MEMORIA PRÁCTICA – 4:
Correlación y regresión.

Escuela Politécnica Superior - Universidad Autónoma de Madrid

GRADO EN INGENIERÍA BIOMÉDICA

BIOESTADÍSITICA

Versión del documento número 1
Práctica realizada por: *Laura Sánchez Garzón*
Fecha: *14 de abril de 2023*
Profesorado de la práctica: *Mercedes Sotos Prieto y María Téllez Plaza*

1. ÍNDICE

Contenido

2. INTRODUCCIÓN

En la práctica anterior se presentaron distintos procedimientos para estudiar la asociación entre variables categóricas. Aquí se describirán funciones diseñadas para analizar variables bivariantes continuas.

3. EJERCICIOS GUIADOS

3.1. Correlación entre variables continuas

Un coeficiente de correlación es una medida de asociación entre dos variables aleatorias, que no es sensible a cambios de escala a diferencia de la covarianza.
Va de −1 a +1, donde **|1| indica correlación perfecta y 0 significa que no hay correlación**. El **signo es negativo cuando los valores grandes de una variable se asocian con valores pequeños de la otra y positivo si ambas variables tienden a ser grandes o pequeñas simultáneamente**.
Esta sección describe el cálculo en R de las dos medidas de correlación más utilizadas, el **coeficiente de correlación linear de Pearson y el de correlación de los rangos de Spearman**.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Figura 1: Definición del Coeficiente de correlación linear de Pearson (S_{xy}/S_xS_y). Es de especial interés porque tiene una conexión con la regresión lineal simple.

Todas las funciones estadísticas elementales en R requieren que todos los valores no falten o que se indique explícitamente qué se debe hacer con los casos con valores faltantes. Para **mean(), var(), sd()** y funciones similares cuya entrada es un vector, se puede proporcionar el argumento **na.rm=T** para indicar que los valores faltantes deben eliminarse antes del cálculo.
El Código 1, muestra ejemplos de uso de las funciones cor() y cor.test() y como indicar que se debe excluir los registros con valores faltantes.

3.1.1. Código 1

```
> library(ISwR)
> data(thuesen)
> ?thuesen
> attach(thuesen)
> cor(blood.glucose,short.velocity,use="complete.obs")
[1] 0.4167546
> cor(thuesen,use="complete.obs")
               blood.glucose short.velocity
blood.glucose      1.0000000      0.4167546
short.velocity     0.4167546      1.0000000
> cor.test(blood.glucose,short.velocity)

        Pearson's product-moment correlation

data:  blood.glucose and short.velocity
t = 2.101, df = 21, p-value = 0.0479
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.005496682 0.707429479
sample estimates:
      cor
0.4167546
```

Figura 2: Salida Código 1.

Se debe cargar la base de datos Thuesen, del paquete "ISwR" (tiene 24 filas y 2 columnas). Ésta contiene velocidad de acortamiento ventricular y glucosa en sangre de pacientes diabéticos tipo 1.

Tras cargar la base de datos, se calcula el coeficiente de correlación de Pearson (función cor) de la relación entre la glucosa en sangre con velocidad de acortamiento ventricular y de la base de datos completa.

Por último, se aplica la función cor.test para indicar si la correlación es significativamente diferente de cero.

Cuando el valor es |1| hay una relación linear perfecta (por eso también se le llama a veces "coeficiente de correlación lineal").
En el caso de la relación entre la glucosa en sangre con velocidad de acortamiento ventricular (función *cor()*), se obtiene una un coeficiente de 0.4167546, es decir, la relación linear está alejada de ser perfecta.
Al aplica la función sobre objeto del data.frame, se obtiene la matriz completa de correlaciones entre todas las variables de la base de datos:

	Glucosa en sangre	Velocidad de acortamiento ventricular
Glucosa en sangre	1.0000000	0.4167546
Velocidad de acortamiento ventricular	0.4167546	1.0000000

Figura 3: Matriz de relaciones.

Observar que, como es de esperar, la relación de la glucosa en sangre consigo misma es 1 (correlación perfecta), como ocurre también en la velocidad de acortamiento ventricular.

Y como se obtuvo anteriormente, de la relación entre la glucosa en sangre con velocidad de acortamiento ventricular, se obtiene una un coeficiente de 0.4167546.

Al aplicar *cor.test()*, se indica si la correlación es significativamente diferente de cero. Los resultados concluyen que se rechaza la hipótesis nula (hay relación), y se acepta la alternativa (la correlación entre glucosa en sangre y velocidad de acortamiento ventricular no es igual a 0). Además, la función indica el estadístico de contraste (2.101), los grados de libertad (21), el p-valor (0.0479) < 0.05, por tanto, existe correlación lineal con un 95% de confianza, en un rango entre 0.005496682 y 0.707429479 (razón por la que se rechaza la hipótesis nula, ya que r=0.4167546, fuera del rango).

```
> plot(blood.glucose,short.velocity)
```

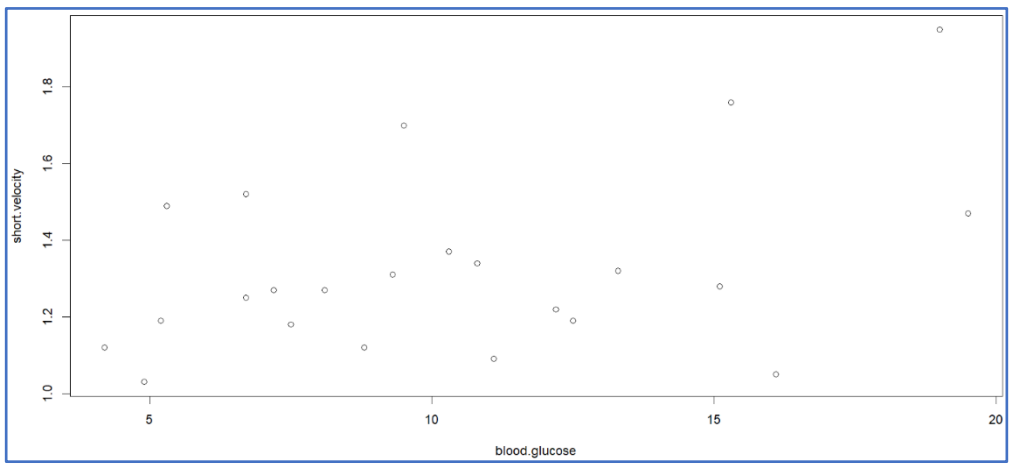

Figura 4: Salida Código 1.

Se aplica la función de gráfico para inspeccionar visualmente la naturaleza de la relación mediante un diagrama de dispersión (el coeficiente de correlación no es una medida de bondad de ajuste al modelo lineal, ya que sólo determina la existencia de un componente lineal, independientemente de la forma de esa relación).

Efectivamente, observando el gráfico, la dispersión es muy notable.

```
> cor.test(blood.glucose,short.velocity,method="spearman")

        Spearman's rank correlation rho

data:  blood.glucose and short.velocity
S = 1380.4, p-value = 0.1392
alternative hypothesis: true rho is not equal to 0
sample estimates:
     rho
0.318002

Warning message:
In cor.test.default(blood.glucose, short.velocity, method = "spearman") :
  Cannot compute exact p-value with ties
```

Figura 5 Salida Código 1.

Se aplica el método de coeficiente de correlación de los rangos de Spearman cuando la relación entre las dos variables es claramente no linear.

Esto se obtiene simplemente reemplazando las observaciones por su rango y calculando la correlación, lo que se especifica a través de una opción para cor.test.

Se obtiene que S de 1380.4, y un p-valor de 0.1392, > 0.05, por lo que se acepta la hipótesis nula, es decir, que rho no es igual a 0 (rho = 0.318002).

3.2. Regresión lineal

A veces es interesante saber si los puntos $\{(x_i, y_i)\}$ de una muestra de una variable continua bivariante pueden ser más o menos bien representados por la recta $y = \beta x + \alpha$.

$$S_{res}^2 = \sum_{i=1}^{n} (y_i - \beta x_i - \alpha)^2$$

Figura 6: Recta de regresión mínimo-cuadrática.

Esta recta de regresión vendrá determinada por aquellos valores de α y β que hagan que la distancia desde los puntos observados en la variable respuesta a los predichos por la recta sean lo más pequeños posibles para todas las observaciones (es decir que minimicen la suma de cuadrados del error, S^2_{res}).

Nótese que **el coeficiente de correlación lineal determinaba el grado de aproximación de los puntos del diagrama de dispersión a una recta**, pero no es una medida de la magnitud de la pendiente de la recta β (o en otras palabras de la magnitud de la asociación) que se estima con la recta de regresión (Figura 7).

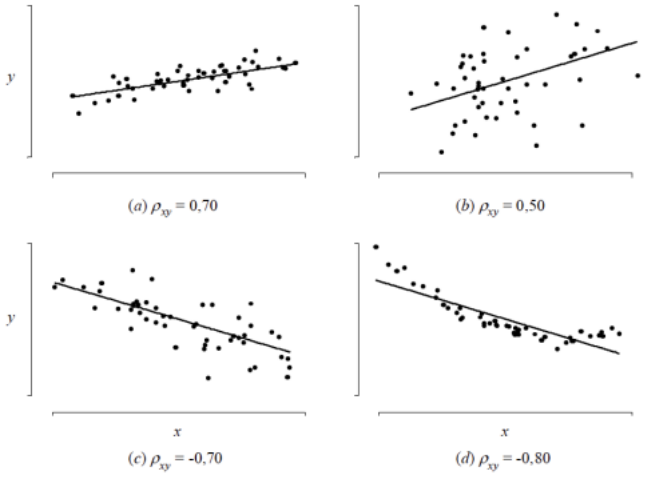

Figura 7: Las variables aleatorias X e Y muestran hipotéticamente distintas pendientes en la recta de regresión (paneles a y b), y distintas formas de la relación subyacente (paneles c y d)

El objetivo principal de esta sección es mostrar cómo realizar un análisis de regresión básico, incluyendo la visualización de gráficos para el diagnóstico del modelo.

3.2.1. Regresión lineal simple

El modelo de regresión simple viene dado por: $\mathbf{y_i = \alpha + \beta x_i + \epsilon_i}$

Donde $\mathbf{y_i}$ **y** $\mathbf{x_i}$ **son, respectivamente, los valores observados de la variable dependiente** (respuesta) **e independiente** (predictora) **para el individuo i**, y la **pendiente** de la recta (el coeficiente de regresión), es decir el cambio promedio (diferencia) en la variable respuesta y por cada incremento en una unidad de la variable predictora x, es **β**. La recta intersecciona con el eje Y en el intercepto (o **constante**) **α**, que también se interpreta como el promedio en la variable respuesta cuando la variable x es igual a 0. El valor $\boldsymbol{\epsilon_i}$ es la **diferencia entre el valor observado** $\mathbf{y_i}$ **y el predicho por el modelo (α + βx$_i$)** para el individuo i, la distribución de los ϵ debe seguir por definición una **distribución N(0, σ²)**.

Existen expresiones cerradas para seleccionar el valor de α y β que minimizan la suma de los mínimos cuadrados (S^2_{res}), en concreto:

$$\hat{\beta} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

Figura 8: A partir de una sola muestra de (xi , yi) se puede calcular el error estándar de las estimaciones calculadas, s.e.(α^) y s.e.(β^) mediante expresiones cerradas conocidas. La pendiente y el intercepto observados en la muestra se desviará del verdadero valor en la población debido a la variabilidad aleatoria del muestreo.

La varianza residual (σ^2) del modelo se estima como $S^2_{res}/(n-2)$, y la desviación estándar (σ) es, por supuesto, la raíz cuadrada de esa cantidad.
Estos **errores estándar** extraídos a partir de la muestra observada son los que veremos que ofrece la salida de la función extractora *summary()* y también se pueden utilizar para calcular los **intervalos de confianza y test de hipótesis**.
El estadístico de contraste consistiría simplemente en dividir la estimación por su error estándar:

$$t = \frac{\hat{\beta}}{s.e.(\hat{\beta})}$$

Figura 9: El estadístico de contraste consiste en probar la hipótesis nula de que β = 0, ya que eso implicaría que la recta es horizontal y, por lo tanto, que las ys son la misma, cualquiera que sea el valor de x (es decir, no hay asociación).
Se distribuye con una distribución t con n - 2 grados de libertad para dicha hipótesis nula.

```
> lm(short.velocity~blood.glucose)

Call:
lm(formula = short.velocity ~ blood.glucose)

Coefficients:
  (Intercept)  blood.glucose
      1.09781        0.02196
```

Figura 9: salida Código 1.

Uso e interpretación del comando lm(), que permite ajustar rectas de regresión en R. Para entenderlo, se ha de saber que el símbolo ~ debe leerse como "en función de".

La salida del comando muestra el intercepto (α^) y la pendiente (coeficiente de regresión β^) estimados a partir de la muestra.

La recta que mejor se ajusta a los datos viene definida por la expresión:
short.velocity = 1.098 + 0.0220 x blood.glucose

La función extractora (resultado de ajustar un modelo está encapsulado en un objeto del cual se pueden obtener ciertas cantidades deseadas) más básica es *summary()*:

```
> summary(lm(short.velocity~blood.glucose))

Call:
lm(formula = short.velocity ~ blood.glucose)

Residuals:
     Min       1Q   Median       3Q      Max
-0.40141 -0.14760 -0.02202  0.03001  0.43490

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.09781    0.11748   9.345 6.26e-09 ***
blood.glucose 0.02196    0.01045   2.101   0.0479 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2167 on 21 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.1737,    Adjusted R-squared:  0.1343
F-statistic: 4.414 on 1 and 21 DF,  p-value: 0.0479
```

Figura 10: salida Código 1.

Residuals se refiere a la **distribución de los residuos** que puede usarse para echar un vistazo a la asunción de que los residuos se deben distribuir de forma normal (la media de los residuos es cero por definición, por lo que la mediana no debe estar lejos de cero, y el mínimo y el máximo deben ser aproximadamente iguales en valor absoluto).

Coefficients devuelve el **coeficiente de regresión** y el intercepto de nuevo, pero esta vez con los correspondientes **errores estándar, pruebas t y p valores**. Los símbolos a la derecha (***) son indicadores gráficos del nivel de significatividad estadística. Por ejemplo, la línea a continuación la tabla muestra la definición de estos indicadores; una estrella significa 0, 01 < p < 0, 05.
Residual standard error ofrece la **variación residual del modelo alrededor de la línea de regresión**.
Nótese que el modelo no se ha ajustado en todos los datos por la presencia de valores perdidos en alguna de las variables. Además, **la salida también ofrece el coeficiente de determinación R^2**, que en la **regresión lineal simple** es equivalente al **coeficiente de correlación de Pearson al cuadrado**. **Mide lo bien que el modelo lineal se ajusta a los datos (bondad de ajuste). Si se multiplica por 100% se puede interpretar como "% de la varianza"** en la variable respuesta original que se explica por las variables del modelo.
El otro coeficiente llamado "**R^2 ajustado**" es una **medida de bondad de ajuste que tiene en cuenta el número de variables que se han incluido en el modelo**, y suele ser de mayor utilidad en el contexto de **regresión lineal múltiple**.
Finalmente, *F-statistic* es el **contraste de la hipótesis H_0: β=0**. Este test en regresión lineal es redundante con la parte de la salida que ofrece el t-test para el coeficiente de regresión. De hecho, el estadístico F es idéntico a la raíz cuadrada del test t en cualquier modelo con sólo un coeficiente de regresión. **El test F tendrá más interés en el caso de regresión lineal múltiple, ya que reflejará la contribución global de todos los coeficientes de regresión**.

3.2.2. Asunciones del modelo de regresión linear

Para garantizar que las estimaciones del modelo son válidas, se deben cumplir las siguientes condiciones:

1. Linealidad

El valor esperado de la variable respuesta Y es una función lineal de la variable explicativa X, de tal forma que **cambios de magnitud constante a distintos niveles de X se asocian con un mismo cambio en el valor medio de Y** . Esto ya se podía ver en los diagramas de dispersión de la Figura 7 (el panel d es un ejemplo claro de una relación no lineal).

2. Homogeneidad de la varianza

La varianza de la variable respuesta Y es la misma para cualquier valor de la variable explicativa X y también en toda la distribución de valores predichos por el modelo. Encontramos un ejemplo en la Figura 11.

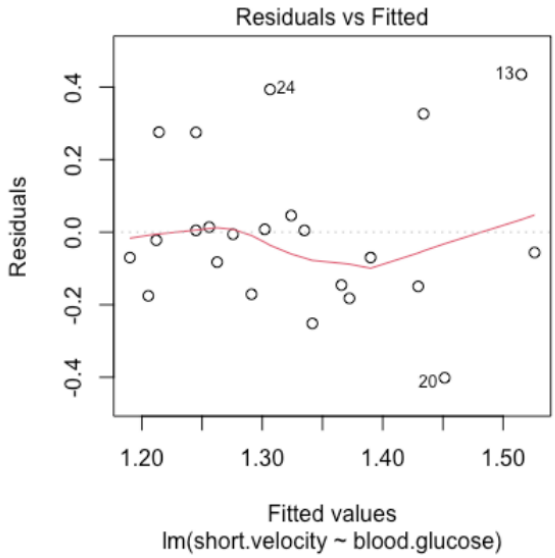

Figura 11: La nube de puntos debe estar centrada en la línea horizontal del cero, y debe ser homogénea a lo largo de todo el rango de valores predichos.

El Código 2 muestra cómo extraer representar y visualizar residuos en R, lo que ayudará a verificar las asunciones fundamentales del modelo de regresión.

2.1. Código 2

```
> lm(short.velocity~blood.glucose)

Call:
lm(formula = short.velocity ~ blood.glucose)

Coefficients:
  (Intercept)  blood.glucose
      1.09781        0.02196

> plot(blood.glucose,short.velocity)
> abline(lm(short.velocity~blood.glucose))
```

Figura 12: Salida del Código 2. Se utiliza la función lm() ("linear model") para realizar análisis de regresión linear. Obtenemos los coeficientes de alpha (intercept, es decir, la constante que define la recta) y beta (blood.glucose, la pendiente de la recta)

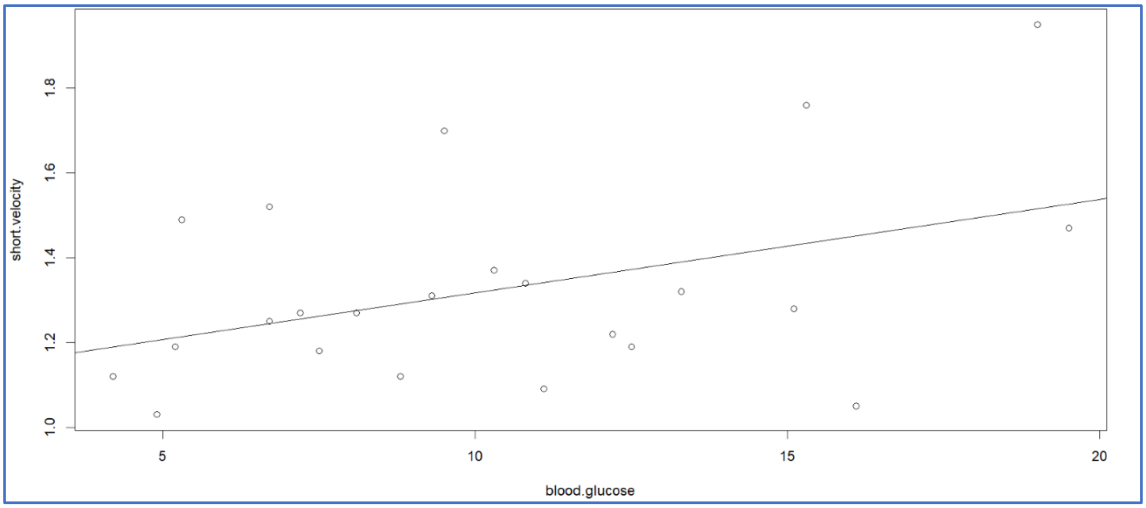

Figura 13: Salida del Código 2. Diagrama de dispersión, incluyendo la línea estimada por el modelo.

abline(), significa "(a, b)-line", dibuja líneas basándose en un intercepto (alpha) y una pendiente (beta), a y b, respectivamente.

Se puede utilizar introduciendo valores numéricos como por ejemplo abline(1.1,0.022), pero también es capaz de extraer automáticamente la información de un objeto creado con la función lm().

*La recta obtenida corresponde a la función 1.098+0.0220*blood.glucose*

```
> lm.velo <- lm(short.velocity~blood.glucose)
> fitted(lm.velo)
       1        2        3        4        5        6        7        8        9       10       11
1.433841 1.335010 1.275711 1.526084 1.255945 1.214216 1.302066 1.341599 1.262534 1.365758 1.244964
      12       13       14       15       17       18       19       20       21       22       23
1.212020 1.515103 1.429449 1.244964 1.190057 1.324029 1.372346 1.451411 1.389916 1.205431 1.291085
      24
1.306459
> resid(lm.velo)
           1            2            3            4            5            6            7
 0.326158532  0.004989882 -0.005711308 -0.056084062  0.014054962  0.275783754  0.007933665
           8            9           10           11           12           13           14
-0.251598875 -0.082533795 -0.145757649  0.005036223 -0.022019994  0.434897199 -0.149448964
          15           17           18           19           20           21           22
 0.275036223 -0.070057471  0.045971143 -0.182346406 -0.401411486 -0.069916424 -0.175431237
          23           24
-0.171085074  0.393541161
> confint(lm.velo)
                    2.5 %     97.5 %
(Intercept)   0.8534993816 1.34213037
blood.glucose 0.0002231077 0.04370194
> plot(blood.glucose,short.velocity)
> lines(blood.glucose,fitted(lm.velo))
Error in xy.coords(x, y) : 'x' and 'y' lengths differ
> lines(blood.glucose[!is.na(short.velocity)],fitted(lm.velo))
```

Figura 14: Salida del Código 2.

En lm.velo se guarda el r análisis de regresión linear obtenido al principio de la salida del Código 2.

La función fitted() devuelve valores ajustados: los valores de y que se esperaría para los valores de x dados de acuerdo con la línea recta que mejor se ajusta; en este caso : 1.098+0.0220*blood.glucose.
Los residuos que devuelve resid() es la diferencia entre el valor de y predicho por el modelo (el que se obtenía con fitted()) y el observado.
La función extractora confint() nos da los intervalos de confianza para los coeficientes.
Es necesario comentar algunos aspectos incómodos que surgen cuando hay valores faltantes en los datos. Por ejemplo, para suponer la línea ajustada en el gráfico de dispersión, se podría, aunque es más fácil de usar abline(lm.velo), también utilizar la funión lines(), sin embargo, se obtiene el siguiente error: "Error in xy.coords(x, y) : 'x' and 'y' lengths differ". Esto ocurre porque hay 24 observaciones pero sólo 23 valores predichos porque unos de los valore en short.velocity es un valor perdido ("NA").
En realidad, lo que necesitábamos era glucosa en sangre, pero sólo para aquellos pacientes cuyos se ha registrado short.velocity. Por tanto, con la última línea de código sí podemos obtener la recta que más se ajusta a nuestra ecuación (ver Figura 15).

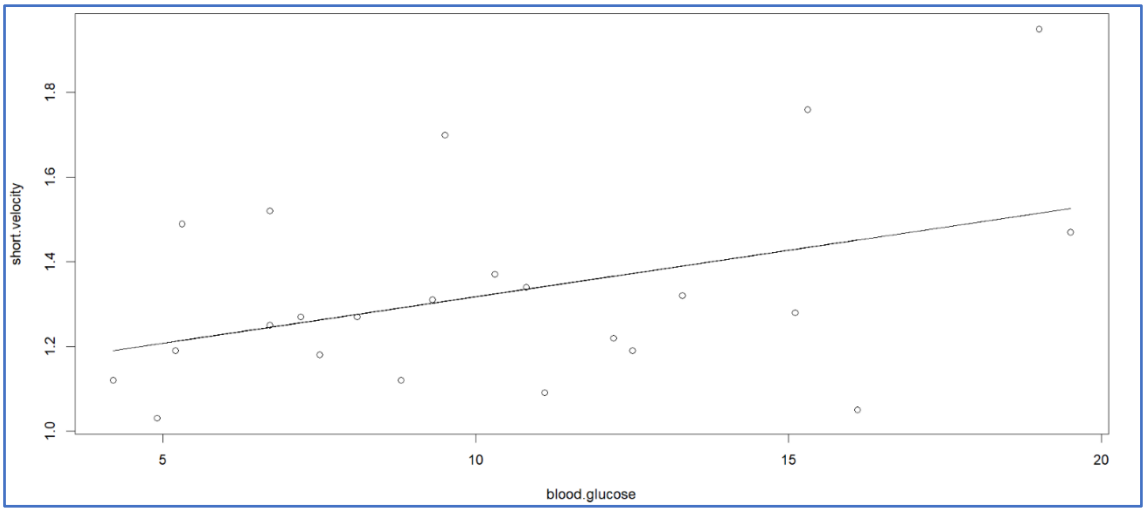

Figura 15: Salida Código 2, implementando la función is.na(), que produce un vector que es "TRUE" siempre que el argumento es "NA" (valor faltante).

Una ventaja de eliminar individuos con valores perdidos es que el ajuste de la recta no se extiende más allá del rango de datos.

```
> options(na.action=na.exclude)
> lm.velo <- lm(short.velocity~blood.glucose)
> plot(blood.glucose,short.velocity)
> lines(blood.glucose,fitted(lm.velo))
> segments(blood.glucose,fitted(lm.velo), blood.glucose,short.velocity)
```

Figura 16: Salida Código 2.

La opción na.exclude se puede utilizar para el manejo de NA. Esto se puede configurar como un argumento para lm() o como una opción.

segments() dibuja segmentos de línea; sus argumentos son las coordenadas del punto final en el orden (x1, y1, x2, y2). Al aplicarlo, se obtiene una gráfica simple de residuos versus valores ajustados.

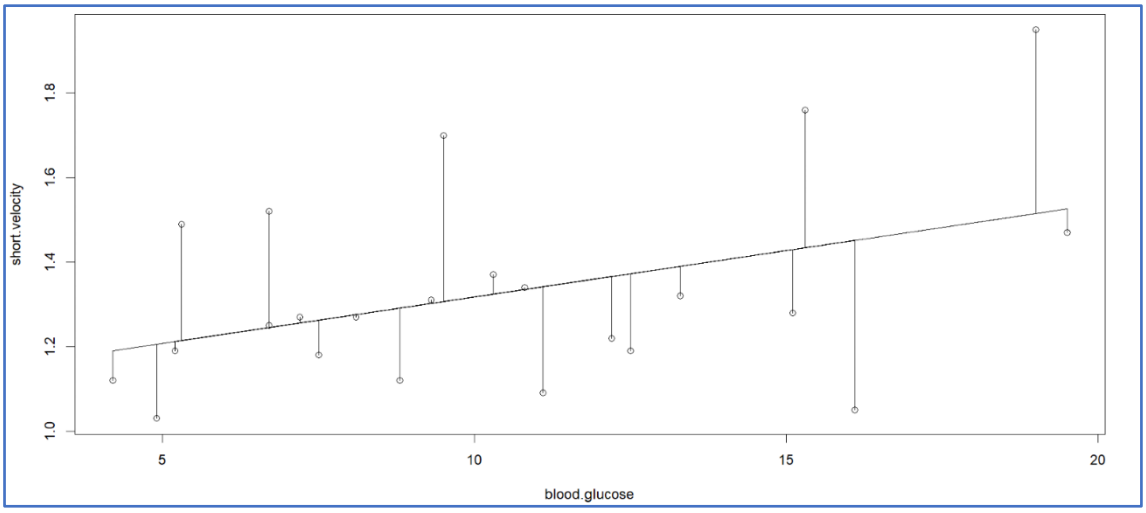

Figura 17: Gráfica en la que se muestren los residuos al conectar las observaciones a los puntos correspondientes en la línea ajustada.

```
> plot(fitted(lm.velo),resid(lm.velo))
> abline(h=0)
```

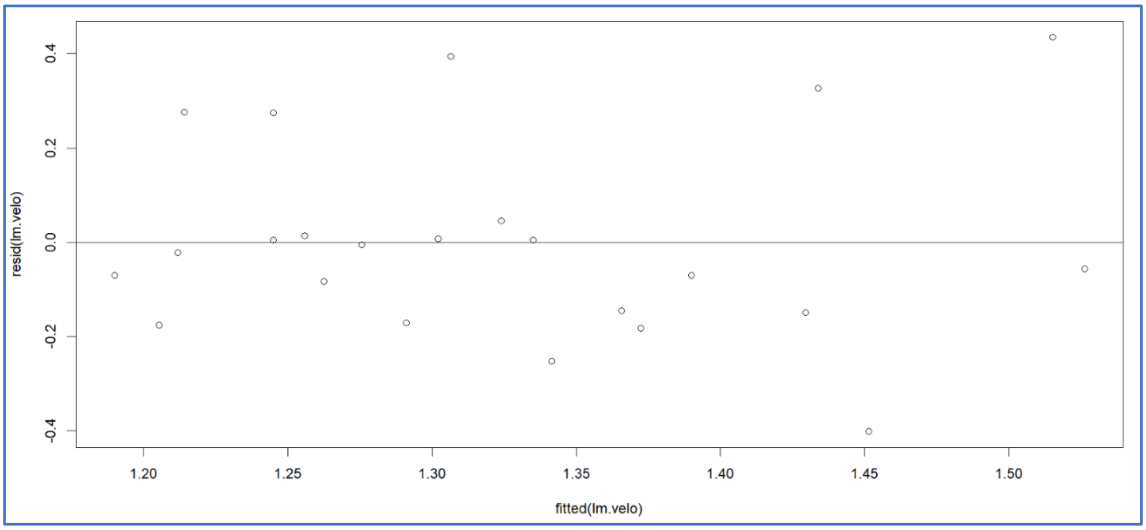

Figura 18: Gráfica simple de residuos versus valores.

```
> qqnorm(resid(lm.velo))
```

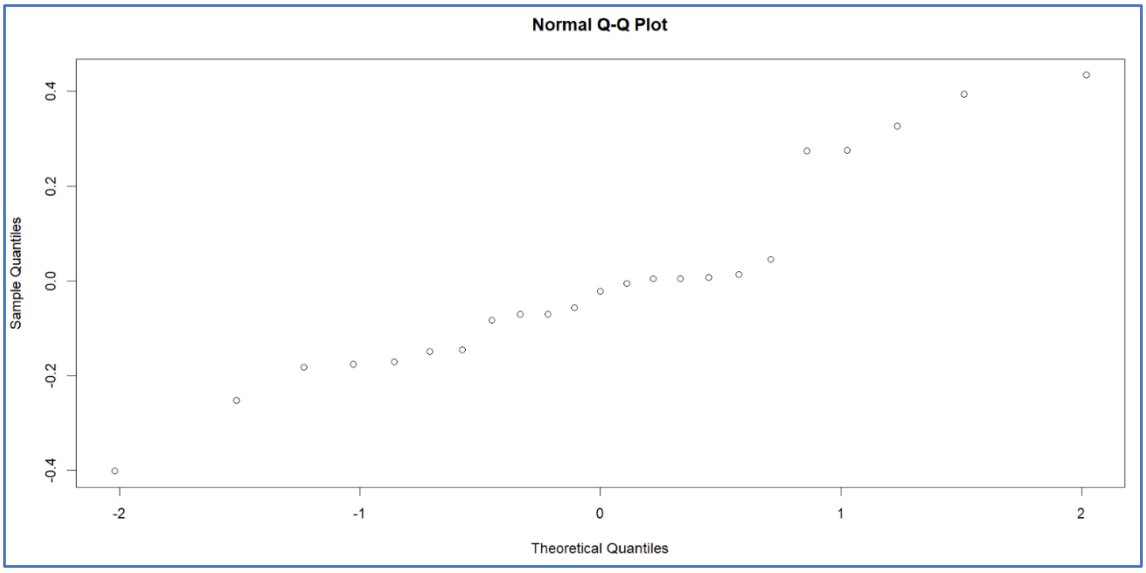

Figura 19: Gráfico Q-Q plot ("quantile-quantile plot"). Con él podemos obtener una indicación de si los residuos siguen una distribución normal comprobando una línea recta.

```
> par(mfrow=c(2,2))
> plot(lm.velo)
```

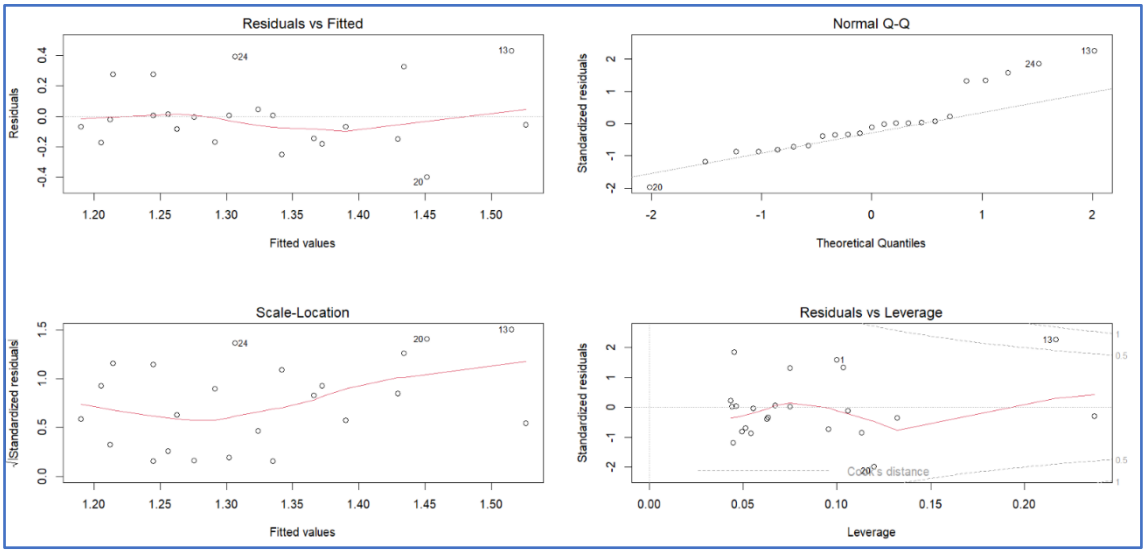

Figura 20: Si pasamos directamente el objeto lm.velo a la función plot() nos genera automáticamente plots para visualizar las asunciones del modelo

3. *Regresión lineal múltiple*

La función lm() maneja modelos mucho más complicados que los modelos lineales simples.
Un análisis de regresión lineal múltiple de, por ejemplo, y según las variables x1, x2 y x3, que se especifica como y ~ x1 + x2 + x3 (es decir, se incluyen **múltiple predictores en vez de sólo uno**).
La especificación de un análisis de regresión múltiple se realiza configurando una fórmula del modelo con "+" entre las variables explicativas.
Para el siguiente ejemplo volveremos a la base de datos juul que ya utilizamos en la Práctica 1. Contiene una muestra de la distribución del factor de crecimiento similar a la insulina (IGF-I), una observación por sujeto, en sujetos de varias edades, con la mayor parte de los datos recopilados en exámenes físicos escolares.

```
> attach(juul)
```

```
> summary(juul)
      age            menarche          sex             igf1           tanner          testvol      
 Min.   : 0.170   Min.   :1.000   Min.   :1.000   Min.   : 25.0   Min.   :1.00   Min.   : 1.000  
 1st Qu.: 9.053   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:202.2   1st Qu.:1.00   1st Qu.: 1.000  
 Median :12.560   Median :1.000   Median :2.000   Median :313.5   Median :2.00   Median : 3.000  
 Mean   :15.095   Mean   :1.476   Mean   :1.534   Mean   :340.2   Mean   :2.64   Mean   : 7.896  
 3rd Qu.:16.855   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:462.8   3rd Qu.:5.00   3rd Qu.:15.000  
 Max.   :83.000   Max.   :2.000   Max.   :2.000   Max.   :915.0   Max.   :5.00   Max.   :30.000  
 NA's   :5        NA's   :635     NA's   :5       NA's   :321     NA's   :240    NA's   :859     
```

```
> cc <- complete.cases(juul[,c(1,3,4,5)])
> attach(juul[cc,])
The following objects are masked from juul (pos = 3):

    age, igf1, menarche, sex, tanner, testvol

The following objects are masked from juul (pos = 4):

    age, igf1, menarche, sex, tanner, testvol

The following objects are masked from juul[cc, ] (pos = 5):
```

...

```
The following objects are masked from juul (pos = 13):

    age, igf1, menarche, sex, tanner, testvol

> lm(igf1~age+as.factor(sex)+as.factor(tanner))

Call:
lm(formula = igf1 ~ age + as.factor(sex) + as.factor(tanner))

Coefficients:
       (Intercept)                age     as.factor(sex)2  as.factor(tanner)2  as.factor(tanner)3  
           243.849              -5.233              16.327             166.239             299.270  
as.factor(tanner)4  as.factor(tanner)5  
           339.519             305.498  
```

```
> summary(lm(igf1~age+as.factor(sex)+as.factor(tanner)))

Call:
lm(formula = igf1 ~ age + as.factor(sex) + as.factor(tanner))

Residuals:
    Min      1Q  Median      3Q     Max
-306.21  -70.41  -10.95   60.14  423.63

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          243.849     10.788  22.605  < 2e-16 ***
age                   -5.233      0.990  -5.285 1.63e-07 ***
as.factor(sex)2       16.327      8.425   1.938    0.053 .
as.factor(tanner)2   166.239     15.892  10.461  < 2e-16 ***
as.factor(tanner)3   299.270     19.087  15.679  < 2e-16 ***
as.factor(tanner)4   339.519     17.839  19.032  < 2e-16 ***
as.factor(tanner)5   305.498     13.555  22.538  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 115.8 on 785 degrees of freedom
Multiple R-squared:  0.5547,    Adjusted R-squared:  0.5513
F-statistic:   163 on 6 and 785 DF,  p-value: < 2.2e-16
```

Figura 20: Se aplica la regresión lineal múltiple a la base juul.

Realmente no hay mucho nuevo aquí, ya que la especificación del modelo y la salida no difiere mucho de lo que se ha descrito para la regresión lineal simple. La diferencia es principalmente la manera de articular los modelos, es decir, **entre un conjunto de posibles variables descriptivas la importancia está en buscar un subconjunto que describe la respuesta suficientemente bien según el objetivo de los análisis que se quieren realizar**.

Merece la pena resaltar que **las pruebas t solamente reflejan lo que sucedería si elimina una variable teniendo en cuenta todas los demás**. No es posible saber qué p-valor tendría una variable concreta en un modelo reducido o con otras variables, si no se ajustan dichos modelos alternativos de novo.

Así mismo **la comparación entre el coeficiente de determinación sin ajustar R^2** (que refleja el cambio en S^2_{res} en comparación con un modelo sin variables de ajuste) **y la el R^2 ajustado** que refleja el cambio en la varianza residual, es decir, $1 - 115.8\ 2/\sigma^2_{igf1}$, donde 115.8 viene de **"residual standard error"** en la salida de *summary()* y $\boldsymbol{\sigma^2_{igf1}}$ se obtiene con *var(igf1)* (nos permite hacernos una **idea de cómo de parsimonioso es el modelo** (en casos donde hay pocos datos y muchas variables estos dos coeficientes serán diferentes, con una atenuación marcada del ajustado)).

4. EJERCICIOS PROPUESTOS

4.1. Ejercicio 1

La tabla siguiente muestra la edad (años) y la presión sanguínea (mmHg) de cada una de doce mujeres.

Edad	56	42	72	36	63	47
Presión	147	125	160	118	149	128
Edad	55	49	38	42	68	60
Presión	150	145	115	140	152	155

a) Utilice las fórmulas apropiadas para calcular la recta de regresión de Y sobre (en función de) X, el coeficiente de correlación lineal, la varianza residual, e interprete los resultados.

Primero, se meten los datos de las tablas en dos variables, edad y PAM (Presión Arterial Media). Y se representa utilizando la función plot, el diagrama de dispersión resultante (ver Figura 22).

```
> #La tabla siguiente muestra la edad (años) y la presión sanguínea (mmHg) de cada una de doce mujeres.
> edad = c(56, 42, 72, 36, 63, 47, 55, 49, 38, 42, 68, 60)
> PAM = c(147, 125, 160, 118, 149, 128, 150, 145, 115, 140, 152, 155)
> n=12
> plot(edad,PAM,main = "Diagrama de dispersión")
```

Figura 21: Entrada de código para el ejercicio propuesto 1.

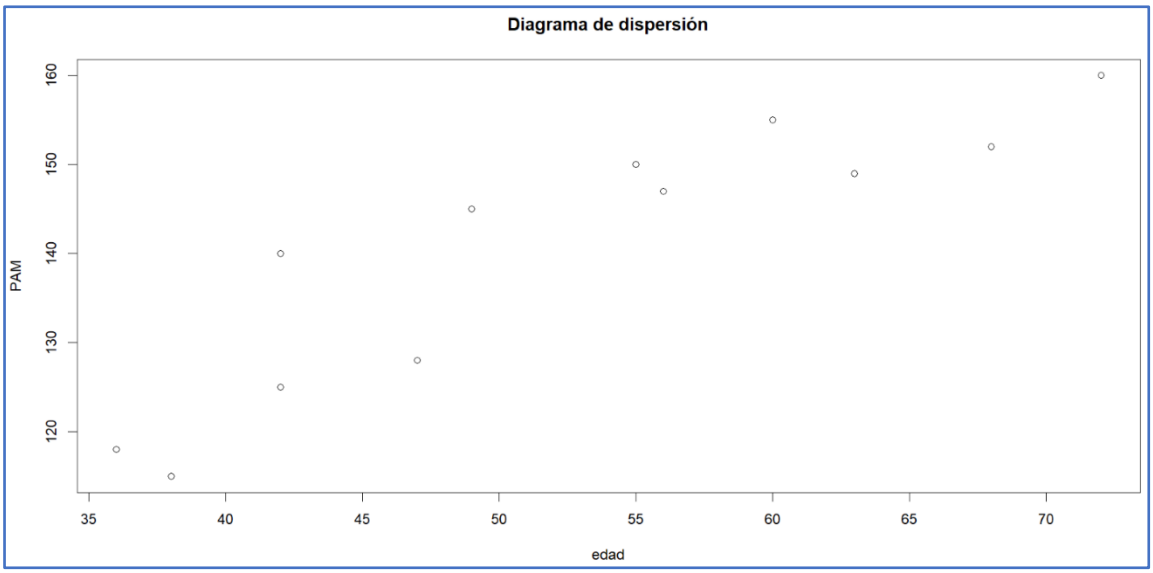

Figura 22: Diagrama de dispersión. Nube de puntos en las que se relaciona la PAM en función de la edad.

Ahora bien, el apartado a pide calcular la recta de regresión lineal de Y sobre X de forma manual (siguiendo las fórmulas). Para facilitar la comprensión de lo que se ejecuta, se le llama x a la edad e y a la PAM. Para facilitar el análisis de datos, se aplica la función data frame a x e y (estructura heterogénea de datos de dos dimensiones).

```
> #apartado a
> #Utilice las fórmulas apropiadas para calcular la recta de regresión de Y sobre (en función de) X,
> #el coeficiente de correlación lineal, la varianza residual, e interprete los resultados.
> x=edad
> y=PAM
> data = data.frame(x,y)
> data
    x   y
1  56 147
2  42 125
3  72 160
4  36 118
5  63 149
6  47 128
7  55 150
8  49 145
9  38 115
10 42 140
11 68 152
12 60 155
```

Figura 23: Ver data (variable donde se ha guardado el data.frame de x e y)

Ahora, se procede a calcular los datos necesarios (medias de x e y, Alpha y beta) para poder aplicar las fórmulas de regresión lineal:

$$\hat{\beta} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

Figura 24: Recordatorio: beta es la pendiente de la recta de regresión lineal, y apha es la constante.

```
> #y depende de x (y es la variable dependiente de x, que es la variable independiente)
> #en la regresión lineal simple se busca encontrar la ecuación de la recta que más se ajusta a la
> #nube de puntos: yi = α + βxi + εi
>
> #β=  Σ(xi - x¯)(yi - y¯)/Σ(xi - x¯)^2
> #α = y¯ - βx¯
>
> media.x = mean(x)
> media.x
[1] 52.33333
> media.y = mean(y)
> media.y
[1] 140.3333
> beta= sum((data$x - media.x)*(data$y - media.y))/sum((data$x - media.x)^2)
> beta
[1] 1.138005
> alpha = media.y - beta*media.x
> alpha
[1] 80.77773
> #recta de regresion
> yi = alpha + beta*data$x
> yi
 [1] 144.5060 128.5739 162.7141 121.7459 152.4721 134.2640 143.3680 136.5400 124.0219 128.5739 158.1621
[12] 149.0580
```

Figura 25: Se ha calculado manualmente las medias de x, y, beta, alpha y la recta de regresión lineal, acorde a la fórmula $y_i = \alpha + \beta x_i$ (recordar que la recta se aplica a cada valor de x_i)

Por otra parte, el enunciado pide calcular el coeficiente de regresión lineal:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Figura 26: Fórmula del Coeficiente de Correlación Lineal.

Y pide también calcular la varianza residual, cuya fórmula es:

$$S_{res}^2/(n-2)$$

Figura 27: Fórmula de la varianza residual (σ^2)

Siendo S^2_{res} la suma de cuadrados del error:

$$S_{res}^2 = \sum_{i=1}^{n}(y_i - \beta x_i - \alpha)^2$$

Figura 28: Fórmula de la S^2_{res}

```
> #el coeficiente de correlación lineal se calcula siguiendo la fórmula:
> #r =  Σ(xi - x¯)(yi - y¯)/√Σ(xi - x¯)^2√Σ(yi - y¯)^2
>
> #coeficiente de correlación lineal
> r= sum((data$x - media.x)*(data$y - media.y))/(sqrt(sum((data$x - media.x)^2))*sqrt(sum((data$y - media.
y)^2)))
> r
[1] 0.8961394
> #antes de calcular la varianza, debemos calcular la suma de cuadrados del error
> sres = sum((data$y - beta*data$x - alpha)^2)
> sres
[1] 492.4669
> #La varianza residual (σ^2) del modelo se estima como S^2res/(n-2)
> var = sres/(n-2)
> var
[1] 49.24669
```

Figura 29: Salida Código del ejercicio propuesto 1, apartado a). El coeficiente de regresión lineal calculado manualmente da 0.8961394, y la varianza residual, 49.24669.

Dado que se ha obtenido un coeficiente de correlación de 0.89, se puede concluir que la correlación es casi perfecta (es perfecta en r=|1|), y que se trata de una relación directa (ambas variables tienden a ser grandes o pequeñas simultáneamente).

La varianza residual, por el otro lado, mide la variabilidad de los valores de y con respecto a la recta de regresión.

Para asegurar haber aplicado adecuadamente la fórmula, se ha utilizado la función cor.test(), que calcula varios datos estadísticos, entre ellos, el coeficiente de correlación lineal, y se obtiene que r= 0.8961394 (exactamente lo mismo que obtuvimos de la primera manera).

```
> #hemos querido comprobar que el valor obtenido como coeficiente de correlación lienal es el correcto
> cor.test(edad, PAM)

        Pearson's product-moment correlation

data:  edad and PAM
t = 6.3858, df = 10, p-value = 7.976e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6634445 0.9707753
sample estimates:
      cor
0.8961394
```

Figura 30: Ver r= 0.8961394 (p-valor < 0.05).

b) Ahora calcule el coeficiente de regresión y el intercepto, y sus correspondientes intervalos de confianza utilizando lm() y confint(). Represente gráficamente el diagrama de dispersión y la recta de regresión estimada.

Este apartado sí permite utilizar las funciones ya implementadas en R para calcular el coeficiente de regresión de manera automática.
El coeficiente de regresión se obtuvo automáticamente en el apartado anterior, al aplicar cor.test. Como p-valor <<< 0.05, se puede objetar que existe correlación lineal, y ésta (r=0. 8961394) es fuerte y directa.

```
> #apartado b
> #Ahora calcule el coeficiente de regresión y el intercepto, y sus correspondientes intervalos de
> #confianza utilizando lm()y confint().
> #Represente gráficamente el diagrama de dispersión y la recta de regresión estimada.
> #se utiliza lm ajustar rectas de regresión.
> lm(data$y~data$x)

Call:
lm(formula = data$y ~ data$x)

Coefficients:
(Intercept)       data$x
     80.778        1.138
```

Figura 31: Se utiliza la función lm (variable dependiente~variable de la que depende) para obtener el intercepto (alpha = 80.778, casi idéntica a la Alpha obtenida manualmente (ver Figura 25), y la pendiente (beta =1.138, ver Figura 25).

```
> lineal.model = lm(data$y~data$x)
>
> #Los valores de y que se esperaría para los valores de x dados de acuerdo con la línea recta que
> #mejor se ajusta; en este caso : 80.77773+1.138005*edad.
> fitted(lineal.model) #La función fitted() devuelve valores ajustados
       1        2        3        4        5        6        7        8        9       10       11
144.5060 128.5739 162.7141 121.7459 152.4721 134.2640 143.3680 136.5400 124.0219 128.5739 158.1621
      12
149.0580
>
> #Los residuos que devuelve resid() es la diferencia entre el valor de y predicho por el modelo
> #(el que se obtenía con fitted()) y el observado.
> resid(lineal.model)
        1         2         3         4         5         6         7         8         9        10
 2.493981 -3.573947 -2.714101 -3.745916 -3.472055 -6.263972  6.631986  8.460017 -9.021926 11.426053
       11        12
-6.162081  5.941960
>
> #La función extractora confint() nos da los intervalos de confianza para los coeficientes.
> confint(lineal.model)
                2.5 %     97.5 %
(Intercept) 59.5129389 102.042521
data$x       0.7409311   1.535079
>
> plot(data$x,data$y, xplot="edad (años)", yplot="presión sanguínea (mmHg)", main="Diagrama de dispersión")
There were 12 warnings (use warnings() to see them)
> lines(x,fitted(lineal.model))
```

Figura 32: Observar los coeficientes de confianza obtenidos al utilizar la función conftint().

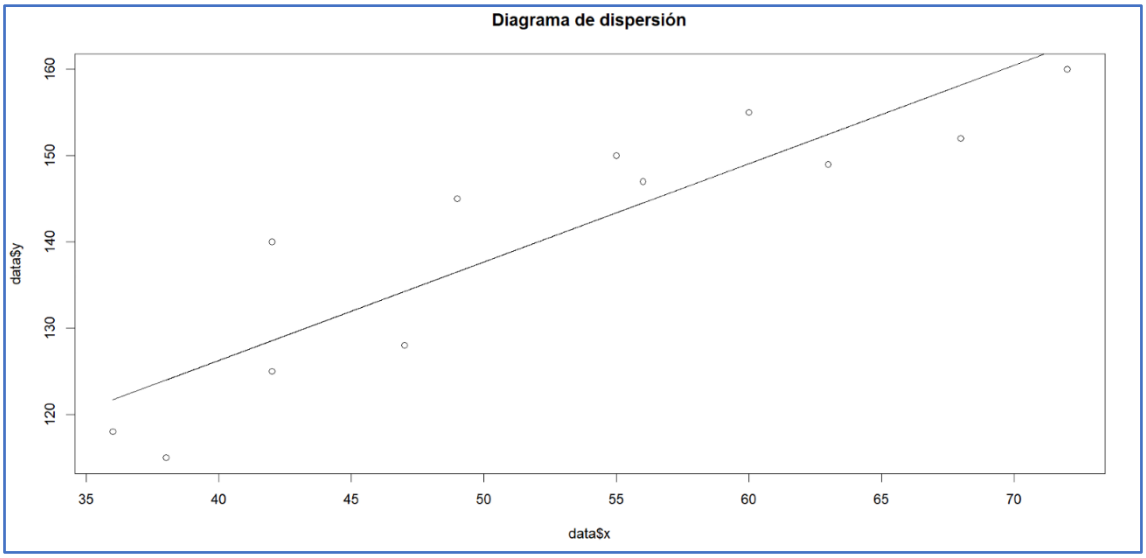

Figura 33: Representación gráfica del diagrama de dispersión y la recta de regresión estimada.

c) Utilice la siguiente línea de código para superponer en el diagrama de dispersión anterior una línea suavizada que se adapte a los puntos.

lines(lowess(presion~edad), col="blue", lwd=3)

¿Cual de las 2 líneas obtenidas le parece que se adapta mejor a la línea de puntos?

Tras observar la Figura 34, se puede concluir que la línea que mejor se adapta a la nube de puntos es la azul.

```
> #Apartado c
> #Utilice la siguiente línea de código para superponer en el diagrama de dispersión anterior una
> #línea suavizada que se adapte a los puntos.
> #lines(lowess(presion~edad), col="blue", lwd=3)
> #¿Cual de las 2 líneas obtenidas le parece que se adapta mejor a la línea de puntos?
>
> lines(lowess(PAM~edad), col="blue", lwd=3)
```

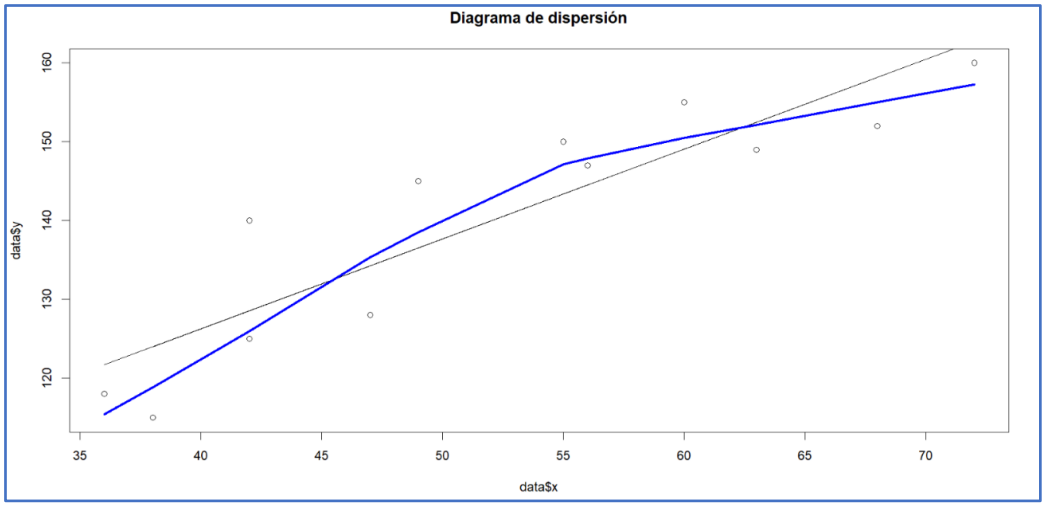

Figura 34: Representación gráfica del diagrama de dispersión, la recta de regresión estimada y la recta que mejor se adapta a la línea de puntos.

4.1.1. Código completo Ejercicio 1

```
#La tabla siguiente muestra la edad (años) y la presión sanguínea (mmHg) de cada una de doce mujeres.
edad = c(56, 42, 72, 36, 63, 47, 55, 49, 38, 42, 68, 60)
PAM = c(147, 125, 160, 118, 149, 128, 150, 145, 115, 140, 152, 155)
n=12

plot(edad,PAM,main = "Diagrama de dispersión")

#apartado a
#Utilice las fórmulas apropiadas para calcular la recta de regresión de Y sobre (en función de) X,
#el coeficiente de correlación lineal, la varianza residual, e interprete los resultados.
x=edad
y=PAM
data = data.frame(x,y)

#y depende de x (y es la variable dependiente de x, que es la variable independiente)
#en la regresión lineal simple se busca encontrar la ecuación de la recta que más se ajusta a la
#nube de puntos: yi = α + βxi + ϵi

#β= ∑(xi – x¯)(yi – y¯)/∑(xi – x¯)^2
#α = y¯ – βx¯
media.x = mean(x)
media.y = mean(y)
beta= sum((data$x - media.x)*(data$y - media.y))/sum((data$x -
media.x)^2)
alpha = media.y - beta*media.x

#recta de regresion
yi = alpha + beta*data$x

#el coeficiente de correlación lineal se calcula siguiendo la fórmula:
#r = ∑(xi – x¯)(yi – y¯)/√∑(xi – x¯)^2√∑(yi – y¯)^2

#coeficiente de correlación lineal
r= sum((data$x - media.x)*(data$y - media.y))/(sqrt(sum((data$x -
media.x)^2))*sqrt(sum((data$y - media.y)^2)))
```

```
#antes de calcular la varianza, debemos calcular la suma de cuadrados del error
sres = sum((data$y - beta*data$x - alpha)^2)
#La varianza residual (σ^2) del modelo se estima como S^2res/(n−2)
var = sres/(n-2)

#hemos querido comprobar que el valor obtenido como coeficiente de correlación lienal es el correcto
cor.test(edad, PAM)

#apartado b
#Ahora calcule el coeficiente de regresión y el intercepto, y sus correspondientes intervalos de
#confianza utilizando lm()y confint().
#Represente gráficamente el diagrama de dispersión y la recta de regresión estimada.

#se utiliza lm ajustar rectas de regresión.
lm(data$y~data$x)
lineal.model = lm(data$y~data$x)

#Los valores de y que se esperaría para los valores de x dados de acuerdo con la línea recta que
#mejor se ajusta; en este caso : 80.77773+1.138005*edad.
fitted(lineal.model) #La función fitted() devuelve valores ajustados
#Los residuos que devuelve resid() es la diferencia entre el valor de y predicho por el modelo
#(el que se obtenía con fitted()) y el observado.
resid(lineal.model)
#La función extractora confint() nos da los intervalos de confianza para los coeficientes.
confint(lineal.model)

plot(data$x,data$y, xplot="edad (años)", yplot="presión sanguínea
(mmHg)", main="Diagrama de dispersión")
lines(x,fitted(lineal.model))

#Apartado c
#Utilice la siguiente línea de código para superponer en el diagrama de dispersión anterior una
#línea suavizada que se adapte a los puntos.
#lines(lowess(presion~edad), col="blue", lwd=3)
#¿Cual de las 2 líneas obtenidas le parece que se adapta mejor a la línea de puntos?
 lines(lowess(PAM~edad), col="blue", lwd=3)
```

4.2. Ejercicio 2

Recordemos la población de estudio que vimos en la Practica 3, que estaba compuesta por participantes de la encuesta de salud de EE.UU que eran nunca fumadores y ex-fumadores . En esta ocasión queremos calcular la la asociación entre el índice de masa corporal -variable independiente, unidades m/kg2- y la filtración glomerular - variable dependiente que es una medida de la función renal (unidades ml/min/1.72m2), teniendo en cuenta la posible confusión introducida por la diabetes (0-No, 1-Sí)), que está relacionada con el índice de masa corporal y la enfermedad renal.

Primero leeremos los datos en formato de texto plano separado por comas, recordemos que para leer datos es indicar correctamente a R donde está el fichero ".csv". Si el fichero está en el directorio de trabajo no es necesario indicar el "path" o "ruta" del archivo. Se leerá así:

```
> # En este caso yo el fichero está en el directorio de trabajo.
> setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P4")
> getwd() # Se puede cambiar con setwd()
[1] "C:/Users/Laura/Desktop/BIOESTADÍSTICA/P4"
> dir() # Se debería ver el fichero que se quiere leer.
[1] "codigo1 P4.R"                          "Mortality_NHANES8894_NonSmokers-1.csv"
> data <- read.csv("Mortality_NHANES8894_NonSmokers-1.csv")
> names(data)
 [1] "X"            "seqn"          "race"          "riagendr"      "ridageyr"      "smoking"
 [7] "bmxbmi"       "hbp"           "highchol"      "diab"          "ckd"           "gfr.epi"
[13] "sedent"       "prev.cvd"      "prev.cancer"   "peryr.exm.8yr" "peryr.age.8yr" "mortstat.8yr"
[19] "cancer.8yr"   "heart.8yr"
> # El codebook con la explicación de lo que es cada variable se encuentra en
> # el fichero Readme.txt que se ha facilitado.
> # Ya hemos visto algunos motivos por los que hay que eliminar valores perdidos
> # cuando se trabaja con modelos de regresión.
> # Otro motivo es para que cuando se realizan modelos progresivos de ajuste,
> # es conveniente que éstos se lleven acabo consistentemente en el subset de los mismos individuos.
> dim(data) #dimensiones
[1] 6195   20
> data <- data[complete.cases(data),]
> dim(data) #dimensiones restando las celdas vacías
[1] 5929   20
> attach(data)
```

Figura 35: Acceso al fichero .csv

a) **Escriba el modelo que le permite calcular el cambio promedio esperado en la función renal por un incremento de una unidad en el índice de masa corporal para un valor fijo de diabetes (E[gfr.epi|bmxbmi, diab]).**
Puede utilizar el nombre de las variables que aparece reflejado en el Codebook.

El objetivo de la Regresión Lineal Múltiple es conocer el valor una variable a partir de otra variable más explicativa. En este caso, la ecuación lineal de la recta aumentará en $b \cdot x_k$, según tantas variables explicativas tengamos (k).

El modelo más adecuado para responder a este primer apartado es el que sigue la Figura 36. En él, cada uno de los coeficientes representa la influencia individual que tiene cada una de las X (imc, *bmxbmi* y diabetes, *diab*) sobre Y (Función Renal, *gfr.epi*).

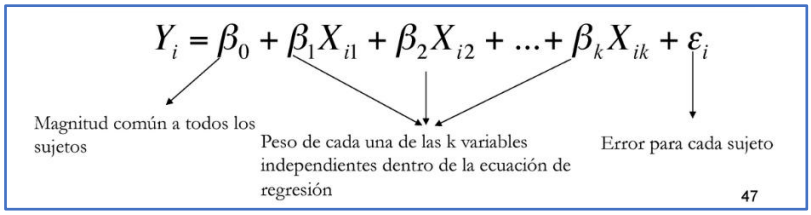

Figura 36: Modelo de RLM.

Por tanto, el modelo que permite calcular el cambio promedio esperado es:

$$gfr.epi = \beta 0 + \beta 1 \cdot bmxbmi + \beta 2 \cdot diab + \varepsilon$$

b) Ajuste el modelo de interés usando lm() y utilice la función extractora summary() para ver los coeficientes. Interprete los elementos clave de la salida.

```
> lm(gfr.epi ~ bmxbmi + as.factor(diab), data = data)

Call:
lm(formula = gfr.epi ~ bmxbmi + as.factor(diab), data = data)

Coefficients:
     (Intercept)            bmxbmi  as.factor(diab)1
         70.3107            0.5203            -5.2994

> rlm=lm(gfr.epi ~ bmxbmi + as.factor(diab), data = data)
> summary(rlm)

Call:
lm(formula = gfr.epi ~ bmxbmi + as.factor(diab), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-81.289 -13.289   1.702  14.596 116.353

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     70.31069    1.39032  50.572  < 2e-16 ***
bmxbmi           0.52033    0.04925  10.566  < 2e-16 ***
as.factor(diab)1 -5.29939    0.69467  -7.629 2.75e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.75 on 5926 degrees of freedom
Multiple R-squared:  0.02383,   Adjusted R-squared:  0.0235
F-statistic: 72.34 on 2 and 5926 DF,  p-value: < 2.2e-16
```

Figura 37: Salida del apartado b del Ejercicio 2.

La función lm() devuelve el intercepto (alpha = 70.3107), $\beta 1$ (0.5203) y $\beta 2$ (-5.2994).

summary() devuelve un resumen completo de los valores que engloban esta medición.

Un residuo es la diferencia entre el valor observado y el valor predicho por el modelo. En este caso, el rango queda entre -81.289 y 116.353.

El intercepto, o valor constante, alfa, es 70.3107, lo que significa que cuando la imc y la diabetes son 0, aún así la función renal es de 70.3107 (unidades ml/min/1.72m^2). La función renal crece en 0.5203 para cada aumento de unidad del IMC, y decrece 5.2994 por cada aumento de unidad de la diabetes.

Además, alpha y beta se pueden ver junto a sus correspondientes errores estándar, pruebas t y p valores (todos menores a 0.05, por lo que aceptamos la hipótesis alternativa: hay relación entre los parámetros). Los símbolos a la derecha son indicadores gráficos del nivel de significatividad estadística.

Además, la salida también ofrece el coeficiente de determinación R^2, que mide lo bien que el modelo lineal se ajusta a los datos (bondad de ajuste). En este caso, 0.02383 no es una medida excesivamente elevada, pero es aceptable. El otro coeficiente llamado "R2 ajustado" es una medida de bondad de ajuste que tiene en cuenta el número de variables que se han incluido en el modelo, por lo que en este tipo de modelos no es más útil (el).

Finalmente, F-statistic es el contraste de la hipótesis H0: β=0. Dado que se refleja la contribución global de todos los coeficientes de regresión, siendo el p-valor tan ínfimo, podemos concluir que el modelo se ajusta mucho a los datos de los que se dispone (el modelo es válido).

c) Realice el diagnóstico del modelo mediante la visualización apropiada de los residuos.

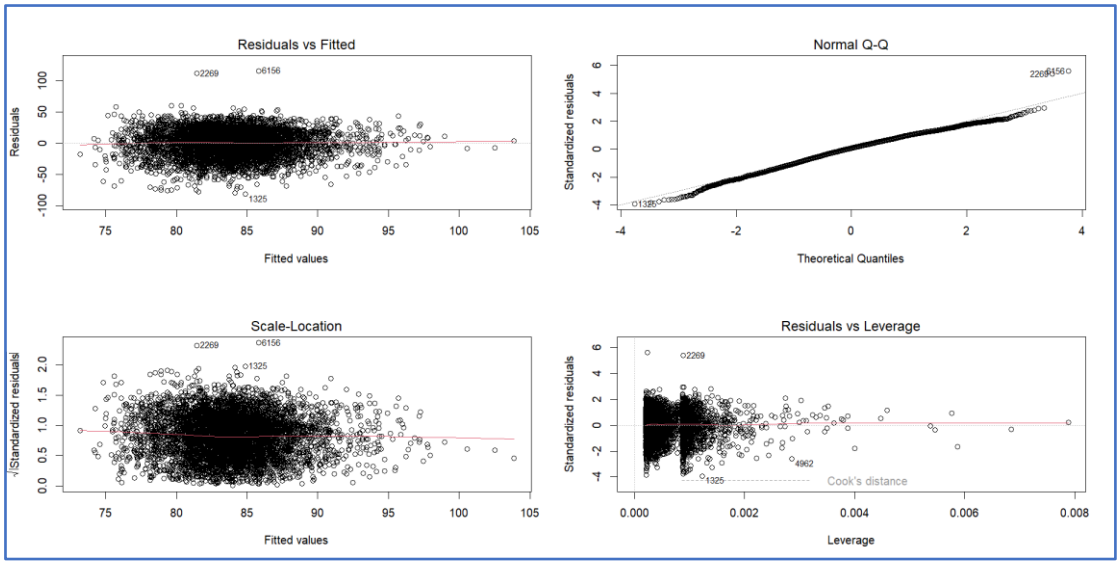

Figura 38: Salida de las diferentes maneras de visualizar los residuos.

En el bloque Residuals vs. Fitted, queda representada la diferencia entre los residuos obtenidos y los predichos por el modelo. Cuanto más amplia sea la nube (más se alejen los valores de la línea roja, 0), peor es la estimación dada por el modelo.

El bloque Normal Q-Q indica cuánto se acerca el modelo a una distribución normal, y puede observarse que se aproxima bastante (salvo en los valores extremos). La línea son los valores predichos por los cuantiles, y los puntos la distancia a la que están.

Scale-location es Residuals vs. Fitted normalizado. Residuals vs. Leverage es la medida que indica cuánto varía el modelo si se elimina alguno de los parámetros de la muestra. En este caso, parece que sí varía bastante.

4.2.1. Código completo Ejercicio 2

```
# En este caso yo el fichero está en el directorio de trabajo.
setwd("C:/Users/Laura/Desktop/BIOESTADÍSTICA/P4")
getwd() # Se puede cambiar con setwd()
dir() # Se debería ver el fichero que se quiere leer.
data <- read.csv("Mortality_NHANES8894_NonSmokers-1.csv")
names(data)
# El codebook con la explicación de lo que es cada variable se encuentra en
# el fichero Readme.txt que se ha facilitado.
# Ya hemos visto algunos motivos por los que hay que eliminar valores perdidos
# cuando se trabaja con modelos de regresión.
# Otro motivo es para que cuando se realizan modelos progresivos de ajuste,
# es conveniente que éstos se lleven acabo consistentemente en el subset de los mismos individuos.
dim(data) #dimensiones
data <- data[complete.cases(data),]
dim(data) #dimensiones restando las celdas vacías
attach(data)


#apartado a
#Escriba el modelo que le permite calcular el cambio promedio esperado en la función renal por un
#incremento de una unidad en el índice de masa corporal para un valor fijo de diabetes (E[gfr.epi|bmxbmi, diab]).
#Puede utilizar elnombre de las variables que aparece reflejado en el Codebook.
E[gfr.epi|bmxbmi, diab]~bmxbmi + as.factor(diab)


#apartado b
#Ajuste el modelo de interés usando lm() y utilice la función extractora summary() para ver los
# coeficientes. Interprete los elementos clave de la salida
lm(gfr.epi ~ bmxbmi + as.factor(diab), data = data)
rlm=lm(gfr.epi ~ bmxbmi + as.factor(diab), data = data)
summary(rlzm)


#apartado c
Realice el diagnóstico del modelo mediante la visualización apropiada de los residuos.
par(mfrow=c(2,2))
plot(rlm)
```

5. CONCLUSIONES

Tal y como era de esperar, RStudio agiliza los cálculos referidos a asociación entre variables categóricas, que hechos a mano pueden resultar engorrosos y puede haber errores de cálculo por aproximaciones (comparar apartados a y b del ejercicio 1).

En esta práctica se han repasado conceptos de la práctica pasada, pero era más importante entender la teoría, que entender cómo funcionaba R (siguiendo los ejercicios guiados, los ejercicios propuestos eran bastante parecidos, y sencillos).