Guía de Estiduo 2: Descripción de los Datos

Análisis de Métodos Cuantitativos
Semana 2
Diplomados Online

Febrero 2019

Índice

En esta guía de estudio, exploramos algunos de los procedimientos disponibles en R para resumir los datos estadśticos, que apr4endimos a introducir en R en la calse anterior, y dar algunos ejemplos de escritura de programas.

1. Medidas de Tendencia Central

Las medidas de tendencia central son puntos típicos o centrales en los datos. Las más utilizadas son la media y la mediana.

Definición 1. *La media es la suma de todos los valores dividida por el número de casos, excluyendo los valores que faltan.*

Para obtener la media en R, cargamos los datos de obesidad y presión sanguinea, *bp.obese*, del paquete *ISwR* (Introductory Statistics with R), y calculemos la media de la variable obesidad (*obese*),

```
> library("ISwR")
> data("bp.obese")
> mean(bp.obese$obese)

[1] 1.313039
```

Por lo tanto, la relación entre el peso real y el peso ideal medio es de 1,313039.

Otra medida de dispersión muy usada es la mediana.

Definición 2. *La mediana es el valor medio del conjunto de datos; el 50 % de las observaciones son menores y el 50 % mayores que este valor.*

En R, si continuamos utilizando los mismos datos, hacemos,

```
> median(bp.obese$obese)
```

```
[1] 1.285
```

lo cual significa que para el 50 % de las personas su tasa de obesidad se encuentra por debajo de 1,285, el 50 % restante, se encuentra por encima.

2. Medidas de Dispersión

Las medidas de dispersión, como su nombre indica, estiman la dispersión o variación de los datos. Hay muchas maneras de hacer esto, y consideramos algunas de las más comunes.

Definición 3. *(Rango)*
El rango se define como la diferencia entre los valores máximos y mínimos.

En R, si continuamos trabajando con los datos de obesidad, podemos calcular el rango de los datos de la forma siguiente, primero calculamos los valores máximo y mínimos con al función *range(9)*

```
> range(bp.obese$obese)
```

```
[1] 0.81 2.39
```

luego, asignamos estos valores a una variable, por ejemplo, *rgobese*, par aluego con la función *diff()*, podamos calcular la diferencia de los valores arrojados por la función *range()*, así,

```
> rgobese<-range(bp.obese$obese)
> diff(rgobese)
```

```
[1] 1.58
```

esto nos dice que, el tamañ del intervalo en el cual se encuentran los datos, es de 1,58.

Definición 4. *(Desviación Estándar)*
*La desviación estándar (*sd*) mide la extensión de los datos o cuánto se desvía de la media. Es la raíz cuadrada de las desviaciones medias cuadradas de la media.*

Una pequeña desviación estándar implica que la mayoría de los valores están cerca del promedio. Una gran desviación estándar indica que los valores están muy extendidos por encima y por debajo de la media. En R, lo calculamos de la siguiente manera,

```
> sd(bp.obese$obese)
```

```
[1] 0.2578387
```

Definición 5. *(Cuantiles)*
Los cuantiles dividen los datos en proporciones, generalmente en cuartos llamados cuartiles, décimas llamadas deciles y porcentajes llamados percentiles.

En R, escribimos los siguiente,

```
> quantile(bp.obese$obese)

    0%    25%    50%    75%   100%
0.8100 1.1425 1.2850 1.4300 2.3900
```

esto nos arroja por defecto el valor de los cuartiles, y no indica que, por ejemplo, el 25 % de los datos se encuentra por debajo de 1,1425, y así sucesivamente.

Para los deciles, tenemos que establecer los puntos deciles, es decir, con la función *seq()*, partimos el intervalo [0,1] en 10, de la forma siguiente,

```
> seq(0,1,0.10)

 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

luego, en la función *quantile()*, establecemos este vector como las probabilidades, es decir,

```
> quantile(bp.obese$obese, probs = seq(0,1,0.10))

   0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
0.810 1.040 1.110 1.173 1.240 1.285 1.326 1.377 1.500 1.589 2.390
```

Mientras, que para los percentiles, hacemos,

```
> quantile(bp.obese$obese, probs = seq(0,1,0.01))

    0%     1%     2%     3%     4%     5%     6%     7%     8%     9%    10%
0.8100 0.8804 0.9202 0.9309 0.9604 0.9720 1.0106 1.0207 1.0300 1.0309 1.0400
   11%    12%    13%    14%    15%    16%    17%    18%    19%    20%    21%
1.0400 1.0424 1.0613 1.0700 1.0715 1.0816 1.0900 1.0936 1.1100 1.1100 1.1142
   22%    23%    24%    25%    26%    27%    28%    29%    30%    31%    32%
1.1300 1.1300 1.1324 1.1425 1.1526 1.1600 1.1628 1.1700 1.1730 1.1831 1.1900
   33%    34%    35%    36%    37%    38%    39%    40%    41%    42%    43%
1.1900 1.1934 1.2000 1.2036 1.2137 1.2238 1.2339 1.2400 1.2400 1.2442 1.2500
   44%    45%    46%    47%    48%    49%    50%    51%    52%    53%    54%
1.2544 1.2600 1.2600 1.2647 1.2748 1.2800 1.2850 1.2900 1.2900 1.2900 1.2954
   55%    56%    57%    58%    59%    60%    61%    62%    63%    64%    65%
1.3055 1.3100 1.3157 1.3200 1.3200 1.3260 1.3300 1.3300 1.3363 1.3400 1.3465
   66%    67%    68%    69%    70%    71%    72%    73%    74%    75%    76%
1.3566 1.3667 1.3700 1.3700 1.3770 1.4013 1.4172 1.4273 1.4300 1.4300 1.4376
   77%    78%    79%    80%    81%    82%    83%    84%    85%    86%    87%
```

```
1.4708 1.4956 1.5000 1.5000 1.5081 1.5346 1.5483 1.5584 1.5600 1.5600 1.5687
   88%    89%    90%    91%    92%    93%    94%    95%    96%    97%    98%
1.5700 1.5789 1.5890 1.6264 1.6392 1.6493 1.6688 1.7270 1.7396 1.7497 2.0342
   99%   100%
2.1984 2.3900
```

Finalmente, una manera rápida de obtener información estadistica de los datos, es utililizar la función *summary()* en R, así

```
> summary(bp.obese)
```

```
      sex             obese             bp
 Min.   :0.0000   Min.   :0.810   Min.   : 94.0
 1st Qu.:0.0000   1st Qu.:1.143   1st Qu.:116.0
 Median :1.0000   Median :1.285   Median :124.0
 Mean   :0.5686   Mean   :1.313   Mean   :127.0
 3rd Qu.:1.0000   3rd Qu.:1.430   3rd Qu.:137.5
 Max.   :1.0000   Max.   :2.390   Max.   :208.0
```

3. Creando Funciones

Ocasionalmente puede que necesitemos algunas funciones estadísticas que no están disponibles en R, por lo que se deberá crear una función propia. Tomemos como ejemplo el coeficiente de asimetría, que mide en qué medida los datos difieren de la simetría. Un conjunto de datos perfectamente simétrico tendrá una asimetría de 0, cuando el coeficiente de asimetría es sustancialmente mayor que cero, los datos son positivamente asimétricos con una larga cola a la derecha, y un coeficiente de asimetría negativo significa que los datos tienen una larga cola a la izquierda[1].

El coeficiente de simetría es definido como,

$$\text{Asimetria} = \frac{\sqrt{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{(\sum_{i=1}^{n}(x_i - \overline{x})^2)^{3/2}} \tag{1}$$

Ejemplo 1. *(Un programa que calcule la asimetría)*
La siguiente sintaxis calcula el coeficiente de asimetría de un conjunto de datos y lo asigna a una función llamada skew *que tiene un argumento* (x).

```
> skew <- function(x) {
+ sum2 <- sum((x-mean(x))^2)
+ sum3 <- sum((x-mean(x))^3)
+ skew <- (sqrt(length(x))*sum3)/(sum2^(1.5))
+ skew}
```

Cuando se ha definido skew*, se puede calcular la asimetría en cualquier conjunto de datos, por ejemplo,*

```
> skew(bp.obese$obese)
```

```
[1] 1.260221
```

Esto indica que los datos están ligeramente positivamente sesgados.

[1] Más adelante en el curso, retomaremos estos conceptos y los discutiremos con profundidad

4. Representaciones Gráficas

Además de los resúmenes numéricos de los datos estadísticos, hay varias representaciones gráficas disponibles en R que tienen un impacto más dramático en el usuario y permiten una mejor comprensión de los datos. La facilidad y velocidad con la que se pueden producir pantallas gráficas es una de las características importantes de R. Ahora examinamos algunas pantallas gráficas comunes.

4.1. Boxplots

Un diagrama de caja es un resumen gráfico basado en la mediana, el cuartil y los valores extremos. Para visualizar los datos de obesidad con boxplot, hacemos,

```
> boxplot(bp.obese$obese)
```

y obtenemos,

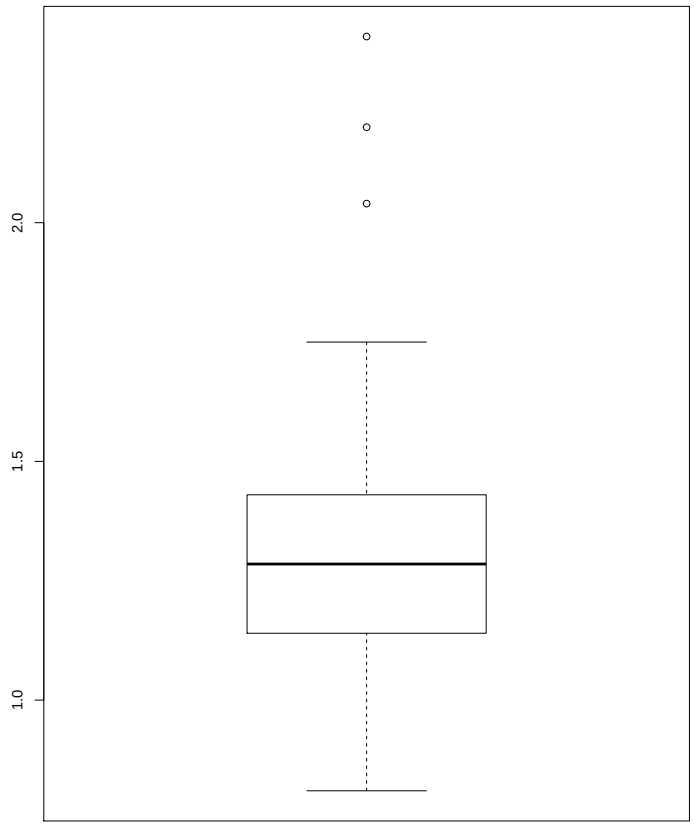

Figura 1: Boxplot de los datos bp.obese

A menudo el plot anterior es llamado de *caja y bigote*, la caja representa el rango intercuartil que contiene el 50 % de los casos. Los bigotes son las líneas que se extienden desde la caja hasta los valores

más altos y más bajos. La línea que atraviesa el recuadro indica la mediana. Para colocarle las leyenda, hacemos,

```
> boxplot(bp.obese$obese, xlab = "Obesidad", ylab = "Tasa")
```

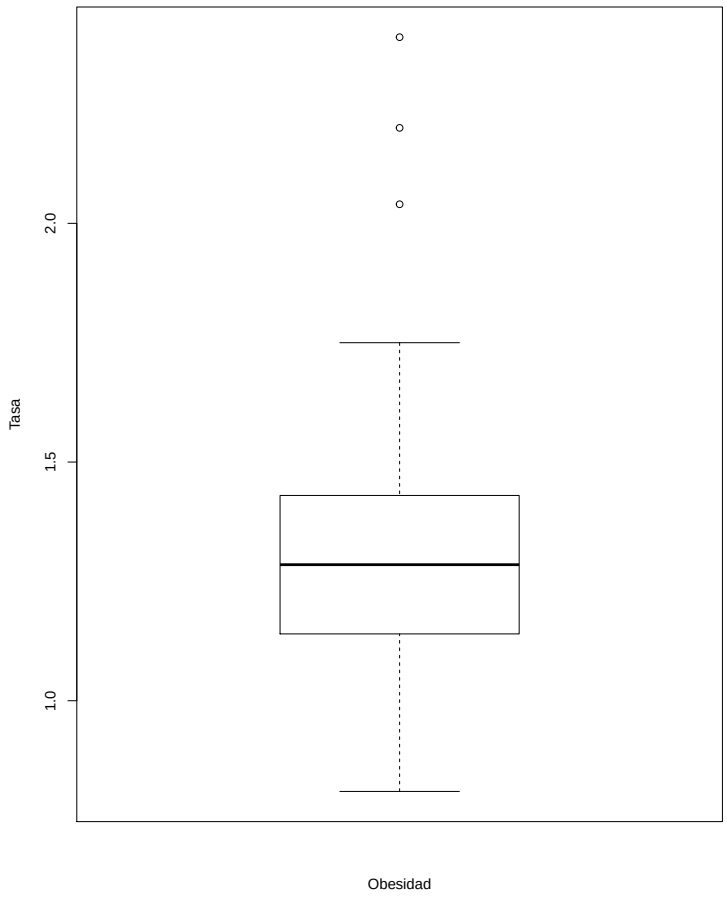

Figura 2: Boxplot de los datos bp.obese con leyendas

Se pueden mostrar varios gráficos de caja en el mismo eje añadiendo argumentos adicionales a la función de gráfico de caja. Por ejemplo, utilizando datos de supervivencia del SIDA en Australia (del paquete de R *MASS*), tenemos

```
> library("MASS")
> boxplot(Aids2$diag, Aids2$death,xlab="Datos de supervivencia del SIDA en Australia")
```

Observe, en el gráfica (3) los puntos fuera de los bigotes, estos valores se denominan valores atípicos y representan casos con longitudes superiores al extremo superior o inferior al extremo inferior de la caja. Estos puntos se consideran atípicos de los datos en general, siendo extremadamente bajos o extremadamente altos en comparación con el resto de los datos.

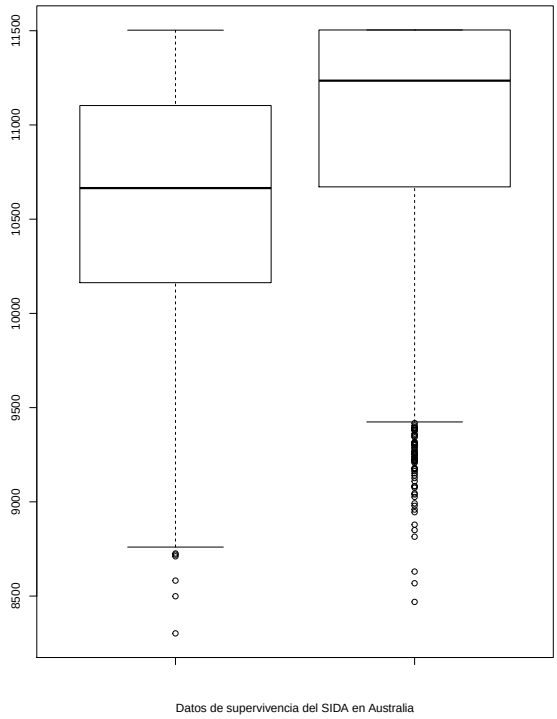

Figura 3: Datos de supervivencia del SIDA en Australia"

4.2. Histogramas

Un histograma es una representación gráfica de las frecuencias en las categorías de una variable y es la forma tradicional de examinar la forma de los datos. En R,

```
> hist(Aids2$age, xlab ="Número de Pacientes", ylab="Edades", main = "Edades de los Pacientes con S
```

Como podemos ver, gráfica (4), eel histograma da el recuento de las observaciones que caen dentro de las categorías. R elige un número adecuado de categorías, a menos que se especifique lo contrario.

Con la función *par()* se puede representar varios histoogramas en un diagrama.

```
> par (mfrow = c(2,2))
> hist(Aids2$diag)
> hist(Aids2$death)
> hist(Aids2$age)
```

4.3. Tallo y Hoja

El diagrama de tallo y hoja, una forma más moderna de mostrar los datos, es una representación de la forma utilizando los números reales observados. Al igual que el histograma, el diagrama de tallo y hoja da las frecuencias de las categorías de la variable, pero va más allá y da los valores reales de cada categoría.

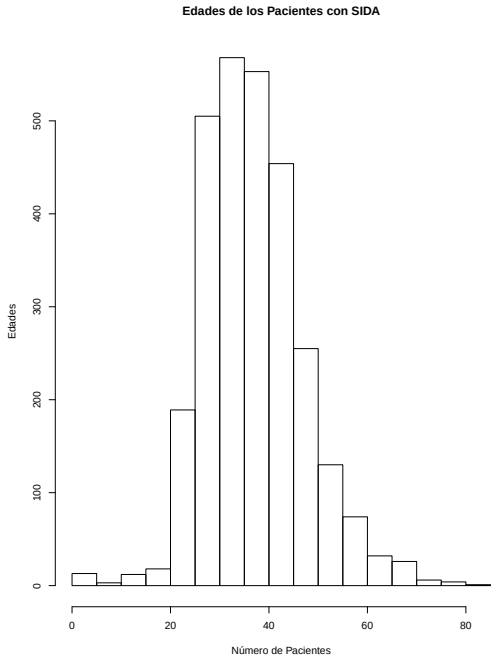

Figura 4: Histograma de supervivencia del SIDA en Australia por edad

Para la serie de datos de *Aids2*, realizamos un diagrama de tallo y hoja para la variable edad, lo cual se realiza con la función *stem()*, es decir,

```
> stem(Aids2$age)

  The decimal point is 1 digit(s) to the right of the |

  0 | 0000111333
  0 | 555669
  1 | 122222344
  1 | 55566677788899
  2 | 00000001111111111112222222222222222233333333333333333333333333344+38
  2 | 55555555555555555555555555555555555555555555555555555555555555555+386
  3 | 00000000000000000000000000000000000000000000000000000000000000000+488
  3 | 55555555555555555555555555555555555555555555555555555555555555555+495
  4 | 00000000000000000000000000000000000000000000000000000000000000000+392
  4 | 55555555555555555555555555555555555555555555555555555555555555555+215
  5 | 00000000000000000000000000000000000011111111111111111111111111111+64
  5 | 55555555555555555555556666666666666666666666666777777777777777777888+9
  6 | 00000000111111222222233333334444444
  6 | 55555566666666666777789999999
  7 | 000123334
  7 | 788
  8 | 02
```

Aquí podemos ver los valores de las observaciones individuales, asćomo la forma de los datos en su

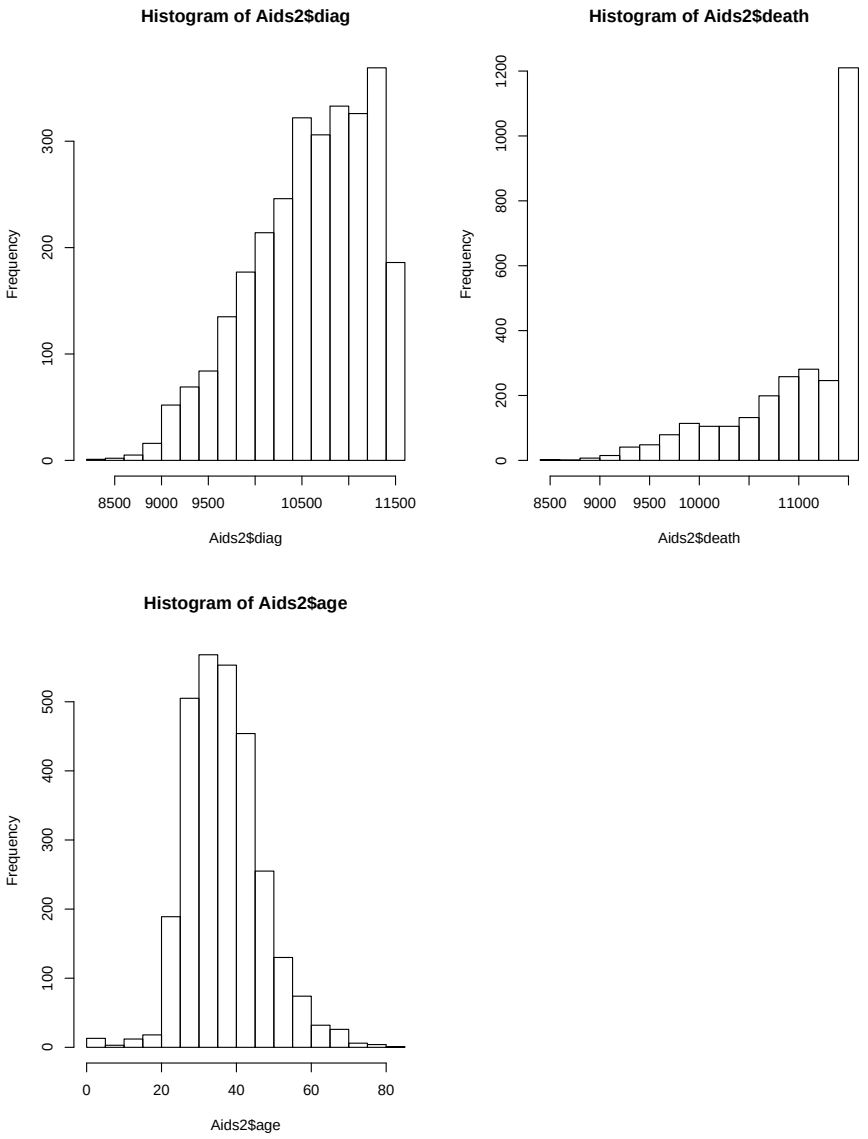

Figura 5: Histograma de supervivencia del SIDA en Australia

conjunto. En el gráfico podemos observar que el grueso de las edades se encuentran entre 20 y 55 años.

4.4. Scatters Plots

Las gráficas de los datos son útiles para investigar las relaciones entre las variables. Por ejemplo, si se desea ver l arelación entre la obesidad y la presión arterial, par alos datos *bp.obese*, hacemos,

```
> plot(bp.obese$obese, bp.obese$bp, xlab = "Obesidad", ylab = "Presi\'on sanguinea")
```

En el gráfico se aprecia, que en apariencia, no existe una relación lineal entre los datos.

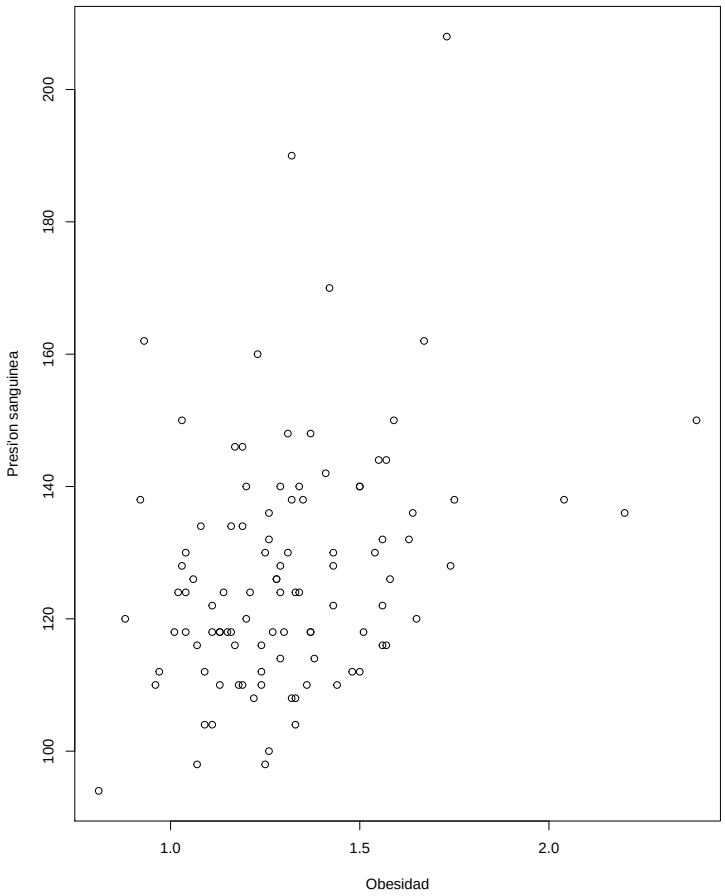

Figura 6: Plot de Obesidad vs. Presión Sanguinea

Cuando hay más de dos variables, utilizamos la librería *car*, y utilizamos la función *scatterplotMatrix()*, es decir,

```
> library("car")
> scatterplotMatrix(bp.obese)
```

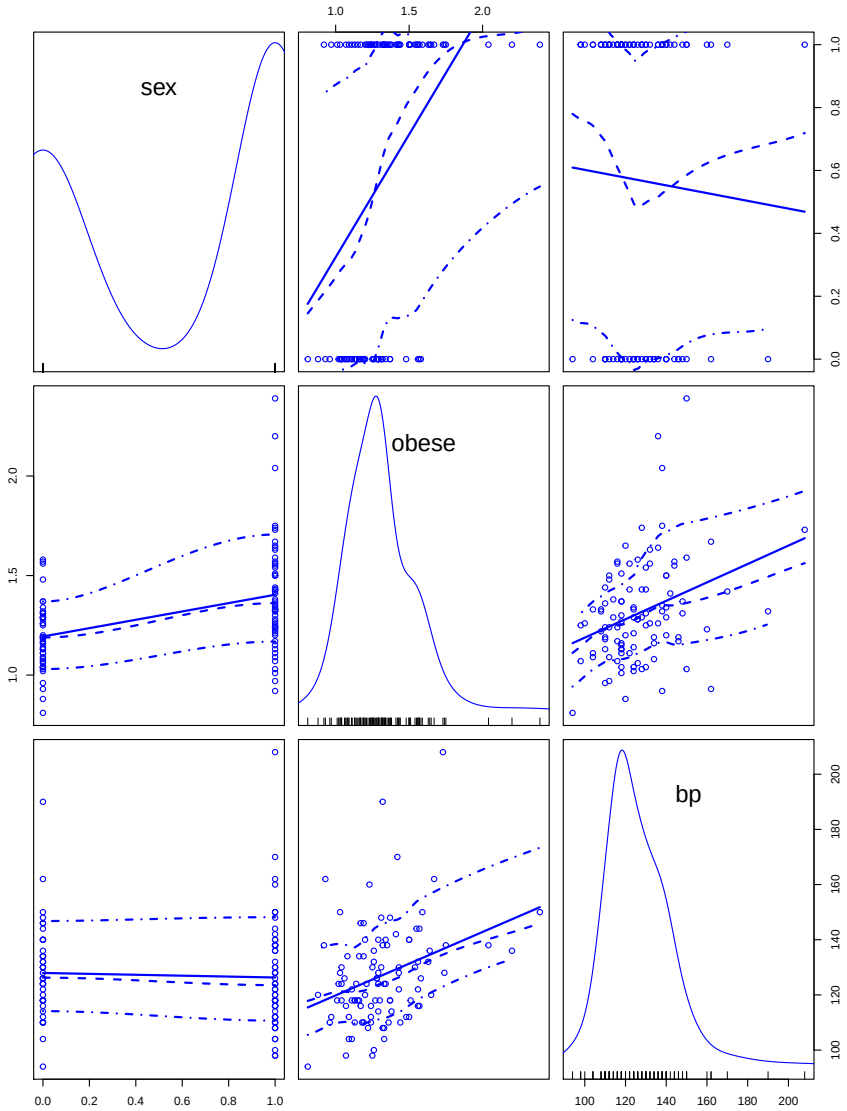

Figura 7: ScatterPlot de la serie de datos bp.obese, donde se observa la comparación entre las variables sexo, obesidad y presión sanguinea