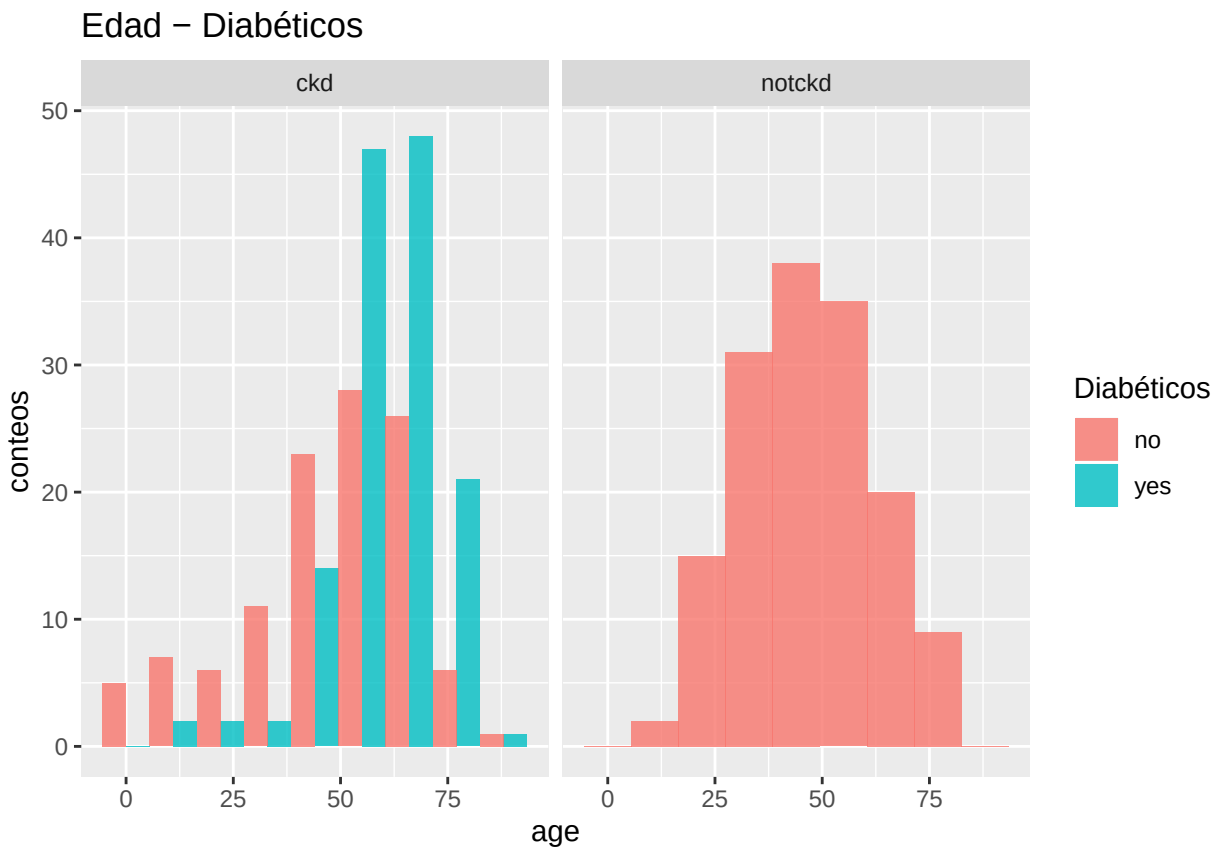
```
lines(density(data_2[data_2$dm=="no","sc"]), col = "red")
lines(density(data_3[data_3$dm=="no","sc"]), col = "purple")
lines(density(data_4[data_4$dm=="no","sc"]), col = "chocolate2")
#points(data_5[data_5$dm=="no","sc"], 0, col="black", cex=2, pch=18)  #No tenemos punto con no
legend("topright", c("al0", "al1", "al2", "al3", "al4", "al5"),
       lty = 1, col = c("blue", "green", "red", "purple", "chocolate2", "black"))
```



Podemos ver que la albúminia, en el caso de los diabéticos, se distribuye casi igual en nivel 0 y nivel 1 (hay una pequeña subida entre 6 y 8 para nivel 0). Para el resto de casos cada vez se aplana más. Esto es que, cada vez que la albúminia está en un estadio superior la creatinina toma un rango mayor, aumentado su valor máximo. Así por ejemplo en un estadio 4 de albúminia el rango de la creatinina puede variar entre 0 y 10, obteniendo su mayor densidad alrededor de 4 (mientras que en un estadio 0 estamos alrededor de 2). Por contra partida, si no son diabéticos, la densidad no se aplana aunque tengamos estemos en estadios más altos de albúminia. Tenemos una densidad muy alta en nivel0.

La franja de edad por donde se mueven los diabéticos son:

```
#Gráfica edad diabéticos

ggplot(data = datos,
       mapping = aes(x = age, fill = factor(dm))) +
  geom_histogram(bins = 9, position = 'dodge', alpha = 0.8) +
  labs(title = 'Edad - Diabéticos',
       fill = 'Diabéticos',  x = 'age', y = 'conteos') +
facet_grid(. ~ classification)
```



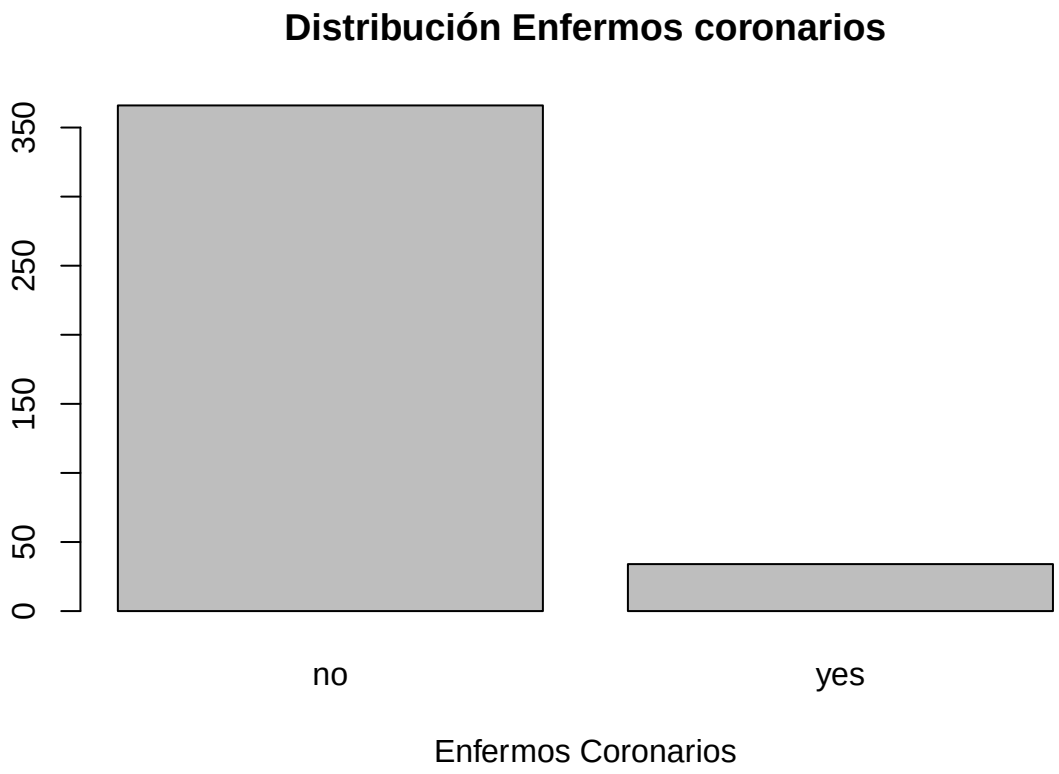
Nos encontramos diabéticos en casi todas las franjas de edad, pero la franja más afectada es entre los 55 y 70 años, donde los diabéticos se disparan.

* ENFERMEDAD CORONARIA

¿Qué pasa con la enfermedad coronaria?

Igual que hemos hecho con hipertensión y diabetes, veámos ahora esta enfermedad. Veámos si hay alguna relación entre las dos enfermedades respecto a las variables de analítica.

```
#registos de enfermos coronarios que tenemos en el dataset
barplot(table(datos$cad), main = "Distribución Enfermos coronarios", xlab = "Enfermos Coronarios")
```



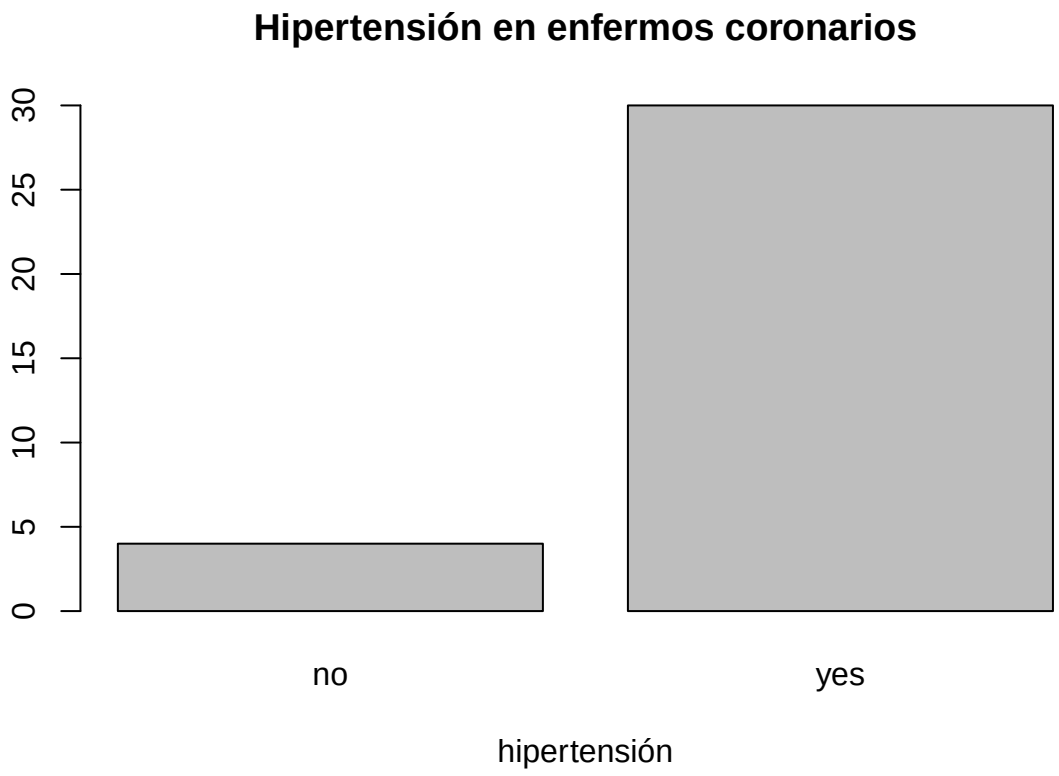

```
table(datos$cad)
```

```
##
##  no yes
## 366  34
```

Tenemos muchísimos más registros que no padecen enfermedad coronaria. De hecho solo la padecen 34 de nuestros pacientes.

La enfermedad coronária puede hacer compotars aumento en la tensión arterial. Veamos de los 34 pacientes que tenemos con enfermedad coronaria cuántos son hipertensos:

```
#Diagrama de Hipertensos entre los pacientes que sufren una enfermedad coronaria
barplot(table(datos[datos$cad == "yes", ]$htn), main = "Hipertensión en enfermos coronarios", xlab="hip
```



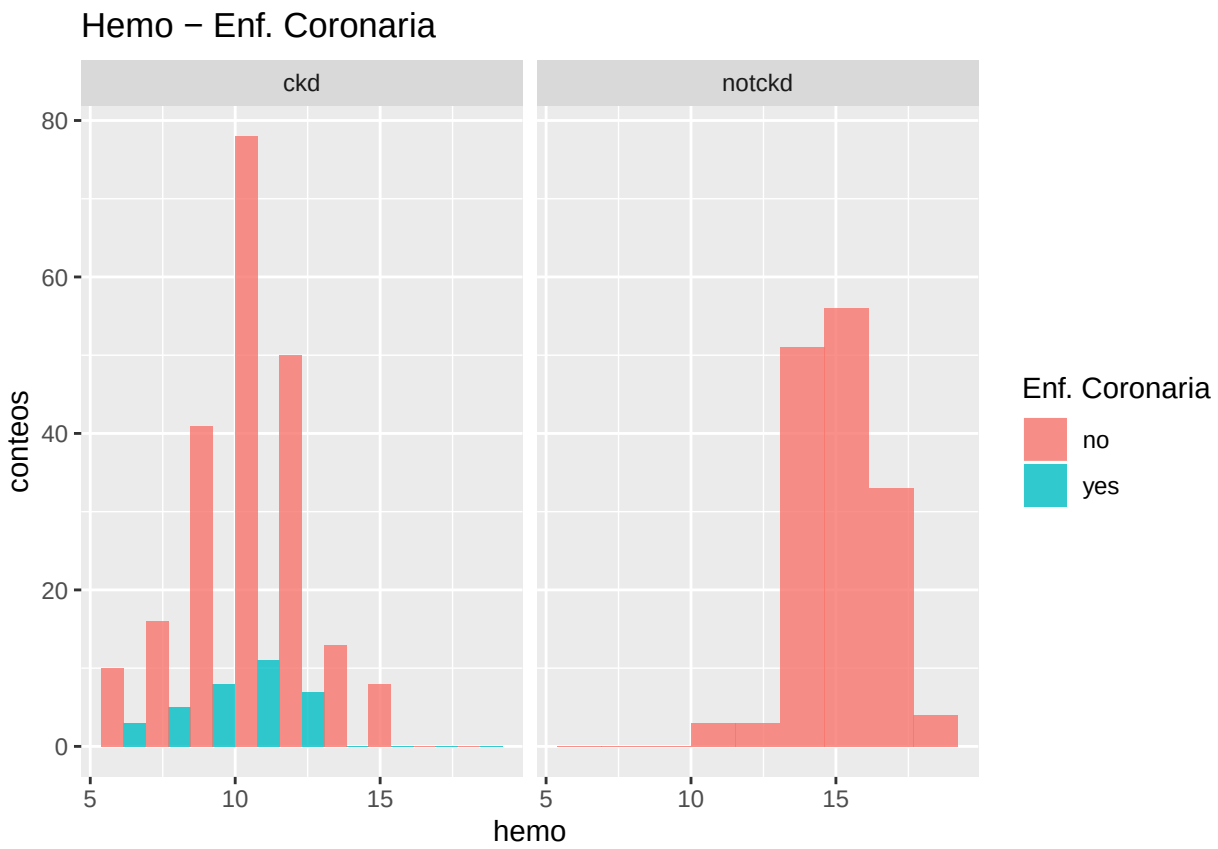

```
#Cantidad
table(datos[datos$cad == "yes", ]$htn)
```

```
##
##  no yes
##   4  30
```

Podemos ver que 30 de los 34 sufren tensión arterial. Segurameente por consecuencia de la enfermedad coronaria.

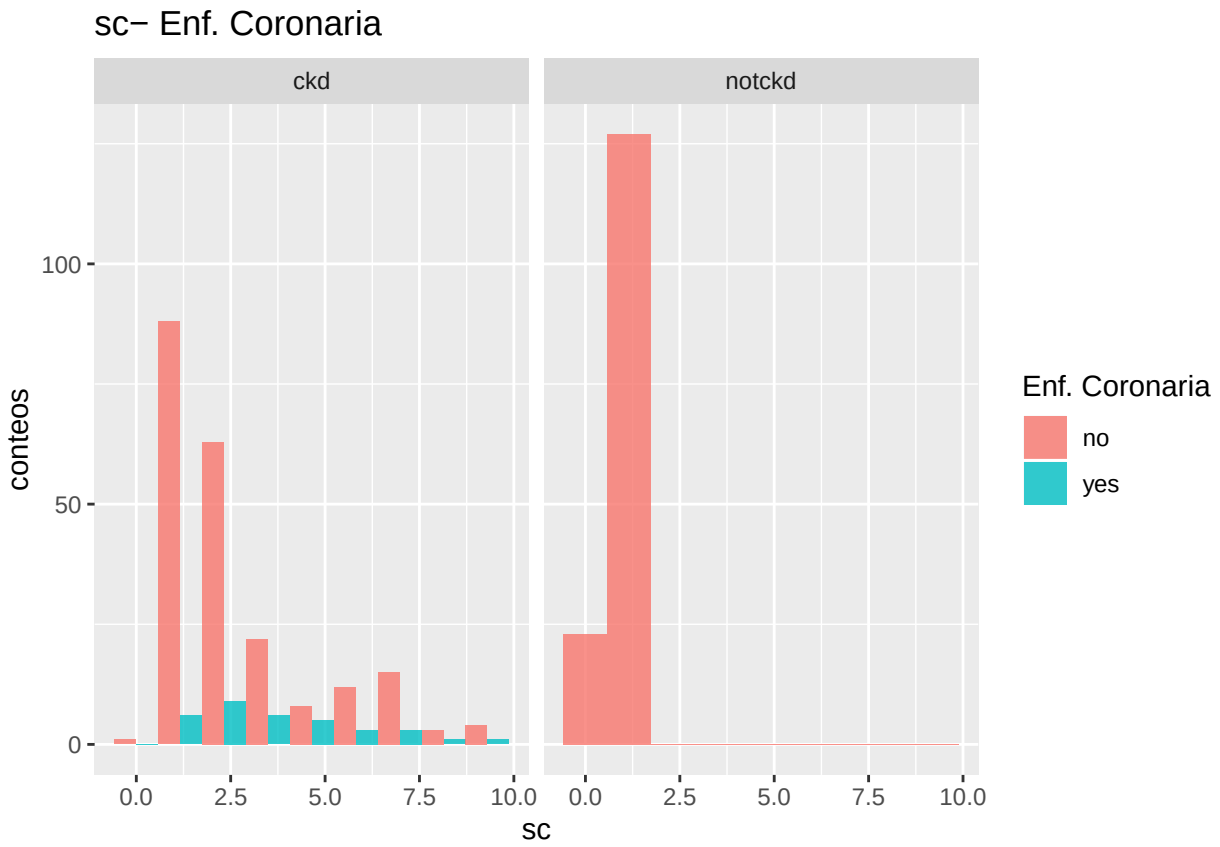
Veámos ahora cómo tenemos los otros parámetros estudiados dentro de los pacientes con enfermedad coronaria:

```
#Gráfico Hemoglobina en enfermedad coronaria
ggplot(data = datos,
       mapping = aes(x = hemo, fill = factor(cad))) +
  geom_histogram(bins = 9, position = 'dodge', alpha = 0.8) +
  labs(title = 'Hemo - Enf. Coronaria',
       fill = 'Enf. Coronaria',  x = 'hemo', y = 'conteos') +
facet_grid(. ~ classification)
```



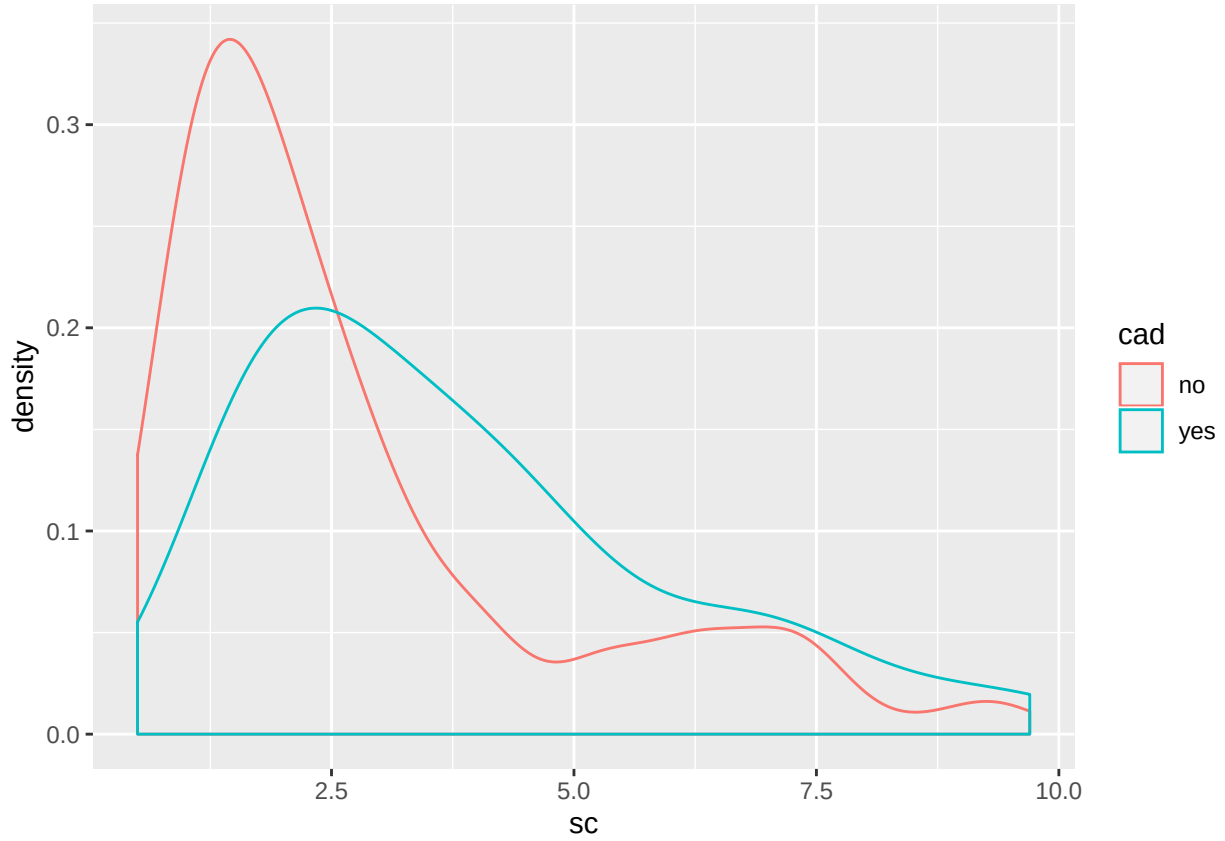
Podemos ver que los pacientes que sufren una enfermedad coronaria son pocos y su nivel de hemoglobina no llega a 14 mg/ml. Además, en nuestros datos, solo los pacientes con enfermedad crónica tiene enfermedad coronaria.

```
#Gráfico rc en enfermedad coronaria
ggplot(data = datos,
       mapping = aes(x = sc, fill = factor(cad))) +
  geom_histogram(bins = 9, position = 'dodge', alpha = 0.8) +
  labs(title = 'sc- Enf. Coronaria',
       fill = 'Enf. Coronaria',  x = 'sc', y = 'conteos') +
facet_grid(. ~ classification)
```



Nos vuelve a pasar como con la diabetes, que no tenemos ningún registro que, no siendo un paciente crónico, tengan una enfermedad coronaria, En nuestra muestra no tenemos, por tanto, pacientes no crónicos con alguna enfermedad coronaria. Parace que solo se vean afectados los pacientes crónicos. Así, dichos pacientes son los que tiene valores de creatinina. Vemos que tenemos fluctuación en la creatinina, ya sea un paciente con enfermedad coronaria o no. Con enfermedad coronaria tenemos pocos datos y por eso nos aparecen pocos registros (solo tenemos 34).

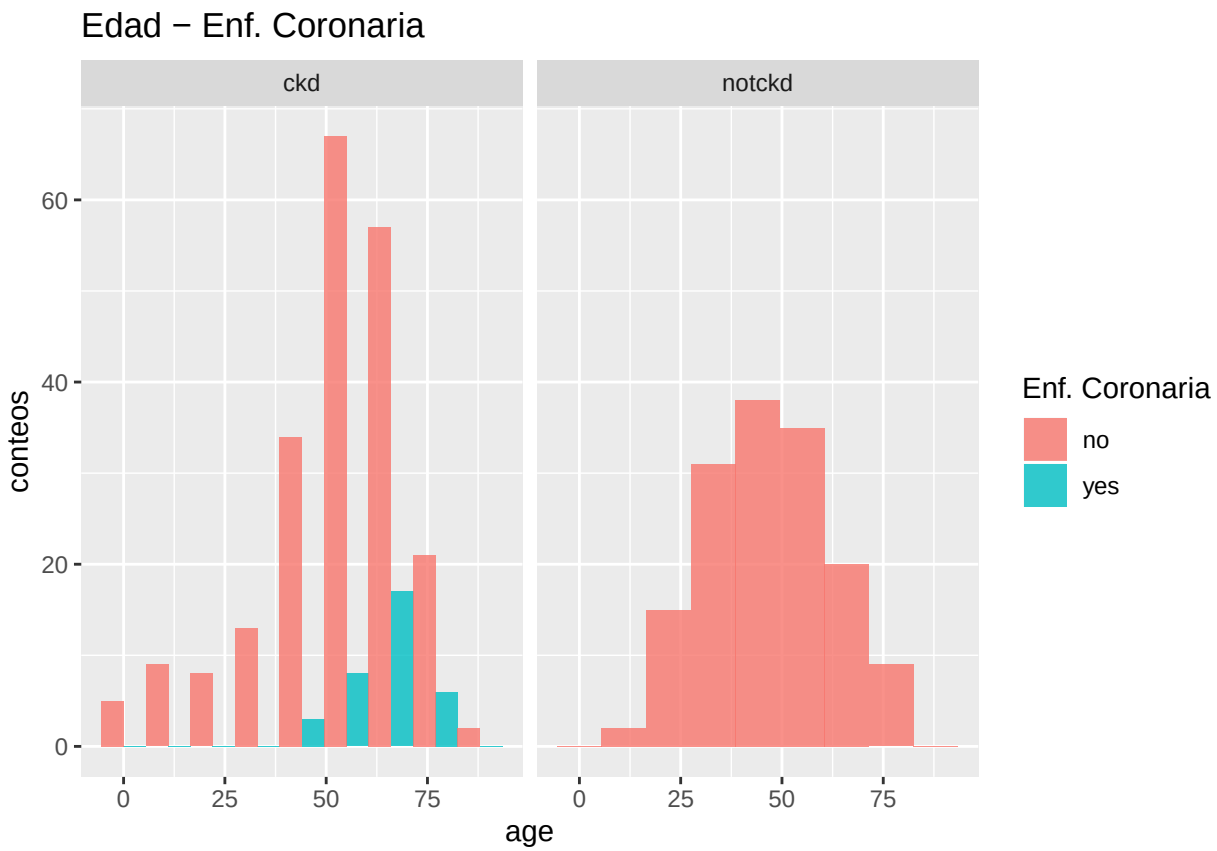
```
#Gráfica de Creatinina en enfermos crónicos según tengan o no enfermedad coronária
ggplot(datos[datos$classification=="ckd",], aes(x=sc, color=cad)) +
  geom_density()
```



Esta gràfica representamos la parte izquierda de la anterior gráfica, es decir, la evolución de la creatinina en los enfermos crónicos según tengan enfermedad coronaria o no (cad es la variable de enfermedad coronaria). Vemos como la Creatinina es más plana en los enfermos con afectación coronaria, mientras que los que no tienen enfermedad coronaria, su creatinina se mantiene más estable entre el 0 y 2.5.

Vamos a visualizar ahora en qué franja de edad nos movemos entre los pacientes crónicos:

```
#Gráfica edad en enfermedad coronaria
ggplot(data = datos,
       mapping = aes(x = age, fill = factor(cad))) +
  geom_histogram(bins = 9, position = 'dodge', alpha = 0.8) +
  labs(title = 'Edad - Enf. Coronaria',
       fill = 'Enf. Coronaria',  x = 'age', y = 'conteos') +
facet_grid(. ~ classification)
```



Los pacientes que sufren una enfermedad coronaria aparecen a partir de los 50 años, además, son todos pacientes crónicos. Así que a partir de dicha edad, puede ser una enfermedad contraída por al enfermedad crónica o puede ser que una enfermedad de este tipo complique la situación en una enfermedad renal.

6.-Conclusiones.

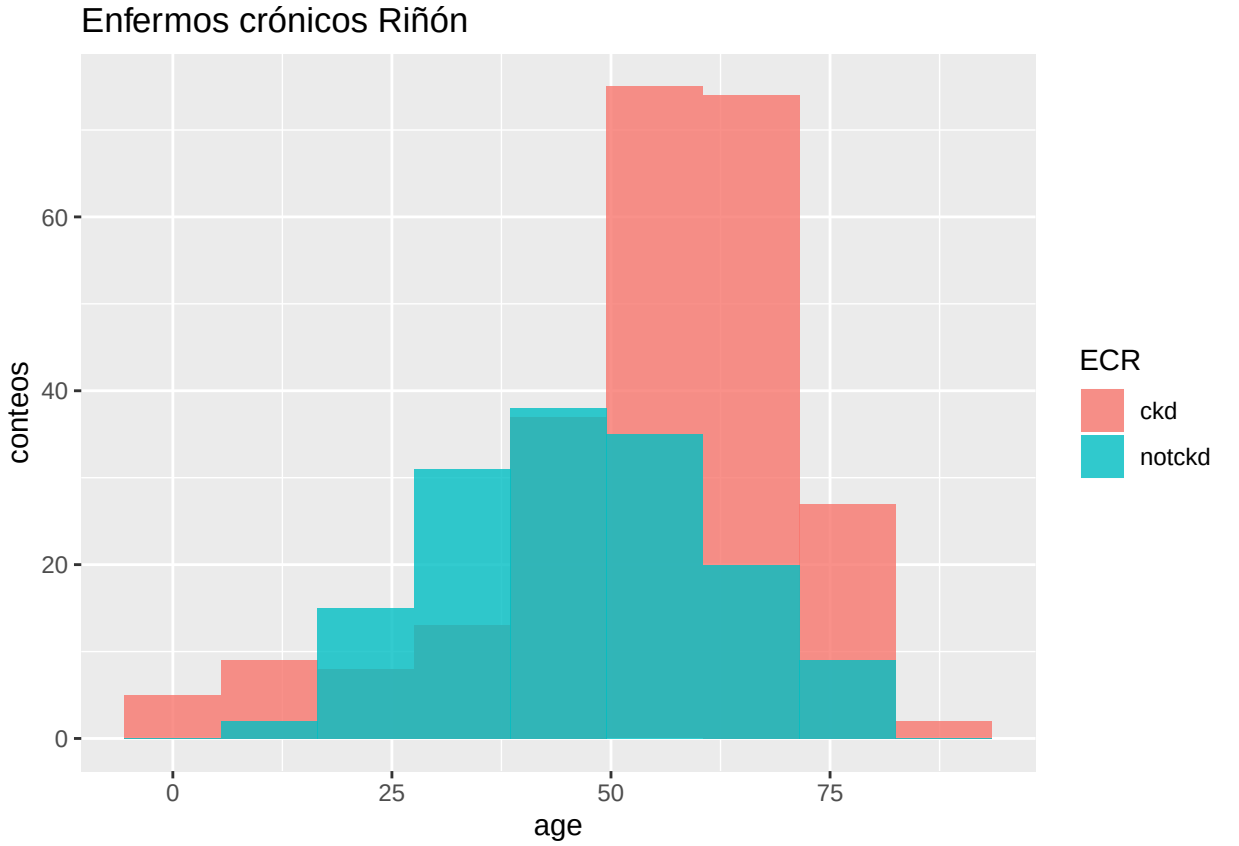
Tras el estudio realizado podemos concluir:

- Existe una relación lineal positiva moderada, entre las variables glóbulos rojos y la hemoglobina.
- La relación es más evidente cuando existe un grado de desnutrición en el paciente.
- Si tenemos en cuenta la hipertensión observamos:
  - Relación lineal positiva moderada de las variables rc y hemo
  - Valores mayores de creatinina en pacientes hipertensos.
  - Valores mayores de globulos rojos en pacientes normotensionados
- No habría diferencias entre las medias de las creatininas de la población con diabetes en relación con aquellos pacientes que no han desarrollado la enfermedad.
- Ser diabético hace predisposición o es factor de riesgo, para contraer la enfermedad. La diabetes aumenta los niveles de creatinina y disminuye los niveles de hematorcritos o glóbulos rojo, provocando anemia. Por lo que la diabetes es un factor que puede afectar a la salud del riñón. De todas maneras, no siendo diabético se puede dar que padezcas enfermedad y los parámetros serán muy parecidos a los diabéticos. Mayoritáriamente se encuetra diabétes entre los pacientes con enfermedad en el riñón entre los 50 y 70 años.

- La enfermedad coronaria comporta una subida de la tensión arterial y por tanto, es factor de riesgo para los pacientes crónicos, sobretodo a partir de los 50 años.

Como podemos ver:

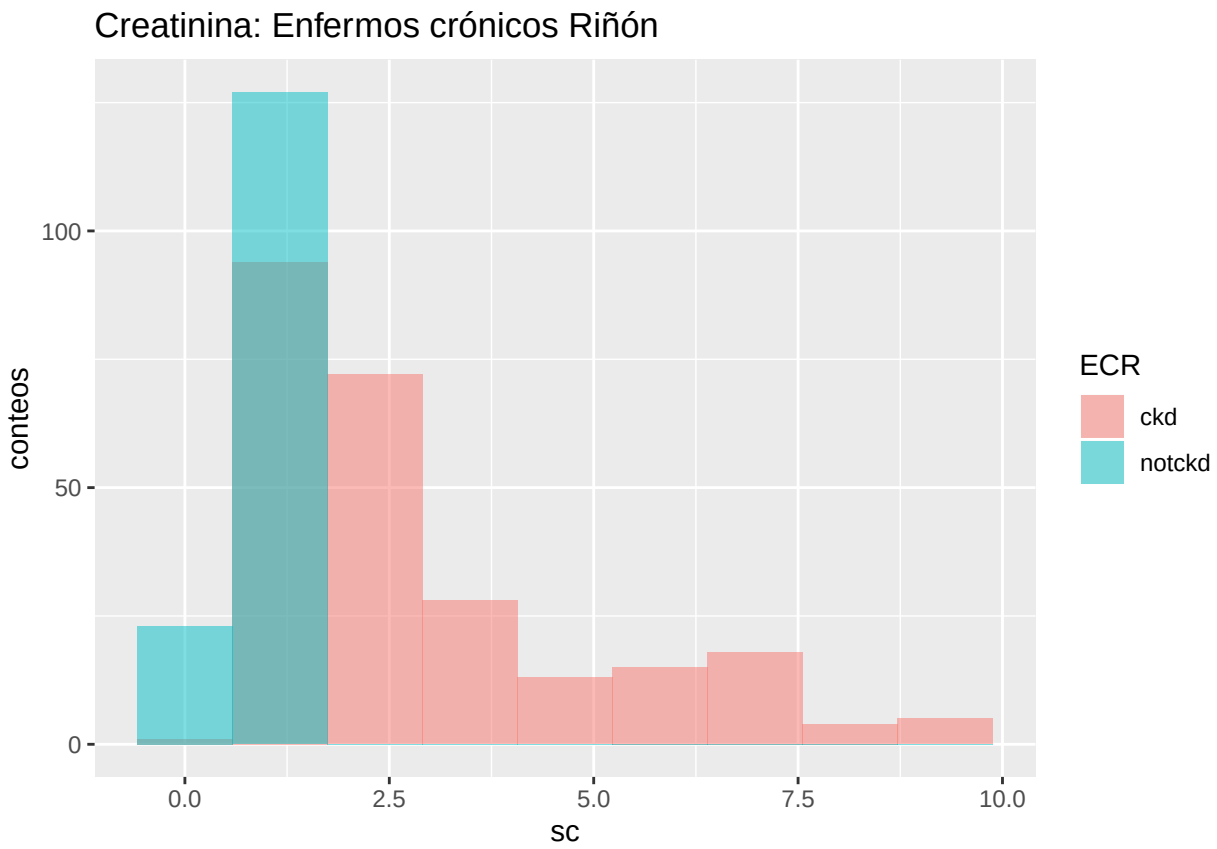
```
#Gráfica edad en enfermos crónicos
ggplot(data = datos,
       mapping = aes(x = age, fill = factor(classification))) +
  geom_histogram(bins = 9, position = 'identity', alpha = 0.8) +
  labs(title = 'Enfermos crónicos Riñón',
       fill = 'ECR',  x = 'age', y = 'conteos')
```



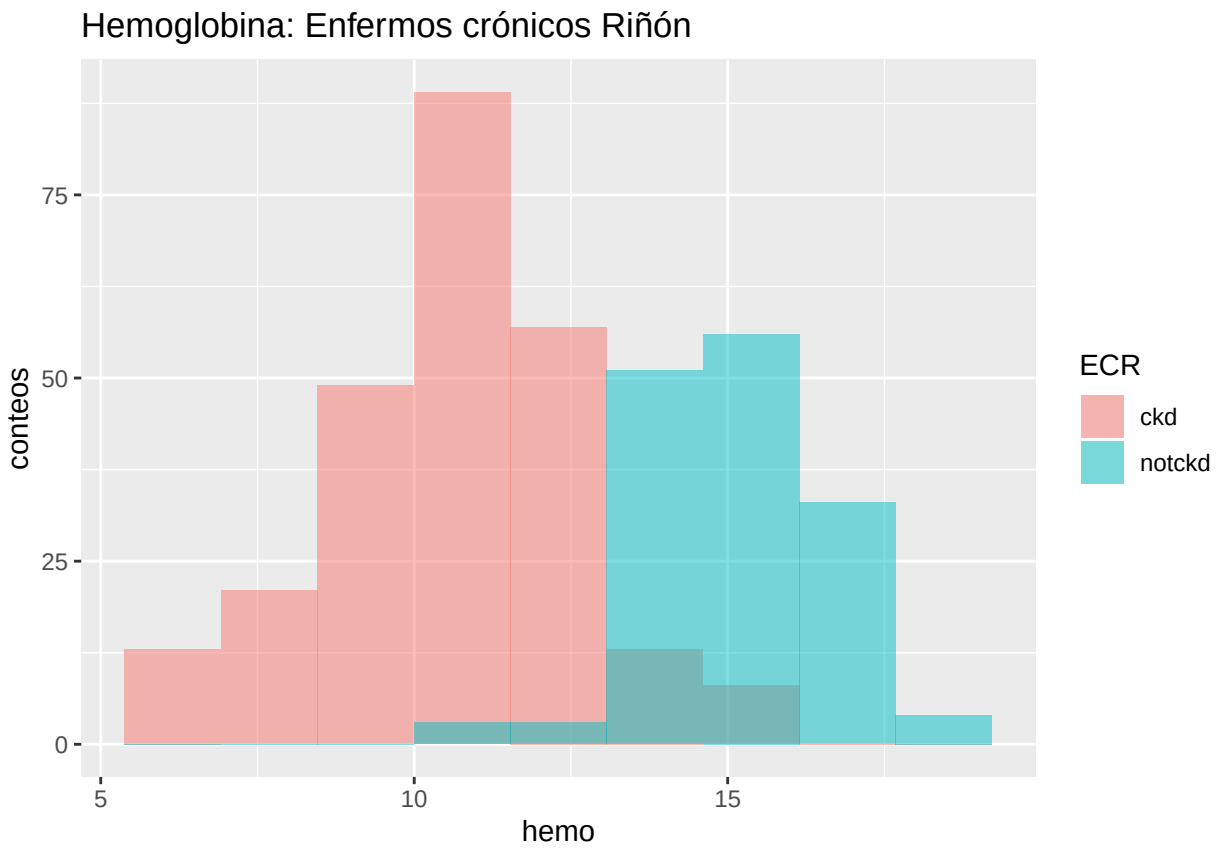
Entre los 50 y 75 años es donde se mueve el gran volumen de enfermos crónicos del riñón. Como ya hemos observado, a esas la hipertensión, las enfermedades coronarias y la diabetes suelen ser más más frecuentes. Siendo, dichas enfermedades, un factor que favorece un mal funcionamiento de los riñones y por tanto, provocando la enfermedad crónica del riñón.

Resumiendo, los enfermos crónicos padecen creatinina alta, los glóbulos rojos (al igual que los hematocritos), suelen estar bajos:

```
par(mfrow=c(1,2))
#Gráfica edad en enfermos crónicos
ggplot(data = datos,
       mapping = aes(x = sc, fill = factor(classification))) +
  geom_histogram(bins = 9, position = 'identity', alpha = 0.5) +
  labs(title = 'Creatinina: Enfermos crónicos Riñón',
       fill = 'ECR',  x = 'sc', y = 'conteos')
```

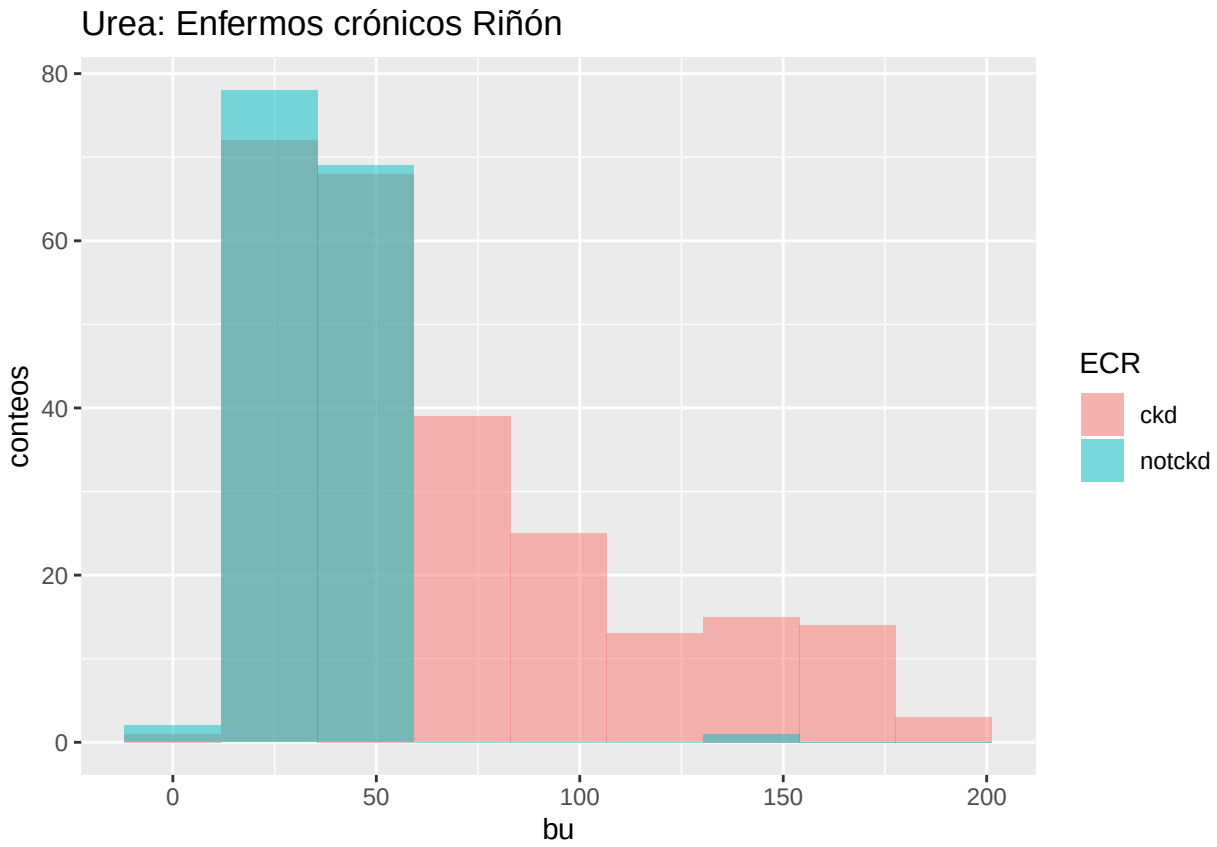



```
#Gráfica edad en enfermos crónicos
ggplot(data = datos,
       mapping = aes(x = hemo, fill = datos$classification))+
  geom_histogram(bins = 9, position = 'identity', alpha = 0.5) +
  labs(title = 'Hemoglobina: Enfermos crónicos Riñón',
       fill = 'ECR',  x = 'hemo', y = 'conteos')
```

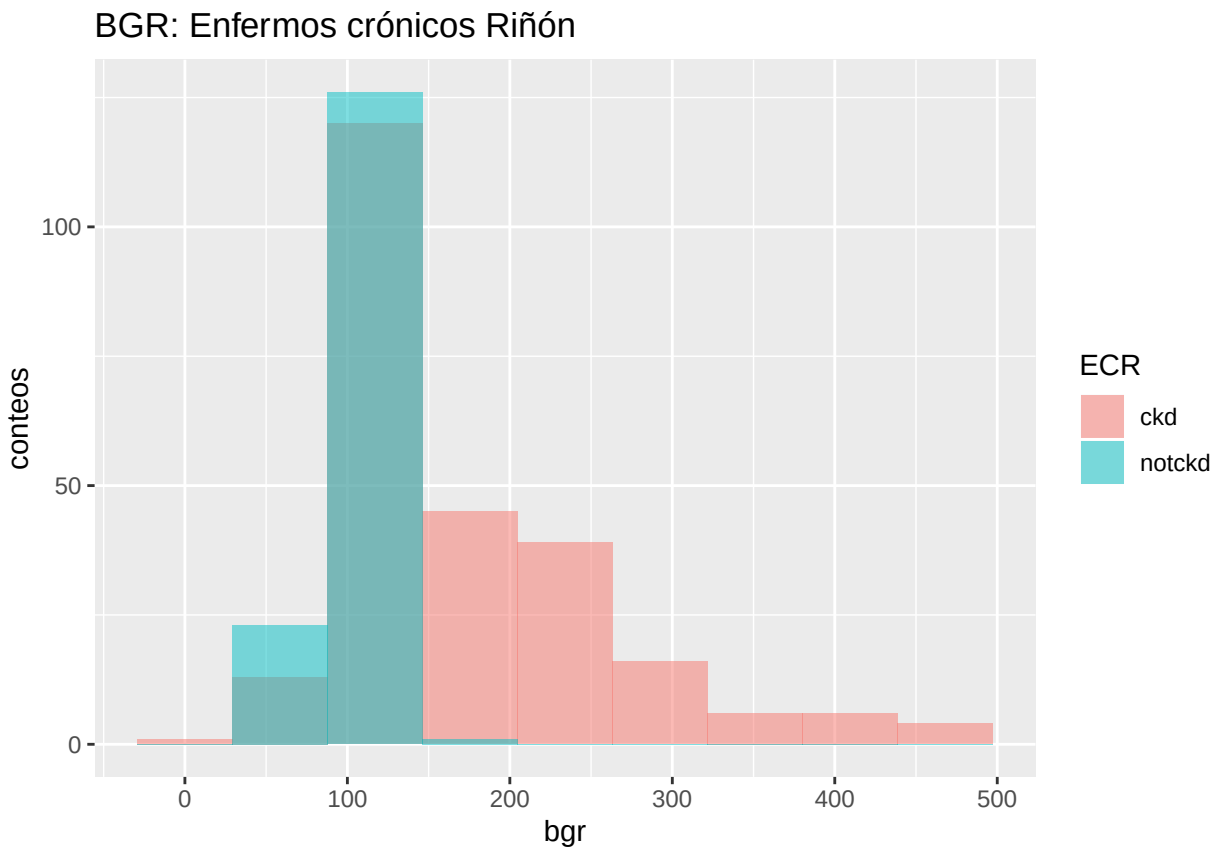


Una alta urea en sangre (a partir de 50 mg) y una alta glucosa en sangre es determinante, ya que parámetros tan altos son indicadores de disfunción renal.

```
par(mfrow=c(1,2))
#Gráfica edad en enfermos crónicos
ggplot(data = datos,
       mapping = aes(x = bu, fill = factor(classification))) +
  geom_histogram(bins = 9, position = 'identity', alpha = 0.5) +
  labs(title = 'Urea: Enfermos crónicos Riñón',
       fill = 'ECR',  x = 'bu', y = 'conteos')
```




```
#Gráfica edad en enfermos crónicos
ggplot(data = datos,
       mapping = aes(x = bgr, fill = datos$classification))+
  geom_histogram(bins = 9, position = 'identity', alpha = 0.5) +
  labs(title = 'BGR: Enfermos crónicos Riñón',
       fill = 'ECR',  x = 'bgr', y = 'conteos')
```

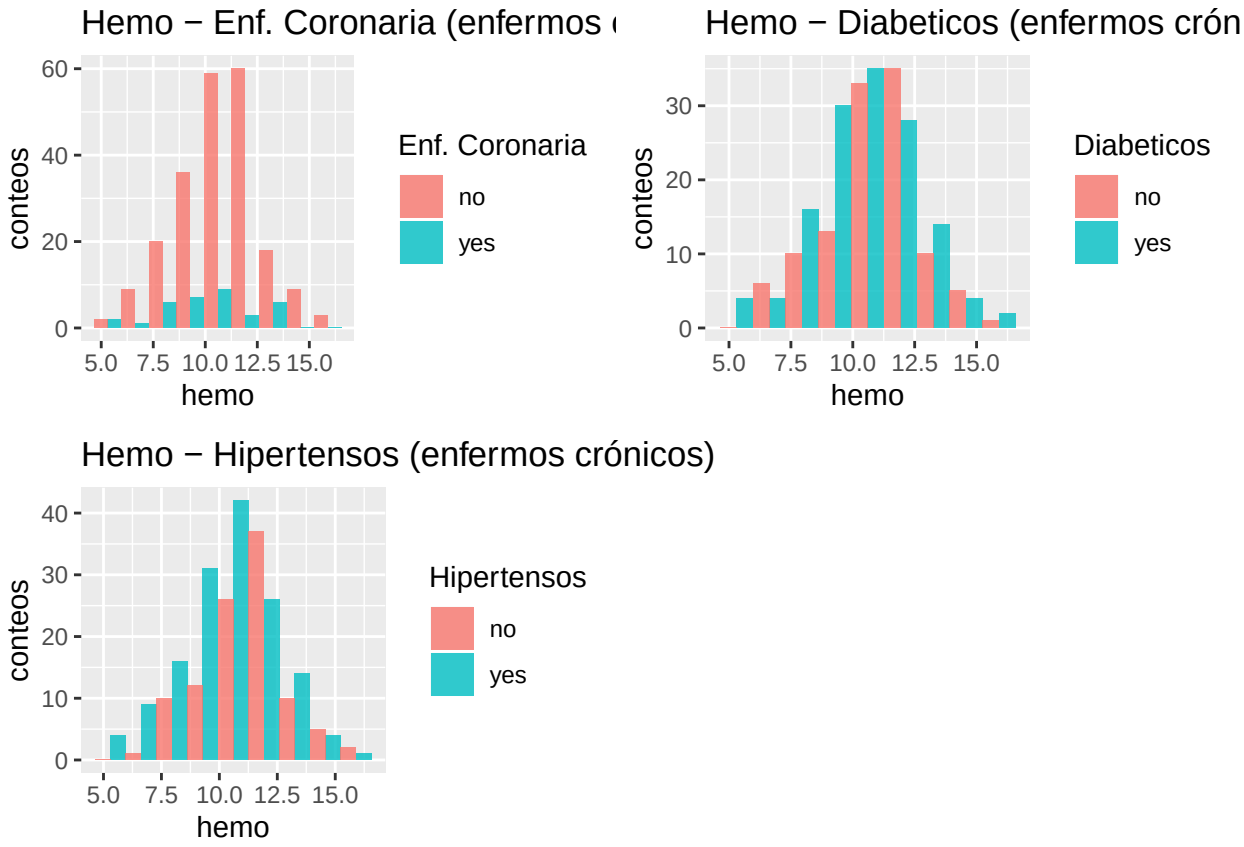


La diabetes, la hipertensión, la anemia, la enfermedad cardiaca son enfermedades que pueden ayudar a la aparición de la enfermedad crónica renal. Hay que decir que en estos datos, la mayoría que tienen algún tipo de enfermedad son pacientes diagnosticados de enfermedad crónica renal. Seguramente, el ser un paciente crónico puede hacer desarrollar cualquiera de las otras enfermedades. Independientemente, se hecha en falta tener enfermos de diabetes, hipertensión, anemia, enfermedad cardiaca que no hayan desarrollado la enfermedad crónica y, poder así, obtener un mejor análisis de las variaciaciones de unos y otros. En estas circumstancias, es difícil diagnosticar una enfermedad crónica cuando, como hemos visto, resulta que tenemos resultados tan parecidos sean o no diabéticos, hipertensos o presenten una enfermedad cardiaca.

Como ejemplo de lo explicado, ponemos los gráficos de enfermos cardiacos que padecen o no padencen las enfermedades nombradas. Como se puede observar, tenemos, tanto pacientes con enfermedad como pacientes sin enfermedad mueven en el mismo rango:

```
par(mfrow=c(1,3))
#Gráfico Hemoglobina enfermos crónicos con enfermedad coronaria
gg1 <- ggplot(data = datos[datos$classification == 'ckd',],
       mapping = aes(x = hemo, fill = factor(cad))) +
  geom_histogram(bins = 9, position = 'dodge', alpha = 0.8) +
  labs(title = 'Hemo - Enf. Coronaria (enfermos crónicos)',
       fill = 'Enf. Coronaria',  x = 'hemo', y = 'conteos')

#Gráfico Hemoglobina enfermos crónicos con enfermedad coronaria
gg2 <- ggplot(data = datos[datos$classification == 'ckd',],
       mapping = aes(x = hemo, fill = factor(dm))) +
  geom_histogram(bins = 9, position = 'dodge', alpha = 0.8) +
  labs(title = 'Hemo - Diabeticos (enfermos crónicos)',
       fill = 'Diabeticos',  x = 'hemo', y = 'conteos')
```

```r
#Gráfico Hemoglobina enfermos crónicos con enfermedad coronaria
gg3 <- ggplot(data = datos[datos$classification == 'ckd',],
       mapping = aes(x = hemo, fill = factor(htn))) +
  geom_histogram(bins = 9, position = 'dodge', alpha = 0.8) +
  labs(title = 'Hemo - Hipertensos (enfermos crónicos)',
       fill = 'Hipertensos',  x = 'hemo', y = 'conteos')

grid.arrange(gg1, gg2, gg3,
          ncol = 2, nrow = 2)
```



Como podemos ver, tengan o no tengan la enfermedad los parámetros son muy parecidos en unos u otros, siendo todos pacientes crónicos.