ANÁLISIS EXPLORATORIO DE DATOS CON R Y MINITAB

Paloma Maín Yaque[1]
Departamento de Estadística e I.O.
Facultad de Ciencias Matemáticas
Universidad Complutense de Madrid

[1]**ISBN: 978-84-692-6255-9**

Índice general

PREFACIO

Este documento contiene el material recopilado durante varios cursos en los que la profesora Paloma Maín ha impartido la asignatura de Análisis Exploratorio de Datos de la Licenciatura en Ciencias Matemáticas, en la Facultad de Matemáticas de la Universidad Complutense de Madrid. En estos años la asignatura ha ido evolucionando como lo está haciendo la práctica del análisis de datos con la incorporación de nuevas técnicas que exigen el manejo de software estadístico para describir bases de datos de gran tamaño. En esta línea, lo que realmente se pretende es la introducción de una forma actual de entender esta primera etapa del análisis estadístico de datos. Efectivamente es un tema que ha tenido un enorme desarrollo y es difícil recoger todos los aspectos en una asignatura cuatrimestral de tres horas semanales o en cursos básicos, pero de alguna manera hemos pretendido que los alumnos se encontrasen cómodos a la hora de describir y efectuar un primer análisis de una base de datos relativamente compleja así como de afrontar el estudio y aplicación de las nuevas técnicas que se les puedan presentar.

Por tanto se incluirá un breve introducción a las dos herramientas utilizadas MINITAB y R, que en nuestra opinión pueden ser representativos del software estadístico en uso. Por un lado MINITAB como software comercial en un entorno Windows, con la utilización de "persianas" y R como software libre con la utilización de comandos estructurados en funciones. En la descripción de los temas se utilizarán ambos para resolver los problemas correspondientes incluyéndose las pantallas de ordenador con las soluciones.

El esquema general del contenido puede resumirse en dos bloques iniciales que estudian las columnas de las bases de datos por separado, mediante las técnicas de análisis univariante y las columnas en su conjunto mediante el análisis multivariante. Para estos dos análisis se utilizarán básicamente dos tipos de procedimientos: gráficos y cuantitativos. Quizá sea en los métodos gráficos en los que se ha experimentado el máximo avance motivado, en gran medida, por el gran aumento de la capacidad gráfica en los ordenadores personales. De todas formas no se puede olvidar la descripción y estudio de las medidas cuantitativas tan relevantes para los procesos posteriores de inferencia. En el apartado dedicado a la descripción de datos multivariantes se tratará también una breve introducción a los métodos de regresión para estudiar una forma de tratar la cuestión de la dependencia entre las columnas de una base de datos.

Dado que en la aplicación de las técnicas anteriores es necesario efectuar ciertas hipótesis de normalidad en las observaciones, se incluirán ciertos procedimientos que permiten evaluar si dichas hipótesis pueden considerarse aceptables sobre las observaciones o si se debe realizar algún tipo de transformación sobre ellas y así poder aplicar las técnicas descritas .

En este punto y una vez estudiadas bases de datos correspondientes a prob-

lemas reales se analizarán de forma particular ciertos tipos de datos que han sido descartados en un primer análisis. Estos serán los datos atípicos, por estar claramente alejados del bloque general de las observaciones y los datos ausentes al no aparecer ningún registro en las celdas correspondientes. Cabe señalar que no se pretenderá más que hacer una breve introducción, dada la gran amplitud que han alcanzado dichos temas componiendo sendas monografías.

Para concluir, queremos dar un agradecido recuerdo a todos nuestros alumnos de estos cursos pasados por su entusiasmo juvenil que ha sido el estímulo más importante para avanzar en nuestro trabajo. Con sus indicaciones se han enmendado errores y perfilado más claramente muchos conceptos y ejercicios. Espero que a los alumnos de los cursos futuros este trabajo les sirva para completar su formación y les resulte agradable afrontar la materia que se les propone.

Parte I

Herramientas computacionales

1. INTRODUCCIÓN A R

El R puede ser considerado una implementación del lenguaje S, desarrollado en Bell Laboratories y que forma la base del producto comercial $S-PLUS$. Los primeros pasos de R se deben a Robert Gentleman y Ross Ihaka del Statistics Department de la University de Auckland, aunque las sucesivas versiones son controladas y desarrolladas por el R Development Core Team del que forman parte muchos colaboradores en todo el mundo. Realmente el objetivo actualmente es la formación de equipos de colaboradores dedicados a la utilización de R para distintas aplicaciones como el grupo de redes bayesianas gR, de visualización de datos multivariantes ggobi. y muchos más que aportan los diferentes paquetes que pueden cargarse gratuitamente desde la página origen de todo este proyecto de software libre www.r-project.org.

Hay una lista de correo para usuarios en español r-help-es, donde consultar y comentar incidencias en el uso de R.

En resumen se podría considerar que los paquetes R proporcionan una forma de manejar conjuntos de funciones o datos y sus documentación.

1.1. Primeros pasos

- R es un lenguaje y un entorno para efectuar cálculos estadísticos y análisis gráficos, de distribución libre que puede cargarse en www.r-project.org. Al igual que la versión comercial S-Plus, está basado en el lenguaje S. Es completamente programable, siendo posible automatizar los procedimientos repetitivos mediante *scripts* del usuario, también es fácil escribir las $funciones$ que se necesiten así como nuevos *paquetes*.

1.1.1. Zona de trabajo ($Workspace$)

- El $Workspace$ en R contiene los datos y procedimientos de un determinado trabajo o proyecto, así que después de arrancar con el fichero $RGui.exe$, es conveniente seleccionar un nuevo directorio de trabajo en la persiana $File/Change\ Directory$, ya que por defecto se posiciona en el directorio en el que se ha instalado R, solapando los sucesivos trabajos. Al cerrar R se preguntará si se quiere *"Save workspace image"*, diciendo *yes* se crearán dos ficheros en el directorio de trabajo: $.Rhistory$ con los comandos y sentencias generadas en la sesión y $.RData$ con los objetos almacenados pero si se le quiere dar un nombre es

preferible guardarlo con ese nombre antes de cerrar en la persiana $File/Save\ Workspace$ y contestar *no* cuando aparece *"Save workspace image"*. La próxima vez con *doble-click* sobre ese fichero $.RData$ arrancará R en dicho espacio de trabajo cerrando con el procedimiento anterior. Se puede ver en que directorio se trabaja tecleando $getwd()$.

- Al arrancar R aparece la ventana de la $RConsole$ de trabajo donde se irán introduciendo los sucesivos comandos que se pueden guardar en un fichero $.Rhistory$. Si se obtienen gráficos aparecerán en otra ventana $RGraphics$ que pueden copiarse o guardarse en diferentes formatos así como imprimirse.

- R trabaja con *"objetos"*, la mayoría de los cuales son funciones. Tecleando $objects()$ (o $ls()$)se obtiene la lista de los que se tienen almacenados. Estos objetos se guardan en el $Workspace$ que es un fichero $.RData$, y que por defecto se guarda en el *directorio de trabajo* en el que está instalado R.

- En R, como en la mayoría de los paquetes basados en UNIX, se consideran nombres diferentes los escritos con mayúsculas o minúsculas, así A y a representarán distintos símbolos. En general se pueden utilizar todos los símbolos alfanuméricos, aunque un nombre nunca podrá comenzar por un número.

- Los comandos en R pueden ser: *expresiones* como $plot(x, y)$ que serán evaluadas impresas y su valor perdido, *o asignaciones* utilizando $<-$ o $=$ como $x < -27$ que también serán evaluadas y su valor transmitido a una variable, pero no automáticamente impreso. Los comandos estarán separados por ; o por una nueva línea, pudiendo agrupar comandos elementales entre { y }. Si un comando no está completo, el R devolverá el símbolo + hasta que se complete.

- Los comentarios en R se pueden poner casi en cualquier sitio, con tal de comenzar con # todo hasta el final de la línea es un comentario.

- Para cerrar la sesión de R se tiene que escribir $q()$ apareciendo la pregunta de si guardar o no "workspace". Si se contesta *yes* se guardará en $.RData$ y $R.History$ del directorio de trabajo.

1.1.2. Ayudas

R tiene una serie de documentos *html* (que arrancan con el navegador) para ayuda y manuales a los que se accede con $help.start()$. Si se conoce el nombre de la función, por ejemplo *nombre* se introduce $help\,(nombre)$ y en una ventana aparte aparecerá toda la información sobre su definición y características. Si no se conoce el nombre exacto de la función es preferible preguntar $help.search\,(\text{"}nombre\text{"})$ y aparecerán los ficheros de ayuda con algún elemento que sea *nombre*. Se puede utilizar también $?help$ o $?help.search$ para obtener más información de estas ayudas. Para no tener que teclear lo mismo varias veces, con las teclas de movimiento, $\uparrow$ o $\downarrow$ se obtienen las sentencias introducidas con anterioridad.

Otra forma de obtener ayuda sobre nuestra función *nombre* es mediante

ejemplos de ayuda tecleando $example(nombre)$. Como prueba escríbase $example\,(plot)$. En la mayoría de las funciones que se describirán, se deberá recurrir a $help(\ )$ para obtener información completa de las mismas.

1.1.3. Scripts

Son ficheros que contienen listas de comandos según se escriben en la consola, pudiendo incluir comentarios (líneas comenzando por #) que explican lo que se pretende hacer. Son muy útiles para repetir en sucesivas ocasiones, ciertos procedimientos que hemos tecleado en un determinado momento y para almacenar los comandos utilizados en las diferentes sesiones

Ejemplo: *Para crear y hacer funcionar un script que genere dos muestras de tamaño 50 de una distribución* $N\,(2,3)$ *calculando su coeficiente de correlación.*

1. Arrancar R

2. En la persiana $File$ seleccionar $New\ Script$, si se quiere utilizar un script almacenado seleccionar $Open\ Script$.

3. Escribir en el script las siguientes líneas

$x < -rnorm(50, 2, 3)$

$y < -rnorm(50, 2, 3)$

$plot(x, y)$

$print(cor.test\,(x, y))$

4. Seleccionar todas las líneas y presionar el botón derecho del ratón y aparecerá una ventana de diálogo en la que se selecciona "Run line or selection", así se ejecutarán todas las sentencias del *script*. Si no se selecciona nada, se ejecuta la sentencia al final de la cual está colocado el cursor. También se puede seleccionar un bloque de sentencias que son las que se ejecutarán.

5. Como salida aparecerá en una ventana $RGraphics$, una nube de puntos correspondiendo a las 50 observaciones pedidas y en en la ventana $RConsole$ el valor del coeficiente de correlación muestral así como el correspondiente p-valor del contraste y los extremos del intervalo de confianza a nivel 0,95.

6 Para guardar y cerrar el script, cuando se está en la ventana del mismo abrir la persiana $File$ y seleccionar $Save$ guardándolo con la extensión $.R$ y $Close\ script$ para cerrarlo

1.1.4. Paquetes R

Los paquetes proporcionan las facilidades para manejar conjuntos de funciones o datos y la documentación correspondiente. Son destacables los siguientes aspectos:

- Se cargan y descargan ocupando memoria sólo cuando son utilizados.
- Se instalan y actualizan fácilmente. Un único comando, ejecutable dentro o fuera de R, coloca en su sitio las funciones datos y documentación.
- Se adapta a los usuarios o administradores pudiendo tener además una o más librerías privadas de paquetes.
- Se pueden validar ya que R posee comandos para verificar que la documentación existe, para eliminar errores comunes y para comprobar que los ejemplos funcionan realmente.

Estructura de los paquetes R

La estructura básica suele contener

- un fichero **descriptivo** del paquete, autor y condiciones de la licencia en formato texto
- un fichero **índice** con las lista de funciones y datos, que puede generarse automáticamente
- un subdirectorio man/ de ficheros de documentación
- un subdirectorio R/ de ficheros de códigos R
- un subdirectorio data/ de conjuntos de datos
- un subdirectorio src/ de fuentes *C, Fortran o C++*

de forma menos frecuente también contiene

- tests/ para contrastes de validación
- exec/ para otros ejecutables, por ejemplo en Java
- inst/ para otro material
- un "script" de configuración para comprobar otro software que se necesite o para manejar diferencias entre sistemas.

Formatos de bases de datos

El comando $data()$ carga datos de los paquetes que se tienen cargados

- Ficheros de texto rectangulares con separador coma o espacio en blanco
- Código fuente S producido por la función $dump()$ de R o $S-Plus$
- Ficheros R binarios que produce la función $save()$

El tipo de fichero se elige automáticamente según la extensión.

Carga opcional de paquetes

- En el $Help$ menú se pueden encontrar los $Manuales$ donde se describen las funciones que se cargan con los paquetes básicos que se instalan al arrancar el R.

Hay otros paquetes suplementarios que no son más que grupos de funciones que se han escrito y hecho públicas por los creadores a través de la familia $CRAN$ de sitios en Internet (vía http://cran.r-project.org). En el menú *Packages* se pueden manejar estos otros paquetes, siendo algunos de los más utilizados el mva y $MASS$. También algunos se pueden cargar mediante $library(mva)$ y $library(MASS)$ y con $library()$ se pueden ver los paquetes disponibles para cargar directamente sin tener que acceder al sitio de R. Para saber cuáles se tienen cargados hay que teclear $search()$.

Dentro del menú *Packages* se tienen las opciones de instalación o actualización de nuevos paquetes ya sea directamente del sitio $CRAN$ o desde un fichero *.zip* que también se puede obtener en ese mismo sitio. Una vez instalado en nuestro directorio R podrá ser cargado con *Load Package* para trabajar con él en nuestra sesión.

Cabe señalar, aunque no sea de interés para la mayoría de los usuarios, que es posible acceder a los códigos fuente de los paquetes de R. No se distribuye de forma automática sino que hay que ir al sitio $CRAN$ y acceder a la opción R *Sources* para cargarla en nuestro sistema. A partir de ahí se puede visualizar el código de cualquier cálculo por ejemplo *cor.test* del paquete *statst* (para saber en qué paquete se encuentra hay que teclear ?*cor.test* y aparece en la primera línea de la ventana de $Help$), dentro del directorio R donde se guardó el código fuente, mediante el camino $src\backslash library\backslash stats\backslash R\backslash cor.test.R$.

1.1.5. Lenguaje R

Como se dijo al principio es un dialecto del lenguaje S, con una estructura común a muchos lenguajes tipo C, pero con unas características especiales que lo hacen especialmente versátil para el manejo de elementos estadísticos en concreto para operaciones con matrices y vectores. Fue diseñado en los 80 y desde entonces ha tenido un enorme desarrollo dentro de la comunidad estadística por sus posibilidades para la modelización estadística y la visualización mediante gráficos. Los comandos elementales en R consisten en **expresiones de cálculo y asignaciones** que pueden separarse por punto y coma ; o por línea nueva y agruparse en una expresión compuesta mediante llaves { }. Si un comando se escribe incompleto por error, R suele devolver + hasta que se completa.

Ejemplo: *Falta el paréntesis final*

$> (media.x < -mean(x)$

+

+

+)

[1]2,206341

Como se vio al utilizar los *scripts*, es posible utilizar comandos almacenados en un fichero externo mediante el procedimiento indicado en dicho apartado o mediante el comando *source*. Análogamente es posible almacenar

todas las salidas de R en un fichero externo, por ejemplo $result1.lis$ mediante $sink$("*ubicación del fichero* $result1.lis$"), que se podrá leer con cualquier procesador de textos. Para ver las salidas en la consola de nuevo se tecleará el comando $sink()$.

Cálculo

Si se plantea un cálculo en la línea de comandos, R efectuará el mismo, apareciendo el resultado en la consola.

Ejemplo: *Para calcular* $\exp(2) + 5$

$> exp(2) + 5$

obteniéndose

[1]12,38906

Los cálculos se efectuarán en la forma habitual, pudiendo utilizar los paréntesis para alterar el orden de los cálculos.

Ejemplo:

$> 7 - 5 * 4 + 3$

[1] -10

o bien

$> (7 - 5) * 4 + 3$

[1] 11

Los espacios se pueden utilizar libremente sin alterar el significado de las operaciones.

Asignación

Como se ha visto en el primer ejemplo, una forma de crear nuevos objetos es mediante el *operador de asignación* $<-$ es decir los símbolos *menor que* y *menos*, tecleados uno a continuación del otro.

Ejemplo: *Para crear el objeto* x *como una muestra de tamaño 50 de una distribución* $N(2, 3)$

$> x <- rnorm(50, 2, 3)$

Si a continuación se teclea

$> x$

se obtendrá la descripción explícita de las 50 observaciones generadas en ese momento

[1] 1.9695542 2.0773641 3.9620694 -0.5074673 6.0494055 3.5239660
[7] 4.0615103 -0.2137977 -3.0360154 7.9673968 0.5715117 -0.4865489
[13] -0.7256490 -1.5236052 2.9982977 4.6363986 1.3246436 -0.1932609
[19] 0.4410943 10.4654333 1.2003788 3.0297676 -3.1974437 1.4925627
[25] 4.3648033 3.7473657 1.5032604 6.9660583 1.9510645 7.4039741
[31] 4.3971533 0.9633926 6.8757369 3.7359458 -1.3833869 1.2585773
[37] 1.7376304 -2.5316654 6.1148239 3.0794651 4.5315920 0.7024664
[43] 1.5024827 -1.5534974 -0.7111865 3.2873565 0.4772716 0.9519583
[49] 6.1283580 -1.0715138

y siempre que se utilice x en cualquier expresión, se estará considerando toda la muestra, así por ejemplo

$> media.x < -mean(x)$

permite calcular la media aritmética de las 50 observaciones y guardarla como objeto $media.x$

$> media.x$

[1] 2,206341

Para conseguir que se cree el objeto y se visualice se debe escribir entre paréntesis

$> (media.x < -mean(x))$

[1] 2,206341

Para **borrar objetos** de la zona de trabajo se utilizará el comando $rm(.)$

Ejemplo: *Para borrar* $media.x$

$> rm(media.x)$

Si se teclea

$> objects(\ )$

se observará la desaparición de $media.x$.

Dado que en la denominación de algunos objetos es frecuente la utilización de valores como $x, y, \ldots$, es recomendable abrir directorios de trabajo distintos al efectuar diferentes tareas en R porque las sucesivas creaciones de estos objetos anularán los anteriores.

Tipos de objetos: **vectores** (numéricos, de caracteres, de índices), **matrices** o más generalmente *arrays*, **factores**, **listas** (vectores cuyos elementos no tienen por qué ser del mismo tipo), **data frames** (generalización de las matrices ya que puede haber columnas de diferente tipo, aunque en cada columna del mismo tipo), **funciones**. De un objeto se puede conocer, por ejemplo, su tipo $mode(\ )$, su longitud $length(\ )$ y su estructura $str(\ )$.

Ejemplo:

$> mode(x)$

[1]"$numeric$"

$> length(x)$

[1]50

$> mode(mean)$

[1]"$function$"

Mediante la función $attributes(\ )$ se pueden obtener las características de un objeto.

Ejemplo: *Los datos "Nile" son una serie temporal del flujo del río Nilo.*

$> Nile$

$TimeSeries$:

$Start = 1871$

$End = 1970$

$Frequency = 1$

[1] 1120 1160 963 1210 1160 1160 813 1230 1370 1140 995 935 1110 994 1020

[16] 960 1180 799 958 1140 1100 1210 1150 1250 1260 1220 1030 1100 774 840

[31] 874 694 940 833 701 916 692 1020 1050 969 831 726 456 824 702

[46] 1120 1100 832 764 821 768 845 864 862 698 845 744 796 1040 759

[61] 781 865 845 944 984 897 822 1010 771 676 649 846 812 742 801

[76] 1040 860 874 848 890 744 749 838 1050 918 986 797 923 975 815

[91] 1020 906 901 1170 912 746 919 718 714 740

$> attributes(Nile)$

$\$tsp$

[1]1871 1970 1

$\$class$

[1]"ts"

Se puede comparar con la función $str(\ )$

$> str(Nile)$

$Time-Series[1:100]from1871to1970:1120\ 1160\ 963\ 1210\ 1160\ 1160\ 813\ 1230\ 1370\ 1140...$

Con $attr(objeto, nombre)$ se puede seleccionar un atributo específico.

Ejemplo: *Para crear una matriz 2×2 de unos*

$> num <- c(1,1,1,1)$

$> num$

[1]1 1 1 1

$> attr(num, "dim") <- c(2,2)$

$> num$

$[,1][,2]$

$[1,]1\ 1$

$[2,]1\ 1$

1.2. Vectores

Uno de las facilidades de R es la manipulación de bases de datos. Dentro de estas estructuras el elemento más simple es la colección ordenada de números, el vector numérico, al que se dedicará este capítulo.

1.2.1. Vectores numéricos

Para crear vectores se suele utilizar la función $c(.)$, introduciendo los valores numéricos que definen el vector.

Ejemplo: *Para crear el vector* $v = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$

$> v < -c(1, 2, 3, 4, 5)$
$> v$
[1] 1 2 3 4 5

También se puede utilizar la función $assign(.)$.
Ejemplo:
$> assign(\text{“}v\text{”}, c(1, 2, 3, 4, 5))$
$> v$
[1] 1 2 3 4 5.

Una vez que el vector v existe como objeto, puede utilizarse para construir otros vectores y para efectuar operaciones sobre todos sus elementos.
Ejemplo:
$> exp(v)$
[1]2,718282 7,389056 20,085537 54,598150 148,413159
$> w < -c(2 * v, 0, 0, v)$
$> w$
[1] 2 4 6 8 1 0 0 0 1 2 3 4 5

1.2.2. Operaciones aritméticas

Como se ha visto anteriormente, las operaciones con vectores se efectúan elemento a elemento. Si alguno de los vectores es más corto, se alarga hasta conseguir la longitud del vector más largo.
Ejemplo:
$> d < -c(1, 2, 3)$
$> d$
[1] 1 2 3
$> e < -c(d, 0, 0, 0, 2 * d)$
$> e$
[1] 1 2 3 0 0 0 2 4 6
$> w < -d + e$
$> w$
[1] 2 4 6 1 2 3 3 6 9

Si en lugar del vector anterior, e, introducimos el siguiente, de longitud menor,
$> e < -c(d, 0, 0, 2 * d)$
$> e$
[1]1 2 3 0 0 2 4 6

aparecerá el siguiente mensaje
$> w < -d + e$
$Warning\ message:$
$longer\ object\ length$
$is\ not\ a\ multiple\ of\ shorter\ object\ length\ in: d + e.$

Las operaciones básicas se realizarán con los símbolos habituales: $+, -, *, /$, y $\hat{}$ para elevar a una potencia. También se pueden utilizar las funciones: $log, exp, sin, cos, tan, sqrt, max, min$, así como $diff(range()), length, sum, prod$ que calculan la diferencia entre el valor max y min, el número de elementos, su suma y su producto, respectivamente.

Ejemplo:
$> w$
[1] 2 4 6 1 2 3 3 6 9
$> range(w)$
[1]19
$> diff(range(w))$
[1]8
$> sum(w)$
[1]36
$> prod(w)$
[1] 46656
$> w\hat{}2$
[1] 4 16 36 1 4 9 9 36 81

También se pueden efectuar las operaciones estadísticas más frecuentes: $mean$ y var, siendo

$$var\,(x) = sum((x - mean\,(x))\hat{}2)/\,(length\,(x) - 1)$$

es decir la varianza muestral si x es una muestra unidimensional. Para el caso en el que x es una matriz de datos $n \times p$, el objeto $var\,(x)$ pasará a ser la matriz de covarianzas muestral de dimensión $p \times p$.

Otra operación interesante es la ordenación de un vector mediante la función $sort(\ )$, si se quiere efectuar una permutación se utilizará $order(\ )$ o $sort.list(\ )$.

Hay ocasiones en las que se necesita construir vectores con una determinada sucesión de valores numéricos, por ejemplo de 1 a 10; ésto puede conseguirse mediante la operación $1 : 10$ asignando el resultado a un objeto, pero es más versátil utilizar la función $seq(\ )$ porque admite varias posibilidades. Por otra parte está la función $rep(\ )$ para repetir un determinado objeto de diferentes formas.

Ejemplo:
$> (f < -seq(1, 10))$
[1] 1 2 3 4 5 6 7 8 9 10
$> (f < -seq(2, length = 4))$
[1] 2 3 4 5

$> (f < -seq(2, length = 4, by = 0{,}5))$
[1] 2,0 2,5 3,0 3,5
$> rep(1:3, 2)$
[1] 1 2 3 1 2 3
$> rep(1:3, 2, each = 2)$
[1] 1 1 2 2 3 3 1 1 2 2 3 3

1.2.3. Vectores lógicos

Son vectores generados por condiciones que pueden darse o no, por lo que sus elementos posibles son $TRUE$(cierto), $FALSE$(falso) y NA(no disponible). Es preferible no simplificar estos valores a T, F porque si se utilizan como objetos pueden ser alterados.

Ejemplo:
$> f$
[1] 2,0 2,5 3,0 3,5
$> logi1 < -f >= 3$
$> (logi1 < -f >= 3)$
[1] $FALSE\ FALSE\ TRUE\ TRUE$

Los operadores utilizados son los habituales,$<, >, <=, >=, ==$(igualdad exacta), $!\ =$(desigualdad), &(intersección de expresiones), |(AltGr-1, para la unión) y !(negación).

1.2.4. Valores ausentes ("missing values")

Hay situaciones en las que un valor está no disponible porque es un resultado imposible o se ha perdido, son los valores ausentes de estadística y en estos casos aparece como valor NA. Cualquier operación con ellos seguirá siendo NA, no disponible, sin embargo con la función $is.na(\ )$ se puede construir un nuevo vector donde aparece $TRUE$ si y sólo si el elemento correspondiente es NA.

Si el resultado imposible surge de un cálculo, el símbolo que aparece es NaN(not a number), siendo otra categoría de valores ausentes que se cambia también con $is.na(\ )$, pero si sólo se quieren cambiar los de esta categoría se debe teclear $is.nan(\ )$.

Ejemplo:
$> l$
[1] NA 0,7071068 0,0000000 NaN 1,0000000 0,7071068 0,0000000
$> is.na(l)$
[1] $TRUE\ FALSE\ FALSE\ TRUE\ FALSE\ FALSE\ FALSE$
$> is.nan(l)$
[1] $FALSE\ FALSE\ FALSE\ TRUE\ FALSE\ FALSE\ FALSE$

1.3. Matrices y arrays

Se puede definir un "array" como una colección de observaciones con subíndices, siendo las matrices casos particulares, en concreto "arrays" bidimensionales ya que sus elementos constan de dos subíndices. R es especialmente adecuado para crear y manejar este tipo de objetos.

Tienen como atributo la dimensión, que puede asignarse o describirse mediante $dim(\ )$. Así un vector numérico puede utilizarse como un "array" si se le asigna una dimensión.

Ejemplo:

$> w$
[1] 2 4 6 8 10 0 0 1 2 3 4 5
$> dim(w) < -c(2,6)$
$> w$

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	2	6	10	0	2	4
[2,]	4	8	0	1	3	5

$> dim(w) < -c(6,2)$
$> w$

	[,1]	[,2]
[1,]	2	0
[2,]	4	1
[3,]	6	2
[4,]	8	3
[5,]	10	4
[6,]	0	5

$> dim(w) < -c(2,3,2)$
$> w$
, , 1

	[,1]	[,2]	[,3]
[1,]	2	6	10
[2,]	4	8	0

, , 2

	[,1]	[,2]	[,3]
[1,]	0	2	4
[2,]	1	3	5

Como se puede observar en el ejemplo anterior, la forma de colocar los datos es rellenando por columnas los sucesivos bloques. A los elementos colocados en cada posición se les puede denominar con el nombre del "array" seguido de su posición entre [ ] separando los subíndices que describen dicha posición entre comas.

Ejemplo: *Si se considera la última descripción del objeto w en el ejemplo anterior, para obtener la segunda fila de la segunda columna del primer bloque*

$> w[2,2,1]$

[1] 8

También se pueden extraer columnas, filas y en general cualquier subconjunto describiendo sus posiciones entre [ ].

Ejemplo: *Con el w anterior se puede obtener la segunda columna del primer bloque*

$> w[,2,1]$

[1] 6 8

y las segundas columnas de los dos bloques

$> w[,2,]$

```
     [,1] [,2]
[1,]    6    2
[2,]    8    3
```

Otra forma de construir "arrays" a partir de vectores, es mediante la función $array(\ )$, especificando el nombre del vector y la dimensión del nuevo objeto. Para construir matrices se utilizará la función $matrix(\ )$.

Ejemplo: *Teniendo en cuenta que x es* $x < -rnorm(50,2,3)$ *del primer ejercicio, se puede convertir en un array formado por dos bloques de cinco filas y cinco columnas*

$> (arx < -array(x, c(5,5,2)))$

```
, , 1
          [,1]       [,2]      [,3]       [,4]       [,5]
[1,] -3.2106300  0.3877666 7.5592096 -0.3403993  1.2039125
[2,] -0.7870847  8.7728136 1.0124324  2.7640037 -0.7761628
[3,]  0.4241971  0.4688422 0.2401416  3.4419892 -1.9599388
[4,]  0.2159416  1.0030262 -1.3389534 3.0205676 -4.2068276
[5,]  4.1927219 -6.1559740 4.4434380  1.6881193 -1.1155851
, , 2
          [,1]      [,2]      [,3]        [,4]     [,5]
[1,] -5.247075  2.850224 3.6138058 -0.04179251 1.423822
[2,]  1.189775 -2.359040 6.5736714  0.67692024 1.260775
[3,]  4.042855 -1.425400 -4.1827371 7.05517180 3.344217
[4,]  3.106876  6.196905 -1.2356566 -2.54117124 1.363164
[5,]  4.199660 -1.659834 0.2650083  0.97560422 4.381154
```

Si el vector no tiene elementos suficientes para rellenar los huecos del *array*, con la función anterior se completa con los primeros valores del vector. También se pueden efectuar distintas operaciones con los "arrays" como permutar, el producto exterior...que se describirán a continuación en el caso particular y claramente más frecuente de las matrices.

1.3.1. Operaciones con matrices

Como se dijo anteriormente, una matriz es un *array* con dos subíndices. Entre las funciones que describen y manejan una matriz están $nrow(\ )$ y $ncol()$ para obtener el número de filas y columnas respectivamente, mientras que $t(\ )$ es la función que traspone la matriz. Otra función interesante es $diag(\ )$ que si se aplica a una matriz devuelve el vector formado con los elementos de la diagonal principal, si se aplica a un vector lo que se obtiene es una matriz diagonal con los valores del vector y por último sobre un escalar construye la matriz identidad de la dimensión dada por el escalar.

Ejemplo: *Sea la matriz* $mat1$ *la obtenida con el primer bloque del array* arx *obtenido en el ejemplo anterior.*

```
> (mat1 < −arx[, , 1])
        [,1]       [,2]      [,3]       [,4]       [,5]
[1,] -3.2106300 0.3877666 7.5592096 -0.3403993 1.2039125
[2,] -0.7870847 8.7728136 1.0124324 2.7640037 -0.7761628
[3,] 0.4241971 0.4688422 0.2401416 3.4419892 -1.9599388
[4,] 0.2159416 1.0030262 -1.3389534 3.0205676 -4.2068276
[5,] 4.1927219 -6.1559740 4.4434380 1.6881193 -1.1155851
> ncol(mat1)
[1] 5
> nrow(mat1)
[1] 5
> (tmat1 < −t(mat1))
        [,1]       [,2]      [,3]       [,4]       [,5]
[1,] -3.2106300 -0.7870847 0.4241971 0.2159416 4.192722
[2,] 0.3877666 8.7728136 0.4688422 1.0030262 -6.155974
[3,] 7.5592096 1.0124324 0.2401416 -1.3389534 4.443438
[4,] -0.3403993 2.7640037 3.4419892 3.0205676 1.688119
[5,] 1.2039125 -0.7761628 -1.9599388 -4.2068276 -1.115585
> (dmat1 < −diag(mat1))
[1] -3.2106300 8.7728136 0.2401416 3.0205676 -1.1155851
> diag(dmat1)
        [,1]       [,2]      [,3]       [,4]       [,5]
[1,] -3.21063 0.000000 0.0000000 0.000000 0.000000
[2,] 0.00000 8.772814 0.0000000 0.000000 0.000000
[3,] 0.00000 0.000000 0.2401416 0.000000 0.000000
[4,] 0.00000 0.000000 0.0000000 3.020568 0.000000
[5,] 0.00000 0.000000 0.0000000 0.000000 -1.115585
> diag(5)
   [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    1    0    0
```

[4,] 0 0 0 1 0
[5,] 0 0 0 0 1

Producto de matrices

El operador para aplicar esta operación entre matrices es $\%*\%$, utilizándose $*$ para el producto elemento por elemento de matrices del mismo tamaño. No obstante la función $crossprod(mat1, mat2)$ es equivalente a $mat1\,\%*\%\,\%mat2$ pero el proceso es más rápido.

Para obtener la inversa de una matriz cuadrada no singular se puede utilizar la solución de una ecuación lineal, por ejemplo de $q = Q\,\%*\%\,\%x$, ya que $x = Q^{-1}\,\%*\%\,\%q$, mediante la función $solve(Q, q)$, pudiendo calcular dicha matriz inversa mediante $solve(Q)$. Para obtener el determinante de Q se utiliza $det(Q)$.

Ejemplo: *Se calcula la inversa de la matriz* $mat1$ *definida en ejemplos anteriores con el vector* $x1$ *de unos*

$> x1 < -rep(1, 5)$
$> b1 < -mat1\,\%*\%\,\%x1$
$> solve(mat1, b1)$
$[, 1]$
$[1,]1$
$[2,]1$
$[3,]1$
$[4,]1$
$[5,]1$
$> solve(mat1)$

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.12877872	0.15323841	-0.2164438	-0.01502533	0.19133400
[2,]	-0.02089566	0.13530681	-0.1513859	0.03348884	0.02299115
[3,]	0.08597911	0.05810572	-0.1229635	0.04817971	0.08670692
[4,]	-0.01601109	-0.04914441	0.5782148	-0.24706520	-0.06726225
[5,]	-0.05045421	-0.01365353	0.4070992	-0.42322674	-0.06058936

Autovalores y autovectores

Aplicando la función $eigen(\ )$ a una matriz simétrica, se obtienen los correspondientes vector de autovalores y matriz de autovectores.

Ejemplo: *Se define la matriz* $mat2 = \begin{pmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{pmatrix}$ *y se determinan sus autovalores y autovectores*

$> d1 < -c(10, 0, 2, 0, 10, 4, 2, 4, 2)$
$> (mat2 < -matrix(d1, ncol = 3, nrow = 3))$
[,1] [,2] [,3]
[1,] 10 0 2

[2,] 0 10 4
[3,] 2 4 2
$> eigen(mat2)$
$\$values$
[1] 1.200000e+01 1.000000e+01 1.776357e-15
$\$vectors$
[,1] [,2] [,3]
[1,] -0.4082483 8.944272e-01 0.1825742
[2,] -0.8164966 -4.472136e-01 0.3651484
[3,] -0.4082483 5.551115e-17 -0.9128709

Si se utiliza $eigen(\ )\$values$ o bien $eigen(\ )\$vectors$ solamente aparecerán los autovalores o bien los autovectores que también pueden almacenarse como objetos.

Otra función aplicable a matrices muy útil es $svd(Q)$ ("singular value descomposition"), mediante la que se obtienen las matrices U, D y V tales que $Q = U\ \%*\%\ D\ \%*\%\ t(V)$, donde las columnas de U y de V son ortogonales y D es una matriz diagonal.

Ejemplo:
$> d3 < -c(1, 2, 5, 3, 7, 9, 2, 7, 1)$
$> mat3 < -matrix(d3, c(3, 3))$
$> mat3$
[,1] [,2] [,3]
[1,] 1 3 2
[2,] 2 7 7
[3,] 5 9 1
$> svd(mat3)$
$\$d$
[1] 14.0785096 4.9778584 0.1284231
$\$u$
[,1] [,2] [,3]
[1,] -0.2628271 -0.1087879 -0.9586903
[2,] -0.6746563 -0.6896073 0.2632121
[3,] -0.6897541 0.7159658 0.1078531
$\$v$
[,1] [,2] [,3]
[1,] -0.3594777 0.4202262 0.8331781
[2,] -0.8323937 0.2591632 -0.4898522
[3,] -0.4217778 -0.8696231 0.2566303

Construcción de matrices particionadas

Las funciones $cbind(\ )$ y $rbind(\ )$ permiten obtener una matriz a base de agregar matrices por columnas y filas respectivamente.

Ejemplo.

$> cbind(mat3, mat3)$
[,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 3 2 1 3 2
[2,] 2 7 7 2 7 7
[3,] 5 9 1 5 9 1
$> rbind(mat3, mat3)$
[,1] [,2] [,3]
[1,] 1 3 2
[2,] 2 7 7
[3,] 5 9 1
[4,] 1 3 2
[5,] 2 7 7
[6,] 5 9 1

Hay que señalar la diferencia con la función $c(\ )$ que encadena objetos eliminando su dimensión.

Ejemplo:

$> c(mat3, mat3)$
[1] 1 2 5 3 7 9 2 7 1 1 2 5 3 7 9 2 7 1

Tablas de contingencia

A partir de la función $table(\ )$ se pueden obtener tablas de frecuencia asociadas a factores de igual longitud. En el ejemplo siguiente se utiliza la función $sample(\ )$ que permite extraer muestras de un conjunto de observaciones y el cruce de dos factores para construir una tabla de doble entrada.

Ejemplo:

$> table(c(mat3, mat3))$
1 2 3 5 7 9
4 4 2 2 4 2
$> (v4 < -c(mat3, mat3))$
[1] 1 2 5 3 7 9 2 7 1 1 2 5 3 7 9 2 7 1
$> (mat4 < -sample(v4, 18))$
[1] 1 1 7 1 2 7 5 3 7 9 2 3 1 9 5 2 7 2
$> table(v4, mat4)$
$mat4$
$v4$ 1 2 3 5 7 9
1 1 1 0 0 1 1
2 1 1 0 1 1 0
3 1 0 0 0 0 1
5 0 1 0 0 1 0
7 0 1 1 1 1 0
9 1 0 1 0 0 0

Para describir un factor mediante la agrupación en clases, se utilizan las funciones $factor(\ )$ y $cut(\ )$ como se indica en el siguiente ejemplo que trabaja con

las columnas 4^a y 5^a de los datos *airquality* y que representan $Temperatura$, 153 valores entre 56 y 97 y $Meses$, incluyendo solamente del 5 al 9. Si se hubiera descrito $Tempf$ solo con $cut(\ )$ y las 11 clases que se han pedido de forma aproximada en lugar de las 9 que efectivamente le corresponden, habrían salido dos filas de ceros en la tabla final.

$> Temp < -airquality[, 4]$
$> Tempf < -factor(cut(Temp, breaks = 55 + 5 * (0 : 11)))$
$> Month < -airquality[, 5]$
$> table(Tempf, Month)$

	Month				
Tempf	5	6	7	8	9
(55,60]	8	0	0	0	0
(60,65]	7	1	0	0	2
(65,70]	9	1	0	0	5
(70,75]	4	5	2	2	6
(75,80]	2	13	1	9	8
(80,85]	1	5	18	6	4
(85,90]	0	3	7	9	1
(90,95]	0	2	3	3	4
(95,100]	0	0	0	2	0

1.4. Bases de datos

En este apartado se pueden incluir los objetos *list*( ), como colección ordenada, a su vez, de objetos y $data.frame(\ )$ quizá la estructura de datos más utilizada y que consiste en una colección de variables de la misma longitud que pueden ser de tipo diferente.

1.4.1. Listas

Con la función *length*( ) aplicada al nombre de la lista, se obtiene el número de objetos de que consta, utilizando [[ ]] para describir dichas componentes y seguidas del número de orden entre [ ] que ocupa cada elemento del objeto para describirlos.

Ejemplo:

$> (lmat2 < -list(mat1, mat2))$

[[1]]

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-3.2106300	0.3877666	7.5592096	-0.3403993	1.2039125
[2,]	-0.7870847	8.7728136	1.0124324	2.7640037	-0.7761628

[3,] 0.4241971 0.4688422 0.2401416 3.4419892 -1.9599388
[4,] 0.2159416 1.0030262 -1.3389534 3.0205676 -4.2068276
[5,] 4.1927219 -6.1559740 4.4434380 1.6881193 -1.1155851
[[2]]
[,1] [,2] [,3]
[1,] 10 0 2
[2,] 0 10 4
[3,] 2 4 2
$> lmat2[[2]][5]$
[1]10
$> lmat2[[1]][7]$
[1] 8,772814

Las listas o elementos de las mismas se pueden enlazar para conseguir nuevas listas.

Ejemplo:
$> lmat2[[3]] < -lmat2[[2]][5]$
$> lmat2$
[[1]]
[,1] [,2] [,3] [,4] [,5]
[1,] -3.2106300 0.3877666 7.5592096 -0.3403993 1.2039125
[2,] -0.7870847 8.7728136 1.0124324 2.7640037 -0.7761628
[3,] 0.4241971 0.4688422 0.2401416 3.4419892 -1.9599388
[4,] 0.2159416 1.0030262 -1.3389534 3.0205676 -4.2068276
[5,] 4.1927219 -6.1559740 4.4434380 1.6881193 -1.1155851
[[2]]
[,1] [,2] [,3]
[1,] 10 0 2
[2,] 0 10 4
[3,] 2 4 2
[[3]]
[1] 10

1.4.2. Data frames

La forma de construir data frames, siempre que los elementos tengan la misma dimensión, es mediante $data.frame(\ )$ y $as.data.frame(\ )$.

Para conseguir que las variables del data frame sean objetos se utilizan las funciones $attach(\ )$ y $detach(\ )$

Ejemplo.
$> mat2$
[,1] [,2] [,3]
[1,] 10 0 2
[2,] 0 10 4

[3,] 2 4 2
> (dfmat2 < −data.frame(mat2))
X1 X2 X3
1 10 0 2
2 0 10 4
3 2 4 2
> X1
Error: Object "X1" not found
> attach(dfmat2)
> X1
[1] 10 0 2
> detach(dfmat2)
> X1
Error: Object "X1" not found

Importación de bases de datos

La mayoría de las veces, las observaciones se encontrarán almacenadas en ficheros externos que se tienen que incorporar mediante las funciones *read.table*( ), que pasa los datos a un *data.frame* y *scan*( ) que los pasa a un vector.

Para poder leer datos con *read.table*( ) éstos deben de tener en la primera línea el nombre de cada variable y en las sucesivas filas una etiqueta y los valores de las variables. Con *read.table*( ) se pueden leer los ficheros directamente si se encuentran en el mismo directorio en el que se arranca R y dando los datos de ubicación del fichero en su caso.

Ejemplo: *Se copia la siguiente base de datos en un fichero de texto, cork.txt* (mediante *copy* y *paste* en Wordpad, por ejemplo) *y se coloca en el mismo directorio desde el que se arranca R*

A B C D
1 72 66 76 77
2 60 53 66 63
3 56 57 64 58
4 41 29 36 38
> (dfcork < −read.table("cork.txt")
A B C D
1 72 66 76 77
2 60 53 66 63
3 56 57 64 58
4 41 29 36 38
Si se copia en un disquete habrá que introducir
> (dfcork < −read.table("a : /cork.txt"))
Para leer una base de datos que se encuentra en una dirección de internet
> (ejURL < −read.table("http : //www.mat.ucm.es/˜palomam/aed/ejURL.dat")
V1 V2 V3 V4 V5 V6 V7
1 1 210 201 -9 130 125 -5

2 2 169 165 -4 122 121 -1
3 3 187 166 -21 124 121 -3
4 4 160 157 -3 104 106 2
5 5 167 147 -20 112 101 -11
6 6 176 145 -31 101 85 -16
7 7 185 168 -17 121 98 -23
8 8 206 180 -26 124 105 -19
9 9 173 147 -26 115 103 -12
10 10 146 136 -10 102 98 -4
11 11 174 151 -23 98 90 -8
12 12 201 168 -33 119 98 -21
13 13 198 179 -19 106 110 4
14 14 148 129 -19 107 103 -4
15 15 154 131 -23 100 82 -18

Si se utiliza la función scan( )

$> (sejURL < -scan(\text{“}http://www.mat.ucm.es/\tilde{}palomam/aed/ejURL.dat\text{”}))$

Read 105 items

[1] 1 210 201 -9 130 125 -5 2 169 165 -4 122 121 -1 3 187 166 -21

[19] 124 121 -3 4 160 157 -3 104 106 2 5 167 147 -20 112 101 -11 6

[37] 176 145 -31 101 85 -16 7 185 168 -17 121 98 -23 8 206 180 -26 124

[55] 105 -19 9 173 147 -26 115 103 -12 10 146 136 -10 102 98 -4 11 174

[73] 151 -23 98 90 -8 12 201 168 -33 119 98 -21 13 198 179 -19 106 110

[91] 4 14 148 129 -19 107 103 -4 15 154 131 -23 100 82 -18

Para leer el fichero cork.txt con scan( ) hay que quitar el nombre de las variables. Por ejemplo guardamos en scork.txt solo los valores

1 72 66 76 77
2 60 53 66 63
3 56 57 64 58
4 41 29 36 38

$> (scork < -scan(\text{“}scork.txt\text{”}))$

Read 20 *items*

[1] 1 72 66 76 77 2 60 53 66 63 3 56 57 64 58 4 41 29 36 38

Se puede utilizar a continuación la función *edit( )* sobre la base de datos que se ha leído, para manejarla con comodidad en una ventana auxiliar.

Ejemplo:

$> edfcork < -edit(dfcork)$

Importación de bases de datos de otros sistemas estadísticos

Es recomendable cargar el paquete *foreign* aunque en algunas ocasiones se podrán traer utilizando *read.table( )*, no obstante se podrá utilizar menos memoria con las siguientes funciones.

Para traer ficheros de MINITAB con extensión *.mtp* (Minitab Portable Worksheet) se tiene *read.mtp( )* que lo trae en forma de lista. Para leer ficheros en formato SAS Transport (XPORT) y pasarlos a una lista de data frames, se

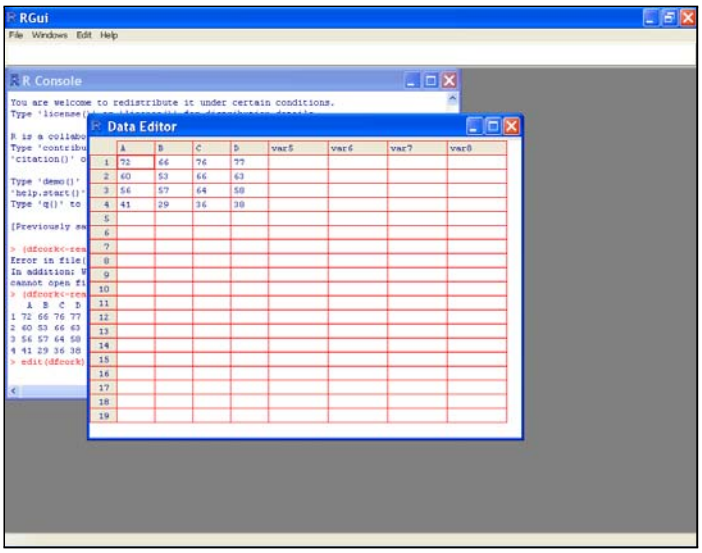


Figura 1.4.1

tiene *read.xport*( ) y análogamente *read.spss*( ) para ficheros exportados por SPSS, *read.S*( ) para muchos objetos de S-PLUS y *read.dta*( ) para ficheros de STATA.

1.5. Distribuciones de probabilidad

Con R es posible determinar algunas características de las distribuciones de probabilidad más utilizadas. teniendo que dar en cada caso los parámetros correspondientes. En concreto según la letra que se coloque delante del nombre de la distribución de probabilidad se podrá obtener la densidad si se coloca d, la CDF(función de distribución) si p, la función cuantílica si q y para simular una muestra habrá que poner r.

Ejemplo: *La distribución Normal se denomina norm, teniendo que especificar los parámetros media y desviación típica así que para obtener la función de densidad, y de distribución en* $z = 1{,}96$ *para una* $N(0,1)$

$> dnorm(1{,}96, 0, 1)$
[1]0,05844094
$> pnorm(1{,}96, 0, 1)$
[1]0,9750021
$> qnorm(0{,}975, 0, 1)$
[1]1,959964
$> qnorm(0{,}9750021, 0, 1)$
[1]1,96

Ejemplo: *Para obtener una muestra de tamaño* 100 *de una* $N(3,2)$ *y otra*

independientemente de una $t-Student$*, a la que se denomina* t*, con* 1 *grado de libertad, y representar ambos histogramas para comparar la diferencia en las colas. Para representar ambos histogramas se utilizará la función* $par(\ )$ *que establece parámetros en los gráficos y de la que se utiliza* $mfrow = c(2,1)$ *para particionar la ventana del gráfico en dos filas y una columna.*

$> par(mfrow = c(2,1))$

$> mt1 < -rt(100,1)$

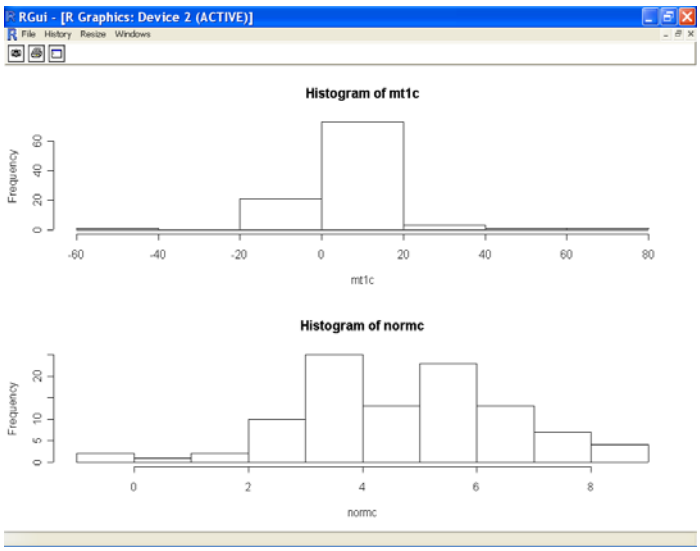
$> mt1c < -mt1 * 2 + 3$

$> hist(mt1c)$

$> norm < -rnorm(100,1)$

$> normc < -norm * 2 + 3$

$> hist(normc)$



1.6. Programación

Los comandos en R, pueden ser agrupados entre "llaves" { } dando como valor el resultado final con la última expresión. A su vez puede integrarse dentro de una expresión más amplia y así sucesivamente. Dentro de estas expresiones puede haber ciertos comandos de control de la ejecución permitiendo condicionar la ejecución de un bloque o ejecutar repetidamente otro.

Ejecución condicionada

Responde al comando if / $else$ de la forma habitual. Hay una ($expresión\_1$) que debe evaluarse, si es $CIERTA$ entonces se evalúa la $expresión\_2$, de lo contrario se evalúa la $expresión\_3$, el valor final de la expresión global es el de la seleccionada finalmente. La sintaxis correspondiente al caso anterior sería

$> if\ (expresión\_1)$
$expresión\_2$
$else$
$expresión\_3$

También pueden enlazarse sucesivas condiciones

$> if\ (expresión\_1)$
$expresión\_2$
$else\ if\ (expresión\_3)$
$expresión\_4$
$else\ if\ (expresión\_5)$
$expresión\_6$
$else$
$expresión\_8$

evaluándose, en orden, las sucesivas expresiones impares hasta que una es $CIERTA$, valorándose la correspondiente expresión par de dicho caso. No hay límites para los $else\ if$ permitidos.

Ejemplo: *Donde se van alterando los valores de una muestra de números aleatorios según sea el valor de su suma.*

$> u < -runif(20)$
$> u$

[1] 0.8274505 0.7492379 0.9971830 0.6657915 0.2522081 0.2581410 0.6137507

[8] 0.1366209 0.8652314 0.6629930 0.6904928 0.8157469 0.7545211 0.9928052

[15] 0.8426746 0.8316724 0.3762088 0.3373698 0.1013896 0.8551755

$> s < -sum(u)$
$> if\ (s < 5)\ u < -1 + u\ else\ if\ (s < 10)\ u < - - u$
$> u$

[1] 0.8274505 0.7492379 0.9971830 0.6657915 0.2522081 0.2581410 0.6137507

[8] 0.1366209 0.8652314 0.6629930 0.6904928 0.8157469 0.7545211 0.9928052

[15] 0.8426746 0.8316724 0.3762088 0.3373698 0.1013896 0.8551755

$> s$

[1] 12.62666

$>if\ (s < 5)\ u < -1 + u\ else\ if\ (s < 10)\ u < - - u\ else\ if\ (s < 15)\ u < -2 * u$

$> u$

[1] 1.6549009 1.4984758 1.9943660 1.3315830 0.5044162 0.5162821 1.2275015

[8] 0.2732418 1.7304629 1.3259860 1.3809856 1.6314938 1.5090422 1.9856104

[15] 1.6853491 1.6633447 0.7524176 0.6747396 0.2027791 1.7103510

Ejecución repetida

Mediante las órdenes for, $while$ and $repeat$ se puede conseguir la valoración repetida de ciertas expresiones. Por ejemplo para for sería

$> for\ (nombre\ en\ expresión\_1)\ \{expresión\_2\}$

donde el $nombre$ es el contador de las repeticiones y en la $expresión\_2$ se incluyen las acciones a realizar, en función de la variable $nombre$.

Ejemplo: *A partir de una matriz* 20×10 *de observaciones aleatorias e independientes* $N(0,1)$ *se obtienen las medias muestrales de las columnas, representándolas con un gráfico de tallo y hojas*

$z < -matrix(rnorm(200), 20, 10)$
$mean.samp < -NULL$
$for(i\ in\ 1:10)\{$
$mean.samp[i] < -mean(z[, i])$
$\}$
$stem(mean.samp)$

1.6.2. Funciones

En lenguaje R las funciones se pueden incorporar como objetos a los procedimientos o almacenar para su uso posterior. La forma general es

$nombre < -function(argumento\_1, argumento\_2, \ldots)expresión$

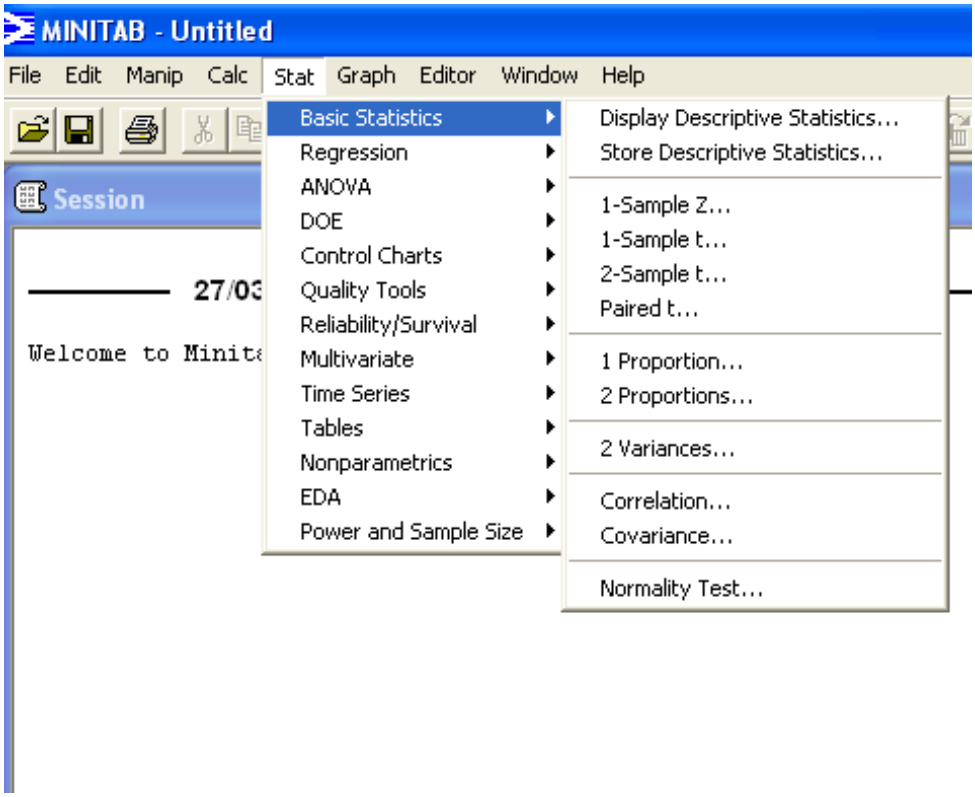
donde la *expresión* se escribe en lenguaje R y utiliza los argumentos introducidos. Para llamar a la función se utilizará el $nombre(valores\ de\ los\ argumentos)$ como se utilizan normalmente las funciones que ya están definidas en los paquetes que se cargan al comienzo.

Ejemplo: *Función para calcular y listar las medias aritméticas de distintas potencias de una colección de* 10 *observaciones aleatorias e independientes* $N(0,1)$.

```
fourmom <- function(x){
m1 <- mean(x)
m2 <- mean(x^2)
m3 <- mean(x^3)
m4 <- mean(x^4)
list(m1 = m1, m2 = m2, m3 = m3, m4 = m4)
}
x <- rnorm(10)
fourmom(x)
```

2. INTRODUCCIÓN A MINITAB

Es un software estadístico comercial, *www.minitab.com*, en un entorno Windows. Su funcionamiento es similar a la mayoría del software comercial estadístico, pero con facilidades particulares es el aspecto metodológico por lo que hemos considerado oportuno incluirlo como representante de esta categoría de herramientas para el análisis de datos. Todos los elementos que se manejan en un determinado trabajo se pueden recoger en *PROYECTOS* para almacenar y recuperar posteriormente toda la información. Los procedimientos en persianas que se despliegan al marcar los distintos elementos de la *Barra de herramientas* (Toolbars) que aparece en la parte de arriba de la pantalla.

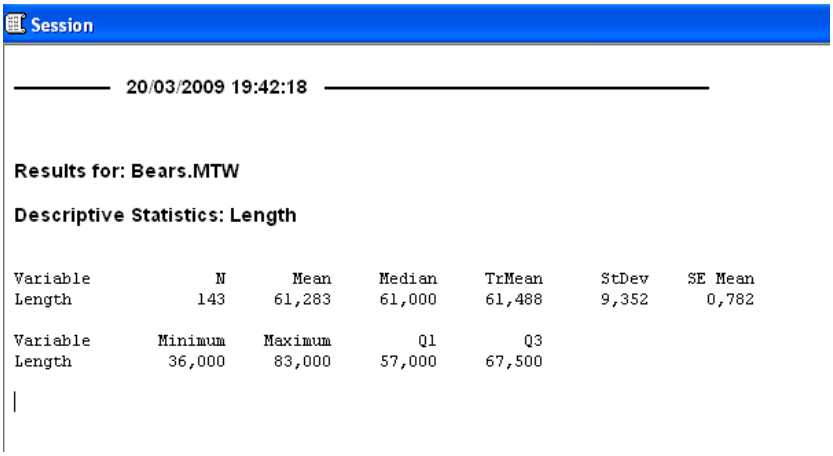


2.1. Ventanas Iniciales

Según se arranca MINITAB aparecen distintas "ventanas" que tienen diferentes funciones.

2.1.1. Ventana de la sesión (Session Window)

Recoge todas las salidas de texto como por ejemplo los resultados finales en la aplicación de los procedimientos.



Session

20/03/2009 19:42:18

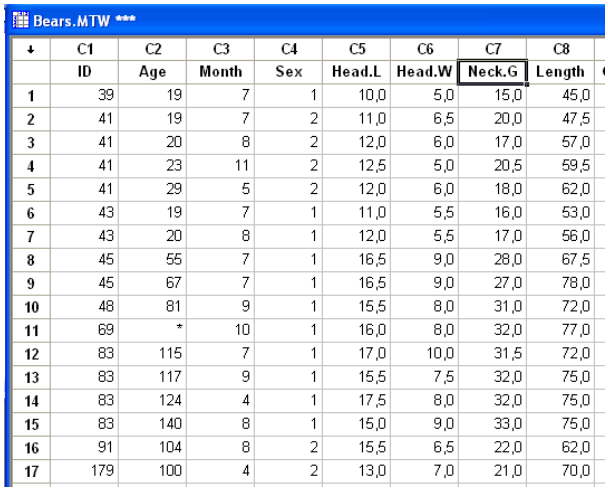
Results for: Bears.MTW

Descriptive Statistics: Length

Variable	N	Mean	Median	TrMean	StDev	SE Mean
Length	143	61,283	61,000	61,488	9,352	0,782

Variable	Minimum	Maximum	Q1	Q3
Length	36,000	83,000	57,000	67,500

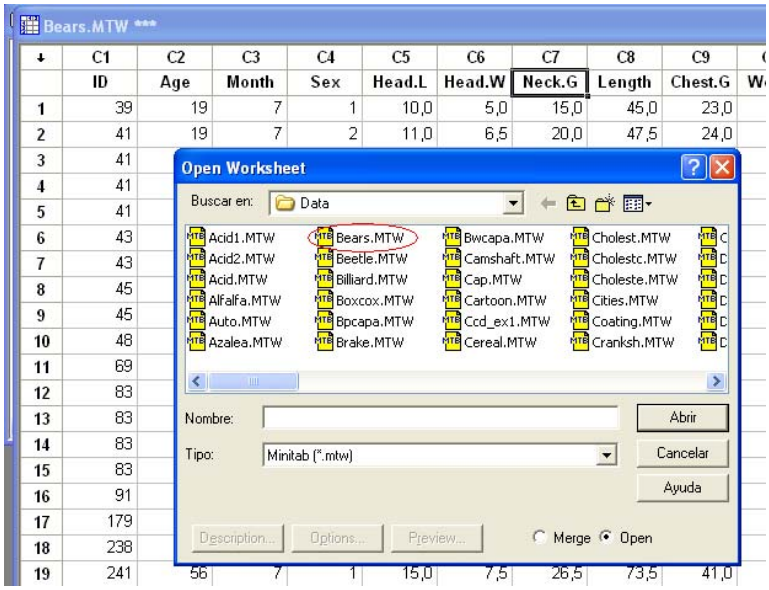
2.1.2. Ventana de datos (Worksheet)



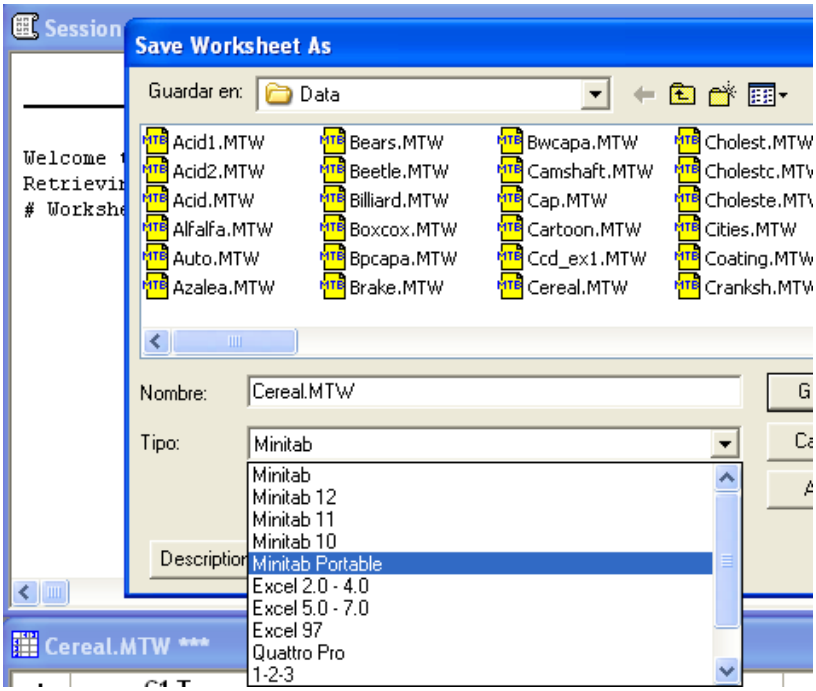
Bears.MTW ***

↓	C1	C2	C3	C4	C5	C6	C7	C8
	ID	Age	Month	Sex	Head.L	Head.W	Neck.G	Length
1	39	19	7	1	10,0	5,0	15,0	45,0
2	41	19	7	2	11,0	6,5	20,0	47,5
3	41	20	8	2	12,0	6,0	17,0	57,0
4	41	23	11	2	12,5	5,0	20,5	59,5
5	41	29	5	2	12,0	6,0	18,0	62,0
6	43	19	7	1	11,0	5,5	16,0	53,0
7	43	20	8	1	12,0	5,5	17,0	56,0
8	45	55	7	1	16,5	9,0	28,0	67,5
9	45	67	7	1	16,5	9,0	27,0	78,0
10	48	81	9	1	15,5	8,0	31,0	72,0
11	69	*	10	1	16,0	8,0	32,0	77,0
12	83	115	7	1	17,0	10,0	31,5	72,0
13	83	117	9	1	15,5	7,5	32,0	75,0
14	83	124	4	1	17,5	8,0	32,0	75,0
15	83	140	8	1	15,0	9,0	33,0	75,0
16	91	104	8	2	15,5	6,5	22,0	62,0
17	179	100	4	2	13,0	7,0	21,0	70,0

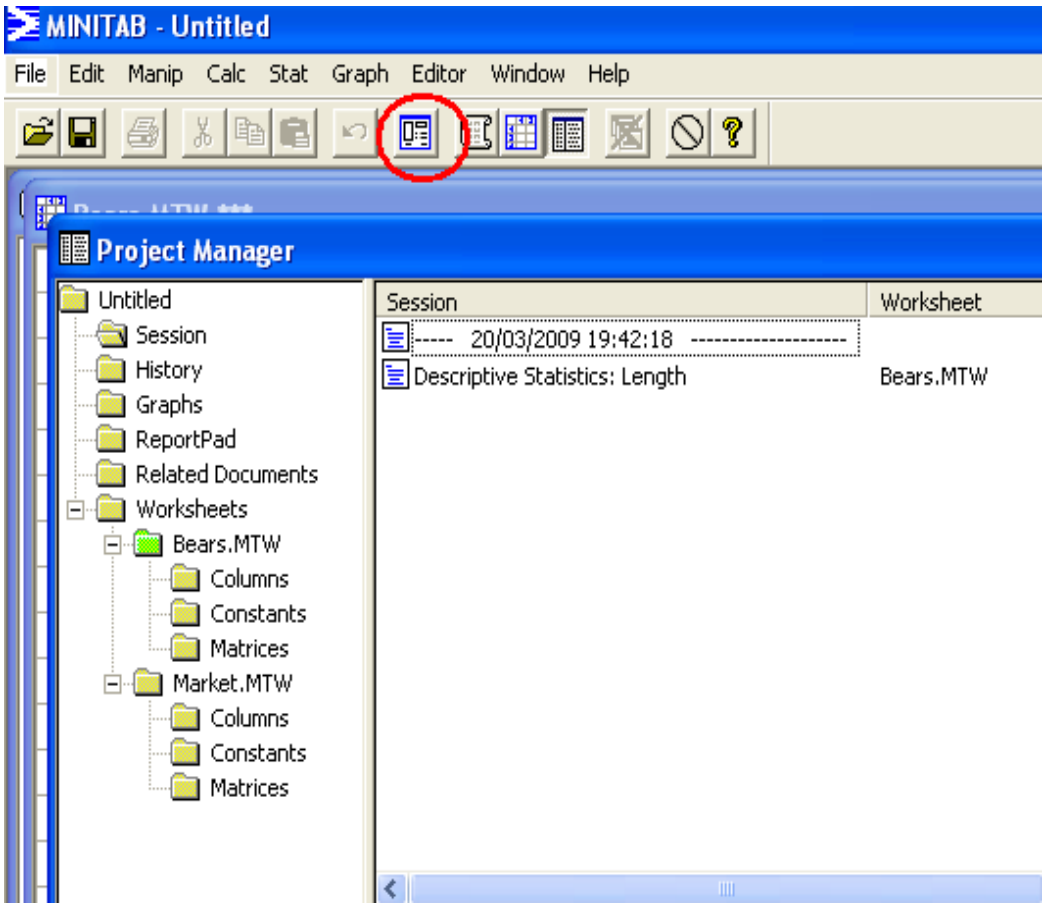
Donde se visualizan y editan las bases de datos. Puede haber varias abiertas. La que está activa, es decir sobre la que se aplican los procedimientos requeridos es la que contiene $***$ junto al nombre. Existen bases de datos almacenadas en el espacio MINITAB. Para abrirlas en la *Barra de Herramientas* File → Open Worksheet y por defecto aparecerán los distintos ficheros *.mtw.



Si se quieren exportar es preferible guardarlas como ficheros *Minitab Portable* con extensión *.mtp.



2.1.3. Ventana de Información General del Proyecto (Project Manager)



Describe todos los elementos utilizados en un Proyecto junto con sus características. Dentro de esta ventana se puede llamar a la *Ventana de Historia* (History) en la que se pueden ver todos los comandos asociados a los procedimientos realizados desde la *Barra de Herramientas*. La posibilidad de utilizar comandos permitirá la personalización de las aplicaciones en un nivel avanzado de utilización del software.

Se pueden almacenar:

Variables con la denominación C*

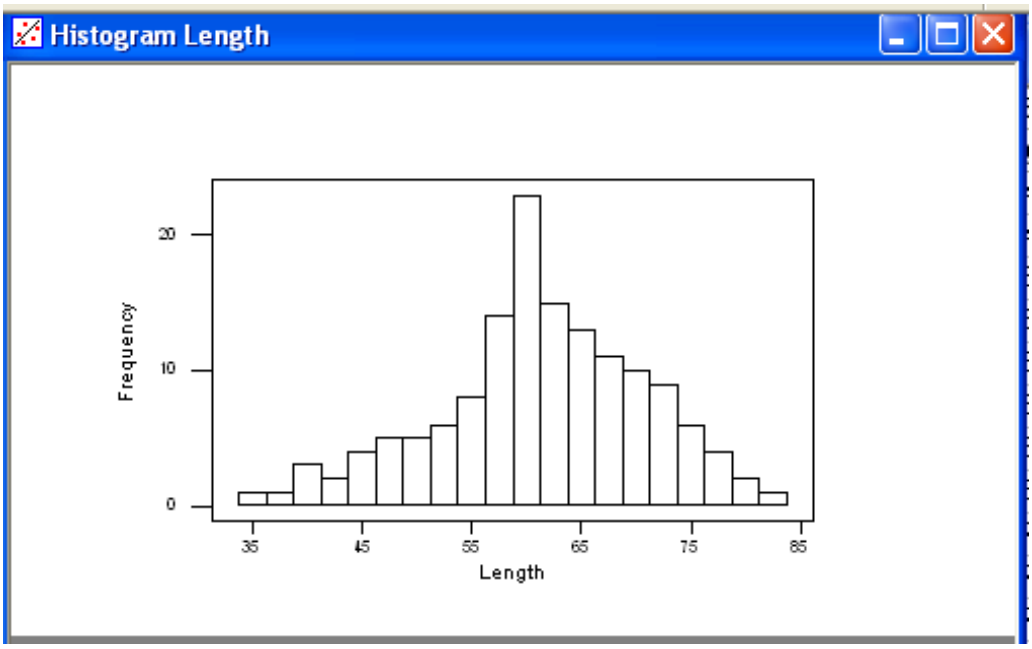
Matrices con la denominación M*

Constantes con la denominación K*.

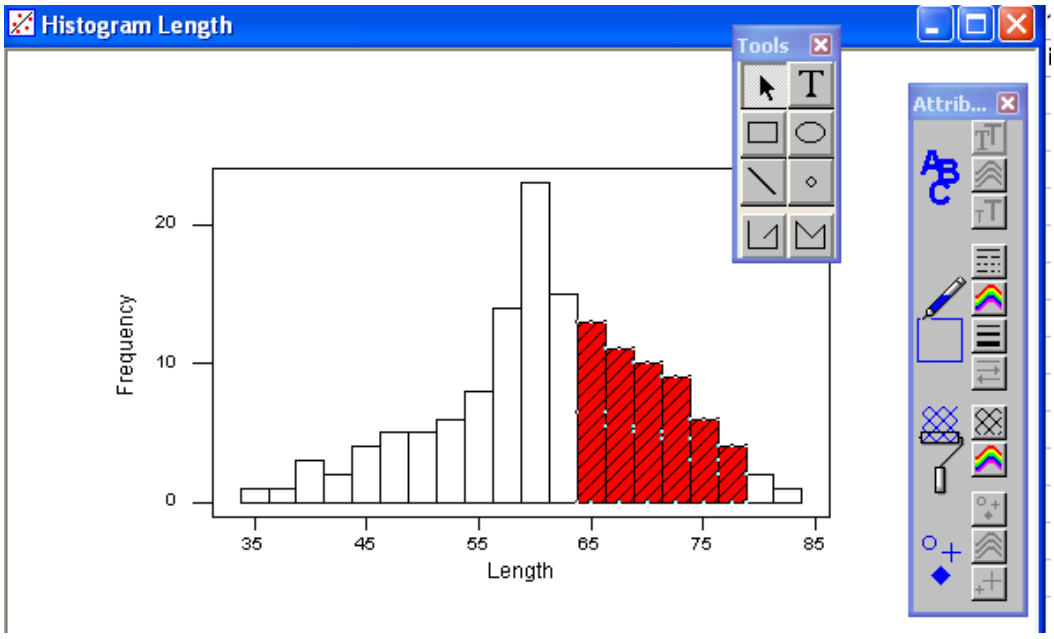
Para recuperar estos elementos en la *Barra de Herramientas* Manip → Display Data.

2.1.4. Ventana de Gráficos (Graphs)

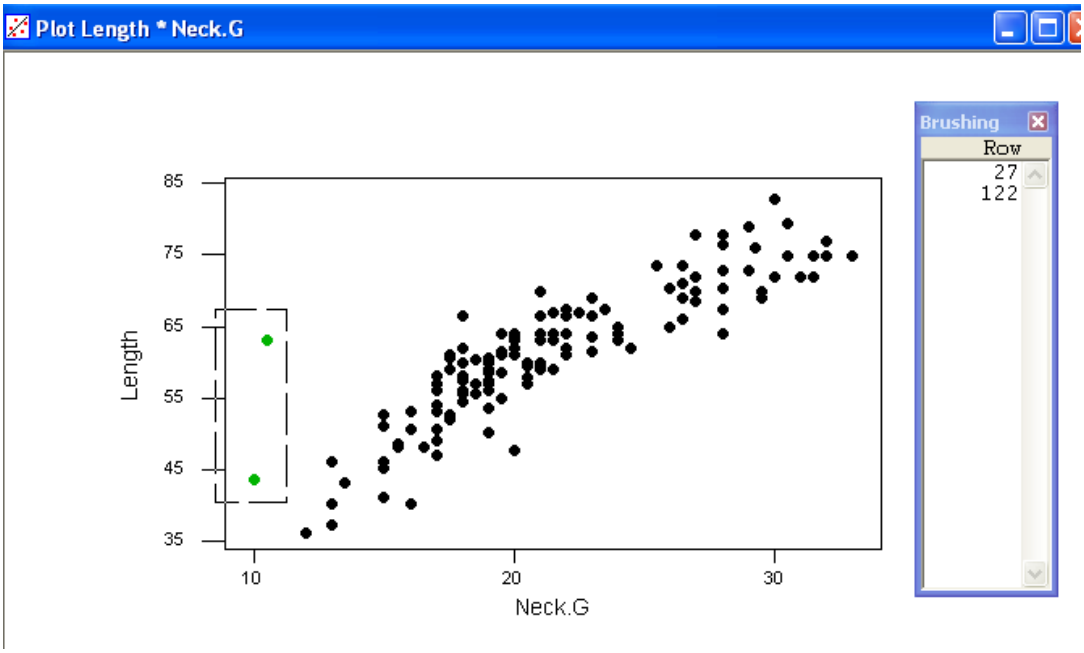
Siempre que se solicite la realización de un gráfico, aparecerá una ventana especial con las figuras.



Si esta ventana está activa, se puede "editar" mediante la persiana *Editor* de la *Barra de Herramientas*

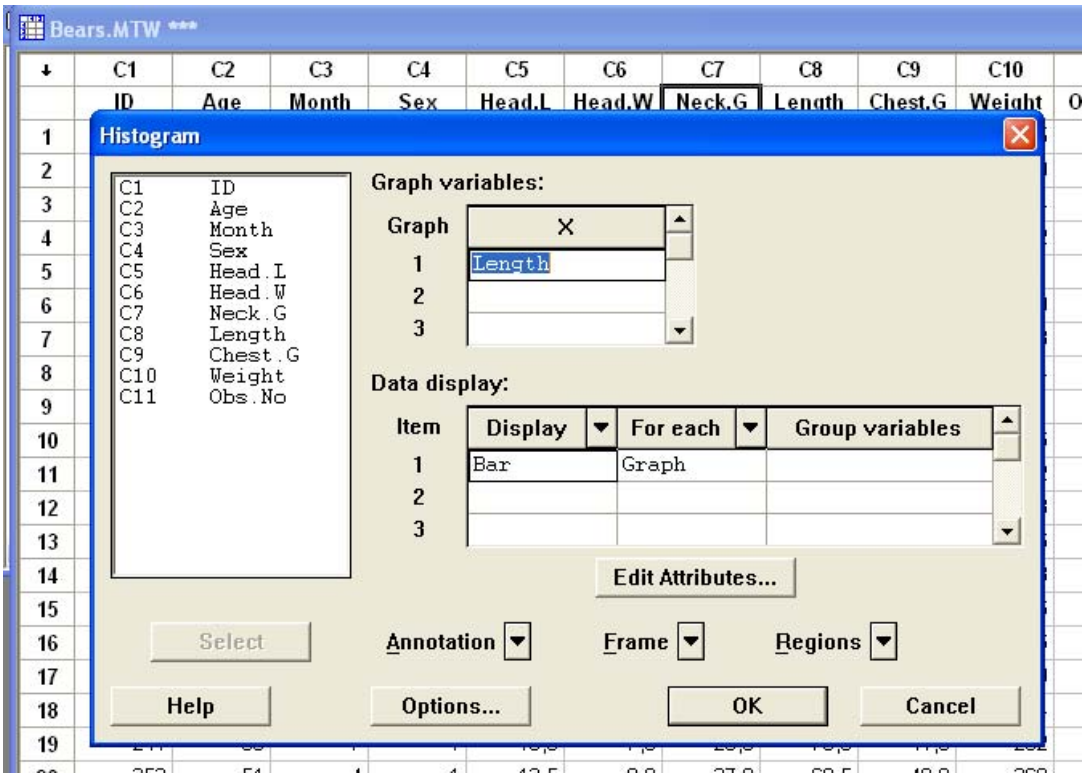


o “barrer” (Brush) detectando los puntos de interés.



2.2. Procedimientos

Marcando los elementos de la *Barra de Herramientas* se despliegan las persianas con los descriptores de las distintas tareas que pueden realizarse. Al señalar un apartado en concreto pueden aparecer nuevas persianas con distintas opciones a realizar o las ventanas de diálogo en las que se definen los parámetros concretos para ejecutar el procedimiento en un caso particular.



También se pueden utilizar comandos para la ejecución de los procedimientos. Para ello se necesita activar esta opción: Con la ventana de Sesión activa ir al *Editor* en la *Barra de Herramientas* marcar la fila de la persiana con *Enable Commands* y aparecerá MTB > en la ventana de Sesión donde se deberán introducir los comandos. Esta forma de trabajo implica un conocimiento avanzado del software pero como se dijo anteriormente permite desarrollar las aplicaciones de una forma personalizada.

3. PRÁCTICAS

3.1. Importar bases de datos

Con R

- Desde una dirección de internet:

```
darwin<-read.table(‘‘http://www.mat.ucm.es/~palomam/aedej1.dat’’)
```

- Desde un fichero de texto

```
darwin2<-read.table(‘‘C:/aed/datos/ejemplo1.dat’’)
```

- Desde un fichero de datos (por ejemplo dBase *.dbf) que puede ser una base de datos abierta en Minitab y guardada como fichero dBase.

cargar el paquete foreign

```
library(foreign)
Peru<-> read.dbf(‘‘F:/aed/datos/Peru.dbf’’)
```

- Desde el Campus Virtual de la UCM (Asignatura AED de la Lic. C.C. Matemáticas)

Ir a Bases de datos → datos.zip → descomprimir → seleccionar el fichero → guardar como fichero de texto → abrir desde fichero de texto

Con MINITAB

- Desde una dirección de internet un fichero Minitab Portable:

http://www.mat.ucm.es/~palomam/ejemplo1.mtp

y directamente arranca Minitab o guardar con Worpad como texto en C:/aed/ej1minitab.mtp

- Desde Minitab:

File → Open Worksheet → Minitab Portable

- Desde un fichero de texto (“C:/aed/datos/ejemplo1.txt”):

File → Other Files → Import Special Text

- Desde el Campus Virtual de la UCM (Asignatura AED de la Lic. C.C. Matemáticas)

Ir a Bases de datos → datos.zip → descomprimir → seleccionar el fichero → guardar como fichero de texto → abrir desde fichero de texto

3.2. Ejercicios

Se van a cargar unos datos que posteriormente servirán para aplicar los procedimienos considerados. Aunque se encuentran en un fichero comprimido dentro del Campus Virtual de la UCM, podrán tambien ser descargados desde *http://www.mat.ucm.es/~palomam/aed/datos.*

Campus Virtual → Bases de datos → clase1 → Open o Save

Importar los siguientes datos

Datos 1

http://www.mat.ucm.es/~palomam/aed/datos/datos1.dat

```
49
-67
8
16
6
23
28
41
14
56
24
75
60
-48
29
```

Minitab (Cortar y Pegar directamente)

R (Cortar y Pegar en Worpad) → Guardar como fichero texto ejemplo1.dat y abrir con *read.table* o con *scan.*

Datos 2

http://www.mat.ucm.es/~palomam/aed/datos/datos2.txt

```
109 137.6
113 147.8
115 136.8
116 140.7
119 132.7
120 145.4
121 135
124 133
126 148.5
129 148.3
```

```
130 147.5
133 148.8
134 133.2
135 148.7
137 152
139 150.6
141 165.3
142 149.9
```

Minitab (Cortar y Pegar directamente sobre dos columnas)

R (Cortar y Pegar en Worpad) → Guardar como fichero de texto resline.txt y abrir con *read.table.*

Parte II

Análisis de datos

4. INTRODUCCIÓN

4.1. Historia

- Utilización de gráficos desde 1786 en que William Playfair representa muy ajustadamente datos comparados de importaciones y exportaciones en Inglaterra.

- Diagramas en el libro *Statistical Methods for Research Workers* escrito en 1925 por Sir Ronald Aylmer Fisher, creador de los fundamentos de la estadística moderna.

- Pero es el texto de Tukey "Exploratory Data Analysis" (EDA) de 1977, el que supone un cambio fundamental en el planteamiento del análisis de datos, introduciendo nuevas técnicas y en definitiva una nueva filosofía en el conocimiento estadístico de las bases de datos.

- El futuro se orienta en el desarrollo de las técnicas de Minería de Datos (Data Mining) utilizando el gran avance de los ordenadores en el manejo de grandes bases de datos así como en el aumento de la potencia de los procedimientos gráficos.

4.2. Cuestiones Generales

Relación con otro tipo de planteamientos

CLÁSICO: Problema → Datos → Modelo → Análisis → Conclusiones

AED: Problema → Datos → Análisis →Modelo → Conclusiones

BAYESIANO: Problema → Datos →Modelo→ Distribución a priori → Análisis → Conclusiones

Logros del AED

- Visión global de los datos
- Aplicación de modelos simples y ajustados
- Detección de datos atípicos
- Conclusiones robustas
- Estimadores de parámetros
- Ordenación de factores que inciden en el problema según su importancia
- Observación de factores significativos

- Planteamiento del escenario óptimo

Procedimientos

- Cuantitativos
- Gráficos

Tipos de Problemas

$$\text{Univariantes}\begin{cases}\text{Variables cuantitativas}\\ \text{Variables cualitativas o categóricas}\end{cases}$$

$$\text{Multivariantes}\begin{cases}(Y, X_1, \ldots, X_k) \rightarrow \text{predicción, optimización, } \ldots\\ (X_1, \ldots, X_p) \rightarrow \text{identificación de estructura}\end{cases}$$

5. DESCRIPCIÓN DE DATOS UNIVARIANTES

5.1. Métodos cuantitativos

5.1.1. Medidas de tendencia central

Dados $(x_1, \ldots, x_k)$ con frecuencias absolutas $(n_1, \ldots, n_k)$ tales que

$$n = n_1 + \ldots + n_k$$

- **Media aritmética**

$$\bar{x} = \frac{\sum_{i=1}^{k} x_i n_i}{n}$$

- **Media geométrica**

$$G = \sqrt[n]{x_1^{n_1} \ldots x_k^{n_k}}$$

- **Media armónica**

$$H = \frac{n}{\sum_{i=1}^{k} \frac{1}{x_i} n_i}$$

- **Mediana** Considerando todas las observaciones ordenadas

$$x_{1:n} \leq \cdots \leq x_{k:n} \leq \cdots \leq x_{n:n}$$

si n es impar $M_e = x_{\left(\left[\frac{n}{2}\right]+1\right):n}$

si n es par $M_e = \frac{x_{(\frac{n}{2}+1):n} + x_{(\frac{n}{2}):n}}{2}$

- **Moda** M_o
  es el valor más frecuente
  no tiene por qué ser única (distribuciones bimodales, multimodales,...)

5.1.2. Medidas de posición

Cuantil de orden p, tal que $0 < p < 1$. Si $p = \frac{1}{4}$ cuartiles, $Q_1, Q_2 = M_e, Q_3$.

Método 1: interpolación entre $x_{(np+1):n}$ y $x_{(np):n}$ si $np \in \mathbb{Z}^+$

$x_{([np]+1):n}$ si $np \notin \mathbb{Z}^+$

Método 2: (R) interpolación entre $x_{[(n-1)p+1]:n}$ y $x_{[(n-1)p+1]+1:n}$ si $(n-1)\,p+1 \notin \mathbb{Z}^+$

$x_{((n-1)p+1):n}$ si $(n-1)\,p+1 \in \mathbb{Z}^+$

Método 3: $(MINITAB)$ interpolación entre $x_{[(n+1)p]:n}$ y $x_{[(n+1)p]+1:n}$ si $(n+1)\,p \notin \mathbb{Z}^+$

$x_{((n+1)p):n}$ si $(n+1)\,p \in \mathbb{Z}^+$

Bisagras (Hinges de Tukey)

Son las medianas de las "mitades" en que divide la mediana a las observaciones, incluyendo la mediana para un número impar de observaciones:

H_1 bisagra inferior

H_2 bisagra superior

Forman parte de los Resúmenes de los Valores Letra (Letter Values Displays), que constituyen un método para describir los datos, una vez ordenados y contabilizados, mediante letras que se adjudican a diferentes valores de posición. Por ejemplo M para la mediana, H para las bisagras y así $E, C, D, B, A, \ldots$

Resumen de los cinco números

	M_e	
H_1		H_2
$x_{n:n}$		$x_{1:n}$

5.1.3. Medidas de dispersión

Varianza

$$s^2 = \frac{\sum_{i=1}^{k} (x_i - \bar{x})^2 n_i}{n-1} = \frac{\sum_{i=1}^{k} x_i^2 n_i - n\bar{x}^2}{n-1}$$

Desviación típica o estándar

$$s = \sqrt{\frac{\sum_{i=1}^{k} (x_i - \bar{x})^2 n_i}{n-1}}$$

Coeficiente de variación

$$CV = \frac{s}{\bar{x}}$$

Rango intercuartílico

$$R_I = Q_3 - Q_1$$

Rango

$$R = x_{n:n} - x_{1:n}$$

Desviación media (respecto de la media)

$$DM = \frac{\sum_{i=1}^{k} |x_i - \bar{x}|\, n_i}{n}$$

Desviación mediana

$$MAD \rightarrow \text{ mediana de } \{|x_i - M_e|\}_{i=1,\ldots,n}$$

Desviación de las bisagras $(H - spread)$

$$H_2 - H_1$$

5.1.4. Medidas de asimetría y curtosis

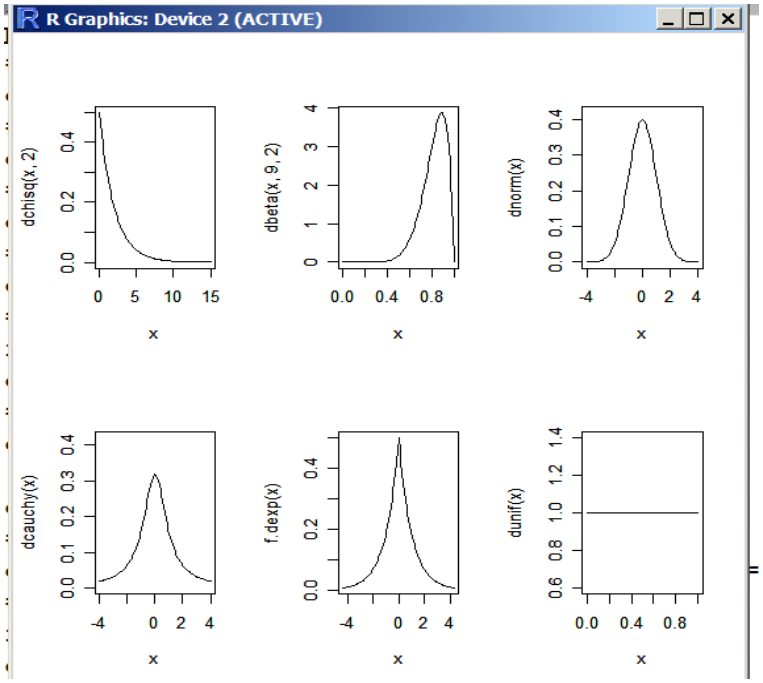
Se pueden representar las gráficas de distintas funciones de densidad con diferentes características de simetría y aplastamiento (curtosis)

R: Funciones de densidad

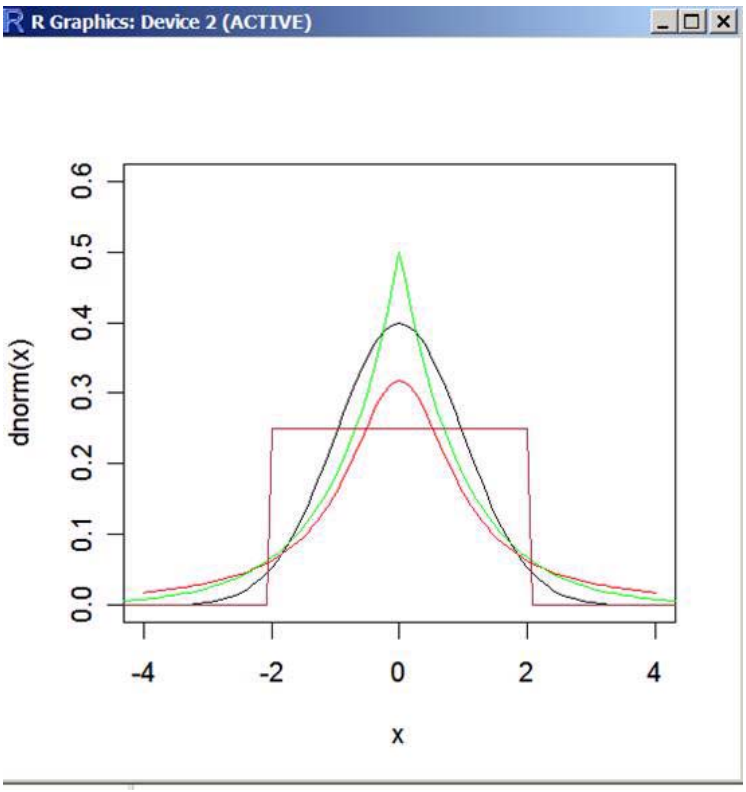
F:\aed0708\ejerciciosclase\clase8\fdens.R - R Editor

```
par(mfrow=c(2,2))
#Chi cuadrado(2)
curve(dchisq(x,2),0,15)
#Beta(9,2)
curve(dbeta(x,9,2),0,1)
#N(0,1)
curve(dnorm(x),-4,4,ylim=c(0,0.42))
#Cauchy
curve(dcauchy(x),-4,4,ylim=c(0,0.42))
#Doble exponencial
f.dexp<-function(x){(1/2)*exp(-abs(x))}
curve(f.dexp(x))

curve(dnorm(x),-4,4,ylim=c(0,0.6))
#Cauchy
curve(dcauchy(x),-4,4,ylim=c(0,0.42),add=TRUE,col="red")
#Doble exponencial
f.dexp<-function(x){(1/2)*exp(-abs(x))}
curve(f.dexp(x),add=TRUE,col="green")
#Uniforme
curve(dunif(x,-2,2),add=TRUE,col="brown")
```

R: Funciones de densidad



y sobreponiendo las simétricas respecto del origen,



Coeficiente de asimetría Pearson

$$\frac{3\left(\bar{x}-M_e\right)}{s}$$

Coeficiente de asimetría muestral

$$g_1=\frac{\sum_{i=1}^{k}\left(x_i-\bar{x}\right)^3 n_i}{\left(n-1\right)s^3}\begin{cases}>0 \text{ asimetría positiva o a la derecha}\\ =0 \text{ simetría}\\ <0 \text{ asimetría negativa o a la izquierda}\end{cases}$$

Coeficiente de curtosis muestral

$$k=\frac{\sum_{i=1}^{k}\left(x_i-\bar{x}\right)^4 n_i}{\left(n-1\right)s^4}-3\begin{cases}>0 \text{ leptocúrtica (“picuda”)}\\ =0 \text{ mesocúrtica}\\ <0 \text{ platicúrtica (“aplastada”)}\end{cases}$$

5.1.5. **Datos 1**

http://www.mat.ucm.es/~palomam/aed/datos/datos1.dat

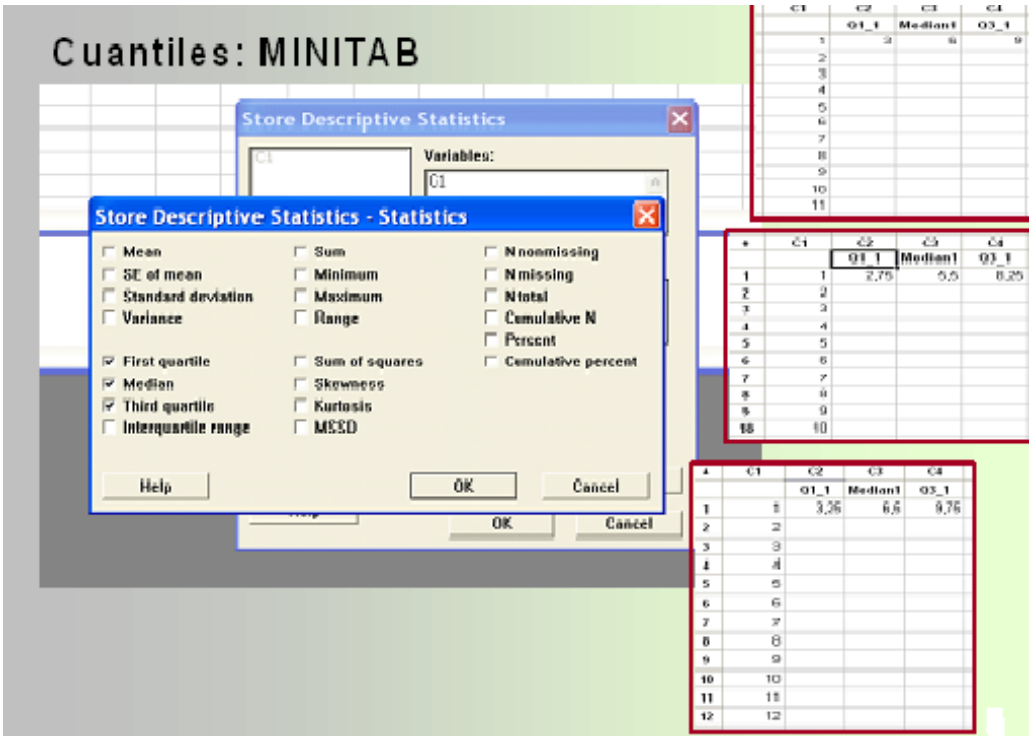
MINITAB: Medidas Descriptivas

Results for: ejemplo1.mtp

Descriptive Statistics: C1

Variable	N	Mean	Median	TrMean	StDev	SE Mean
C1	15	20,93	24,00	23,54	37,74	9,75

Variable	Minimum	Maximum	Q1	Q3
C1	-67,00	75,00	8,00	49,00



MINITAB:

	C1	C2	C3	C4
		Q1_1	Median1	Q3_1
1	3	9,25	24	31,5
2	7			
3	10			
4	22			
5	23			
6	25			
7	28			
8	31			
9	33			
10	35			

Porque

$(n+1)/4 = 2.75 \rightarrow Q_1 = X_{2:10} + (0.75)(X_{3:10} - X_{2:10}) = 7 + (0.75)3 = 9.25$

$3(n+1)/4 = 8.25 \rightarrow Q_3 = X_{8:10} + (0.25)(X_{9:10} - X_{8:10}) = 31 + (0.25)2 = 31.5$

Funciones de R

#base de datos

```
ejemplo1<-
read.table(‘‘http://www.mat.ucm.es/~palomam/aed/datos/datos1.dat’’)
```

#función media aritmética

```
ej1.m<-mean(ejemplo1)
```

#cálculo de la media aritmética

```
sum(ejemplo1)/dim(ejemplo1)
```

#vector de datos

```
ejemplo1.s<-
scan(‘‘http://www.mat.ucm.es/~palomam/aed/
datos/datos1.dat’’)
```

#función media aritmética

```
ej1.m.s<-mean(ejemplo1.s)
```

#cálculo de la media aritmética

```
sum(ejemplo1.s)/length(ejemplo1.s)
```

#cálculo de la media geométrica

```
ej1.g<-exp(mean(log(abs(ejemplo1))))
ej1.g.2<-(prod(ejemplo1))^(1/15)
28.81678
```

#cálculo de la media armónica

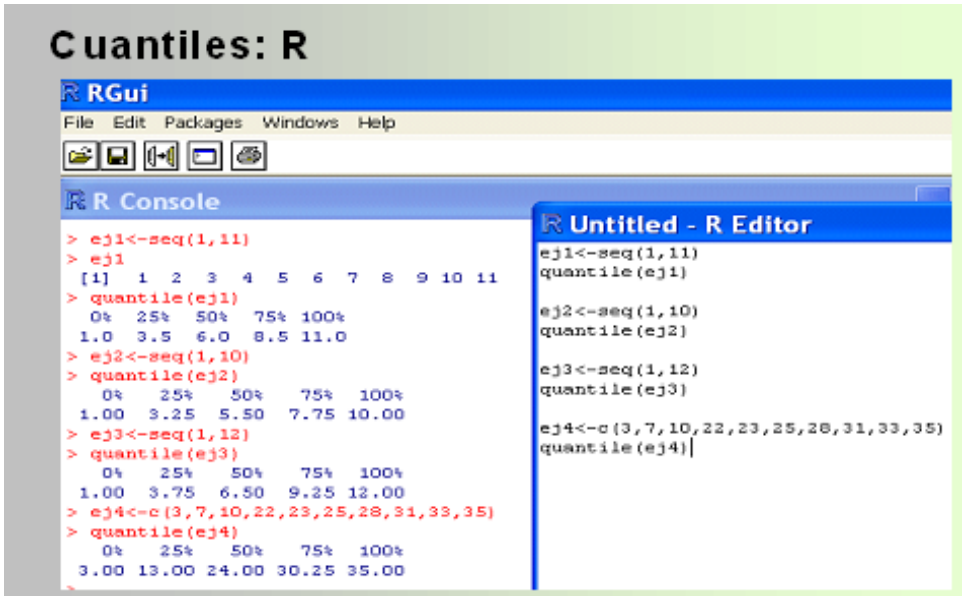
```
ej1.h<-1/(mean(1/ejemplo1))
ej1.2.2<-15/sum(1/ejemplo1)
23.51709
```

#cuartiles

```
quantile(ejemplo1.s)
0% 25% 50% 75% 100%
-67 11 24 45 75
```

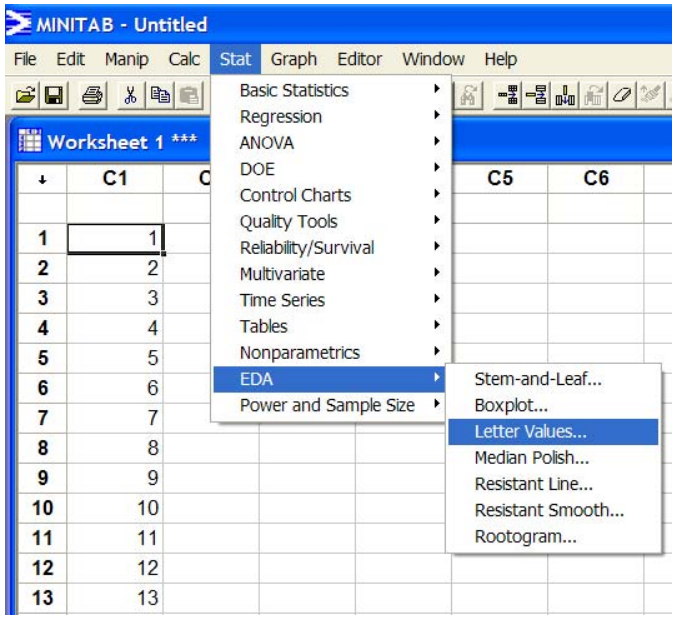
#mediana

```
median(ejemplo1.s)
24
#deciles
quantile(ejemplo1.s,probs=seq(0,1,by=1/10))
0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
-67.0 -26.4 7.6 14.4 20.2 24.0 28.4 38.6 50.4 58.4 75.0
```

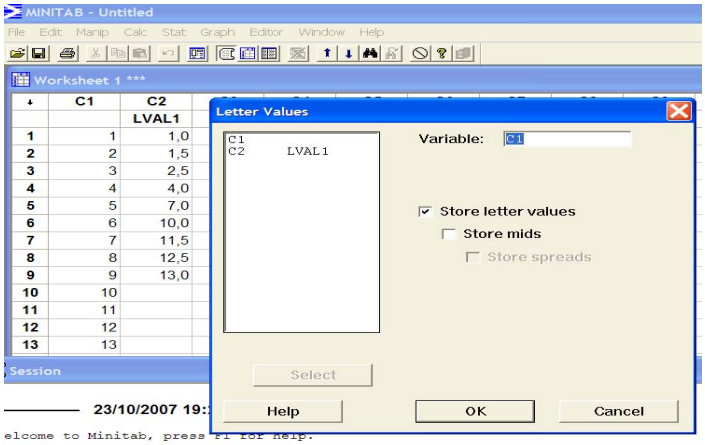


5.1.6. Ejercicios

MINITAB



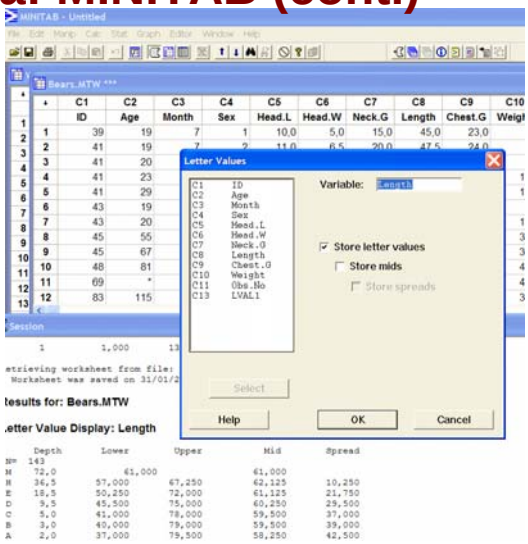
Valores letra: MINITAB (cont.)



	Depth	Lower	Upper	Mid	Spread
N=	13				
M	7,0	7,000		7,000	
H	4,0	4,000	10,000	7,000	6,000
E	2,5	2,500	11,500	7,000	9,000
D	1,5	1,500	12,500	7,000	11,000
	1	1,000	13,000	7,000	12,000

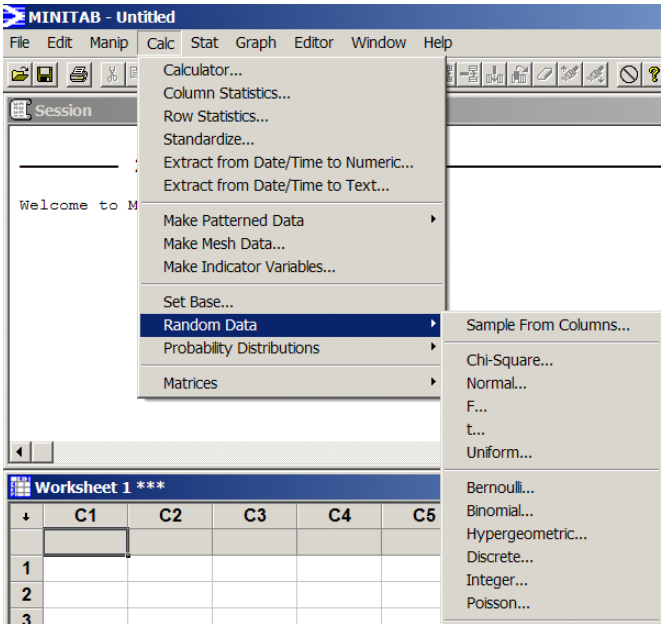
Valores letra: MINITAB (cont.)

Datos: Bears.MTW

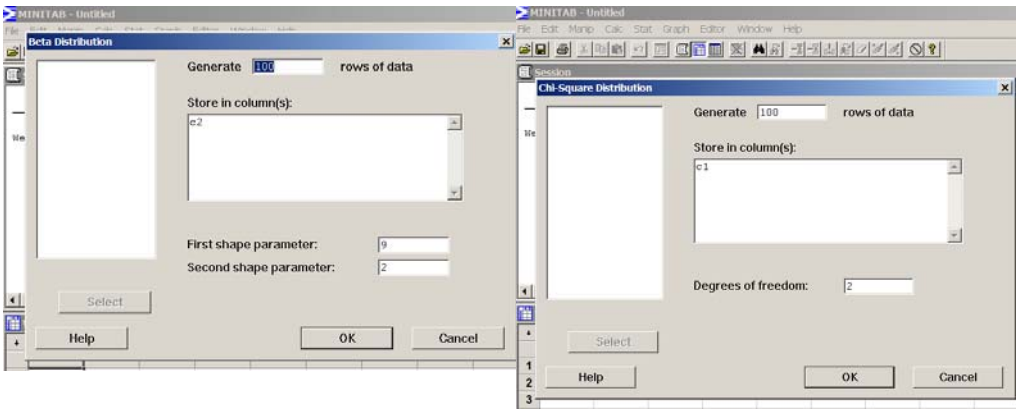


	Depth	Lower	Upper	Mid	Spread
N=	143				
M	72,0	61,000		61,000	
H	36,5	57,000	67,250	62,125	10,250
E	18,5	50,250	72,000	61,125	21,750
D	9,5	45,500	75,000	60,250	29,500
C	5,0	41,000	78,000	59,500	37,000
B	3,0	40,000	79,000	59,500	39,000
A	2,0	37,000	79,500	58,250	42,500

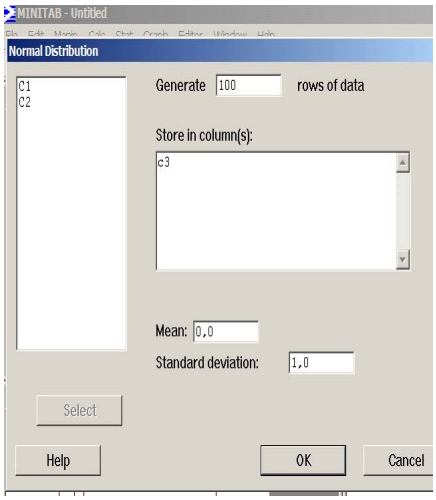
ASIMETRÍA Y CURTOSIS

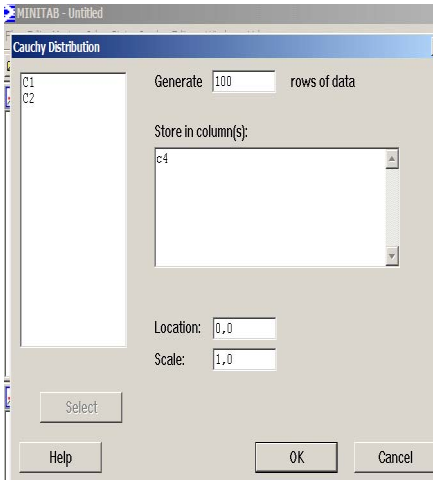


ASIMETRÍA Y CURTOSIS

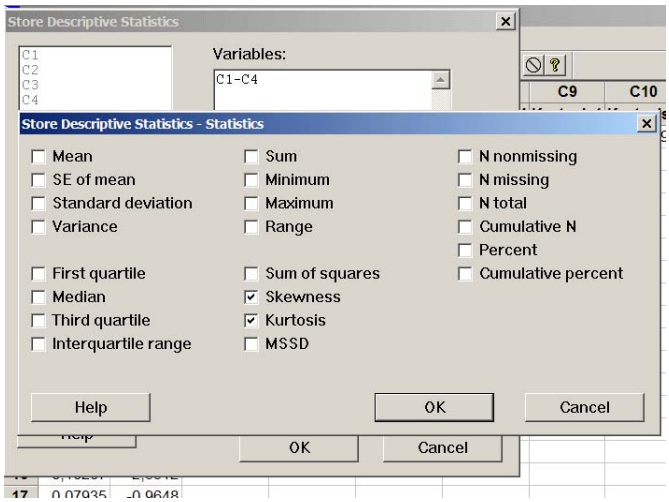


ASIMETRÍA Y CURTOSIS





ASIMETRÍA Y CURTOSIS



R

R: Resumen de los cinco números

```
#ej1 n=20
fivenum(c(680,685,685,700,700,705,705,715,730,730,735,740,745,750,755,780,780,785,800,800))

quantile(c(680,685,685,700,700,705,705,715,730,730,735,740,745,750,755,780,780,785,800,800))

#ej2 n=21
fivenum(c(680,685,685,700,700,705,705,715,730,730,735,740,745,750,755,780,780,785,800,800,801))

quantile(c(680,685,685,700,700,705,705,715,730,730,735,740,745,750,755,780,780,785,800,800,801))
```

Si se generan observaciones, mediante simulación, de los diferentes modelos cuyas funciones de densidad se representaron anteriormente, se obtendrán los diferentes valores de las medidas de asimetría y aplastamiento según su forma.

```
# generar 100 observaciones de Chi2 y 100 de Beta(9,2)
m.chisq<-rchisq(100,2)
m.beta<-rbeta(100,9,2)
m.norm<-rnorm(100)
m.cau<-rcauchy(100)
summary(m.chisq)
summary(m.beta)
summary(m.norm)
summary(m.cau)
```

ASIMETRÍA Y CURTOSIS

5.2. Métodos gráficos

5.2.1. Variables cualitativas

- **Diagramas de barras o rectángulos con la misma base** (barplot)

la altura proporcional a la frecuencia de la categoría

- **Diagramas de sectores** (pie charts)

el ángulo proporcional a la frecuencia de la categoría

- **Pictogramas, Cartogramas, . . .**

el tamaño de la figura o de la zona sombreada proporcional a la frecuencia de la categoría

5.2.2. Variables cuantitativas

- **Histogramas** (de rectángulos con la misma o diferente amplitud de base)

el área del rectángulo proporcional a la frecuencia de la categoría

- **Diagramas de puntos** (dotplot)

las observaciones $\longrightarrow$ puntos

- **Diagramas de caja** (boxplot)

la caja limitada por Q_1 y Q_3 con una línea sobre M_e
los "bigotes" hasta el máximo y el mínimo de las observaciones si están dentro de $\begin{cases} Q_1 - \alpha(Q_3 - Q_1) \\ Q_3 + \alpha(Q_3 - Q_1) \end{cases}$
(α suele ser 1,5)
$*$ $\longrightarrow$ datos atípicos (outliers) fuera de ese rango

También se puede construir (Tukey) con las bisagras H_1 y H_2 y los límites

$$\begin{cases} F_1 = H_1 - 1{,}5(H_2 - H_1) \longrightarrow \text{ valla ("fence") inferior} \\ F_2 = H_2 + 1{,}5(H_2 - H_1) \longrightarrow \text{ valla ("fence") superior} \end{cases}$$

- **Diagramas de tallo y hojas** (stem and leaf plots)

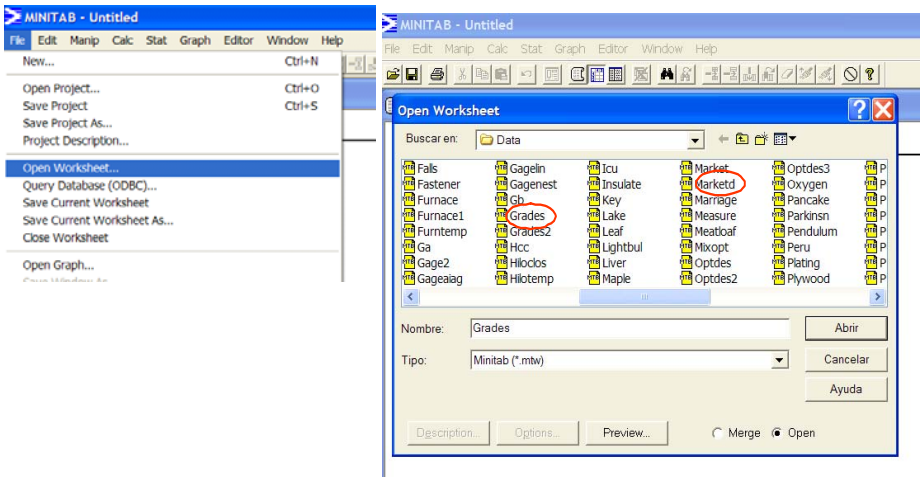
tallo $\longrightarrow$ primerdígito o primer bloque de dígitos
hojas $\longrightarrow$ resto de dígitos hasta completar la observación

5.2.3. Ejercicios

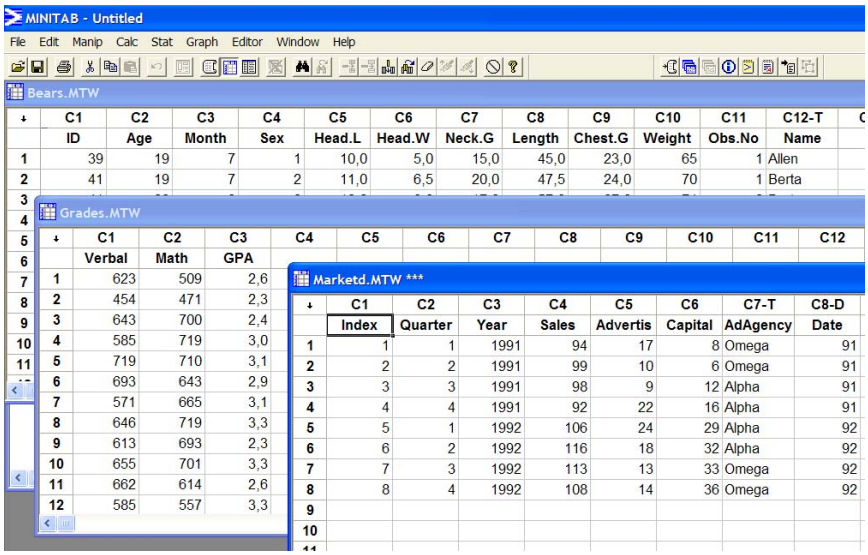
MINITAB

MINITAB: Gráficos

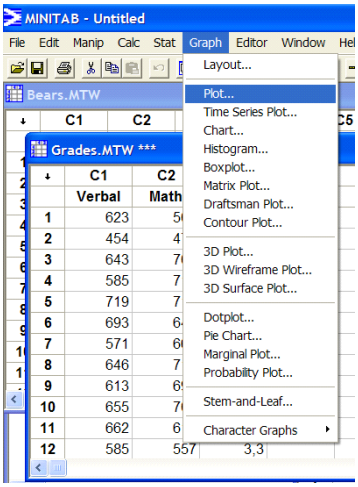
Abrir bases de datos: Grades, Marketd y Bears

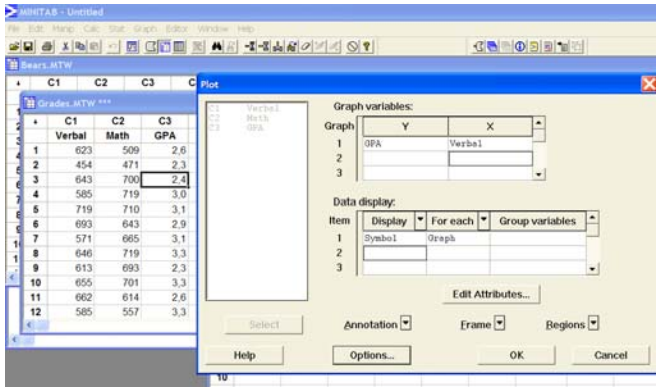


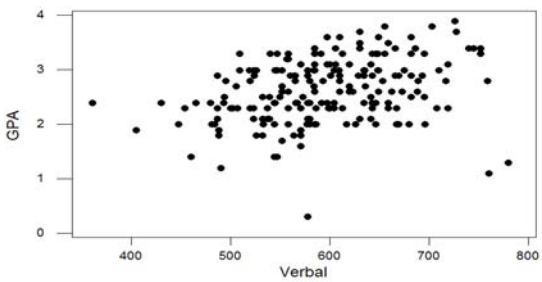
MINITAB: Gráficos



MINITAB: "PLOT"



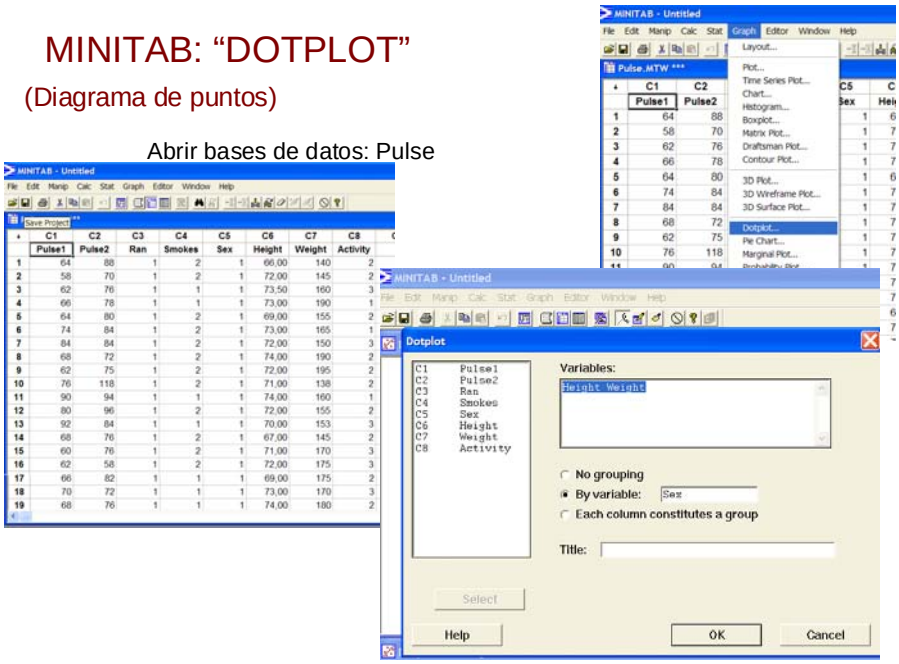




MINITAB: “DOTPLOT”

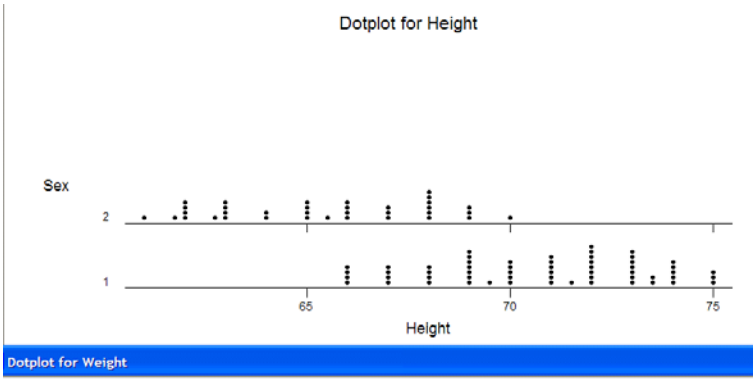
(Diagrama de puntos)

Abrir bases de datos: Pulse

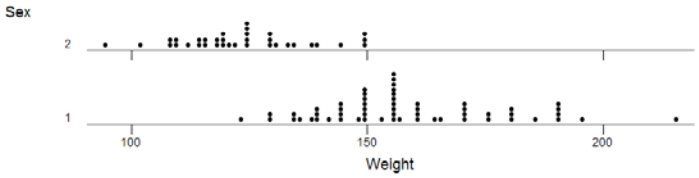


MINITAB: “DOTPLOT”

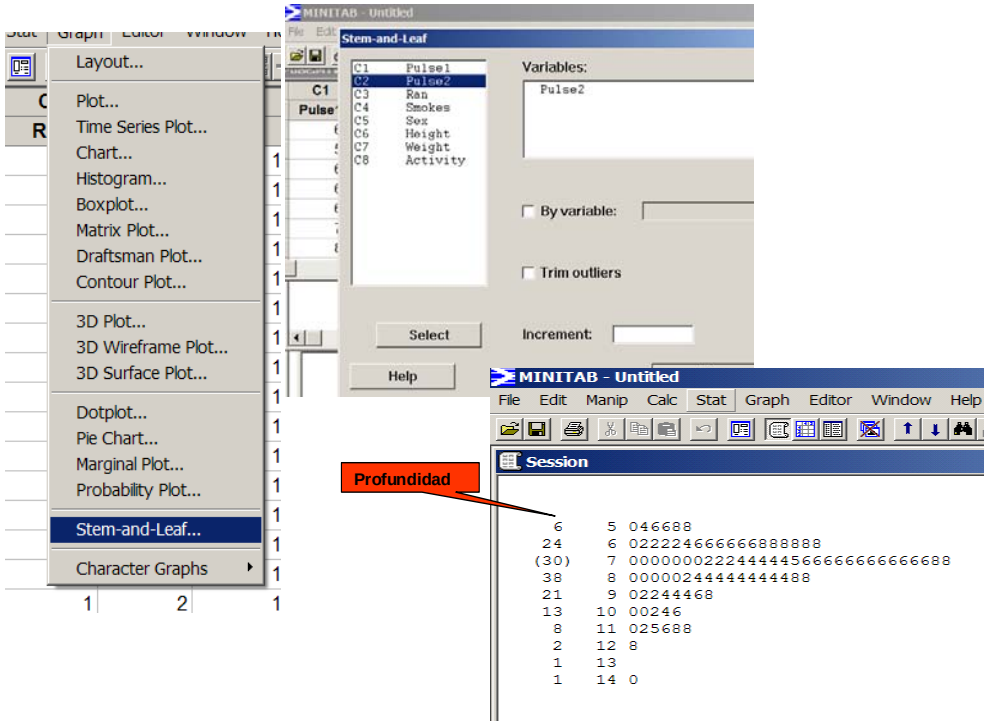
(Diagrama de puntos)



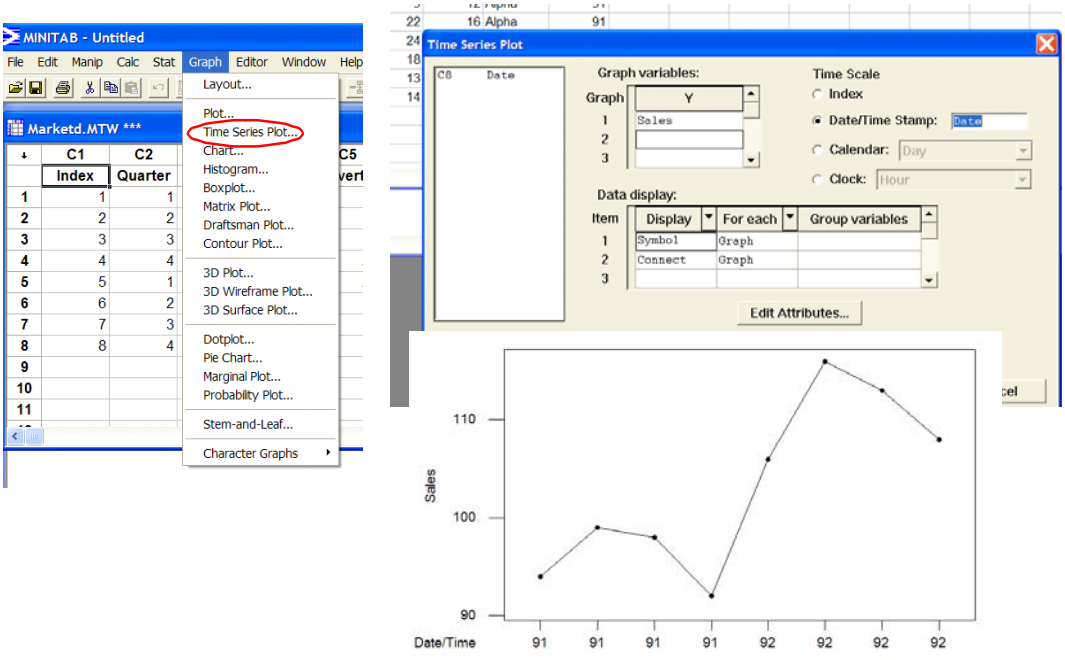
Dotplot for Weight



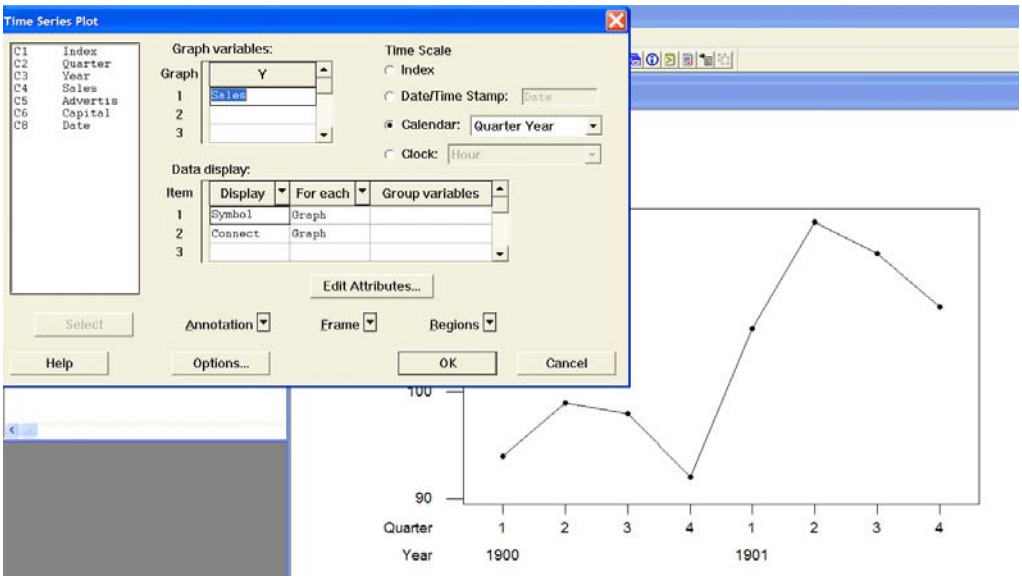
MINITAB: “STEM-AND-LEAF” (Diagrama de tallo y hojas)



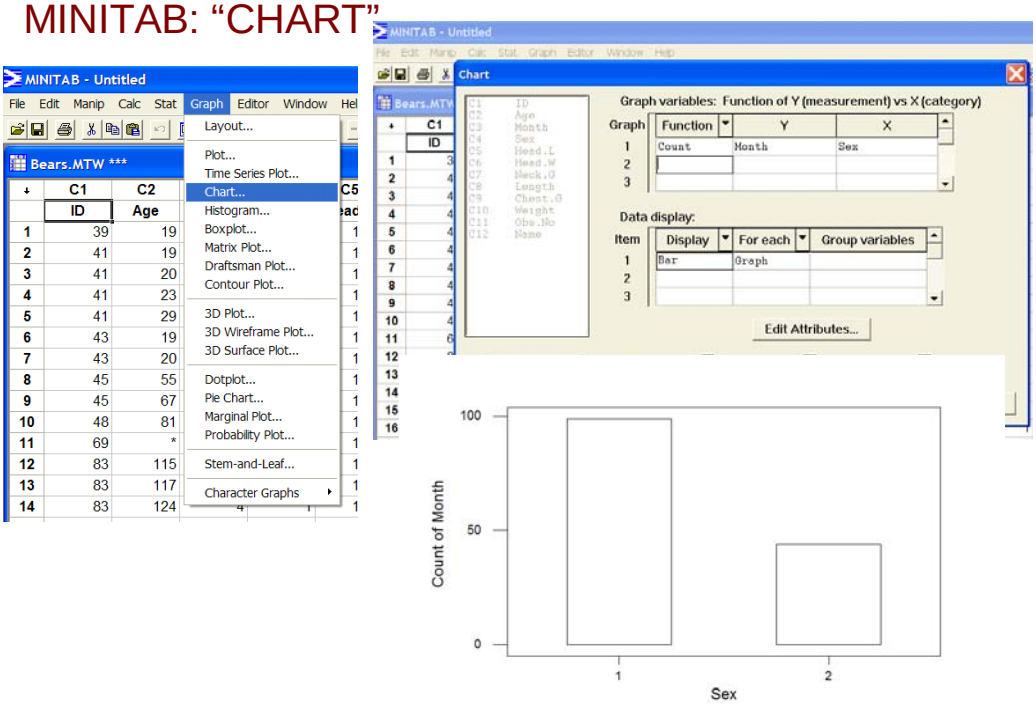
MINITAB: “TIME SERIES PLOT”

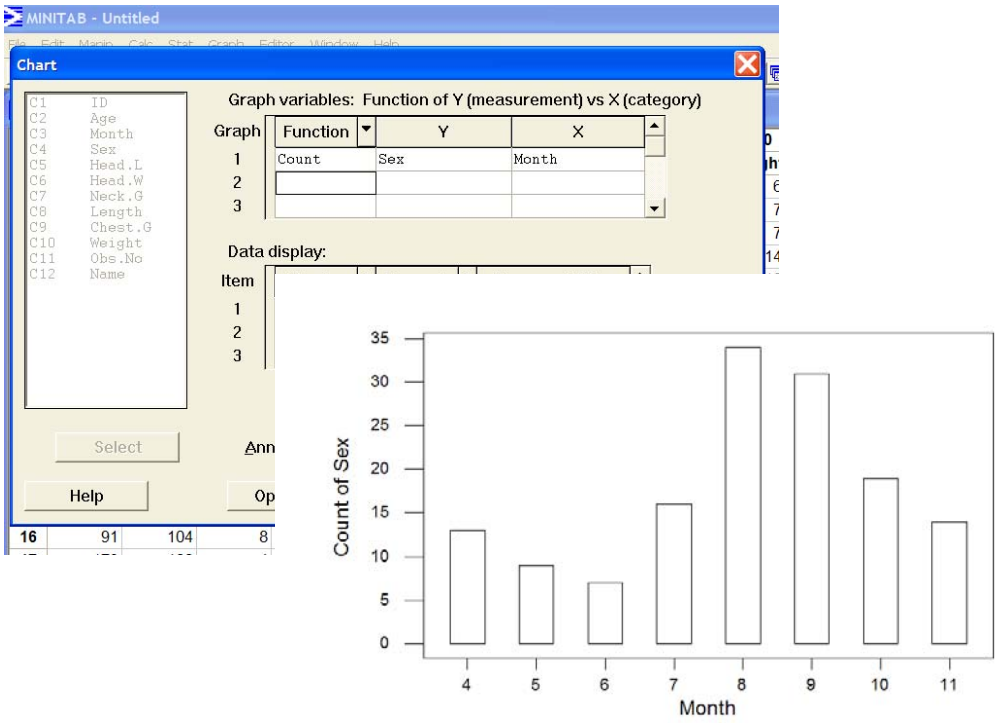


MINITAB: “TIME SERIES PLOT”

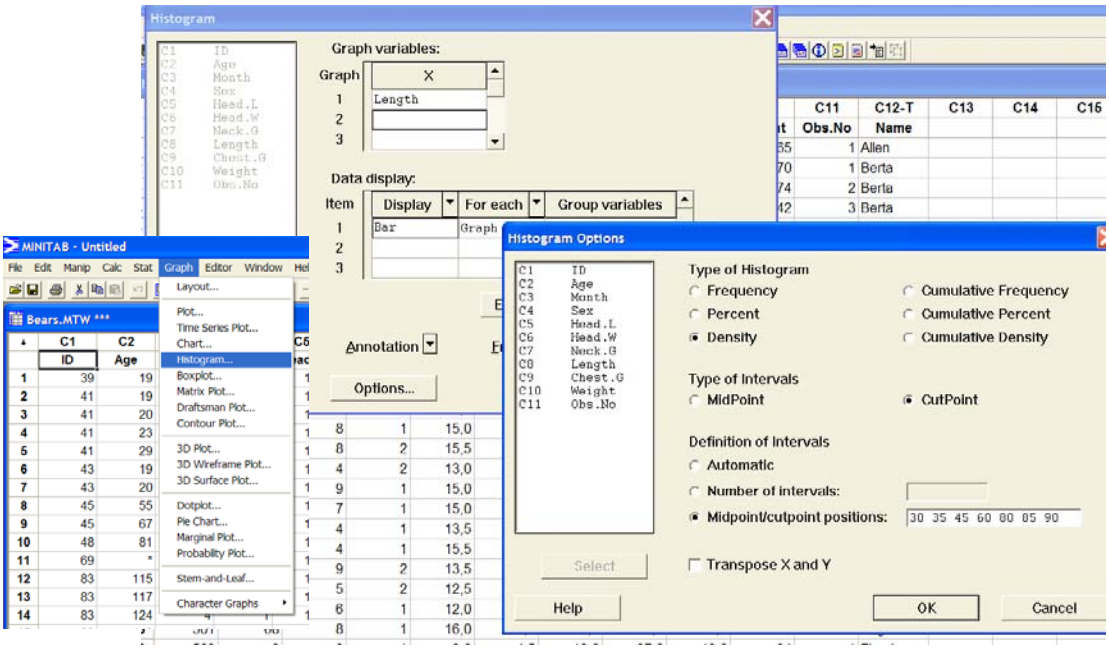


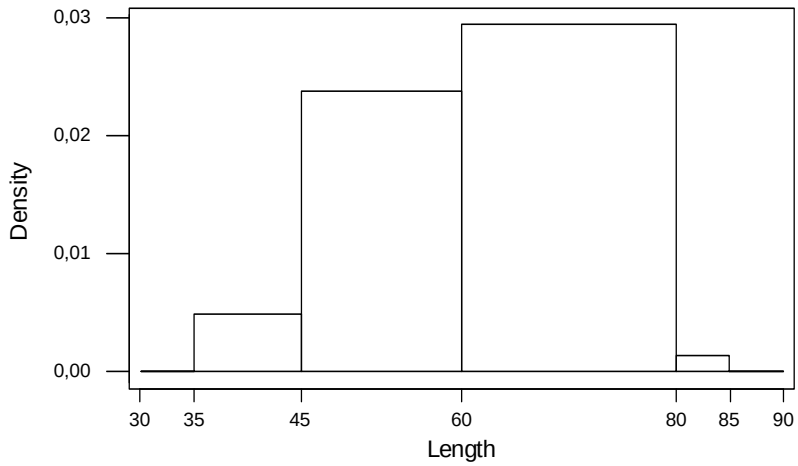
MINITAB: “CHART”





MINITAB: "HISTOGRAM"





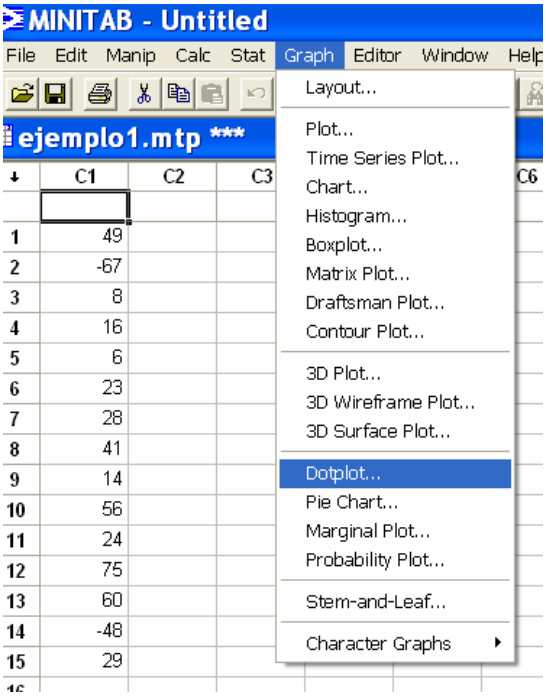
Ejemplo 1
http://www.mat.ucm.es/~palomam/aed/datos/ejemplo1.dat

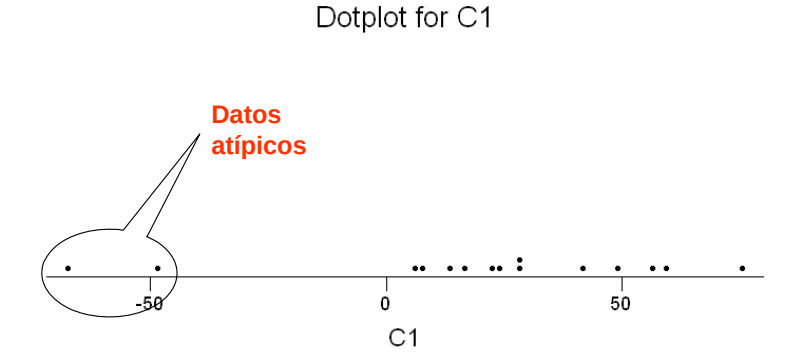
```
49 -67 8 16 6 23 28 41 14 56
24 75 60 -48 29
```

Ejemplo 2
http://www.mat.ucm.es/~palomam/aed/datos/ejemplo2.dat

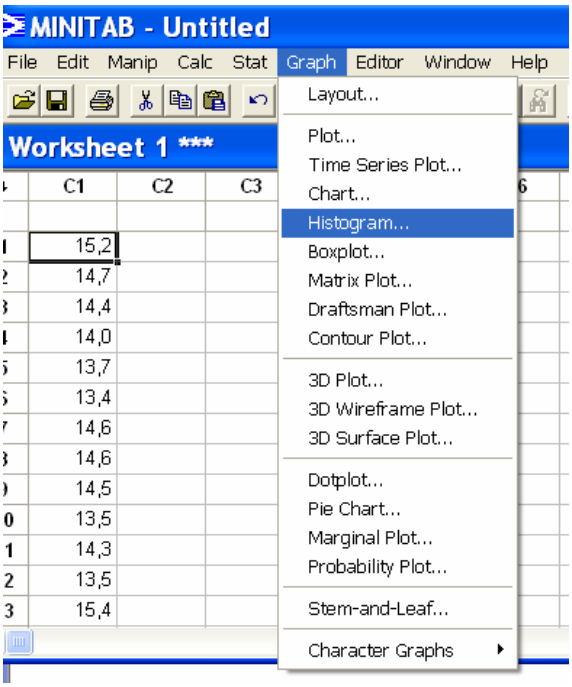
```
15.2   14.7   14.4   14.0   13.7   13.4   14.6   14.6   14.5
13.5   14.3   13.5   15.4   14.6   13.7   13.5   14.2   13.8
13.8   13.9   13.7   13.5   13.0   13.7   14.4   14.0   13.7
13.5   13.0   15.0   14.2   14.0   14.6   15.8   13.5   15.1
```

MINITAB: Gráfico de puntos

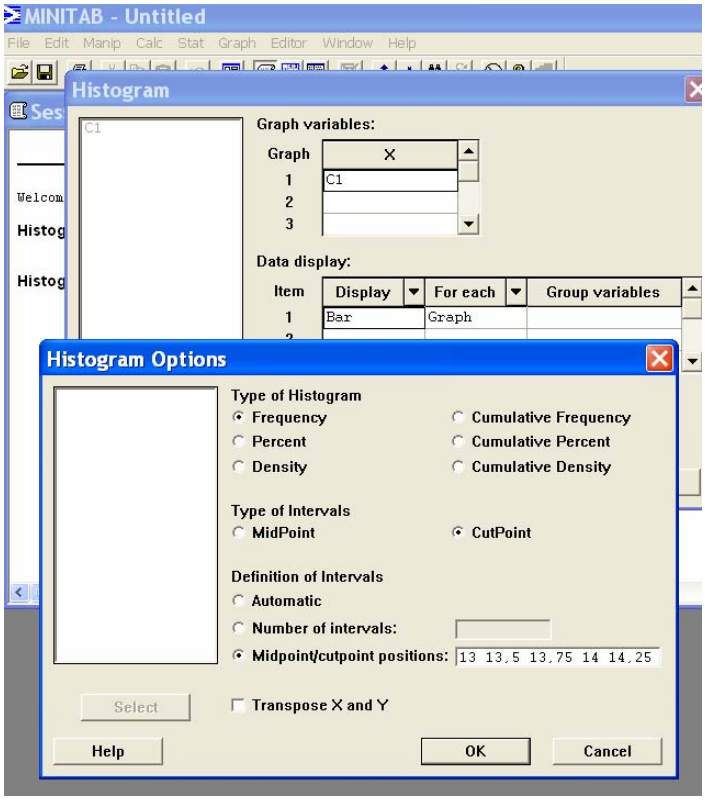




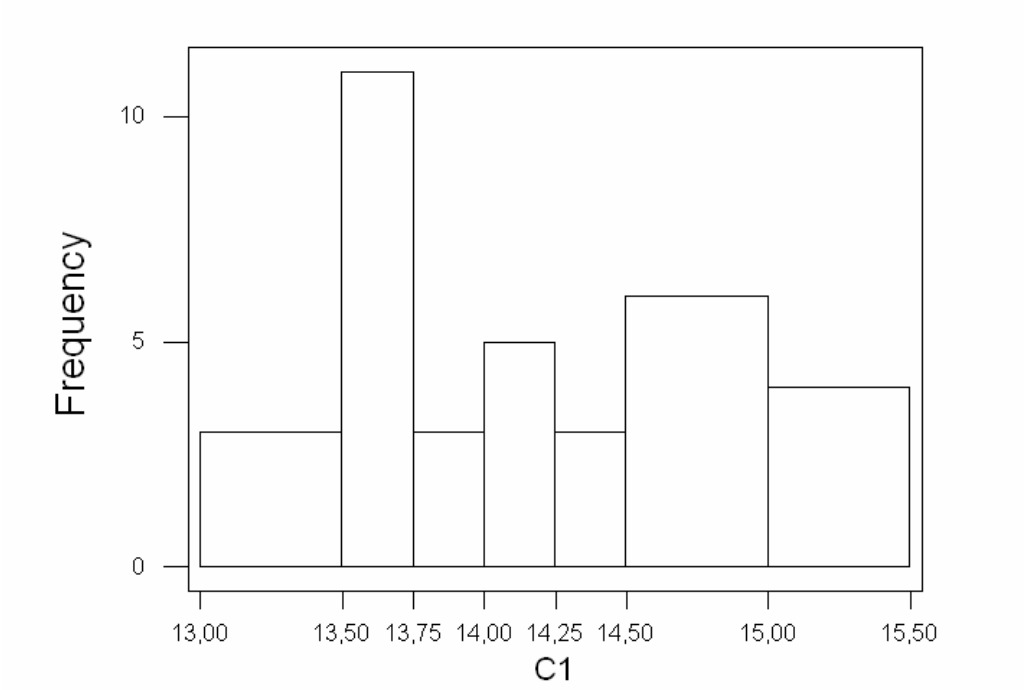
MINITAB: Histograma



MINITAB: Ventana diálogo de histograma



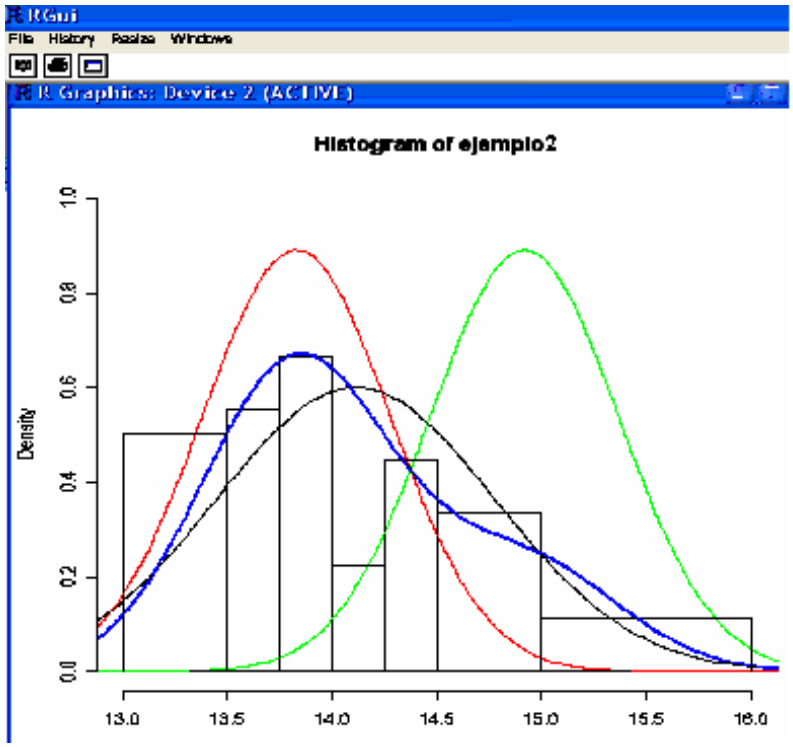
MINITAB: Graph →Histogram→Frame→Tick →Labels(0,75)



Funciones de R

```
#ejemplo2
ejemplo2<-
scan(‘‘http://www.mat.ucm.es/~palomam/aed/datos/ejemplo2.dat’’)
#Tabla de frecuencias
ej2.c<-cut(ejemplo2,breaks=c(13,13.5,13.75,14,14.25,14.5,15,16)
,include.lowest =TRUE, right = FALSE)
class(ej2.c)
table(ej2.c)
class(table(ej2.c))
median(ejemplo2)
median(ej2.c)
median(table(ej2.c))
attributes(ej2.c)
#histograma
hist(ejemplo2)
hist(ejemplo2,freq=FALSE)
hist(ejemplo2,breaks=c(13,13.5,13.75,14,
14.25,14.5,15,16),ylim=c(0,1))
#Se pueden ajustar diferentes ‘‘normales’’
```

```
curve(dnorm(x,mean(ejemplo2),sd(ejemplo2)),add=TRUE)
curve(dnorm(x,13.828,0.4482),add=TRUE,col=‘‘red’’)
curve(dnorm(x,14.916,0.4482),add=TRUE,col=‘‘green’’)
#Finalmente se puede ajustar una ‘‘mixtura de normales’’
curve(0.74*dnorm(x,13.828,0.4482)+0.26*dnorm(x,14.916,0.4482)
,add=TRUE,col=‘‘blue’’,lwd=‘‘2’’)
```



Mediante simulación se pueden generar muestras de diferentes modelos, cuyas funciones de densidad se representaron anteriormente, y comparar con sus histogramas

ASIMETRÍA Y CURTOSIS

- R

```
# generar 100 observaciones de Chi2 y 100 de Beta(9,2)
m.chisq<-rchisq(100,2)
m.beta<-rbeta(100,9,2)
m.norm<-rnorm(100)
m.cau<-rcauchy(100)
summary(m.chisq)
summary(m.beta)
summary(m.norm)
summary(m.cau)


par(mfrow=c(2,2))
hist(m.chisq)
abline(v=mean(m.chisq),col="red")
abline(v=median(m.chisq),col="green")
hist(m.beta)
abline(v=mean(m.beta),col="red")
abline(v=median(m.beta),col="green")
hist(m.norm)
abline(v=mean(m.norm),col="red")
abline(v=median(m.norm),col="green")
hist(m.cau)
abline(v=mean(m.cau),col="red")
abline(v=median(m.cau),col="green")
```

RGui

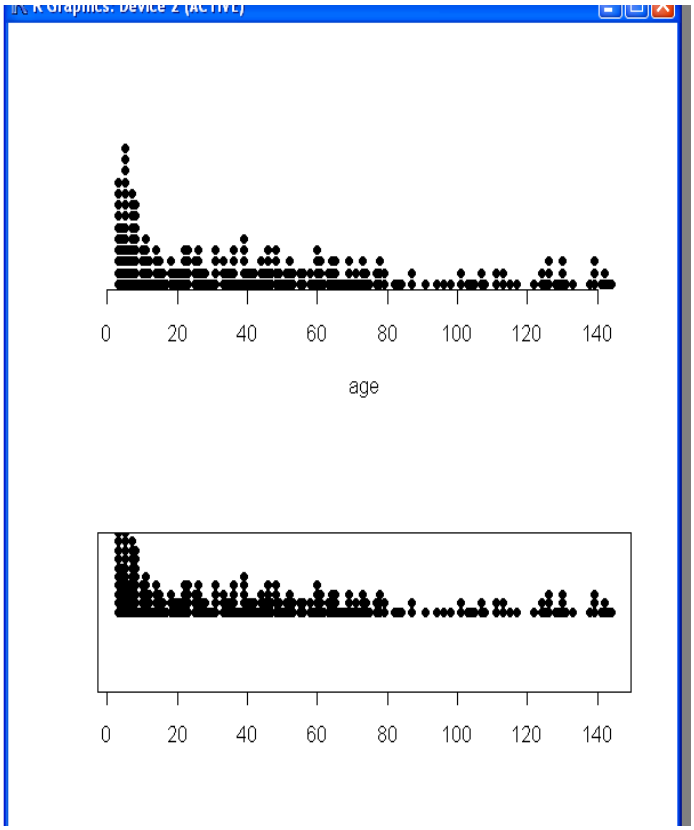
Archivo Editar Paquetes Ventanas Ayuda

R Console

```
 [26] "Floyd"
 [31] "Oliver"
 [36] "Pete"
 [41] "Tozia"
 [46] "Walt"
 [51] "Allison"
 [56] "Christophe"
 [61] "Gary"
 [66] "Ken"
 [71] "Quinn"
 [76] "U-Sam"
 [81] "Vanessa"
 [86] "Albert"
 [91] "Michele"
 [96] "Mary"
[101] "Davy"
[106] "Friday"
[111] "Ralph"
[116] "Ozz"
[121] "Curt"
[126] "Chet"
[131] "Larry"
[136] "Molly"
[141] "Curt"
>
```

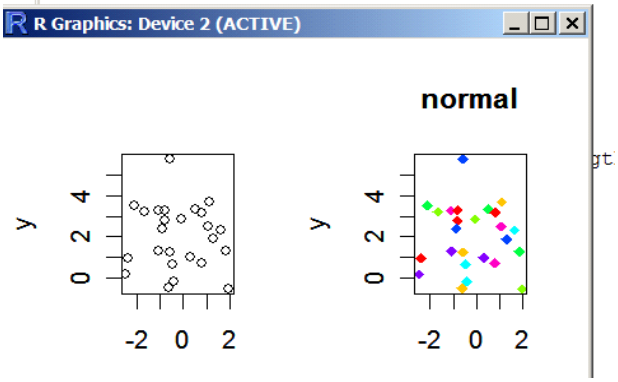
```
cat1<-scan("http://www.mat.ucm.es/~palomam/aed/datos/ososnom.txt")#error
cat1<-scan("http://www.mat.ucm.es/~palomam/aed/datos/ososnom.txt"
,what="character")
class(cat1)
tcat1<-table(cat1)

#cargar UsingR
library(UsingR)
central.park.cloud
class(central.park.cloud)
ta1<-table(central.park.cloud)

h1p2<-read.table("http://www.mat.ucm.es/~palomam/aed/datos/h1p2.txt")
class(h1p2)
cat2<-h1p2[,6]
class(cat2)
tcat2<-table(cat2)

barplot(tcat2)
par(mfrow=c(2,1))
barplot(tcat2,main="h1p2")
barplot(ta1)
barplot(tcat2,main="h1p2",ylim=c(0,30),xpd=F)
barplot(tcat2,main="h1p2",ylim=c(0,30))
barplot(VADeaths)
barplot(VADeaths,beside=T,horiz=T,space=c(0.1,0.5))#por defecto c(0,1)
barplot(VADeaths,beside=T,horiz=T,space=c(0.2,0.5)
,names.arg=c("RurH","RurM","UrbH","UrbM"),col=rainbow(5))

pie(tcat2,main="h1p2")
h1p2.nom<-c("bueno","malo","regular","excelente")
pie(tcat2,main="h1p2",labels=h1p2.nom)
pie(tcat2,main="h1p2",labels=c("bueno","malo","regular","excelente"))
pie(ta1)
```

```
dotchart(ta1)
dotchart(central.park.cloud) ¡error!
dotchart(cat2,pch=18,col=1:3)
attach(kid.weights)
kid3<-kid.weights[,3]
stripchart(kid3,pch=19)
stripchart(kid3,method="stack",pch=17)

kid4<-kid.weights[45<=age & weight>30,3]
#es lo mismo que
kid4<-height[45<=age & weight>30]
stripchart(kid4,method="stack")
stripchart(height,method="stack")

kid2<-kid.weights[,1]

par(mfrow=c(2,1))
DOTplot(age)
stripchart(age,method="stack",pch=19)
```



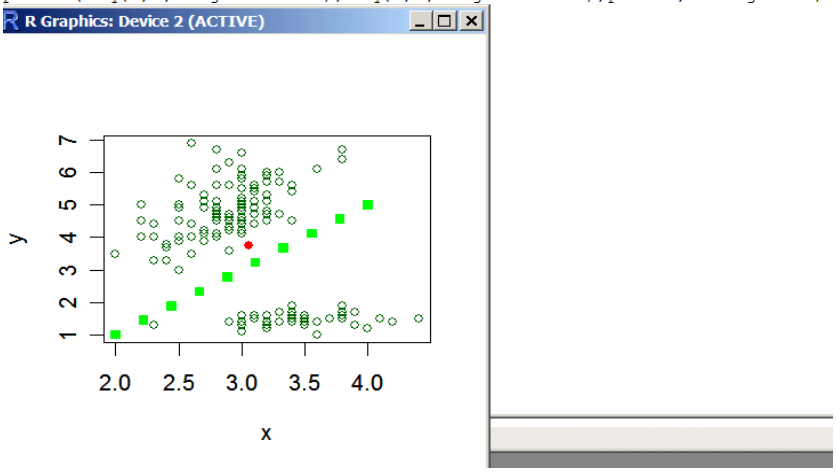
R: Añadir elementos a un gráfico

```
>
> par(mfrow=c(1,2))
> #cambiar colores y tipo de símbolo
> x<-rnorm(25)
> y<-rnorm(25,2,2)
> plot(x,y)
> plot(x,y,main="normal",pch=18,col=rainbow(8))
```



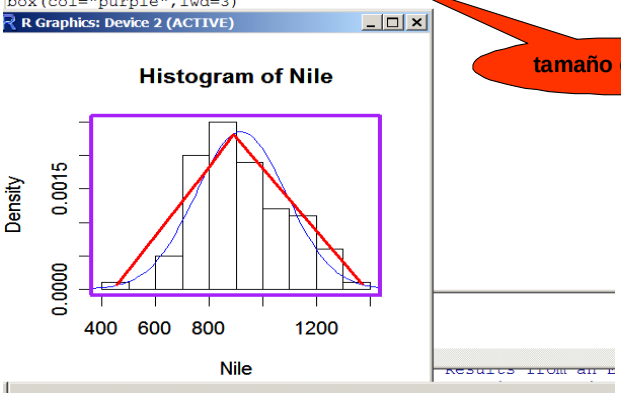
R: Añadir puntos

```
#puntos
x<-iris[,2]
y<-iris[,3]
plot(x,y,col="dark green")
points(mean(x),mean(y),col="red",pch=19)
points(seq(2,4,length.out=10),seq(1,5,length.out=10),pch=15,col="green")
```



R: Añadir líneas

```
#líneas
hist(Nile,freq=FALSE)
curve(dnorm(x,mean(Nile),sd(Nile)),col="blue",add=TRUE)
attributes(Nile)
v.Nile<-as.vector(Nile)
sv.Nile<-sort(v.Nile)
lines(sv.Nile[c(1,25,50,100)],dnorm(sv.Nile[c(1,25,50,100)]
,mean(v.Nile),sd(v.Nile)),col="red",lwd=2)
#caja
box(col="purple",lwd=3)
```

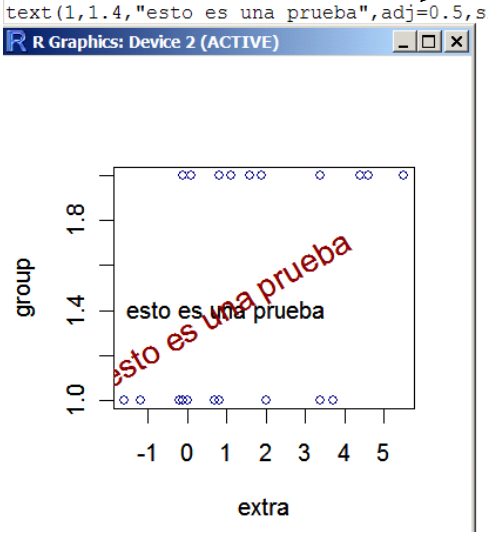


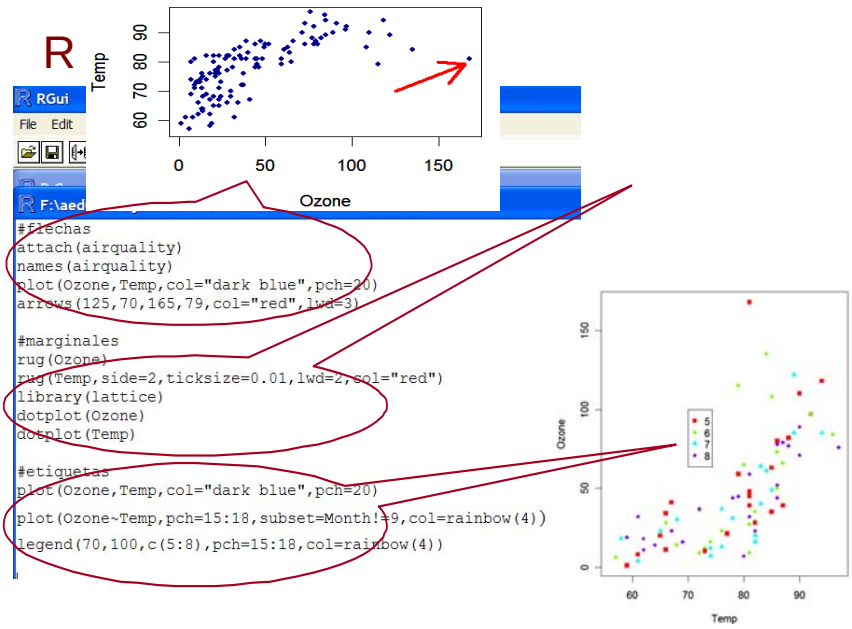
tamaño de línea

R: Añadir texto

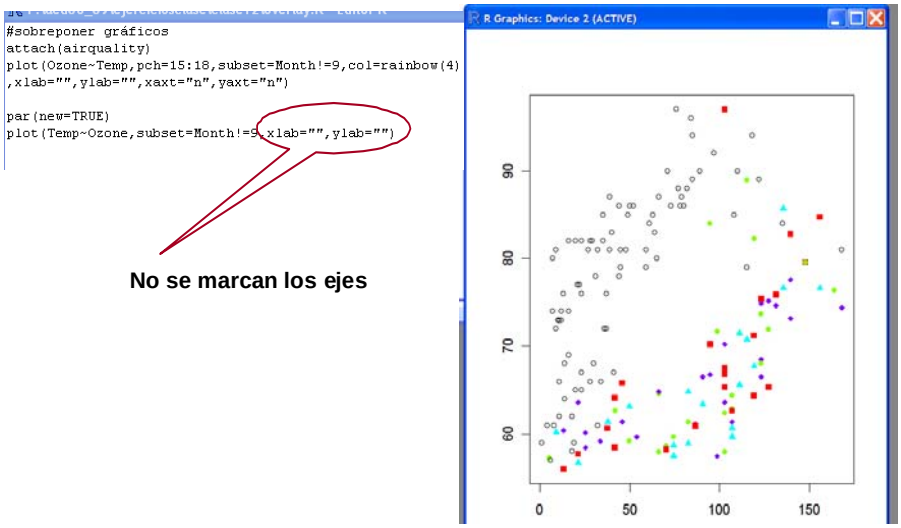
texto en el centro

```
#texto
plot(sleep,col="dark blue")
text(1,1.4,"esto es una prueba")
text(1,1.4,"esto es una prueba",adj=0.5,srt=30,col="dark red",cex=1.4)
```





R: Sobreponer gráficos



6. DESCRIPCIÓN DE DATOS MULTIVARIANTES

6.1. Métodos cuantitativos

6.1.1. Matriz de datos

$$\mathbf{X}_{n\times p} = \begin{pmatrix} X_{11} & \dots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \dots & X_{np} \\ \uparrow & \dots & \uparrow \\ variable\ Z_1 & & variable\ Z_p \end{pmatrix} = \begin{pmatrix} X_1' \\ \vdots \\ X_n' \end{pmatrix} \begin{matrix} \longleftarrow \text{ individuo } \mathbf{1} \\ \vdots \\ \longleftarrow \text{ individuo } \mathbf{n} \end{matrix}$$

6.1.2. Población (Modelo)

Variable aleatoria p-dimensional

$$Z = \begin{pmatrix} Z_1 \\ \vdots \\ Z_p \end{pmatrix}$$

$$\textbf{media}: E[Z] = \boldsymbol{\mu} = \begin{pmatrix} \mu_1 = E[Z_1] \\ \vdots \\ \mu_p = E[Z_p] \end{pmatrix}$$

$$\textbf{matriz de covarianzas}: Cov[Z] = \boldsymbol{\Sigma} = E\left[(Z - E[Z])(Z - E[Z])'\right] = \\ = E[ZZ'] - \boldsymbol{\mu}\boldsymbol{\mu}' = \\ = \begin{pmatrix} Var[Z_1] & \cdots & Cov[Z_i, Z_j] \\ & \ddots & \\ Cov[Z_i, Z_j] & & \\ & \cdots & Var[Z_p] \end{pmatrix}$$

donde $Cov[Z_i, Z_j] = E[(Z_i - E[Z_i])(Z_j - E[Z_j])] = E[Z_i Z_j] - E[Z_i]E[Z_j]$.

Subconjuntos de variables

Si se consideran los grupos de variables $Z = \begin{pmatrix} U \\ V \end{pmatrix}$ donde $U = \begin{pmatrix} Z_1 \\ \vdots \\ Z_r \end{pmatrix}$ y $V = \begin{pmatrix} Z_{r+1} \\ \vdots \\ Z_p \end{pmatrix}$ se tiene que

$$\textbf{media}: E[Z] = \begin{pmatrix} \boldsymbol{\mu}_U = E[U] \\ \boldsymbol{\mu}_V = E[V] \end{pmatrix}$$

$$\textbf{matriz de covarianzas}: Cov[Z] = \begin{pmatrix} \boldsymbol{\Sigma}_{UU} & \boldsymbol{\Sigma}_{UV} \\ \boldsymbol{\Sigma}_{VU} & \boldsymbol{\Sigma}_{VV} \end{pmatrix}$$

6.1.3. Estadísticos

Vector de medias:

$$\bar{\mathbf{X}} = \begin{pmatrix} \bar{X}_1 \\ \vdots \\ \bar{X}_p \end{pmatrix}$$

donde $\bar{X}_j = \frac{1}{n}\sum_{k=1}^{n} X_{kj} = \frac{1}{n}\begin{pmatrix} X_{1j} & \dots & X_{nj} \end{pmatrix} \underbrace{\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{1_n}$, por tanto

$$\bar{\mathbf{X}} = (1_n' 1_n)^{-1} \mathbf{X}' 1_n$$

Matriz de covarianzas muestral:

$$\mathbf{S} = \begin{pmatrix} S_{11} & \cdots & S_{ij} \\ & \ddots & \\ S_{ij} & \cdots & S_{pp} \end{pmatrix}$$

donde

$$\begin{aligned} S_{ij} &= S_{ji} = \frac{\sum_{k=1}^{n} \left(X_{ki} - \bar{X}_i\right)\left(X_{kj} - \bar{X}_j\right)}{n-1} = \frac{\left(\sum_{k=1}^{n} X_{ki}X_{kj}\right) - n\bar{X}_i\bar{X}_j}{n-1} \\ S_{ii} &= \frac{\sum_{k=1}^{n} \left(X_{ki} - \bar{X}_i\right)^2}{n-1} \end{aligned}$$

son las varianzas y covarianzas muestrales.

Por lo tanto

$$\mathbf{S} = \frac{1}{n-1} \sum_{k=1}^{n} \left(X_k - \bar{\mathbf{X}}\right)\left(X_k - \bar{\mathbf{X}}\right)'.$$

Además se puede demostrar que $E[\mathbf{S}] = \mathbf{\Sigma}$.

Cálculo a partir de la matriz de datos centrados

Sea

$$\mathbf{X}_C = \begin{pmatrix} X_{11} - \bar{X}_1 & \dots & X_{1p} - \bar{X}_p \\ \vdots & \ddots & \vdots \\ X_{n1} - \bar{X}_1 & \dots & X_{np} - \bar{X}_p \end{pmatrix}$$

entonces

$$\begin{aligned} \mathbf{X}_C &= \mathbf{X} - \mathbf{1}_n \bar{\mathbf{X}}' \\ \mathbf{S} &= \frac{1}{n-1} \mathbf{X}_C' \mathbf{X}_C. \end{aligned}$$

También $\mathbf{X}_C = \mathbf{P}\mathbf{X}$, donde $\mathbf{P} = \mathbf{I}_n - \mathbf{1}_n \left(1_n' 1_n\right)^{-1} 1_n'$ es una matriz simétrica e idempotente y así

$$\mathbf{S} = \frac{1}{n-1} \mathbf{X}'\mathbf{P}\mathbf{X}$$

Matriz de correlación muestral:

$$\mathbf{R} = \begin{pmatrix} 1 & \cdots & R_{ij} \\ & \ddots & \\ R_{ij} & & \\ & \cdots & 1 \end{pmatrix}$$

donde

$$R_{ij} = R_{ji} = \frac{S_{ij}}{\sqrt{S_{ii}S_{jj}}}$$

son los coeficientes de correlación muestral, $0 \leq R_{ij}^2 \leq 1$.

Si $\mathbf{D_S} = \begin{pmatrix} \sqrt{S_{11}} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sqrt{S_{pp}} \end{pmatrix}$ se tiene

$$\mathbf{R} = \mathbf{D_S^{-1}} \mathbf{S} \mathbf{D_S^{-1}}$$

$$\mathbf{S} = \mathbf{D_S} \mathbf{R} \mathbf{D_S}$$

Subconjuntos de variables

Si $X = \underbrace{(B_1}_{\substack{observaciones \\ de\ U}} \quad | \quad \underbrace{B_2)}_{\substack{observaciones \\ de\ V}}$ se tiene

$$\textbf{vector de medias: } \bar{\mathbf{X}} = \begin{pmatrix} \bar{X}_U \\ \bar{X}_V \end{pmatrix}$$

$$\textbf{matriz de covarianzas muestral: } \mathbf{S} = \begin{pmatrix} \mathbf{S}_{UU} & \mathbf{S}_{UV} \\ \mathbf{S}_{VU} & \mathbf{S}_{VV} \end{pmatrix}$$

6.1.4. Combinaciones lineales de variables

Supongamos que dada la variable aleatoria $Z = \begin{pmatrix} Z_1 \\ \vdots \\ Z_p \end{pmatrix}$ tal que $E[Z] = \boldsymbol{\mu}_Z$ y $Cov[Z] = \boldsymbol{\Sigma}_Z$, se define la variable aleatoria $Y = a_1 Z_1 + \ldots + a_p Z_p = a'Z$ para $a = \begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}$, entonces

$$\begin{aligned} E[Y] &= a' \boldsymbol{\mu}_Z \\ Cov[Y] &= a' \boldsymbol{\Sigma}_Z a. \end{aligned}$$

Las observaciones $\mathbf{X} = \begin{pmatrix} X_{11} & \ldots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \ldots & X_{np} \end{pmatrix}$ se transforman en $\mathbf{W} = \mathbf{X}a$ y por lo tanto

$$\text{media muestral: } \bar{w} = a' \bar{\mathbf{X}}$$

$$\text{varianza muestral: } s_w^2 = a' \mathbf{S}_X a.$$

En general sea $Y = AZ$ donde $A = \begin{pmatrix} a_{11} & \ldots & a_{1p} \\ \vdots & \ddots & \vdots \\ a_{q1} & \ldots & a_{qp} \end{pmatrix}$ entonces

$$E[Y] = A \boldsymbol{\mu}_Z$$

$$Cov\,[Y] = A\boldsymbol{\Sigma}_Z A'$$

En este caso, las observaciones pasan a ser $\mathbf{W} = \mathbf{X}A'$ por lo que

$$\text{vector de medias: } \bar{W} = A\bar{\mathbf{X}}$$

$$\text{matriz de covarianzas muestral: } \mathbf{S}_W = A\mathbf{S}_X A'.$$

6.1.5. Medidas de variabilidad alrededor de $\bar{\mathrm{X}}$

Sea la matriz de datos

$$\mathbf{X}_{n\times p} = \begin{pmatrix} X_{11} & \dots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{n1} & \dots & X_{np} \end{pmatrix} = \begin{pmatrix} X_1' \\ \vdots \\ X_n' \end{pmatrix}$$

con matriz de covarianzas $\mathbf{S}$, entonces las cantidades que suelen definirse para describir el comportamiento de la "nube de puntos" alrededor de su media, se basan en la evalución de distintas características de dicha matriz de covarianzas.

Varianza Total

$$VT = \text{suma(elementos de la diagonal de } \mathbf{S})$$

Varianza Media

$$VM = \frac{1}{p}\,\text{suma(elementos de la diagonal de } \mathbf{S})$$

Varianza Generalizada

$$VG = \text{determinante}(\mathbf{S})$$

Otra forma de tratar el problema conlleva la introducción de la distancia de Mahalanobis entre un punto x_i y la media $\bar{\mathbf{X}}$ definida por

$$\Delta^2\,(x_i, \bar{x}) = (x_i - \bar{x})'\ \mathbf{S}_x^{-1}\ \ (x_i - \bar{x})\,.$$

6.1.6. Coeficientes de asimetría y curtosis multivariantes

Sean $d_{ij} = (x_i - \bar{x})'\ \mathbf{S}_x^{-1}\ (x_j - \bar{x})$ para cualquier par de observaciones x_i y x_j. Se puede demostrar que

$$\Delta^2\,(x_i, x_j) = \Delta^2\,(x_i, \bar{x}) - 2d_{ij} + \Delta^2\,(x_j, \bar{x})\,.$$

Los coeficientes más utilizados se basan en estas cantidades calculadas para todas las observaciones.

Coeficiente de asimetría

$$\Delta_p = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^3$$

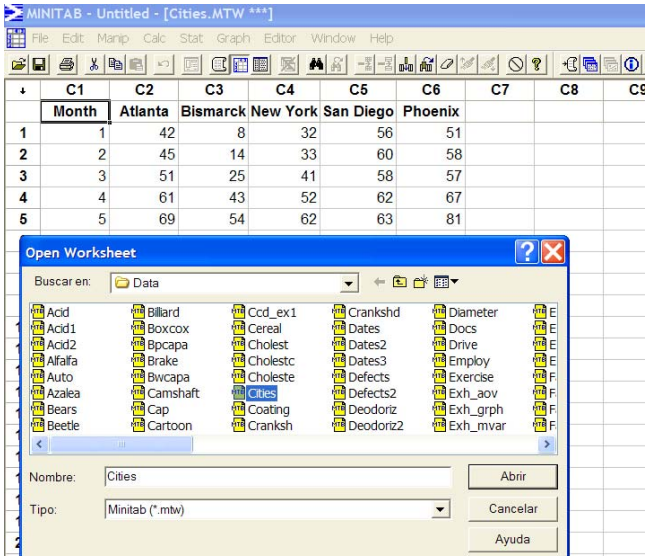
Coeficiente de curtosis

$$K_p = \frac{1}{n} \sum_{i=1}^{n} d_{ii}^2$$

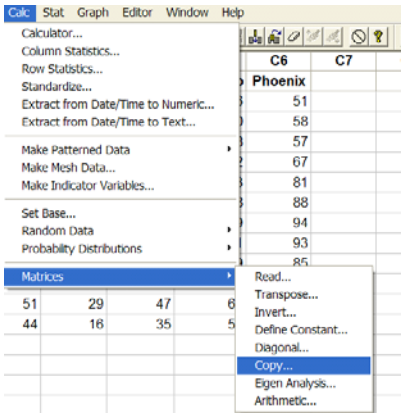
6.1.7. Manipulación de matrices con MINITAB

MINITAB : Matrices desde “Worksheet”

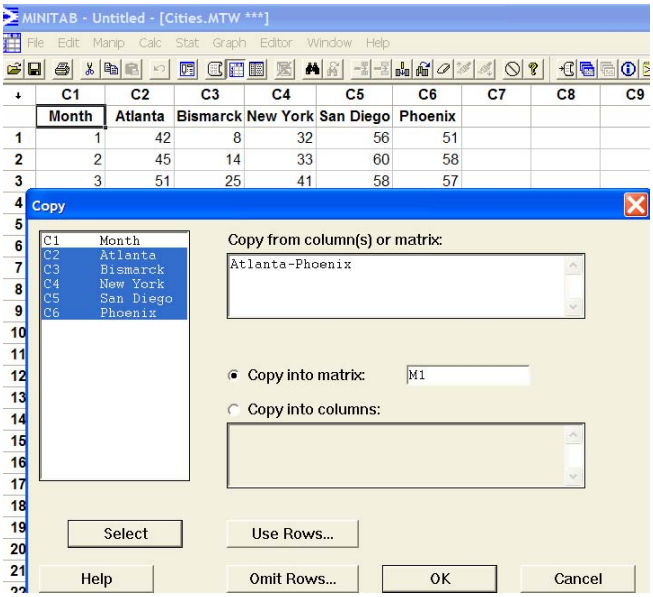
Datos Cities



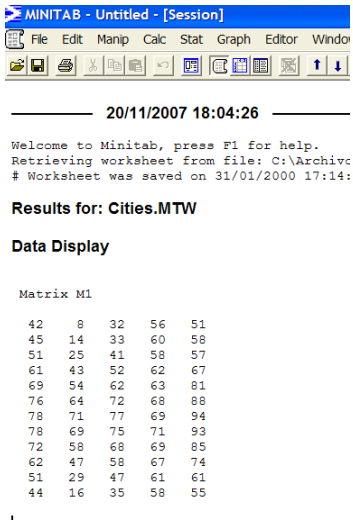
Construcción de la matriz M1 con las columnas C2 a C6



MINITAB : Matrices desde “Worksheet”

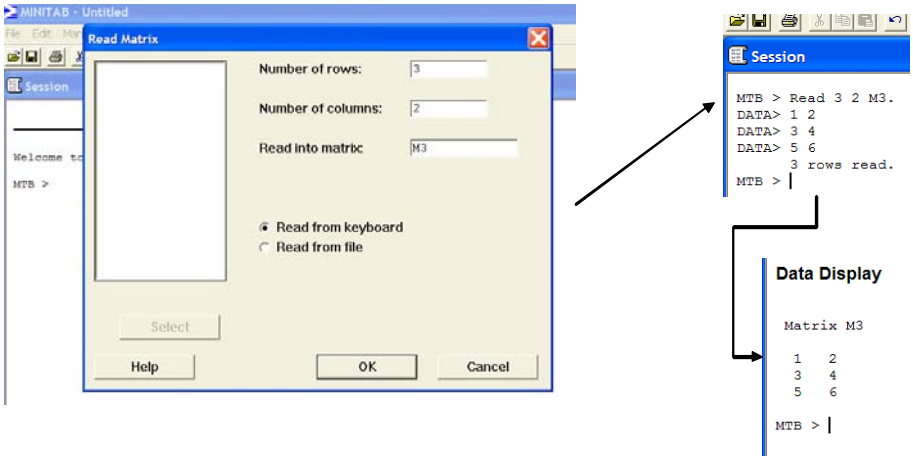


Manip→Display Data



MINITAB : Matrices desde “keyboard”

Dede ventana Session → Editor → Enable Commands

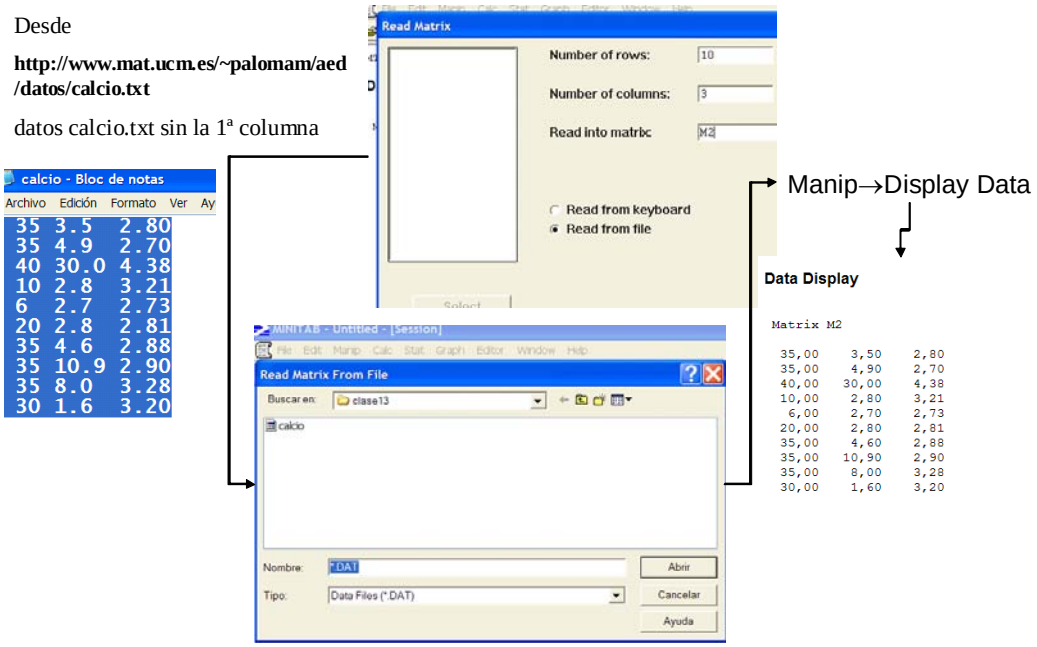


MINITAB : Matrices desde fichero

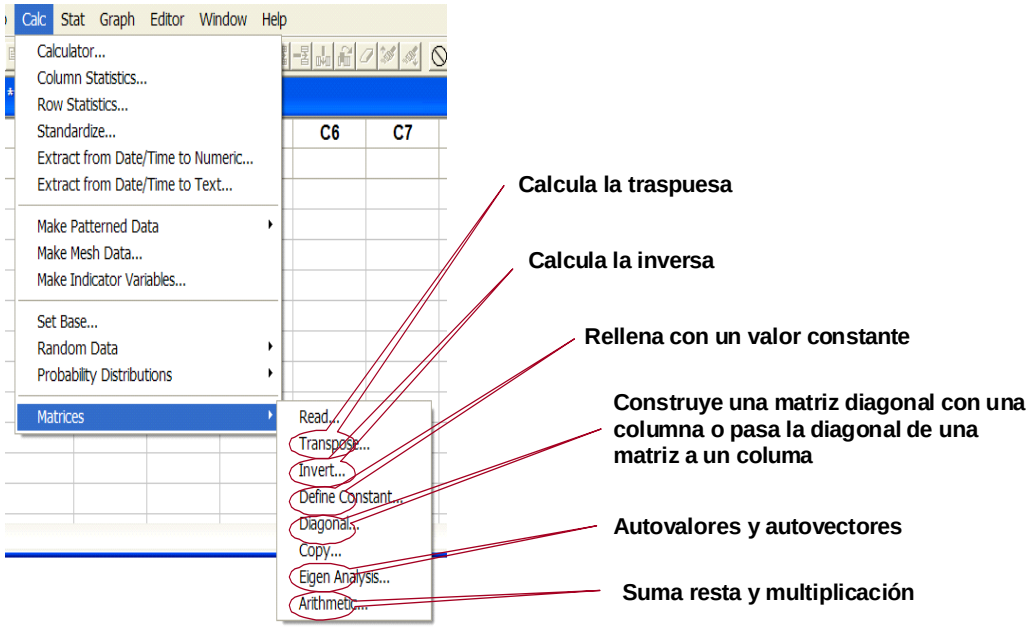
Desde

http://www.mat.ucm.es/~palomam/aed/datos/calcio.txt

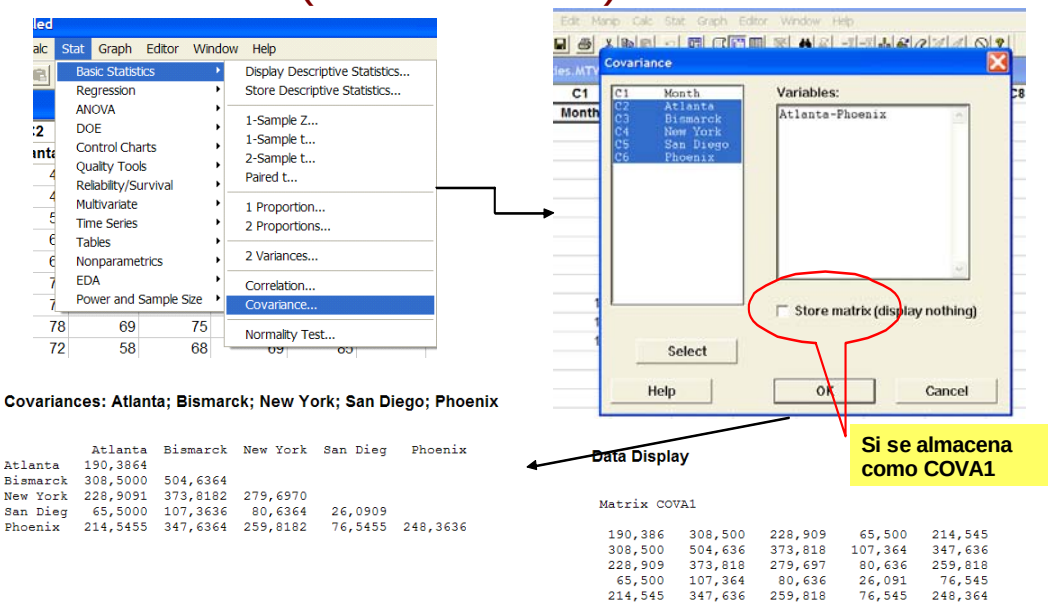
datos calcio.txt sin la 1ª columna



MINITAB : Operaciones con matrices



MINITAB: Vector de medias y matriz de covarianzas (Datos “Cities”)



6.1.8. Ejercicios con R

Ejercicio 1 : Con los datos Pulse de Minitab

http://www.mat.ucm.es/~palomam/aed/datos/pulse.txt

y utilizando Minitab y R, se pide:

- Seleccionar las variables Pulse 1, Pulse 2, Weight y Height
- Determinar las matrices de covarianzas y correlación
- Calcular el vector de medias a partir de la matriz de datos X
- Determinar X_C y P
- Comprobar que P es simétrica e idempotente
- Determinar la matriz de covarianzas a a partir de X_C

```
pulse<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/pulse.txt",
header=TRUE)

    #se elimina la primera columna
```

```
pulse<-pulse[,-1]
```

#se convierte en una matriz

```
m.pulse<-as.matrix(pulse)
```

#la media

```
pulse.mean<-apply(pulse,2,mean)
```

#la matriz de covarianzas

```
cov(pulse)
```

#la matriz de correlaciones

```
cor(pulse)
```

#la matriz de datos centrados

```
pulse.c<-scale(pulse,scale=FALSE)
class(pulse.c)
```

#cálculo de cov(pulse)

```
(1/(nrow(pulse)-1))*t(pulse.c)%*%pulse.c
```

#cálculo de la media

```
t(pulse)
t(pulse)%*%seq(1,1,length=nrow(pulse))
(1/nrow(pulse))*t(m.pulse)%*%seq(1,1,length=nrow(pulse))
```

#cálculo de la matriz P

```
P<-diag(nrow(pulse))-(1/nrow(pulse))*
seq(1,1,length=nrow(pulse))%*%t(seq(1,1,length=nrow(pulse)))
P%*%m.pulse
```

#Propiedades de P

```
(P%*%P)-P
t(P)-P
```

La salida de los resultados

```
> cov(pulse)
 Pulse2 Height Weight
Pulse2 292.197802 -8.928571 -68.41758
Height -8.928571 13.390409 68.18084
Weight -68.417582 68.180841 563.55901
> cor(pulse)
 Pulse2 Height Weight
Pulse2 1.0000000 -0.1427403 -0.1686008
Height -0.1427403 1.0000000 0.7848664
Weight -0.1686008 0.7848664 1.0000000
> (1/nrow(pulse))*t(m.pulse)%*%seq(1,1,length=nrow(pulse))
            [,1]
Pulse1 72.86957
Pulse2 80.00000
Height 68.71739
Weight 145.15217
>
```

Ejercicio 2 : Con los datos diab.txt

http://www.mat.ucm.es/~palomam/aed/datos/diab.txt

y utilizando Minitab y R se pide:

Determinar vector de medias y matriz de covarianzas

•Separar las variables 1 y 2 y determinar sus características

•Determinar, a partir de todos los datos, X_C

•Si interesa la variable $Z=1.2X_1+0.4X_2+0.1X_3+0.1X_4+1.2X_5$

determínese la media y la varianza muestral asociada a los datos

```
diab<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/diab.txt")
diab<-diab[,-1]
m.diab<-as.matrix(diab)
```

#la media

```
diab.mean<-apply(diab,2,mean)
```

#solo de las dos primeras columnas

```
diab.mean.i<-diab.mean[c(1,2)]
```

#la matriz de covarianzas

```
diab.cov<-cov(diab)
```

#solo de las dos primeras columnas

```
diab.cov.i<-diab.cov[c(1,2),c(1,2)]
```

#la matriz de correlaciones

```
cor(diab)
```

#los datos centrados

```
scale(diab,scale=FALSE)
```

$\#Z = 1.2X_1 + 0.4X_2 + 0.1X_3 + 0.1X_4 + 1.2X_5$

```
a<-c(1.2,0.4,0.1,0.1,1.2)
```

#la media

```
med.y<-t(a)%*%diab.mean
```

#la varianza

```
var.y<-t(a)%*%diab.cov%*%a
```

#en efecto

```
Y<-m.diab%*%a
mean(Y)
var(Y)
```

La salida de los resultados

```
> diab.mean
        V2          V3          V4          V5          V6
  0.917826  90.413043 340.826087 171.369565  97.782609
> diab.mean.i
       V2        V3
 0.917826 90.413043
> diab.cov
            V2          V3           V4           V5          V6
V2  0.01618184   0.216029    0.7871691   -0.2138454    2.189072
V3  0.21602899  70.558937   26.2289855  -23.9560386  -20.841546
V4  0.78716908  26.228986 1106.4135266  396.7323671  108.383575
V5 -0.21384541 -23.956039  396.7323671 2381.8826087 1142.637681
V6  2.18907246 -20.841546  108.3835749 1142.6376812 2136.396135
> diab.cov.i
           V2        V3
V2 0.01618184  0.216029
V3 0.21602899 70.558937
```

```
> cor(diab)
            V2          V3        V4          V5          V6
V2  1.00000000  0.20217252 0.1860353 -0.03444497  0.37231056
V3  0.20217252  1.00000000 0.0938743 -0.05843578 -0.05368006
V4  0.18603532  0.09387431 1.0000000  0.24438735  0.07049590
V5 -0.03444497 -0.05843578 0.2443873  1.00000000  0.50653268
V6  0.37231056 -0.05368006 0.0704959  0.50653268  1.00000000
> med.y
         [,1]
[1,] 205.8253
> var.y
         [,1]
[1,] 3417.609
> mean(Y)
[1] 205.8253
> var(Y)
         [,1]
[1,] 3417.609
>
```

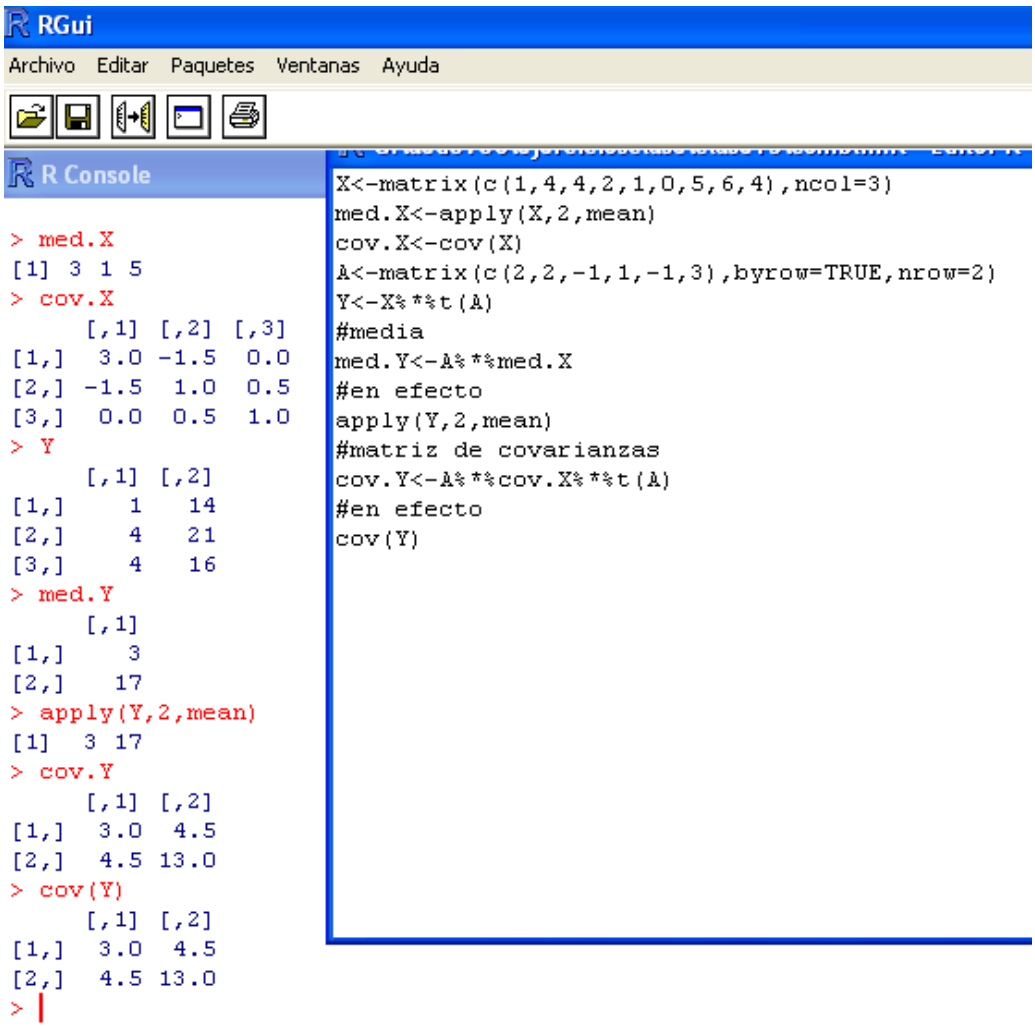
Ejercicio 3 : Sea $\mathbf{Z} = \begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \end{pmatrix}$ de la que se obtienen los datos

$$\mathbf{X} = \begin{pmatrix} 1 & 2 & 5 \\ 4 & 1 & 6 \\ 4 & 0 & 4 \end{pmatrix}$$

y sean $W_1 = 2Z_1 + 2Z_2 - Z_3$

$W_2 = Z_1 - Z_2 + 3Z_3$

Se pide: la nueva matriz de datos, vector de medias y matriz de covarianzas



6.1.9. Estudio de la variabilidad

Se pueden comparar las medidas introducidas anteriormente o utilizar el concepto de distancia estadística (Mahalanobis).

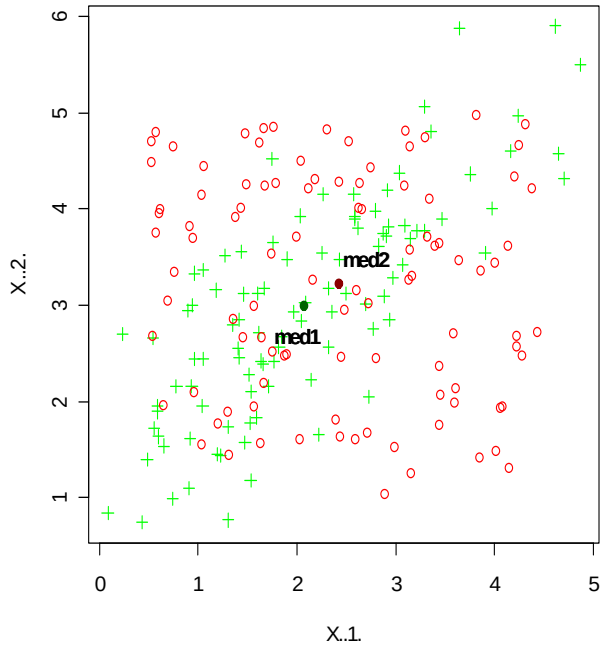
EJEMPLO

Con los datos

"http://www.mat.ucm.es/~palomam/aed/datos/dat1.txt" (cruces verdes)
"http://www.mat.ucm.es/~palomam/aed/datos/dat2.txt" (círculos rojos)
se obtienen dos tipos de "nubes de puntos" que pueden representarse con *R*.

```
dat1<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/dat1.txt")
plot(dat1,col="green",pch=3)
dat2<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/dat2.txt")
points(dat2[,1],dat2[,2],col="red")
med1<-apply(dat1,2,mean)
med2<-apply(dat2,2,mean)
points(rbind(med1,med2),col=c("dark green","dark red"),pch=19)
text(2,2.7,"med1",font=2)
text(2.7,3.5,"med2",font=2)
```

Obteniéndose el siguiente gráfico



y las siguientes características

```
cov1<-cov(dat1)
cov2<-cov(dat2)
cor1<-cor(dat1)
cor2<-cor(dat2)
#Varianza total
VT1=sum(diag(cov1))
VT2=sum(diag(cov2))
#Varianza media
VM1=(1/ncol(dat1))*sum(diag(cov1))
VM2=(1/ncol(dat2))*sum(diag(cov2))
#Varianza generalizada
VG1=det(cov1)
VG2=det(cov2)
```

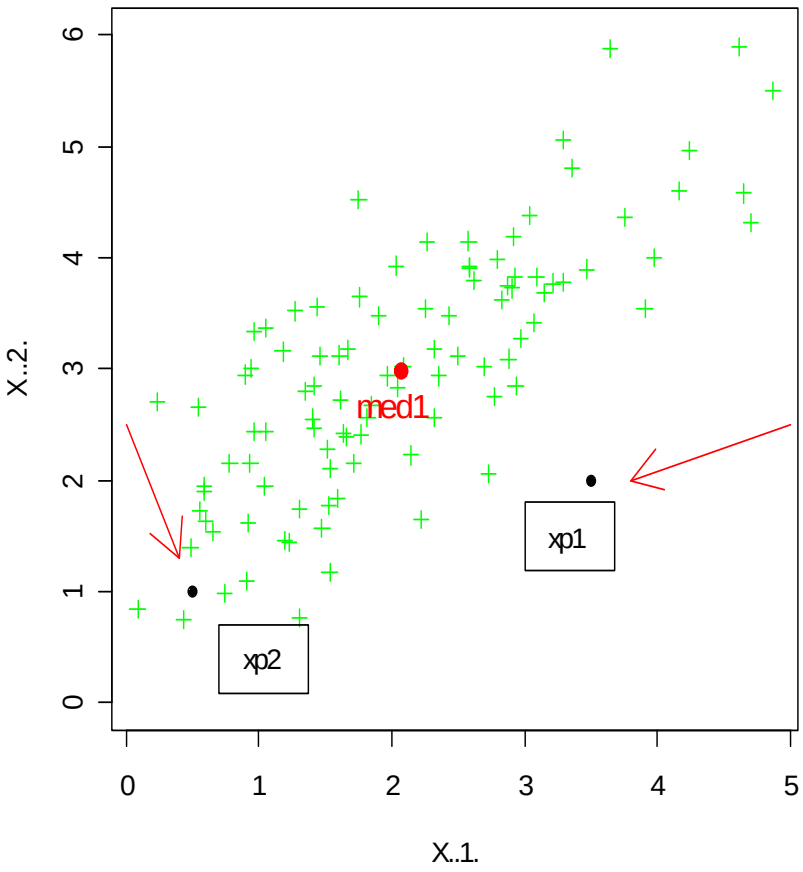
que dan como salida

```
> cov1
       X..1.     X..2.
X..1. 1.2286660 0.9760124
X..2. 0.9760124 1.2610625
> cov2
       x.2       y.2
x.2 1.3551807 -0.1775550
y.2 -0.1775550 1.2930154
> cor1
       X..1.     X..2.
X..1. 1.0000000 0.7840976
X..2. 0.7840976 1.0000000
> cor2
       x.2       y.2
x.2 1.0000000 -0.1341320
y.2 -0.1341320 1.0000000
> VT1
[1] 2.489729
> VT2
[1] 2.648196
> VM1
[1] 1.244864
> VM2
[1] 1.324098
> VG1
[1] 0.5968244
> VG2
[1] 1.720744
```

Distancias entre los puntos y las medias

Consideramos la posición de los puntos $xp1$ y $xp2$ respecto de la media de los datos *dat1.txt*.

```
#Distancias a la media
plot(dat1,col="dark blue",ylim=c(0,6))
points(med1[1],med1[2],col="red",pch=19,cex=1.2)
text(2,2.7,"med1",col="red",cex=1.2)
xp1<-c(3.5,2)
points(xp1[1],xp1[2],pch=20,cex=1.2)
arrows(5,2.5,3.8,2,col="red")
legend(3,1.8,"xp1",adj=0.5)
xp2<-c(0.5,1)
points(xp2[1],xp2[2],pch=20,cex=1.2)
arrows(0,2.5,0.4,1.3,col="red")
legend(0.7,0.7,"xp2",adj=0.5)
```



que puede cuantificarse mediante la evaluación de la "distancia de Mahalanobis" y la "distancia euclídea".

```
dist<-function(x,med)
{
dist.mah<-sqrt(t(x-med)%*%solve(cov1)%*%(x-med))
print("distancia de Mahalanobis")
print(dist.mah)
dist.euc<-sqrt(t(x-med)%*%(x-med))
print("distancia euclidea")
print(dist.euc)
}
dist(xp1,med1)
dist(xp2,med1)
```

que efectivamente reflejan la posición de $xp2$ más cercana a la "nube de puntos" (menor "distancia de Mahalanobis") que $xp1$, a pesar de estar más alejado de la media (mayor "distancia euclídea")

```
> dist(xp1,med1)
[1] "distancia de Mahalanobis"
 [,1]
[1,] 3.324378
[1] "distancia euclidea"
 [,1]
[1,] 1.74491
> dist(xp2,med1)
[1] "distancia de Mahalanobis"
 [,1]
[1,] 1.779564
[1] "distancia euclidea"
 [,1]
[1,] 2.540302
```

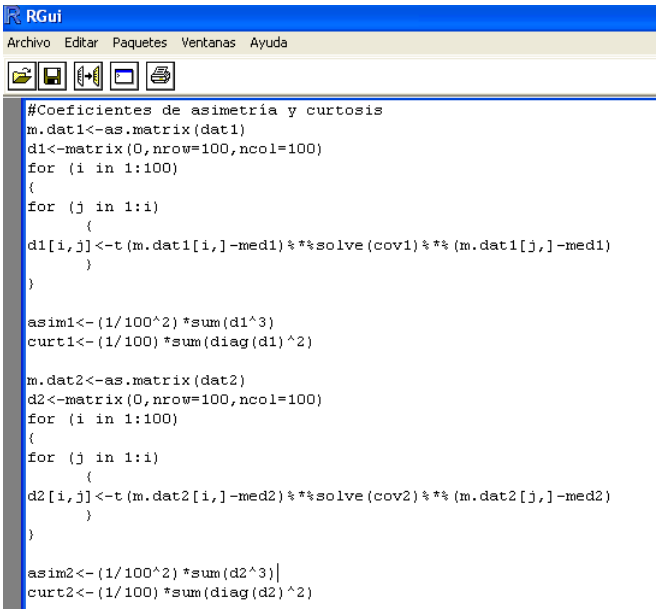
6.1.10. Comparación de asimetría y curtosis

A partir de los datos

"http://www.mat.ucm.es/~palomam/aed/datos/dat1.txt" (cruces verdes)
"http://www.mat.ucm.es/~palomam/aed/datos/dat2.txt" (círculos rojos)

se pueden comparar sus aspectos de asimetría alrededor la media y de curtosis.

```
> asim1
[1] 0.3265256
> curt1
[1] 7.116727
> asim2
[1] 0.1090851
> curt2
[1] 5.377523
>
```



6.2. Métodos gráficos

6.2.1. Gráficos para dos variables

- **Gráficos de dispersión o nubes de puntos** (scatterplots)
Cada observación un punto en el plano.
- **Matrices de gráficos de dispersión** (matrix plots)
Matrices bidimensionales compuestas de gráficos de dispersión individuales para cada par de variables.
- **Gráficos marginales**
Como complemento a los anteriores, con representación, en los bordes, de histogramas, diagramas de caja,... para cada variable.

6.2.2. Gráficos para tres variables

- **Gráficos de dispersión o nubes de puntos** (scatterplots)
Cada observación un punto en el espacio.
- **Gráficos de contorno** (contour plots)
Puntos representando las observaciones de dos de las variables para valores fijos de las observaciones de la tercera variable.
- **Gráficos de superficie y de líneas** (surface and wireframe plots)

Generan una red de polígonos formados por caras y lados que representan las observaciones de las tres variables.

6.2.3. Operaciones con los gráficos anteriores

- **Barrido** (brushing)
Permite señalar puntos y analizarlos por separado.
- **"Spinnig"** herramienta incorporada en cierto tipo de software (por ejemplo S-Plus), que permite girar los tres ejes para profundizar en las relaciones de las tres variables.

6.2.4. Perfiles Multivariantes

Se pretende recoger toda la información de la base de datos con más de dos columnas, en una figura de dos dimensiones.
- **Curvas de Fourier o de Andrews**
Método propuesto por Andrews y que consiste en representar cada observación p-variante $(x_1, \ldots, x_p)$ mediante la curva de Fourier

$$f(t) = \frac{x_1}{\sqrt{2}} + x_2 \sin t + x_3 \cos t + x_4 \sin 2t + x_5 \cos 2t + \ldots$$

representada en el intervalo $(-\pi, \pi)$
- **Caras de Chernoff**
Cada observación se representa por una cara donde el valor de cada variable representa la medida de un rasgo de la misma, por ejemplo tamaño de nariz, de boca..., posición de ojos,...
- **Estrellas**
Cada observación (de variables no negativas) se representa en dos dimensiones mediante círculos de radio fijo con p líneas emanando del centro.
La longitud de cada línea es el valor que tiene cada variable en dicha observación. Si se unen los extremos finales de los radios cada observación representada por una estrella.

6.2.5. Gráficos condicionados (Coplots)

Permiten la representación de dos variables condicionando por los valores de una tercera variable. Considerando distintas particiones del rango de la variable que condiciona se pueden obtener "nubes de puntos" de las otras dos variables, obteniéndose información sobre la relación entre las tres variables.

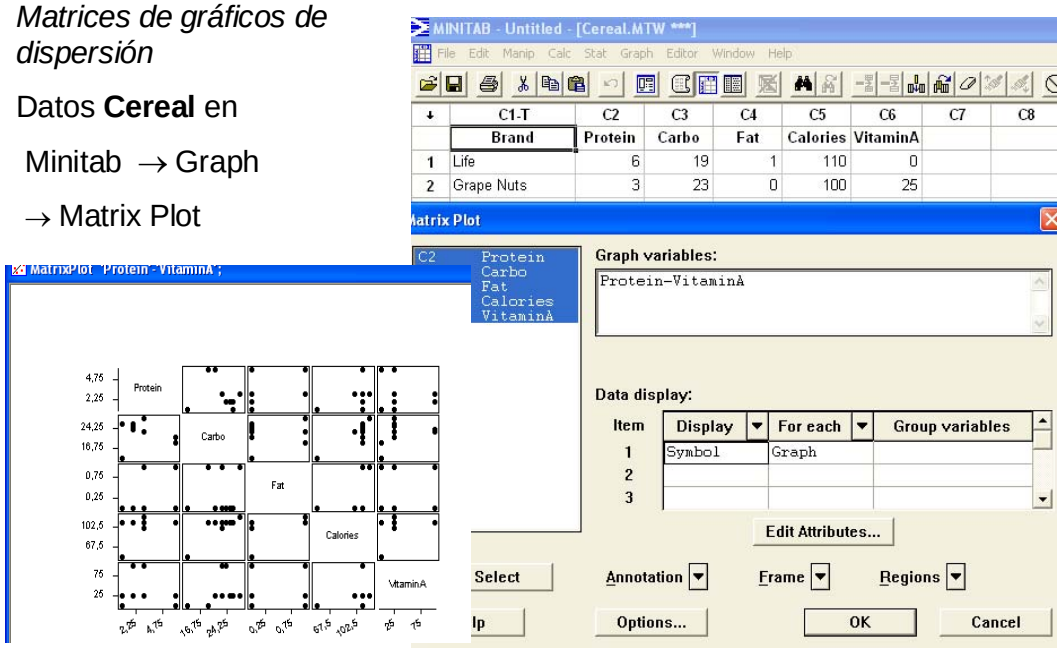
GRÁFICOS MULTIVARIANTES: MINITAB

Matrices de gráficos de dispersión

Datos **Cereal** en

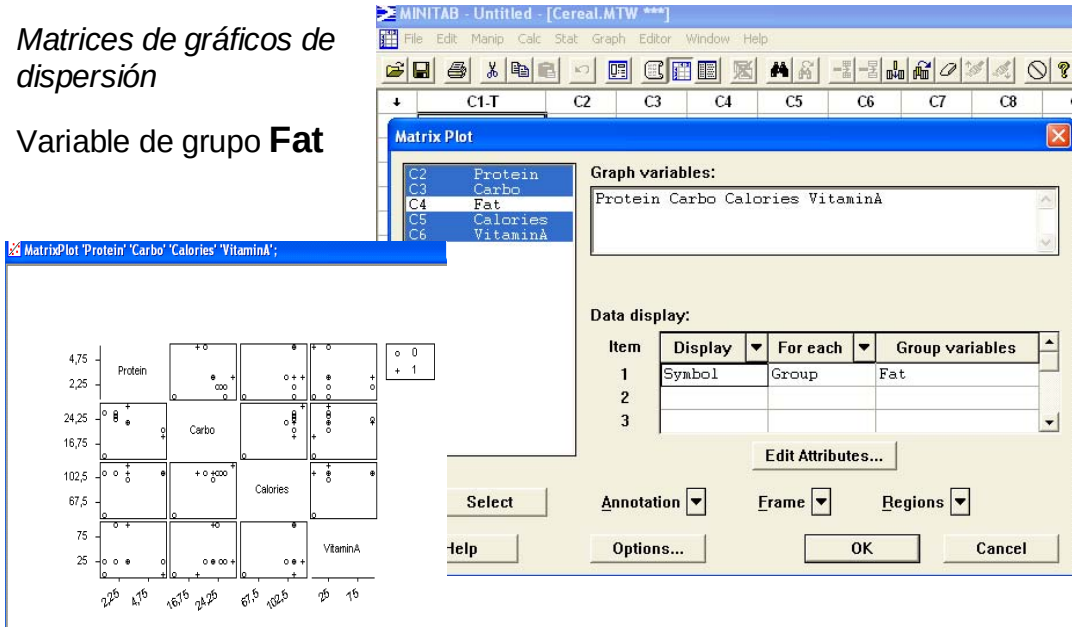
Minitab → Graph

→ Matrix Plot



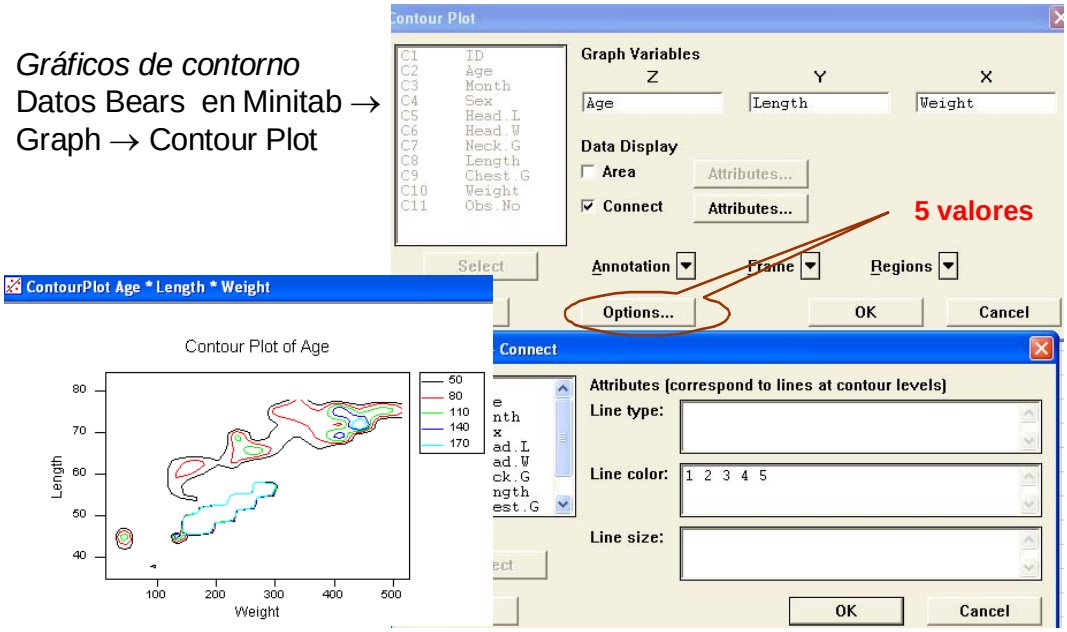
Matrices de gráficos de dispersión

Variable de grupo **Fat**



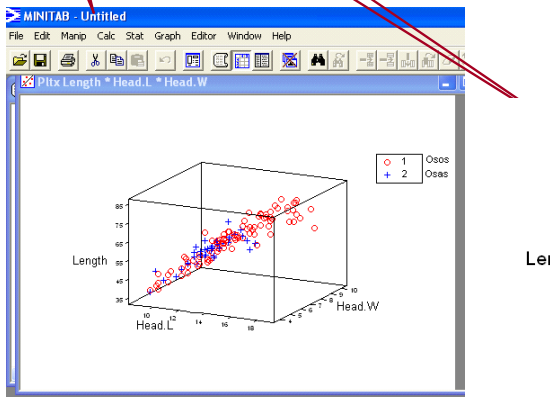
Gráficos de contorno

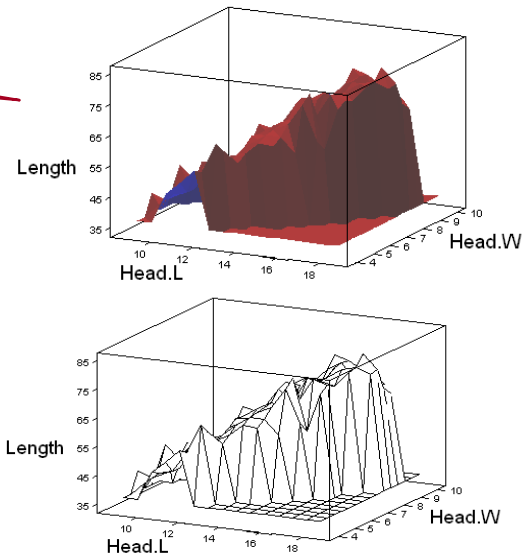
Datos Bears en Minitab → Graph → Contour Plot



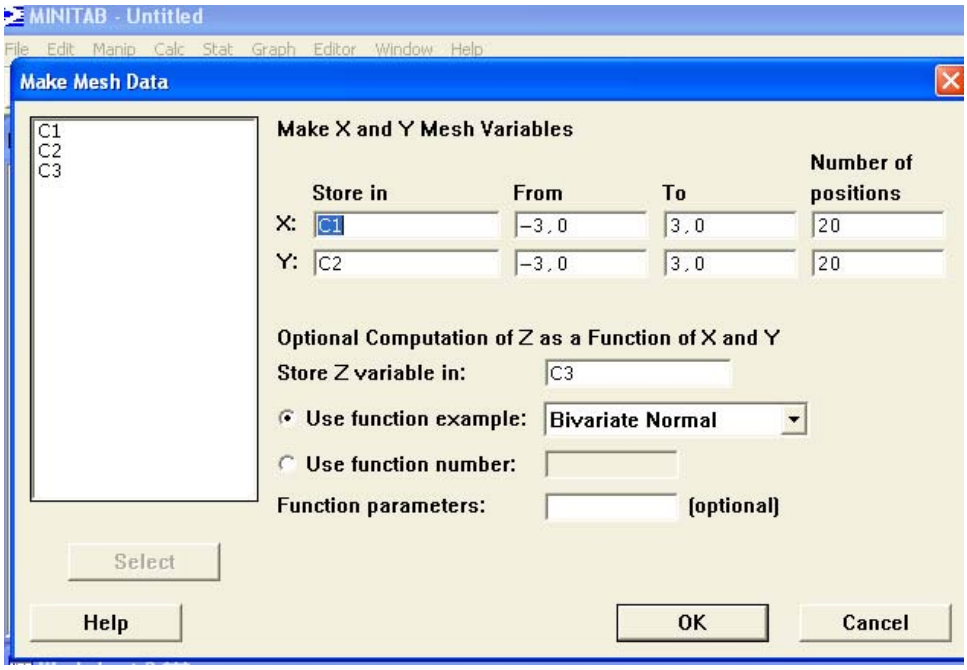
Gráficos de dispersión

Datos Bears en Minitab → Graph → 3D, 3DWireframe, 3DSurface Plot





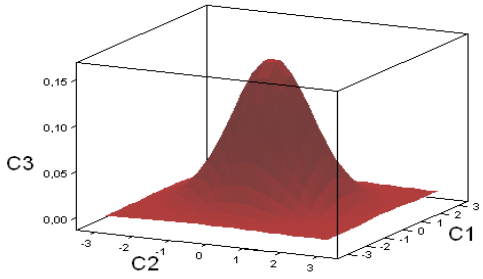
Gráficos de dispersión Make Mesh Data

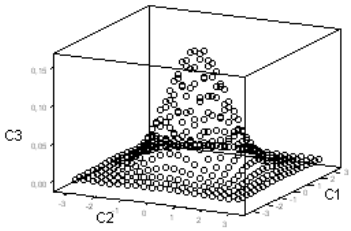


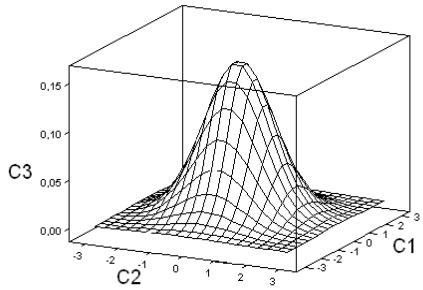
Gráficos de dispersión

Normal Bivariante → Graph →

3D, 3DWireframe, 3DSurface Plot







GRÁFICOS MULTIVARIANTES: R

Paquetes fundamentales: grDevices, graphics, grid y lattice

"máquina" gráfica

sistemas gráficos

paquete gráfico construido sobre el sistema grid para producir gráficos Trellis

Trellis Display

Cleveland, W. (1993), *Visualizing Data*

Tipos de funciones gráficas:

- •ALTO NIVEL: producen gráficos completos
- •BAJO NIVEL: añaden elementos a un gráfico existente
- •INTERACTIVAS: permiten trabajar con una salida gráfica de forma interactiva

PAQUETE GRAPHICS

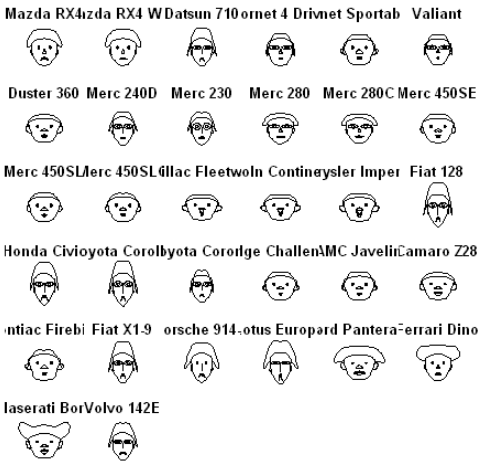
```
attach(airquality)

cloud(Month~Wind*Temp)
wireframe(Month~Wind+Temp)

levelplot(Month~Wind+Temp)
contourplot(volcano)
contour(volcano)
filled.contour(volcano)

matplot(iris[,c(1,3)],iris[,c(2,4)])
pairs(iris)

library(UsingR)
attach(kid.weights)
plot(height,weight)
identify(height,weight)
kid.weights[c(9,70,158,163,207,228),]
locator(20)
```

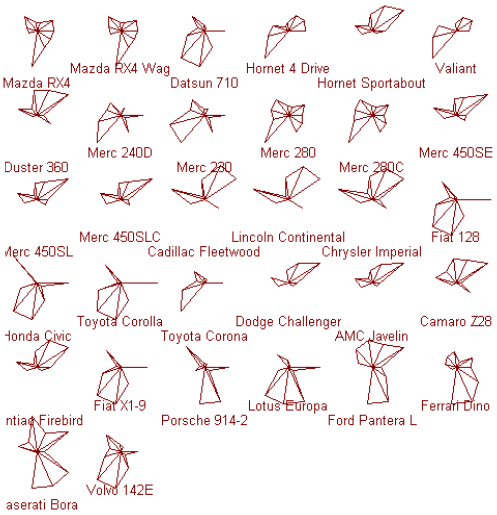
Identifica puntos de forma interactiva

Localiza hasta 20 puntos

PAQUETE GRAPHICS: Perfiles Multivariantes

```
library(TeachingDemos)
?mtcars
faces(mtcars)
stars(mtcars)
```





Con los datos en *http://www.mat.ucm.es/~palomam/aed/datos/frenos2.dat*

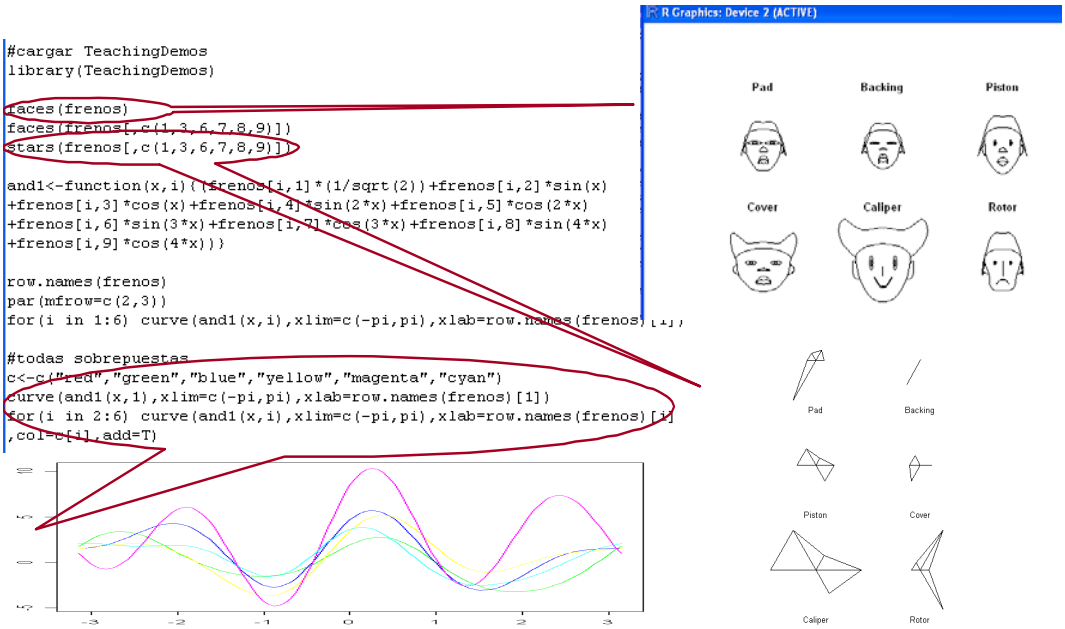
```
Pad       0.750   -1   0.0047    2   1.8   0.735   0.765   0.5   0.0
Backing   0.062   -1   0.0015    2   1.8   0.057   0.067   0.3   0.0
Piston    1.550   -1   0.0034    2   1.8   1.540   1.560   0.0   0.2
Cover     0.950    1   0.0012    2   1.8   0.945   0.955   0.2   0.0
Caliper   3.700    1   0.0059    1   1.8   3.680   3.720   0.0   0.3
Rotor     0.750   -1   0.0142    1   1.8   0.715   0.785   0.0   0.5
```

se define el objeto *frenos* que representan las características de seis marcas de frenos

```
frenos<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/frenos2.dat"
,row.names=1)
```

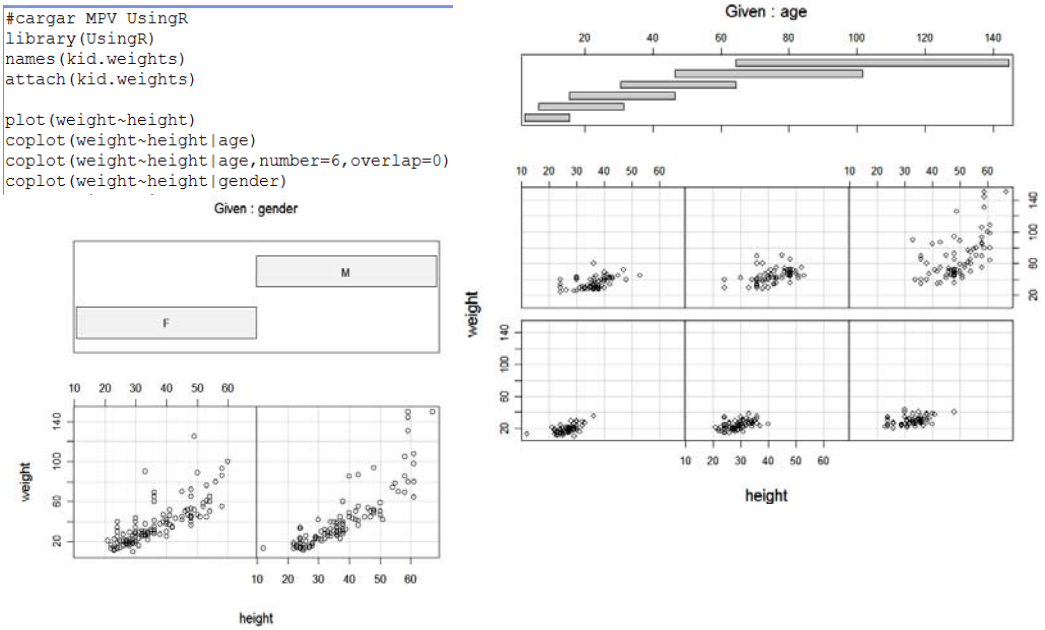
obteniendo su representación mediante Perfiles Multivariantes: **Caras de Chernoff** (faces), **Estrellas** (stars) y **Curvas de Fourier o de Andrews**

```
#cargar TeachingDemos
library(TeachingDemos)

faces(frenos)
faces(frenos[,c(1,3,6,7,8,9)])
stars(frenos[,c(1,3,6,7,8,9)])

and1<-function(x,i){frenos[i,1]*(1/sqrt(2))+frenos[i,2]*sin(x)
+frenos[i,3]*cos(x)+frenos[i,4]*sin(2*x)+frenos[i,5]*cos(2*x)
+frenos[i,6]*sin(3*x)+frenos[i,7]*cos(3*x)+frenos[i,8]*sin(4*x)
+frenos[i,9]*cos(4*x)}

row.names(frenos)
par(mfrow=c(2,3))
for(i in 1:6) curve(and1(x,i),xlim=c(-pi,pi),xlab=row.names(frenos)[i])

#todas sobrepuestas
c<-c("red","green","blue","yellow","magenta","cyan")
curve(and1(x,1),xlim=c(-pi,pi),xlab=row.names(frenos)[1])
for(i in 2:6) curve(and1(x,i),xlim=c(-pi,pi),xlab=row.names(frenos)[i]
,col=c[i],add=T)
```

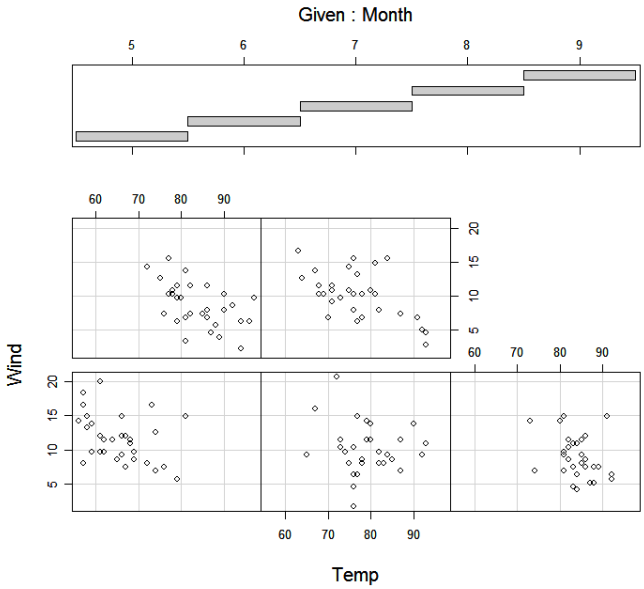


PAQUETE GRAPHICS: “Coplots”

```
#cargar MPV UsingR
library(UsingR)
names(kid.weights)
attach(kid.weights)

plot(weight~height)
coplot(weight~height|age)
coplot(weight~height|age,number=6,overlap=0)
coplot(weight~height|gender)
```



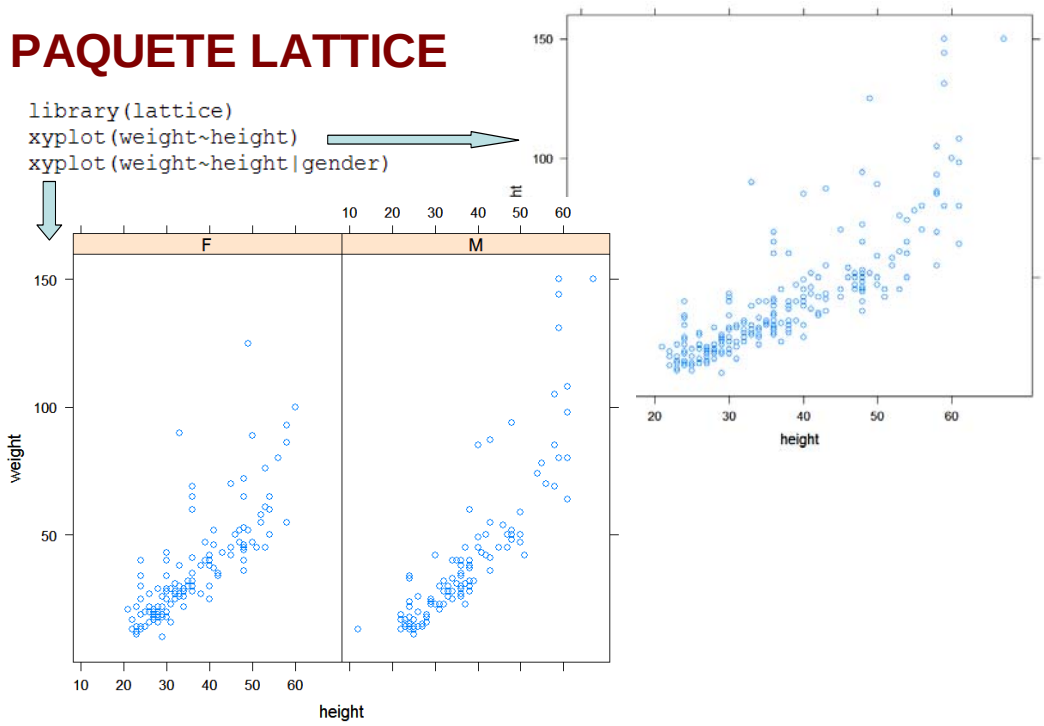
```
attach(airquality)
coplot(Wind~Temp|Month,number=5,overlap=-0.5)
```



Los *Trellis Graphics* son un nuevo sistema para la realización de gráficos con S-PLUS. El paquete *Lattice* de *R* consiste en la implementación de los "gráficos Trellis" en *R*. Por tanto es un paquete gráfico de segunda generación en el que se incluyen funciones de alto nivel para la creación de gráficos como "objetos trellis".

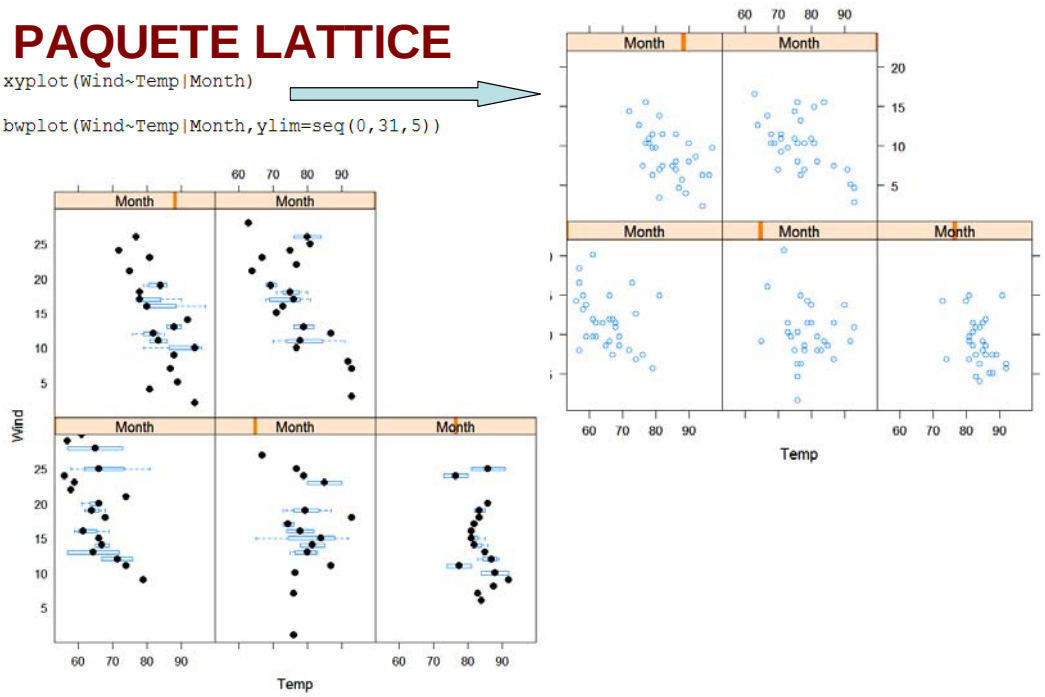
PAQUETE LATTICE

```
library(lattice)
xyplot(weight~height)
xyplot(weight~height|gender)
```



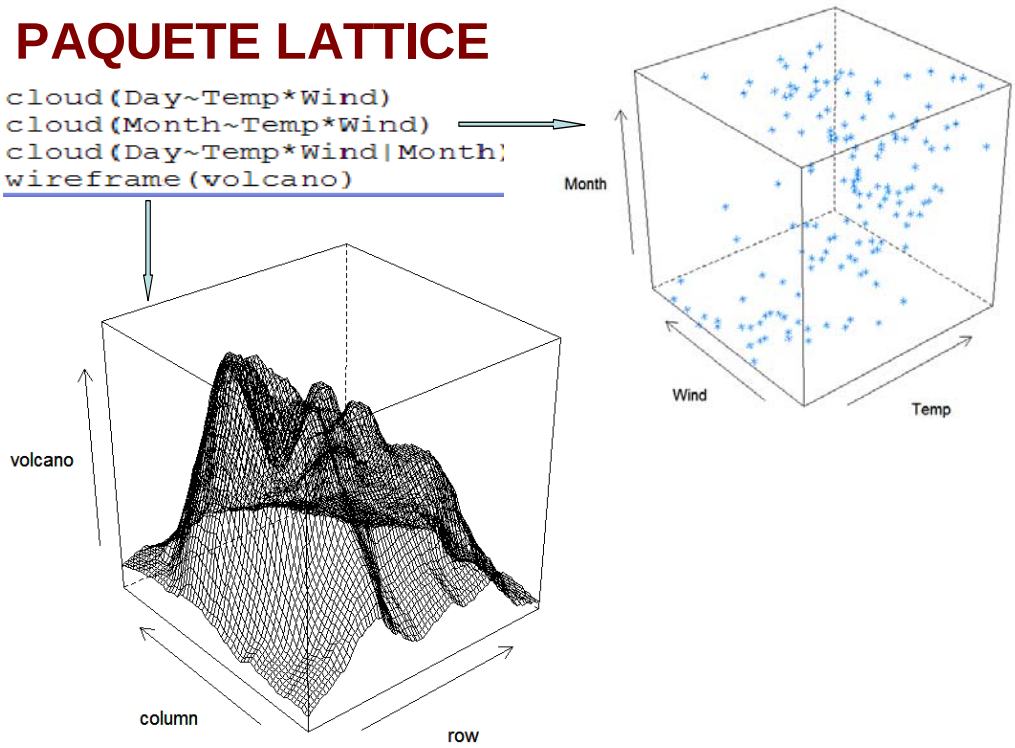
PAQUETE LATTICE

```
xyplot(Wind~Temp|Month)

bwplot(Wind~Temp|Month,ylim=seq(0,31,5))
```

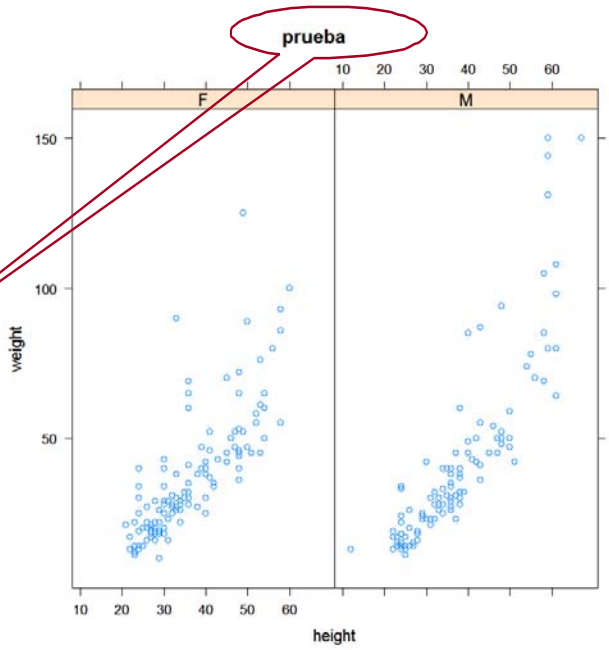


PAQUETE LATTICE

```
cloud(Day~Temp*Wind)
cloud(Month~Temp*Wind)
cloud(Day~Temp*Wind|Month)
wireframe(volcano)
```



```
#crear objetos trellis
tplot1<-xyplot(weight~height|gender)
plot(tplot1)
tplot2<-update(tplot1,main="prueba")
plot(tplot2)
class(tplot2)
```

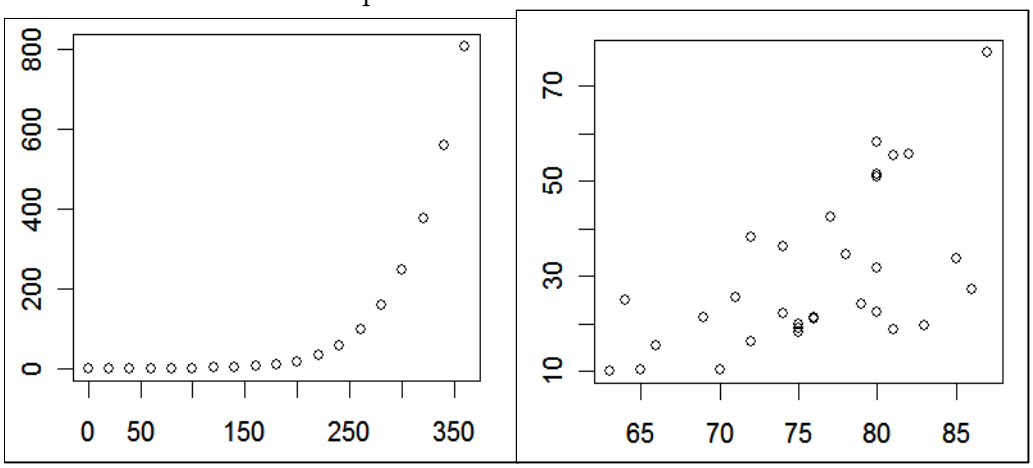


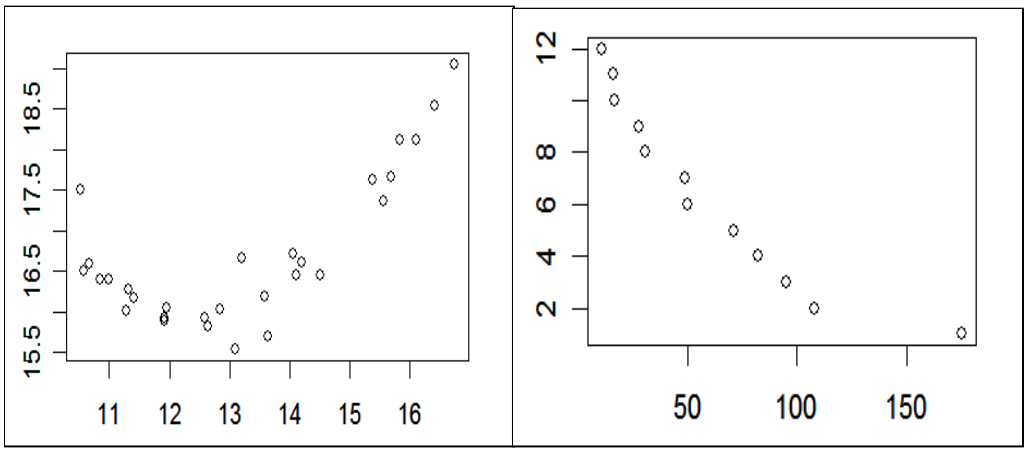
Se puede añadir un título en un objeto nuevo

6.3. Dependencia entre pares de variables cuantitativas

6.3.1. Métodos gráficos

Estudio de la nube de puntos





6.3.2. Medidas cuantitativas

Coeficiente de correlación (lineal) mide el grado de dependencia lineal entre las variables

$$r_{ij} = \frac{s_{ij}}{s_i s_j}$$

siendo

$$s_{ij} = s_{ji} = \frac{\sum_{k=1}^{n} (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{n-1} = \frac{\left(\sum_{k=1}^{n} x_{ki}x_{kj}\right) - n\bar{x}_i\bar{x}_j}{n-1}$$

$$s_i^2 = \frac{\sum_{k=1}^{n} (x_{ki} - \bar{x}_i)^2}{n-1}.$$

Es tal que $-1 \leq r_{ij} \leq 1$ y si:

$r_{ij} = 1$ existe relación lineal positiva entre las variables i y j
$r_{ij} = -1$ existe relación lineal negativa entre las variables i y j
$r_{ij} = 0$ no existe relación lineal entre las variables i y j

Coeficiente de correlación de Spearman detecta dependencia monótona entre las variables y se calcula mediante el coeficiente de correlación entre los rangos de las observaciones.

Ejemplo: Sean las observaciones

X 70.0 65.0 63.0 72.0 81.0 83.0 66.0 75.0 80.0 75.0 79.0 76 76.0 69.0 75.0 74.0 85.0 86.0 71.0 64.0 78.0 80.0 74.0 72.0 77.0 81.0 82.0 80.0 80.0 80 87

Y 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 24.2 21 21.4 21.3 19.1 22.2 33.8 27.4 25.7 24.9 34.5 31.7 36.3 38.3 42.6 55.4 55.7 58.3 51.5 51 77

entonces $r_{xy} = 0.598$ y considerando los rangos de las variables

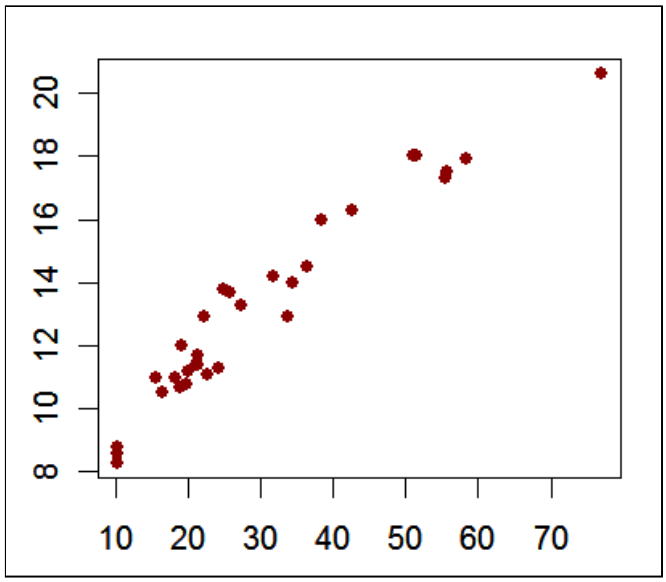
rango (X) 6 3 1 8.5 25.5 28 4 13 22 13 19 15.5 15.5 5 13 10.5 29 30 7 2 18 22 10.5 8.5 17 25.5 27 22 22 22 31

rango (Y) 2.5 2.5 1 5 7 9 4 6 15 10 16 11 13 12 8 14 21 19 18 17 22 20 23 24 25 28 29 30 27 26 31

el coeficiente de correlación de Spearman es $\rho_{xy} = 0.579$

6.3.3. Regresión lineal simple

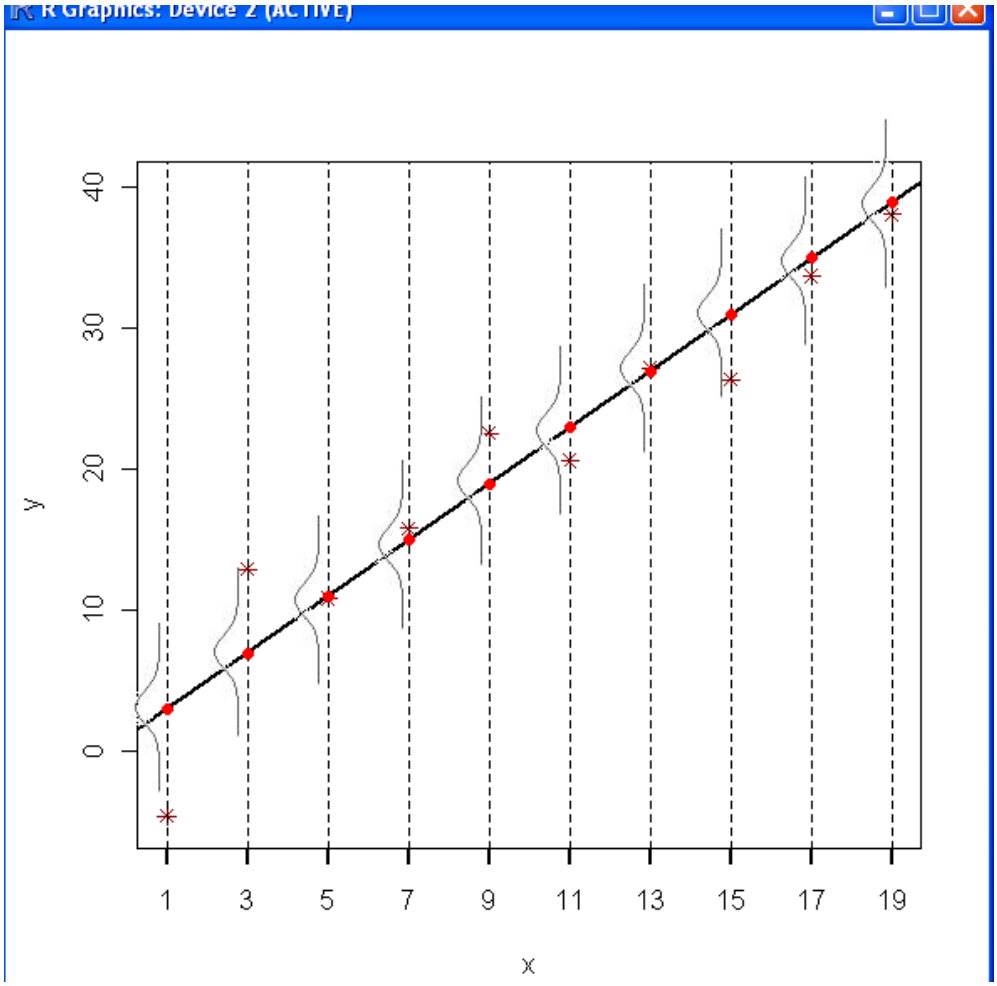
Sean las observaciones $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ de (X, Y) cuya nube de puntos tiene una "forma lineal",



para ajustar una recta a la nube de puntos utilizaremos el modelo de regresión lineal simple.

Modelo

Suponemos que existe una recta recta $y = \alpha + \beta x$ sobre la que se encuentran las medias condicionadas $E[Y \mid X = x_i]$ para $i = 1, \ldots, n$, de forma que las observaciones son $y_i = \alpha + \beta x_i + \varepsilon_i$ $i = 1, \ldots, n$ donde $\{\varepsilon_i\}$ son "ruido" o sea observaciones independientes de una $N(0, \sigma)$.



entonces se determina la recta $y = \hat{\alpha} + \hat{\beta}x$ de tal forma que

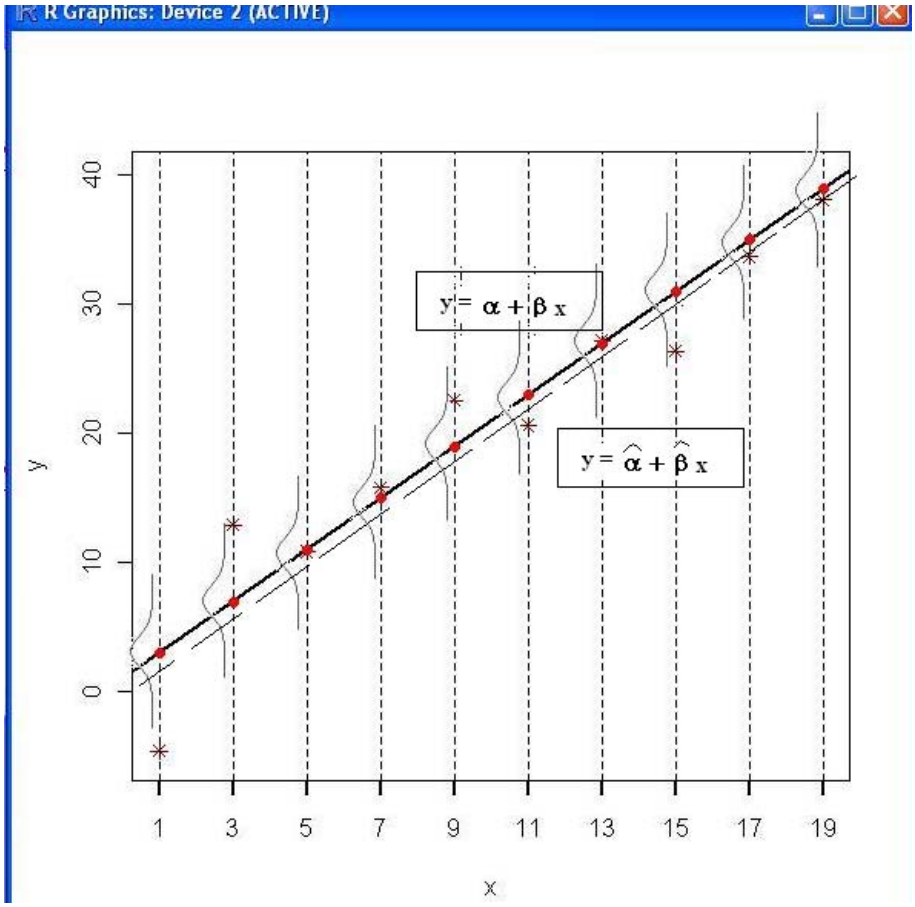
$$\min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - (\alpha + \beta x_i))^2 = \sum_{i=1}^{n} \left(y_i - \left(\hat{\alpha} + \hat{\beta} x_i\right)\right)^2$$

(criterio de mínimos cuadrados)

obteniéndose los estimadores

$$\begin{aligned} \hat{\alpha} &= \bar{y} - \hat{\beta}\bar{x} \\ \hat{\beta} &= \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2}. \end{aligned}$$

Gráficamente podría obtenerse la siguiente representación



Residuos

Se definen

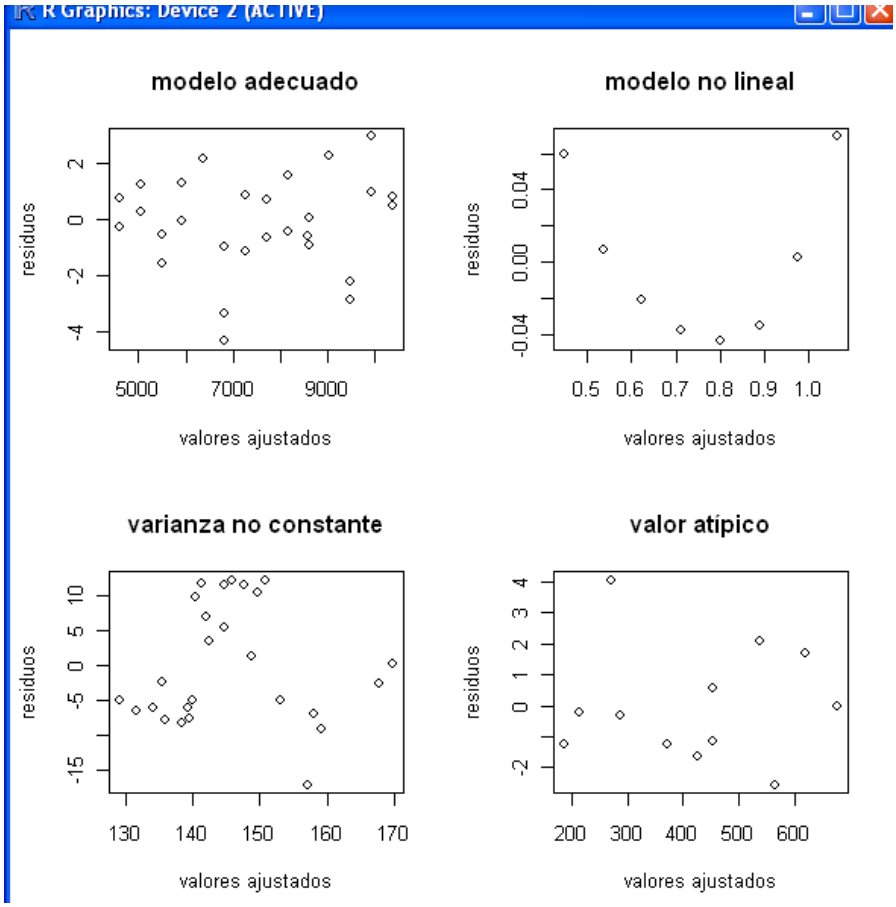
$$\hat{\varepsilon}_i = y_i - \left(\hat{\alpha} + \hat{\beta} x_i\right) = y_i - \hat{y}_i; i = 1, \ldots, n$$

y son una herramienta esencial para el diagnóstico y validación de las hipótesis establecidas en el planteamiento inicial del modelo de regresión lineal. También permiten definir el estimador de σ^2 mediante

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum_{i=1}^{n} \hat{\varepsilon}_i^2$$

Análisis de residuos

Se utilizan los gráficos que representan los puntos $\{(\hat{y}_i, \hat{\varepsilon}_i)\,; i = 1, \ldots, n\}$ analizando los patrones que describen para verificar si el modelo de regresión lineal es el adecuado. Así, por ejemplo se pueden obtener las siguientes nubes de puntos



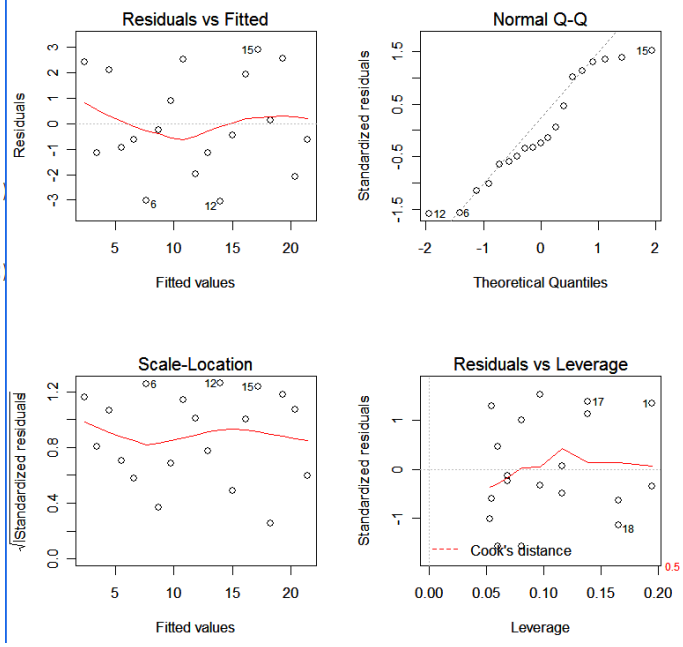
6.3.4. Regresión con R

Todos los elementos de la regresión en R se recogen en un "*objeto lm*" cuyos elementos se especifican si se solicitan sus "*attributes*" o su "*summary*".

Si se obtiene, mediante simulación, una nube de puntos alrededor de $y = 2x+1$ con errores $N(0, 2)$, se podrá ajustar la correspondiente recta de mínimos cuadrados.

REGRESIÓN en R

```
x<-seq(1,10,0.5)
y<-2*x+1+rnorm(length(x),0,2)
plot(x,y)
abline(1,2,col="red")
abline(v=x,lty=2)
n<-length(x)
beta<-(sum(x*y)-n*mean(x)*mean(y))
alpha<-mean(y)-beta*mean(x)
ypred<-alpha+beta*x
sigma2<-(1/(n-2))*sum((y-ypred)^2)
sigma<-sqrt(sigma2)
#objeto lm
reg1<-lm(y~x)
summary(reg1)
abline(reg1,col="green")
attributes(reg1)
reg1$coefficients
yg<-reg1$fitted.values
y-yg
reg1$residuals
par(mfrow=c(2,2))
plot(reg1)
```




```
attributes(reg1)
reg1$coefficients
yg<-reg1$fitted.values
y-yg
reg1$residuals
```

```
> attributes(reg1)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"

> reg1$coefficients
(Intercept)           x
   0.405698    2.125474
> reg1$fitted.values
        1         2         3         4         5         6         7         8
 2.531172  3.593910  4.656647  5.719384  6.782121  7.844859  8.907596  9.970333
        9        10        11        12        13        14        15        16
11.033070 12.095808 13.158545 14.221282 15.284019 16.346756 17.409494 18.472231
       17        18        19
19.534968 20.597705 21.660443
> reg1$residuals
         1          2          3          4          5          6          7
 2.6075010 -1.9016683 -0.1685592  0.2802252 -1.5832786  0.1051743 -1.4021545
         8          9         10         11         12         13         14
-1.4495365 -1.4645084  1.8493617 -0.8092907  2.9735677  0.4691531  1.9102155
        15         16         17         18         19
 1.8883164  0.6604750 -0.1335029 -0.5171967 -3.3142940
```

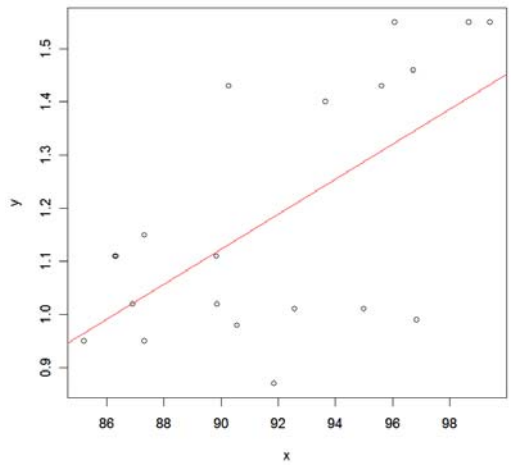
REGRESIÓN Y CORRELACIÓN en R

```
#residuos
plot(reg1$fitted.values,reg1$residuals)
abline(h=c(-3,3),col="red")
identify(reg1$fitted.values,reg1$residuals)
plot(x,y)
abline(1,2,col="red")
abline(v=x,lty=2)
identify(x,y)

#correlación
cor(x,y)
cor(x,y,method="spearman")
r1<-rank(x)
r2<-rank(y)
cor(r1,r2)
```

```
#modelo no lineal
x<-seq(0.5,20,1)
y<-log(x)+rnorm(length(x),0,2)
plot(x,y)
curve(log(x),add=TRUE)
abline(v=x,lty=2)
cor(x,y)
cor(x,y,method="spearman")
n<-length(x)
beta<-(sum(x*y)-n*mean(x)*mean(y))/(sum(x^2)-n*(mean(x)^2))
alpha<-mean(y)-beta*mean(x)
abline(alpha,beta,col="brown")

reg2<-lm(y~x)
summary(reg2)
abline(reg2,col="green")
attributes(reg2)
plot(reg2)
#residuos
plot(reg2$fitted.values,reg2$residuals)
abline(h=c(-2,2),col="red")
identify(reg2$fitted.values,reg2$residuals)
plot(x,y)
abline(1,2,col="red")
abline(v=x,lty=2)
identify(x,y)
```

ANÁLISIS DE RESIDUOS

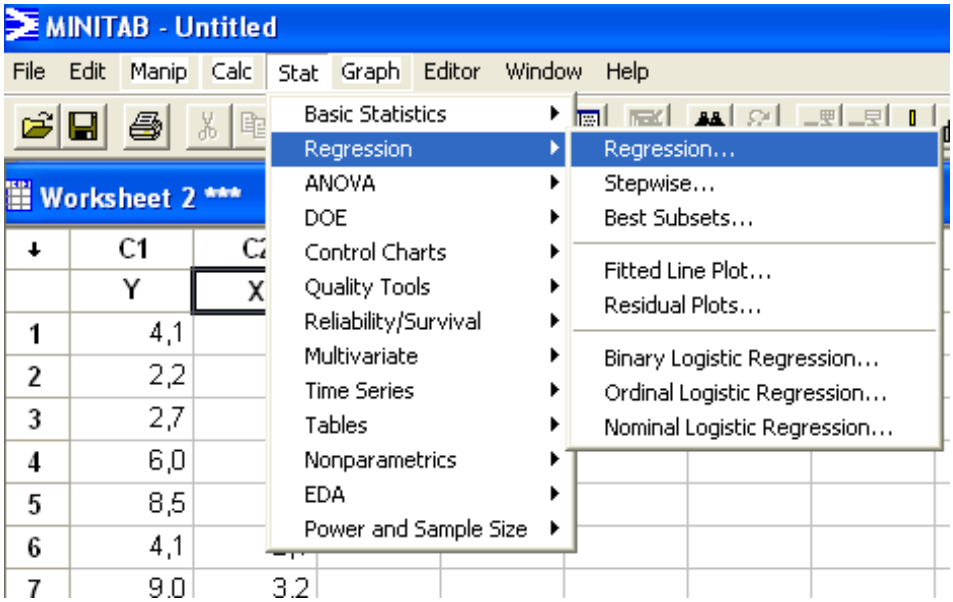
```
library(MPV)
help(package=MPV)
#?p2.7
x<-p2.7[,1]
y<-p2.7[,2]
plot(x,y)
n<-length(x)
beta<-(sum(x*y)-n*mean(x)*mean(y))/(sum(x^2)-n*(mean(x)^2))
alpha<-mean(y)-beta*mean(x)
abline(alpha,beta,col=2)
#objeto lm
reg1<-lm(y~x)
summary(reg1)
abline(reg1,col="green")
attributes(reg1)
plot(reg1)

#residuos
plot(reg1$fitted.values,reg1$residuals)

identify(reg1$fitted.values,reg1$residuals)
plot(x,y)


identify(x,y)
#correlación
cor(x,y)
cor(x,y,method="spearman")
```



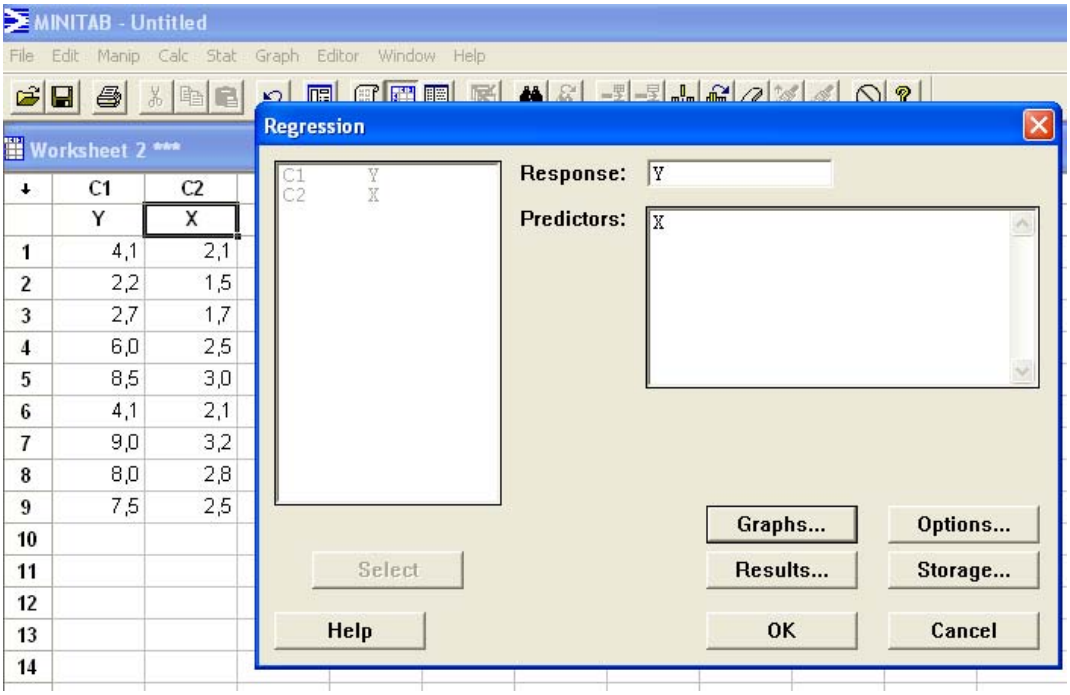

```
> cor(x,y)
[1] 0.6237968
> cor(x,y,method="spearman")
[1] 0.4675275
```

6.3.5. Regresión con MINITAB

Se selecciona la persiana



obteniéndose la ventana de diálogo donde se especificarán las variables consideradas y la opciones de resultados requeridos



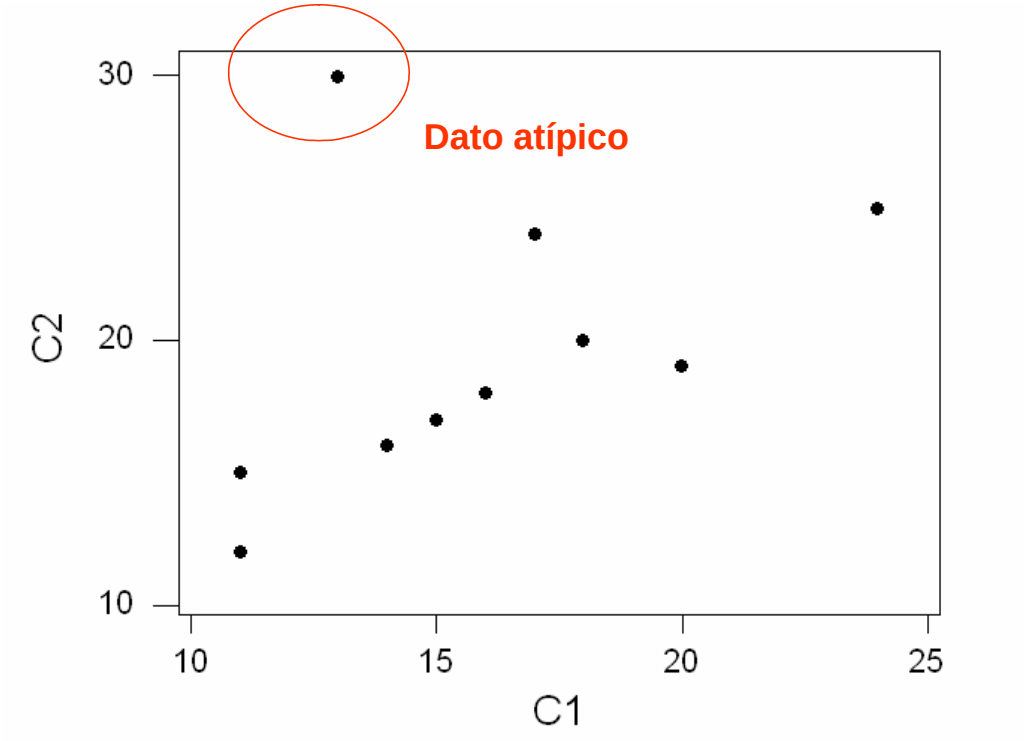
6.3.6. Ejemplo 3

Con los datos de

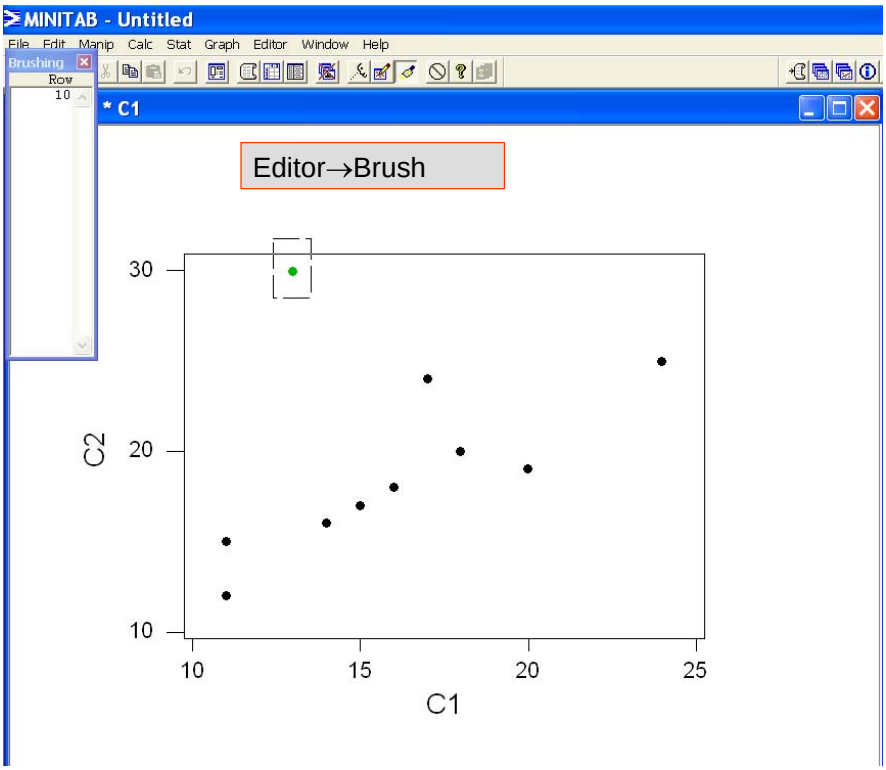
http://www.mat.ucm.es/~palomam/aed/datos/ejemplo3.txt

```
15 17
11 15
16 18
24 25
14 16
11 12
18 20
17 24
20 19
13 30
```

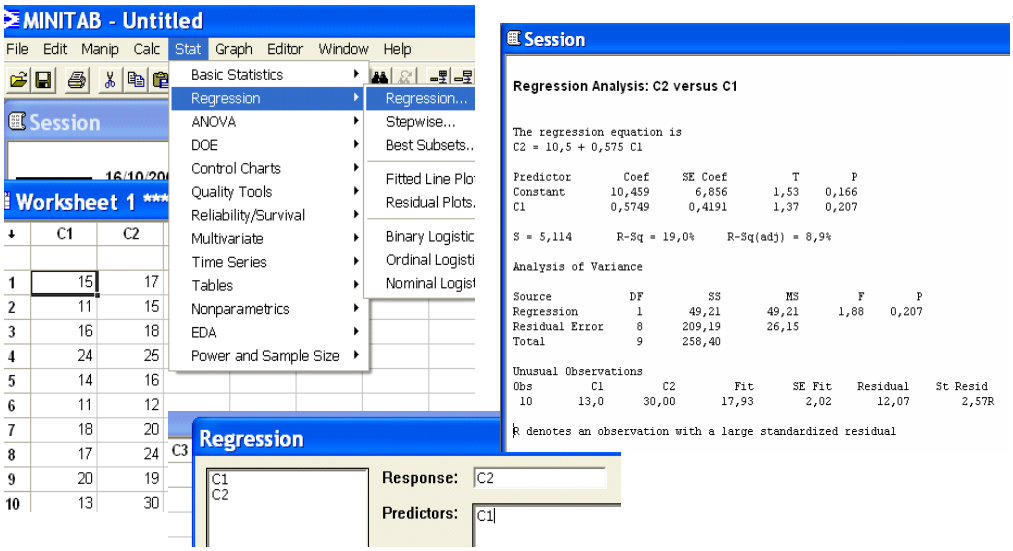
MINITAB: Nube de puntos



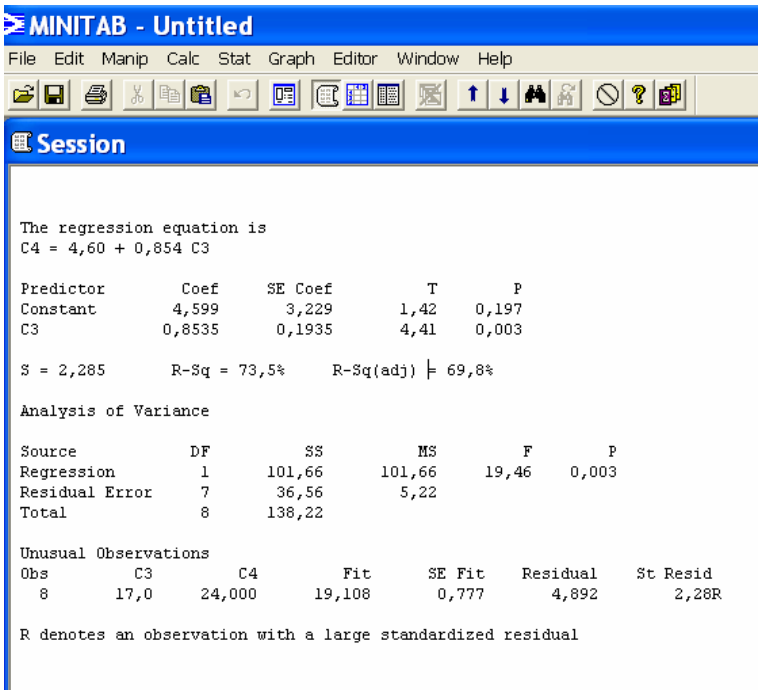
MINITAB: Barrido ("Brush")



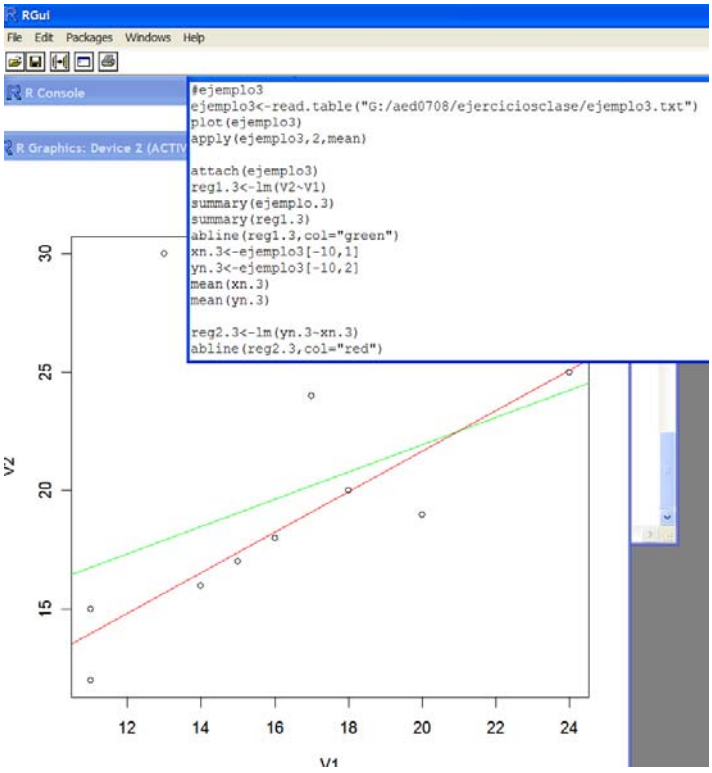
MINITAB: Regresión con todos los datos



MINITAB: Regresión sin dato atípico



R

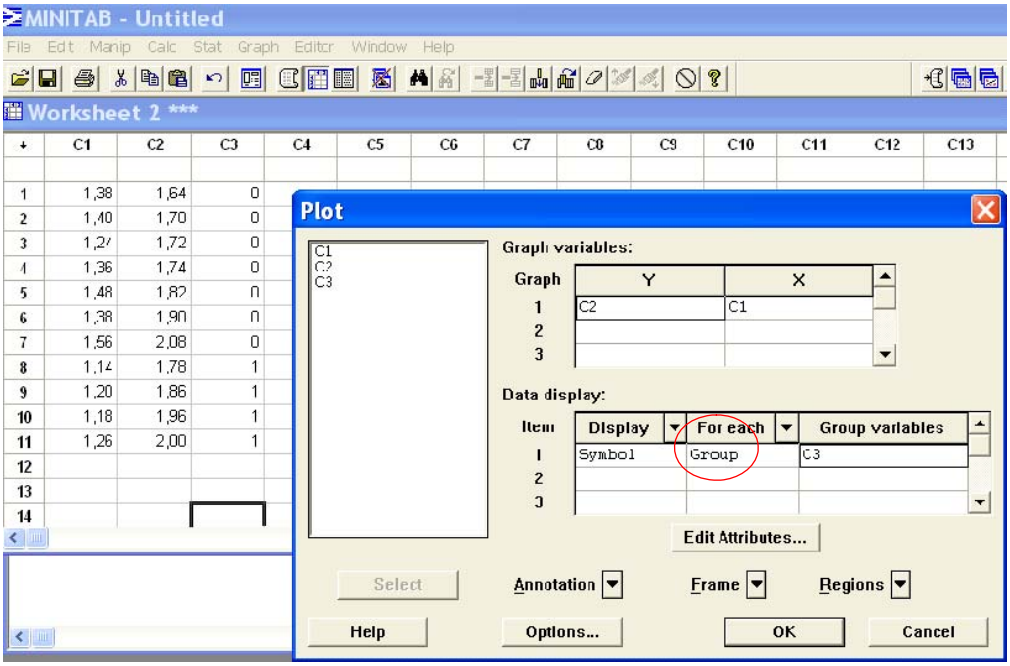


6.3.7. Ejemplo 4

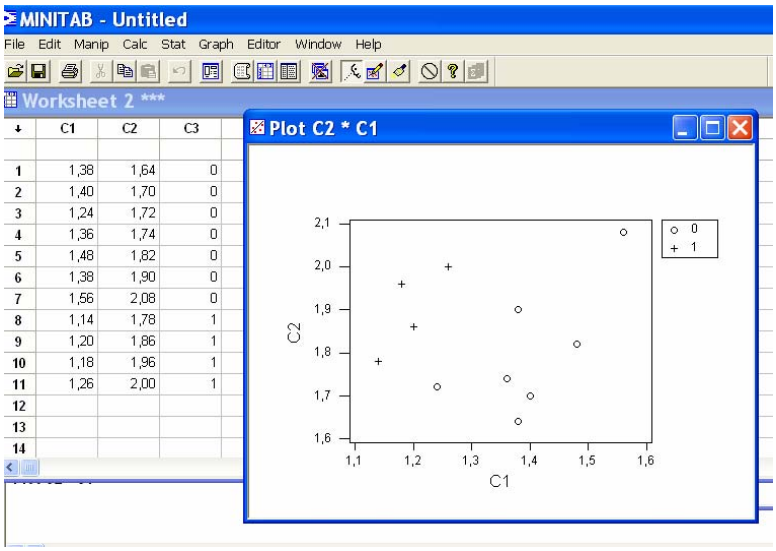
Con los datos de

http://www.mat.ucm.es/~palomam/aed/datos/ejemplo4.txt

```
1.38 1.64
1.4  1.7
1.24 1.72
1.36 1.74
1.48 1.82
1.38 1.9
1.56 2.08
1.14 1.78
1.2  1.86
1.18 1.96
1.26 2
```

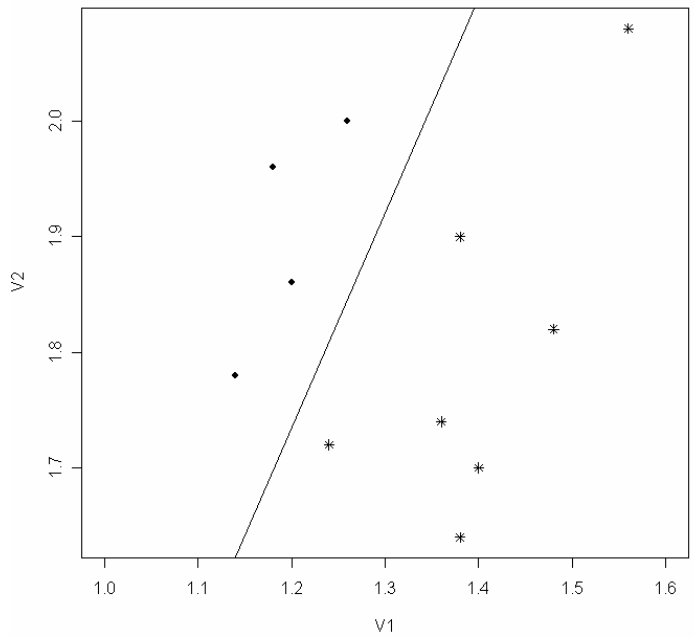


MINITAB: Nube de puntos separados



R

```
ejemplo4<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/ejemplo4.txt")
plot(ejemplo4)
plot(ejemplo4,pch=19)
plot(ejemplo4[1:7,],pch=8,xlim=c(1,1.6))
points(ejemplo4[8:11,],pch=16)
abline(-0.5,1.8617,col=red")
```



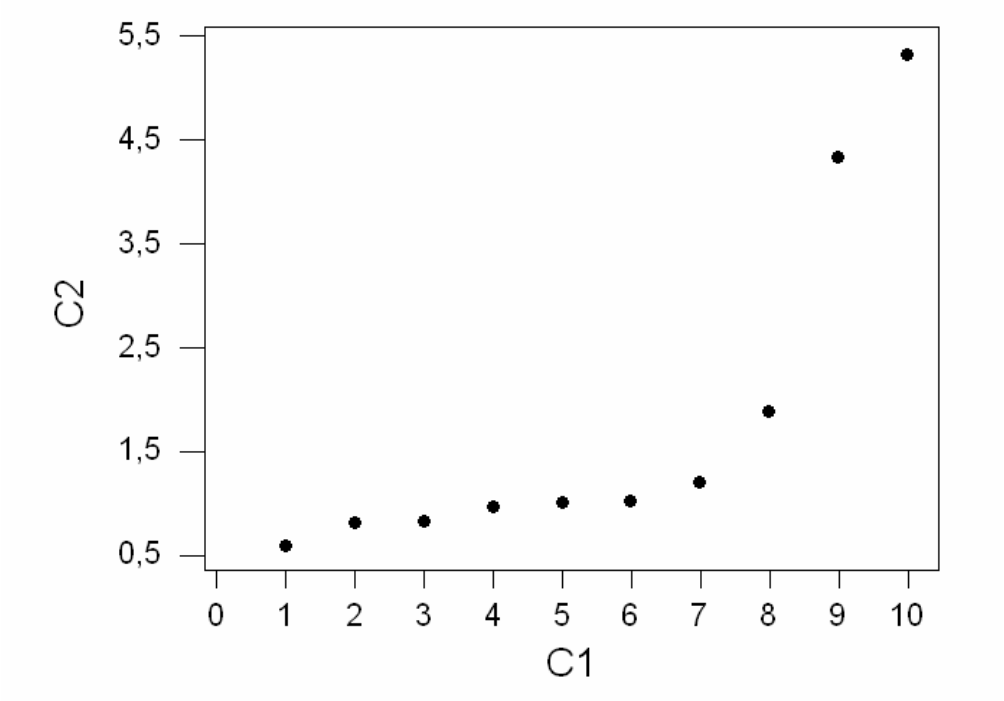
6.3.8. Ejemplo 5

Con los datos de

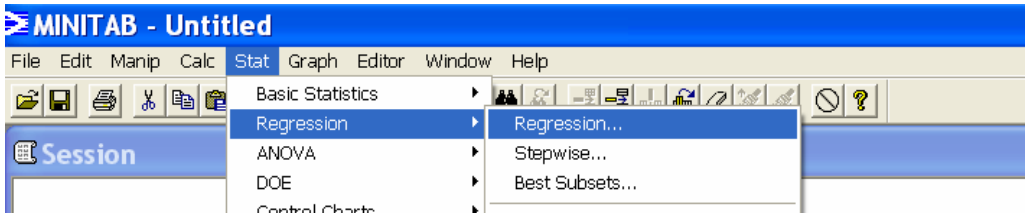
http://www.mat.ucm.es/~palomam/aed/datos/ejemplo5.txt

```
1 0.59
2 0.81
3 0.83
4 0.97
5 1.01
6 1.02
7 1.2
8 1.88
9 4.34
10 5.33
```

MINITAB: Nube de puntos



MINITAB: Regresión con todos los datos



Session

```
Welcome to Minitab, press F1 for help.

Regression Analysis: C2 versus C1


The regression equation is
C2 = - 0,646 + 0,444 C1

Predictor        Coef     SE Coef          T        P
Constant      -0,6460      0,6956      -0,93    0,380
C1             0,4444      0,1121       3,96    0,004

S = 1,018       R-Sq = 66,3%     R-Sq(adj) = 62,0%

Analysis of Variance

Source            DF          SS          MS         F        P
Regression         1      16,290      16,290     15,71    0,004
Residual Error     8       8,295       1,037
Total              9      24,585
```

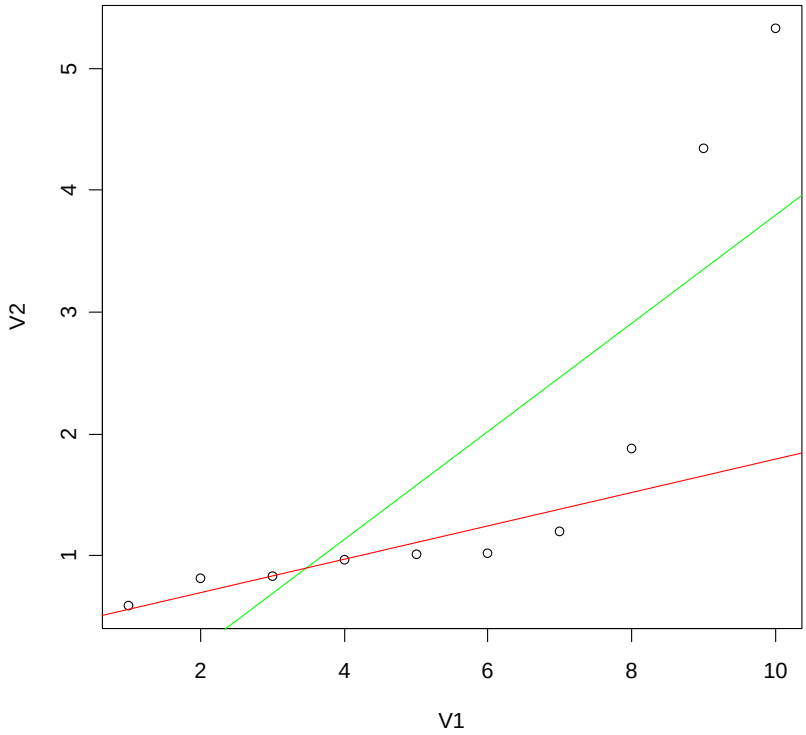
MINITAB: Regresión sin datos atípicos

Regression Analysis: C4 versus C3

```
The regression equation is
C4 = 0,418 + 0,138 C3

Predictor        Coef     SE Coef          T        P
Constant       0,4179      0,1550       2,70    0,036
C3            0,13798     0,03069       4,50    0,004

S = 0,1989      R-Sq = 77,1%     R-Sq(adj) = 73,3%

Analysis of Variance

Source            DF          SS          MS         F        P
Regression         1     0,79957     0,79957     20,22    0,004
Residual Error     6     0,23732     0,03955
Total              7     1,03689

Unusual Observations
Obs         C3         C4         Fit      SE Fit    Residual    St Resid
  8       8,00     1,8800      1,5217      0,1284      0,3583        2,36R

R denotes an observation with a large standardized residual
```

R

```
ejemplo5<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/ejemplo5.txt")
apply(ejemplo5,2,mean)
plot(ejemplo5)
attach(ejemplo5)
reg1<-lm(V2~V1)
summary(ejemplo5)
summary(reg1)
abline(reg1,col="green")
xn<-ejemplo5[-c(9,10),1]
yn<-ejemplo5[-c(9,10),2]
mean(xn)
mean(yn)
reg5<-lm(yn~xn)
abline(reg5,col=red")
```



La idea fundamental radica en huir de un marco paramétrico, como el que acabamos de mencionar en el Modelo de Regresión Lineal Simple y hacer, únicamente, la hipótesis de que la media condicionada es una función f que se comporta con cierta *suavidad* —*"smoothness"*—; lo suficiente como para que su valor en puntos X *próximos* pueda suponerse *muy similar*. Sobre la dispersión de Y respecto a nuestra función objetivo, mantenemos la hipótesis de que es *constantemente igual a* σ^2.

Casi seguro que ante la pregunta de *¿cómo actuaría sobre los datos para "estimar" la función* f ? una respuesta muy frecuente sería del tipo: *asociando a cada* X *un resumen estadístico de los valores de* Y *en las proximidades del* X *considerado*; es decir, algo así como calcular medias locales. Si hiciéramos un gráfico de dichas medias sobre el diagrama de dispersión del par (X, Y), tendríamos una versión *suavizada* de la nube de puntos; de ahí que los métodos reciban el nombre de *suavizadores* —en inglés "*smoothers*"—.

Regresión Local (LOESS, LOWESS)

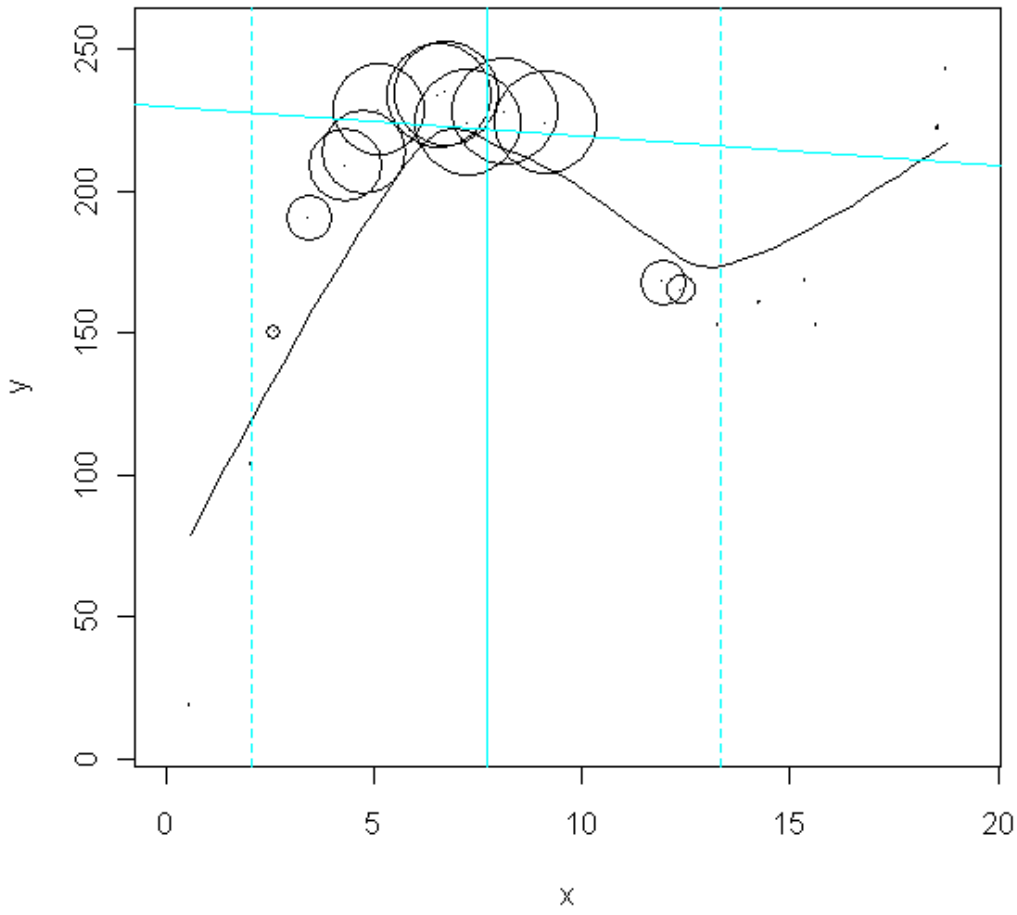
Este método calcula, para cada valor $X = x_0$, el promedio de Y que se obtiene ajustando un modelo de regresión lineal, por mínimos cuadrados ponderados, en un *entorno* de x_0. Evidentemente, habrá que concretar lo que entendemos por un entorno de x_0 y los *pesos* que utilizaremos en las ponderaciones para los mínimos cuadrados; lo demás está perfectamente definido. Se suele tomar como *vecindad* —entorno— de x_0 los k *vecinos más próximos*, es decir, los puntos que dan los k valores más pequeños a la distancia $|x - x_0|$. Habitualmente, el valor de k se especifica mediante un porcentaje —conocido como "*span*"— del número total de observaciones del par (X, Y); los valores más frecuentes son los del intervalo $[0.3, 0.5]$, aunque el valor que se toma en R por defecto es $2/3$. Respecto a los pesos, a cada punto x del entorno de x_0 se le asigna el peso

$$w\left(\frac{|x - x_0|}{\Delta_{x_0}}\right)$$

siendo Δ_{x_0} la máxima distancia encontrada en el entorno y $w(\cdot)$ una función —denominada *tri-cubo*— definida por

$$w(t) = \begin{cases} (1 - t^3)^3 & , \ si \ 0 \leq t < 1 \\ 0 & en \ los \ demás \ casos. \end{cases} .$$

Gráficamente, el procedimiento podría explicarse mediante la siguiente representación donde el tamaño de los círculos refleja el peso de cada punto en el entorno considerado para $x_0 = 8$.



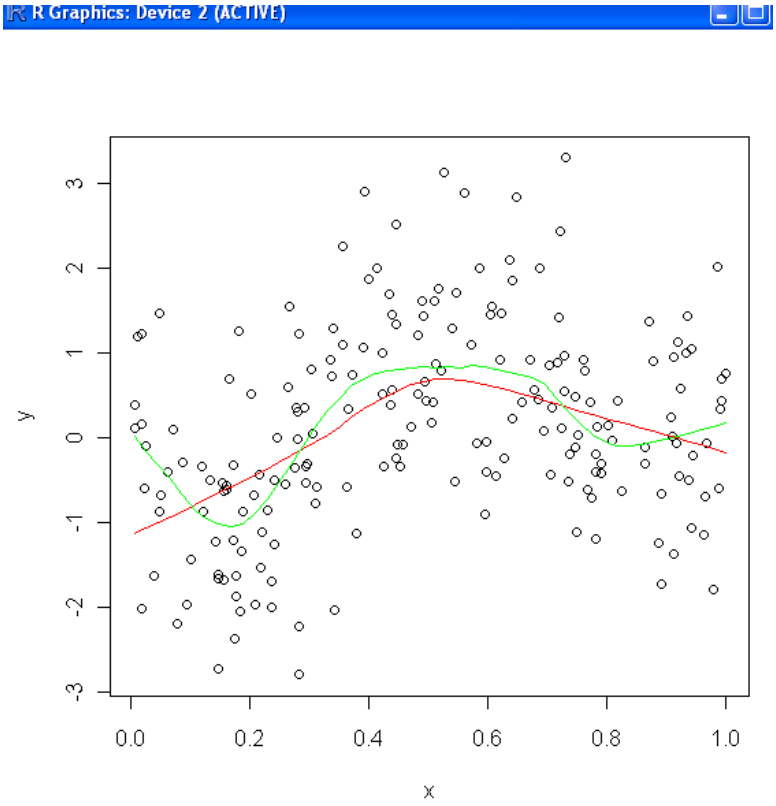
Puede aplicar este procedimiento a los datos

http://www.mat.ucm.es/~palomam/aed/datos/loess2.txt

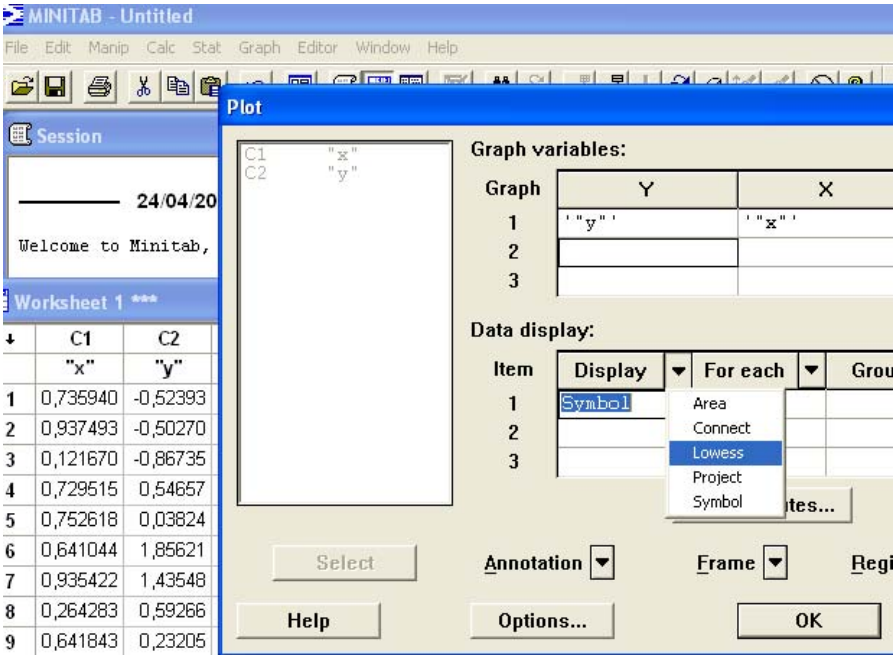
sin más que ejecutar, en R, las instrucciones siguientes:

```
lo<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/loess2.txt"
,header=T)
attach(lo)
plot(x,y)
lines(loess.smooth(x,y),col="red")
lines(loess.smooth(x,y,span=0.25),col="green")
```

El resultado es el que se muestra en la siguiente figura; en ella aparecen los ajustes locales obtenidos con un "*span*" de 0.25, la línea verde, y por defecto con un "*span*" de 2/3, la roja.

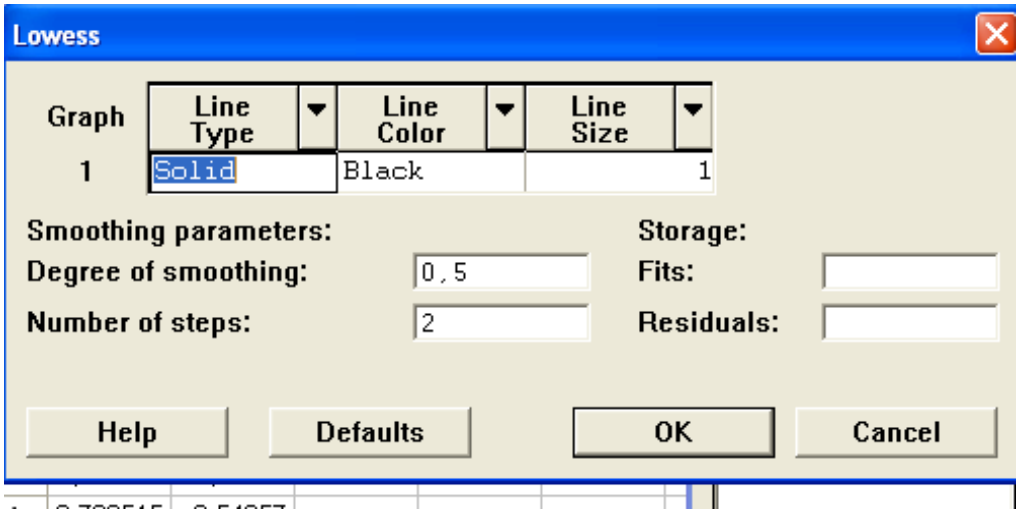


En MINITAB se puede representar este ajuste como opción al solicitar un "Plot" de los puntos. Así, con el ejemplo anterior aparece la siguiente ventana al solicitar **Graph→Plot**

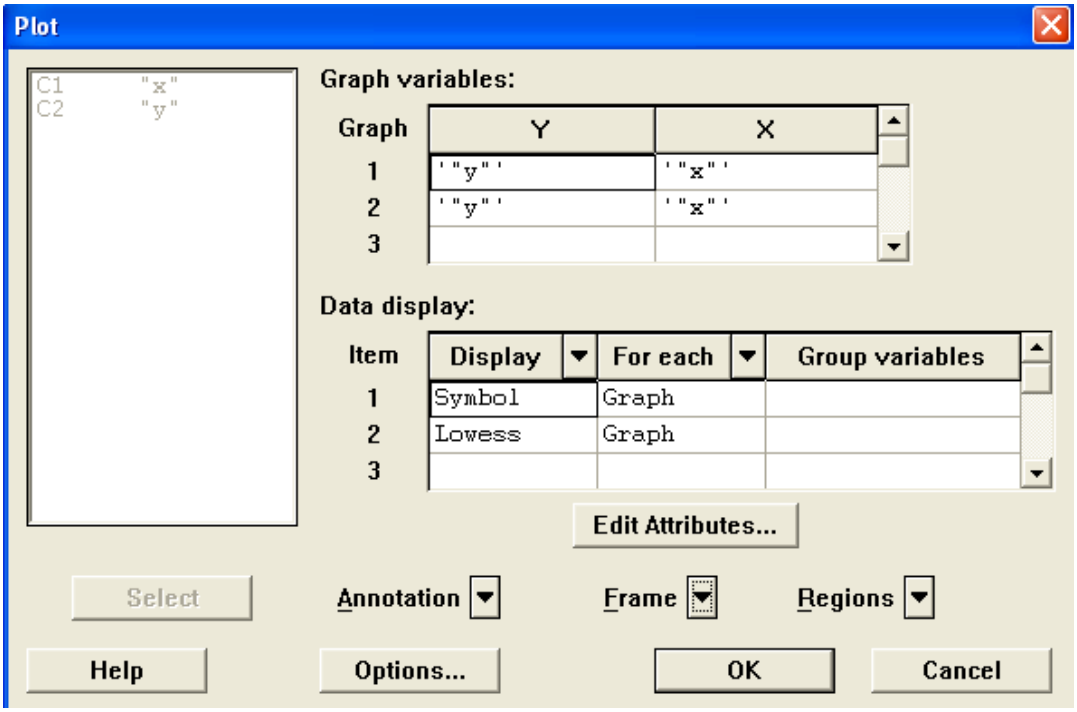


lo que daría solo la representación de la línea ajustada con un "*span*" de 0.5, por defecto. Si se marca 

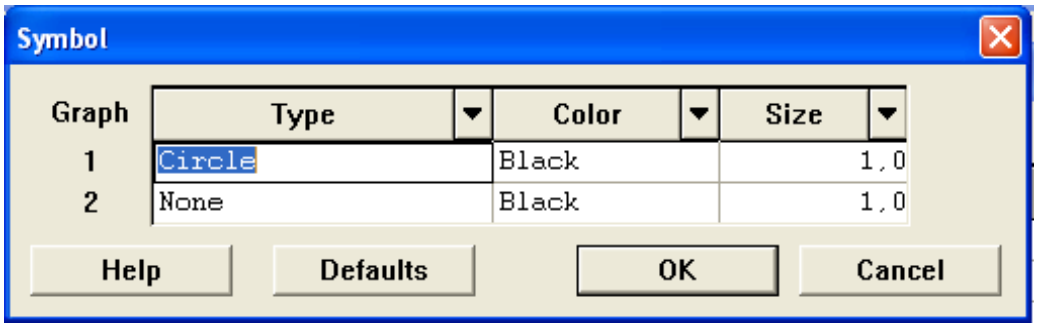
aparece la ventana



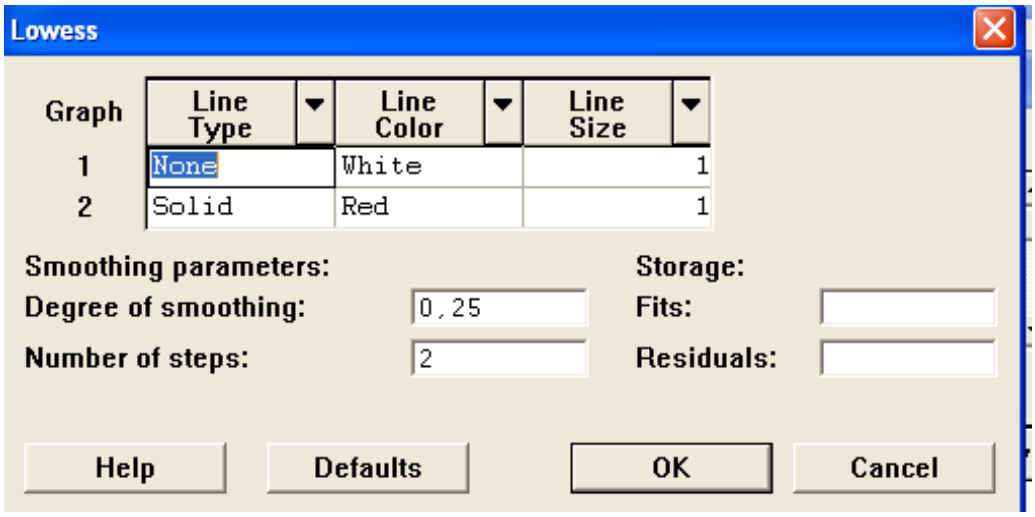
donde se pueden cambiar los parámetros. Si además se requiere la aparición de los puntos, se necesitan dos gráficos



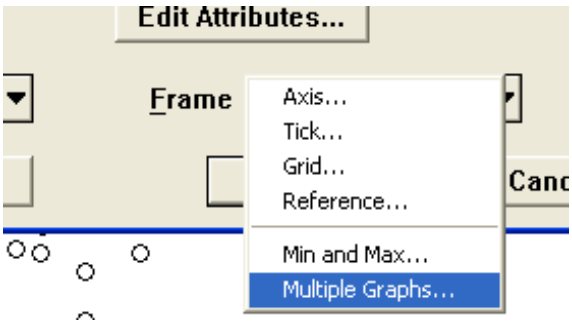
con sus respectivos 



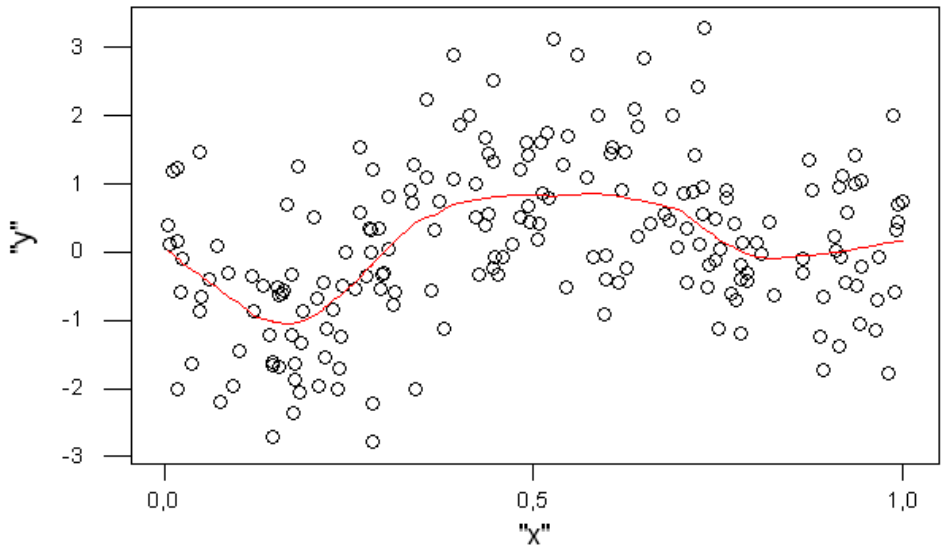
y



junto con la condición de "gráficos múltiple superpuestos" en **Frame**



lo que finalmente nos daría la siguiente figura



6.4. Dependencia entre pares de variables cualitativas

Las variables cualitativas bidimensionales se representan mediante "*Tablas de Contingencia*" que recogen el número de observaciones con los diferentes pares de atributos cruzados. Así para las variables (Z_1, Z_2) cuyos atributos son respectivamente, $A_1, \ldots, A_r$ y $B_1, \ldots, B_c$ se tendrá la siguiente tabla

$Z_1 \backslash Z_2$	B_1	$\cdots$	B_c	
A_1	n_{11}	$\cdots$	n_{1c}	$n_{1\bullet}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_r	n_{r1}	$\cdots$	n_{rc}	$n_{r\bullet}$
	$n_{\bullet 1}$	$\cdots$	$n_{\bullet c}$	n

donde n_{ij} es el número de observaciones con las características A_i y B_j.

Por ejemplo de la base datos en

"*http://www.mat.ucm.es/~palomam/aed/datos/simpson.dat*"

se puede obtener

trabajo \ *género*	hombre	mujer
administración	276	229
técnico	461	34

.

6.4.1. Contraste de dependencia

Se puede utilizar el contraste de la χ^2 para evaluar si existe dependencia entre dos variables cualitativas mediante el estadístico

$$V = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{\left(n_{ij} - \frac{n_{i\bullet} n_{\bullet j}}{n}\right)^2}{\frac{n_{i\bullet} n_{\bullet j}}{n}}$$

cuya distribución, bajo la hipótesis de independencia, es una $\chi^2_{(r-1)(c-1)}$. Por lo tanto se rechazará la hipótesis de independencia a nivel α si el valor del estadístico es $V > \chi^2_{(r-1)(c-1);\alpha}$.

En el ejemplo anterior se obtiene

```
V=188.9591, grados de libertad = 1, p-valor < 2.2e-16,
```

por lo que se rechaza la hipótesis nula, H_0, de independencia.

6.4.2. Paradoja de Simpson

Surge en algunas ocasiones, al introducir una nueva variable que permite comparar las anteriores variables en las diferentes subpoblaciones que determinan las categorías de la variable introducida.

Ejemplo

Se tiene un grupo de enfermos al que se aplican dos tratamientos $\boxed{I}$ y $\boxed{II}$. A continuación se observa si sobreviven $\boxed{S}$ o no $\boxed{M}$ obteniéndose la siguiente tabla

$Supervivencia \backslash Tratamiento$	I	II	
S	108	153	261
M	123	120	243
	231	273	**504**

De donde se obtiene que "parece mejor el tratamiento II" porque la proporción condicionada de supervivencia $\boxed{S}$ dado el tratamiento $\boxed{I}$, $\hat{p}_{S|I} = \frac{108}{231} = 0.47$ **es menor que** la de supervivencia $\boxed{S}$ dado el tratamiento $\boxed{II}$, $\hat{p}_{S|II} = \frac{153}{231} = 0.56$.

Pero resulta que si se obtiene información acerca del "*género*" del enfermo

se obtienen las siguientes tablas

MUJERES

$Supervivencia \backslash Tratamiento$	I	II	
S	57	32	89
M	100	57	157
	157	89	**246**

HOMBRES

$Supervivencia \backslash Tratamiento$	I	II	
S	51	121	172
M	23	63	86
	74	184	**258**

De forma que las proporciones condicionadas en cada subpoblación nos recomiendan el I en las dos subpoblaciones porque

$$\textbf{MUJERES} \quad : \quad \hat{p}_{S|I} = \frac{57}{157} = 0.363 \textbf{ mayor que } \hat{p}_{S|II} = \frac{32}{89} = 0.36.$$
$$\textbf{HOMBRES} \quad : \quad \hat{p}_{S|I} = \frac{51}{74} = 0.689 \textbf{ mayor que } \hat{p}_{S|II} = \frac{121}{184} = 0.657.$$

Este es un ejemplo de la "Paradoja de Simpson", cuya causa habrá que explicar en cada caso, pero que en general nos previene frente a la tendencia de agrupar subpoblaciones, con sus características particulares en una única poblacion.

En el ejemplo se puede observar que los **HOMBRES** sobreviven más que las **MUJERES** con los dos tratamientos y este efecto, al incluir más **HOMBRES** en el II, se traslada a la población total.

7. ESTUDIO DE LA HIPÓTESIS DE NORMALIDAD

En muchos procedimientos de inferencia se necesita considerar observaciones de una población Normal para garantizar la eficiencia de los mismos. Así, por ejemplo, en el modelo de regresión lineal se ha supuesto que los errores son Normales. En este apartado se tratarán diversas técnicas que permiten validar dicho supuesto.

7.1. Normalidad univariante

Sean $(x_1, \ldots, x_n)$ observaciones, supuestamente, de una $N(\mu, \sigma)$. Para comprobarlo se pueden utilizar procedimientos gráficos o contrastes de hipótesis.

7.1.1. Gráficos de probabilidad (QQ-plots)

En general son representaciones que enfrentan a los cuantiles teóricos y observados para evaluar patrones anómalos de comportamiento. Para construir el correspondiente gráfico de probabilidad o gráfico cuantil-cuantil, se deben seguir los siguientes pasos:

1.- Ordenar las oservaciones $x_{1:n} \leq, \ldots \leq x_{j:n} \leq \ldots \leq x_{n:n}$ obteniendo así los **cuantiles observados**.

2.- Calcular $\frac{1-\frac{1}{2}}{n}, \ldots, \frac{j-\frac{1}{2}}{n}, \ldots, \frac{n-\frac{1}{2}}{n}$.

3.- Calcular los **cuantiles teóricos** —de una $N(0,1)$—, $q_{1:n}, \ldots$, $q_{j:n}, \ldots, q_{n:n}$ tales que $\Phi(q_{j:n}) = \frac{j-\frac{1}{2}}{n}$, donde $\Phi(x)$ es la función de distribución de la $N(0,1)$.

4.- Representar $(q_{j:n}, x_{j:n})$, $j = 1, \ldots, n$ examinando la obligada "rectitud" de las observaciones si la hipótesis de normalidad es cierta.

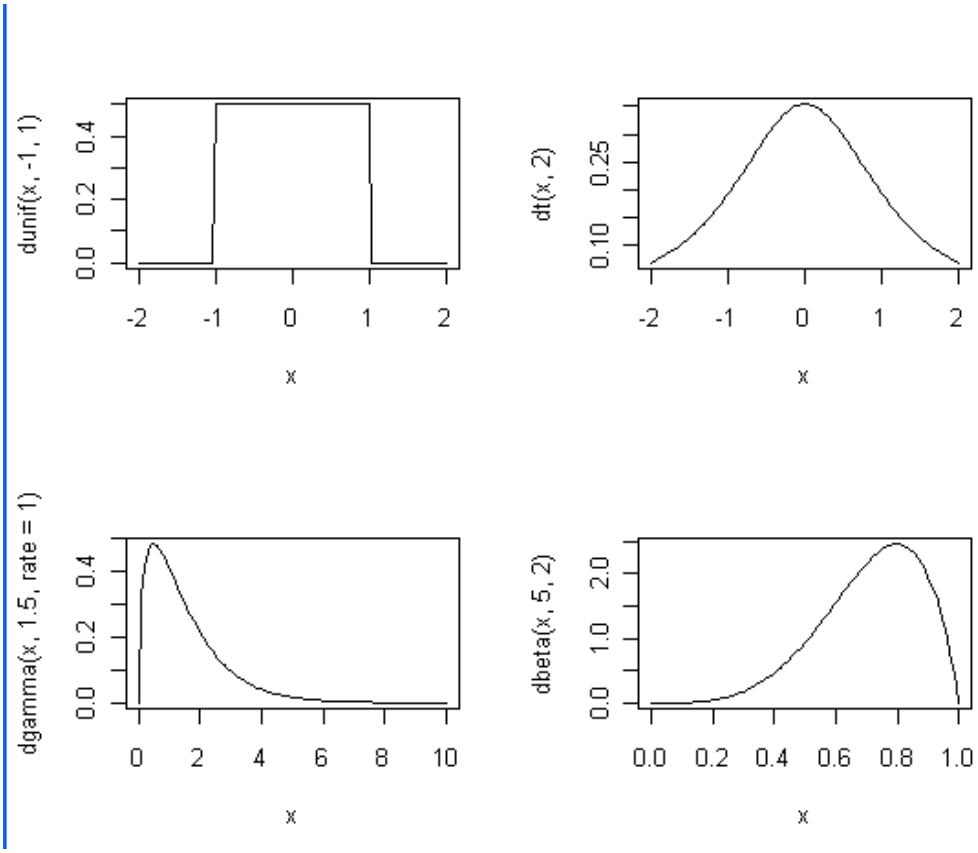
Nota: Si $(x_1, \ldots, x_n)$ son observaciones de una $N(\mu, \sigma)$ en general, con función de distribución $F(x)$, se tiene que $F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right)$, por lo que si $q^*_{j:n}$ es tal que $F\left(q^*_{j:n}\right) = \frac{j-\frac{1}{2}}{n}$ y como $F\left(q^*_{j:n}\right) = \Phi\left(\frac{q^*_{j:n}-\mu}{\sigma}\right)$, resulta que $\frac{q^*_{j:n}-\mu}{\sigma} = q_{j:n}$ de donde $(q_{j:n}, x_{j:n})$ deben de estar sobre la recta $y = \sigma x + \mu$.

Patrones de no normalidad

Este método tiene la venataja de que además de detectar no normalidad, dá una idea de como es la distribución No Normal correspondiente a las observaciones. Si simulamos observaciones de una uniforme $U(-1,1)$, una $t-Student$, una Gamma $\gamma(p=1{,}5, a=1)$ y una Beta $B(p=9, q=2)$

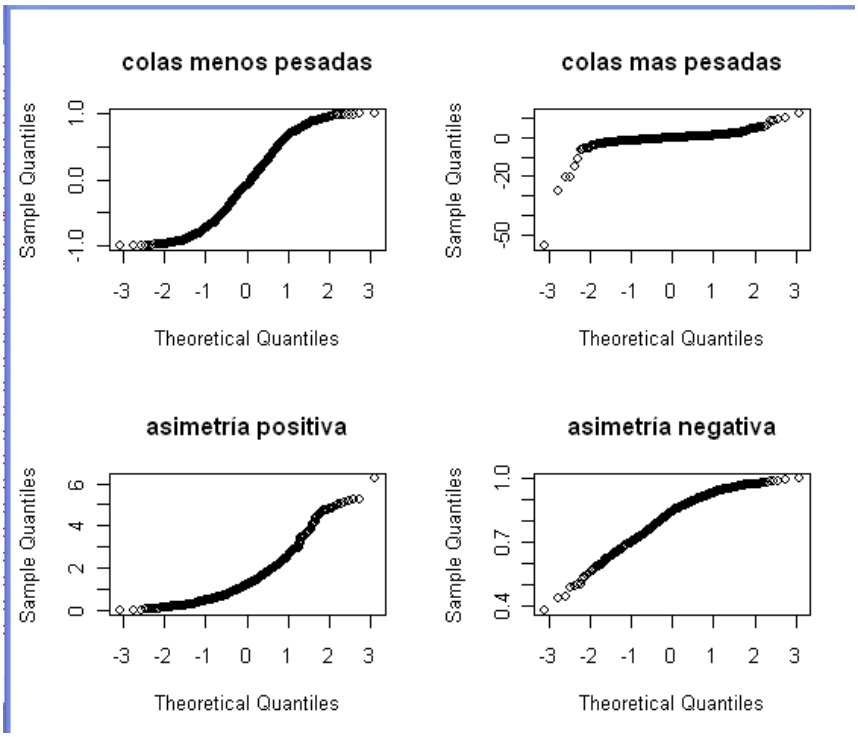
```
munif<-runif(500,-1,1)
mt<-rt(500,2)
mgamma<-rgamma(500,1.5,rate=1)
mbeta<-rbeta(500,9,2)
```

con funciones de densidad



Gráficos de Probabilidad con R

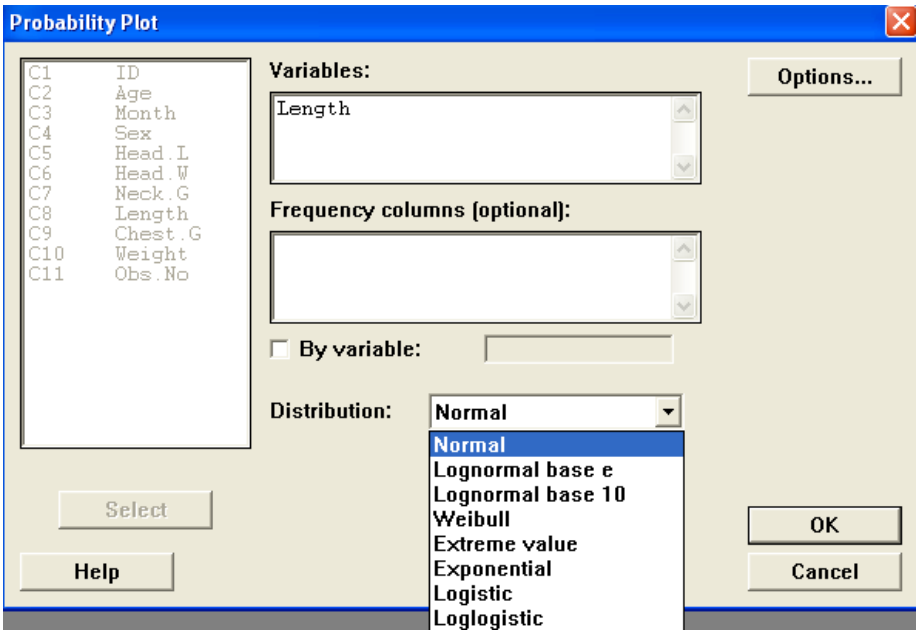
```
qqnorm(munif,main="colas menos pesadas")
qqnorm(mt,main="colas mas pesadas")
qqnorm(mgamma,main="asimetria positiva")
qqnorm(mbeta,main="asimetria negativa")
```



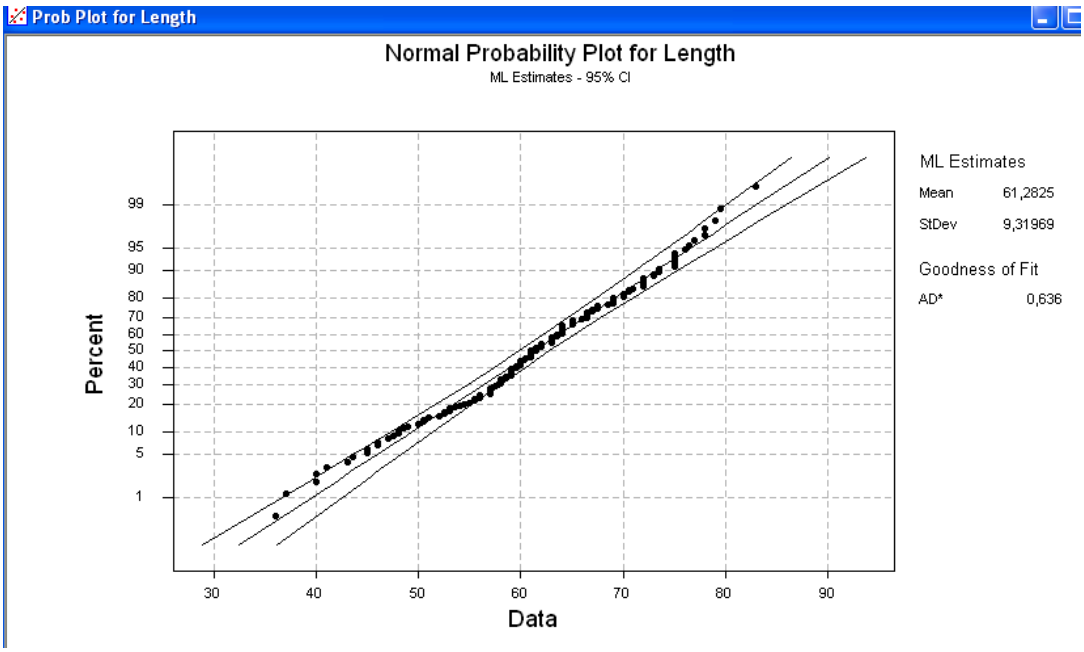
Gráficos de Probabilidad con MINITAB

Hay dos posibilidades:

a) **Graph→Probability Plot** y se abrirá un ventana de diálogo que permite seleccionar entre varias distribuciones, no solo la Normal.

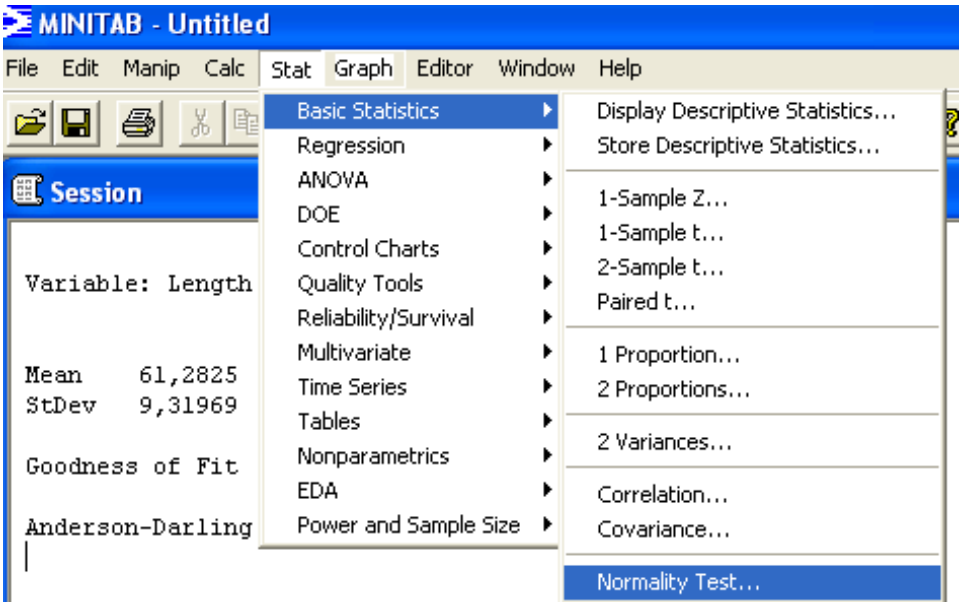


obteniéndose el gráfico

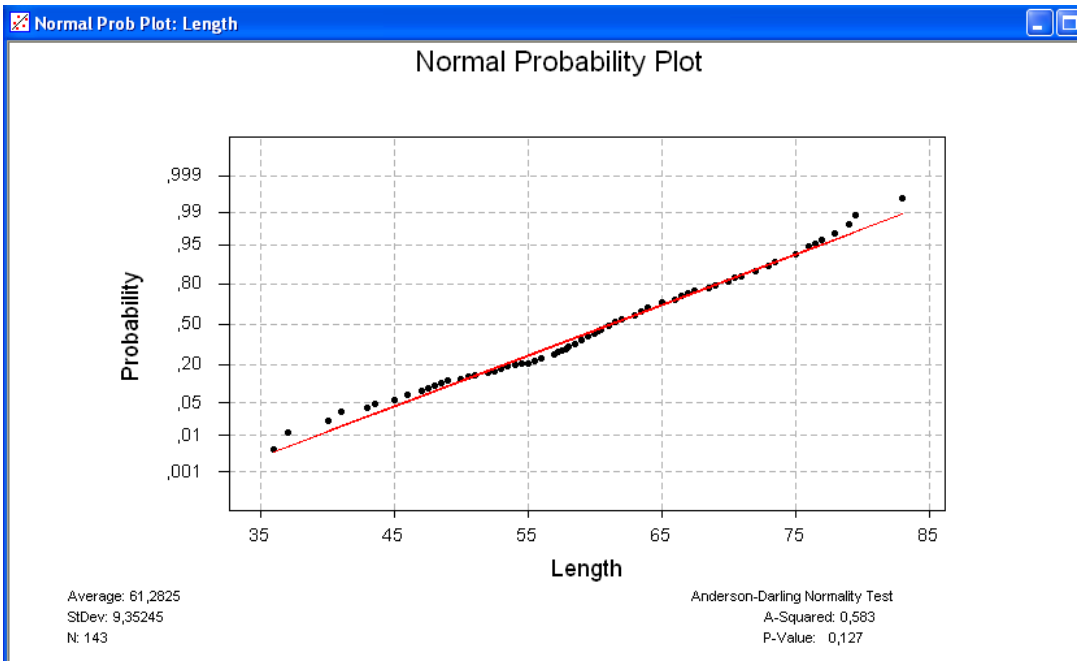


junto con información sobre la bondad del ajuste a la distribución seleccionada.

b) La persiana **Stat**



que proporciona el siguiente gráfico



7.1.2. Contrastes de bondad de ajuste

Los contrastes más utilizados para evaluar el ajuste a una Normal $N(\mu, \sigma)$ son los siguientes:

Contraste de la χ^2

Sean $\{A_1, \ldots, A_k\}$ tales que $\cup A_i = \mathbb{R}$ y $A_i \cap A_j = \emptyset$ para $i \neq j$, donde $A_i = (e_{i-1}, e_i]$.

Se determinan:

n_i, el número de observaciones de entre $(x_1, \ldots, x_n)$ que están en $A_i, i = 1, \ldots, k$, son las frecuencias observadas,

$p_i^0 = P_{N(0,1)}\{A_i\} = \Phi\left(\frac{e_i - \mu}{\sigma}\right) - \Phi\left(\frac{e_{i-1} - \mu}{\sigma}\right)$, donde $\Phi(x)$ es la función de distribución de la $N(0,1)$,

$n_i^0 = np_i^0$, son las frecuencias teóricas.

Estadístico

$$V = \sum_{i=1}^{k} \frac{(n_i - n_i^0)^2}{n_i^0}$$

Región crítica: Rechazar la hipótesis de normalidad a nivel α si $V > \chi^2_{k-1;\alpha}$.

Contraste de Kolmogorov-Smirnov

Se determina la **Función de Distribución Empírica**, $F_n^*(x)$ definida a partir de las observaciones ordenadas $x_{1:n} \leq \ldots \leq x_{n:n}$ mediante

$$F_n^*(x) \begin{cases} 0 & x < x_{1:n} \\ \frac{j}{n} & x_{j:n} \leq x < x_{j+1:n} \\ & j = 1, \ldots, n-1 \\ 1 & x \geq x_{n:n} \end{cases}$$

Estadístico

$$\begin{aligned} D_n &= \sup_x \left| F_n^*(x) - \Phi\left(\frac{x-\mu}{\sigma}\right) \right| = \text{máx}\left\{D_n^+, D_n^-\right\} \\ D_n^+ &= \sup_x \left[F_n^*(x) - \Phi\left(\frac{x-\mu}{\sigma}\right) \right] \\ D_n^- &= \sup_x \left[\Phi\left(\frac{x-\mu}{\sigma}\right) - F_n^*(x) \right] \end{aligned}$$

Región crítica: Rechazar la hipótesis de normalidad a nivel α si $D_n > D_{n;\alpha}$.

Ejercicio con R

Simulando 100 observaciones de una $N(0,1)$, se les aplican los contrastes anteriores.

Contraste de la χ^2

```
mnorm1<-rnorm(100)
hist(mnorm1)
cortes<-c(-Inf,seq(-1.2,1.2,by=0.3),Inf)
#creamos un factor asociando cada observacion a un intervalo
mnorm1.cut<-cut(mnorm1,breaks=cortes)
tmnorm1<-table(mnorm1.cut)
#frecuencias esperadas
length(tmnorm1)
frespe=NULL
for(i in 1:length(tmnorm1)) frespe[i]<-100*(pnorm(cortes[i+1])
-pnorm(cortes[i]))
sum(frespe)
#frecuencias observadas
frobs=vector()
for(i in 1:10) frobs[i]<-tmnorm1[i]
sum(frobs)
#construccion del estadistico
estmnorm1<- sum(((frobs-frespe)^2)/frespe)
```

```
gdl<-length(frespe)-1
pval<-1-pchisq(estmnorm1,gdl)

#utilizacion de chisq.test
#construccion del vector de probabilidades teoricas
prob=vector()
for(i in 1:10) prob[i]<-(pnorm(cortes[i+1])-pnorm(cortes[i]))
chisq.test(x=frobs,p=prob)
```

que produce la siguiente salida

```
> pval
[1] 0.9640306
> chisq.test(x=frobs,p=prob)

        Chi-squared test for given probabilities

data:  frobs
X-squared = 3.0067, df = 9, p-value = 0.964
```

Contraste de Kolmogorov-Smirnov

```
#construccion del estadistico
empi<-seq((1/length(mnorm1)),1,by=(1/length(mnorm1)))
smnorm1<-sort(mnorm1)
F0<-pnorm(smnorm1)
dif.plus<-max(empi-F0)
dif.menos<-max(F0-empi)
dif<-max(dif.plus,dif.menos)

#utilizacion de ks.test
ks.test(mnorm1,"pnorm")
```

que produce la siguiente salida

```
> dif
[1] 0.06579159
> ks.test(mnorm1,"pnorm")

        One-sample Kolmogorov-Smirnov test

data:  mnorm1
D = 0.0658, p-value = 0.7797
alternative hypothesis: two-sided
```

7.2. Normalidad Bivariante

Se utilizará el siguiente resultado teórico relativo a la distribución de una variable aleatoria X con distribución Normal $p-$variante $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ donde $\boldsymbol{\Sigma}$ es definida positiva.

$$(X-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (X-\boldsymbol{\mu}) \text{ es una variable aleatoria } \chi^2_p.$$

7.2.1. Procedimientos

a) Como se tiene que

$$P\left\{(X-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (X-\boldsymbol{\mu}) \leq \chi^2_{p;\alpha}\right\} = 1-\alpha$$

se puede considerar que, aproximadamente y para $p=2$, el $100(1-\alpha)\,\%$ de las observaciones $x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \end{pmatrix}$ "deben quedar dentro" de la elipse

$$(x-\bar{\mathbf{x}})' \mathbf{S}^{-1} (x-\bar{\mathbf{x}}) = \chi^2_{2;\alpha}.$$

Es un procedimiento bastante inestable pero que puede ser útil en un primer paso, para detectar grandes desviaciones de la normalidad.

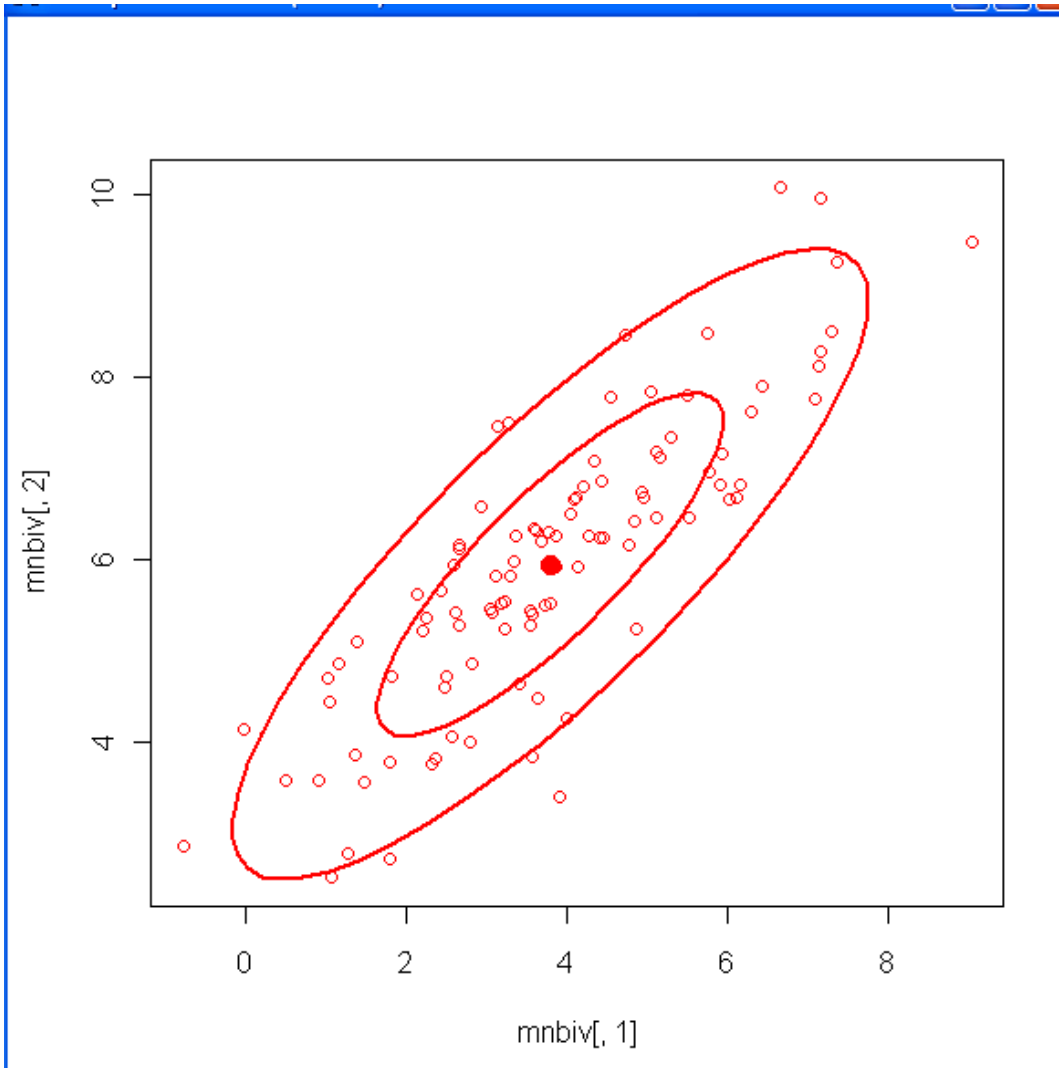
Ejemplo

Se simulan 100 observaciones de una $N_2\left(\begin{pmatrix} 4 \\ 6 \end{pmatrix}, \begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix}\right)$, por lo que se deben satisfacer las condiciones para $\alpha = 0.5$ y $\alpha = 0.1$. Efectivamente, mediante las siguientes funciones de R.

```
library(MASS)
library(car)
mu<-c(4,6)
sigma<-matrix(c(3,2,2,2),ncol=2)
mnbiv<-mvrnorm(100,mu,sigma)
plot(mnbiv)
samplemean<-apply(mnbiv,2,mean)
cov(mnbiv)
dato<-vector(length=100)
for (i in 1:100)
{
dato[i]<-
t(mnbiv[i,]-samplemean)%*%solve(cov(mnbiv))%*%(mnbiv[i,]
-samplemean)
```

```
}
hist(dato,freq=F,ylim=c(0,0.5))
curve(dchisq(x,2),add=T)
data.ellipse(mnbiv[,1],mnbiv[,2])
comp1<-dato>qchisq(0.5,df=2,lower.tail=FALSE)
length(dato[comp1=="TRUE"])
comp2<-dato>qchisq(0.10,df=2,lower.tail=FALSE)
length(dato[comp2=="TRUE"]
```

Una forma de visualizar los resultados sobreponiendo, a los puntos que representan las observaciones, las elipses correspodientes a $\alpha = 0.5$ y $\alpha = 0.1$.



permite comprobar la salida que , efectivamente, no rechaza la normalidad

```
> length(dato[comp1=="TRUE"])
[1] 52
> comp2<-dato>qchisq(0.10,df=2,lower.tail=FALSE)
> length(dato[comp2=="TRUE"])
```

```
[1] 12
```

y

b) Si las observaciones $X = \begin{pmatrix} x_1' \\ \vdots \\ x_n' \end{pmatrix}$ corresponden a una Normal $N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ y se determinan las siguientes distancias de Mahalanobis de cada observación x_j a la media $\bar{x}$,

$$d_j^2 = (x_j - \bar{x})' S^{-1} (x_j - \bar{x}) ; j = 1, \ldots, n$$

los valores $d_1^2, \ldots, d_n^2$ deben ser observaciones de una χ_2^2 (suponiendo que n y $n-p$ son mayores que 30). Por tanto se construirá un Gráfico de Probabilidad de ajuste a una χ_2^2.

1.- Ordenar las oservaciones $d_{1:n} \leq, \ldots \leq d_{j:n} \leq \ldots \leq d_{n:n}$ obteniendo así los **cuantiles observados**.

2.- Calcular $\frac{1-\frac{1}{2}}{n}, \ldots, \frac{j-\frac{1}{2}}{n}, \ldots, \frac{n-\frac{1}{2}}{n}$.

3.- Calcular los **cuantiles teóricos** de una χ_2^2, $q_{1:n}, \ldots, q_{j:n}, \ldots, q_{n:n}$ tales que $F_{\chi_2^2}(q_{j:n}) = \frac{j-\frac{1}{2}}{n}$, donde $F_{\chi_2^2}(x)$ es la función de distribución de la χ_2^2.

4.- Representar $(q_{j:n}, d_{j:n})$, $j = 1, \ldots, n$ examinando la obligada "rectitud" de las observaciones si la hipótesis es cierta.

8. TRANSFORMACIÓN DE VARIABLES

8.1. Transformaciones lineales

8.1.1. Datos univariantes

Sean $(x_1, \ldots, x_n)$, si se define $y_i = ax_i + b, i = 1, \ldots, n$ se obtiene:

$\bar{y} = a\bar{x} + b$

$s_y^2 = a^2 s_x^2$

$M_{e_Y} = aM_{e_X} + b$ (mediana)

$M_{o_Y} = aM_{o_X} + b$ (moda)

$y_{i:n} = ax_{i:n} + b, , i = 1, \ldots, n$

$rango_Y = a\ rango_X$

$g_{1_Y} = g_{1_X}$ (coeficiente de asimetría)

$k_Y = k_x$ (coeficiente de curtosis)

8.1.2. Datos bivariantes

Sean $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, si se definen $\{(v_1, w_1), \ldots, (v_n, w_n)\}$ de forma que

$$\begin{aligned} v_i &= ax_i + b \\ w_i &= cy_i + d \end{aligned}$$

se obtiene:

$s_{vw} = acs_{xy}$ (covarianza muestral)

$r_{vw} = r_{xy}$ (coeficiente de correlación lineal)

$\beta_{W|V} = \frac{c}{a}\beta_{Y|X}$ (coeficiente de regresión)

8.2. Tranformaciones hacia la normalidad

8.2.1. Logit

Si $0 < x_i < 1$,

$$logit\,(x_i) = \log \frac{x_i}{1 - x_i}$$

de forma que -$\infty < logit\,(x_i) < \infty$.

8.2.2. Probit

Si $0 < x_i < 1$ y se corresponden con probabilidades de una Normal,

$$probit\,(x_i) = \Phi^{-1}\,(x_i)$$

siendo $\Phi\,(x)$ la función de distribución de una $N\,(0, 1)$.

8.2.3. Box-Cox

Son las más utilizadas y se definen,

$$x_i^{(\lambda)} \begin{cases} \frac{x_i^{\lambda}-1}{\lambda} & \text{si } \lambda \neq 0 \\ \ln x_i & \text{si } \lambda = 0 \end{cases}$$

estimando el valor de λ, por el método de la máxima verosimilitud.

Ejemplo con R

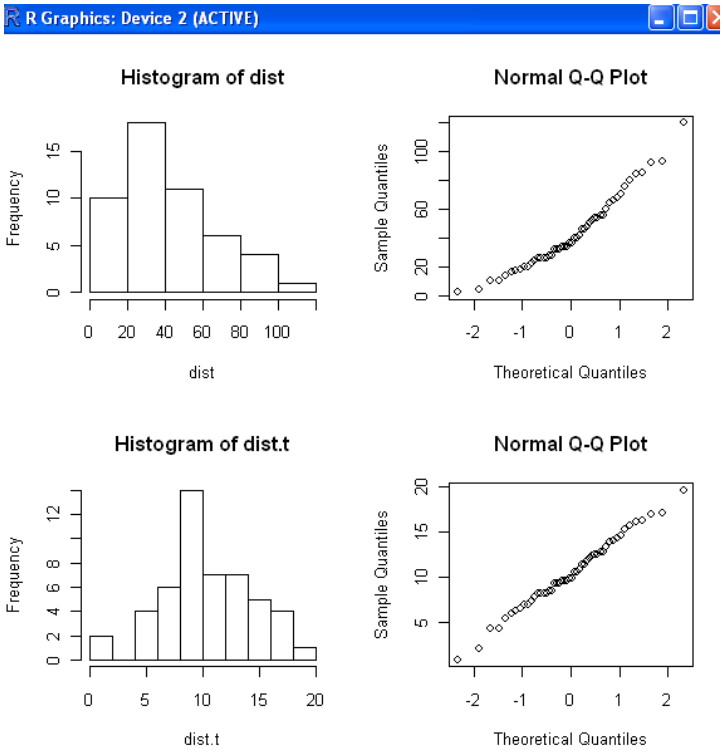
#Datos "cars" cuya variable "dist" no se ajusta a una normal

```
library(car)
attach(cars)
par(mfrow=c(2,2))
hist(dist)
qqnorm(dist)
ks.test(dist,"pnorm")
```

#Se determina el valor de λ y se transforman los datos

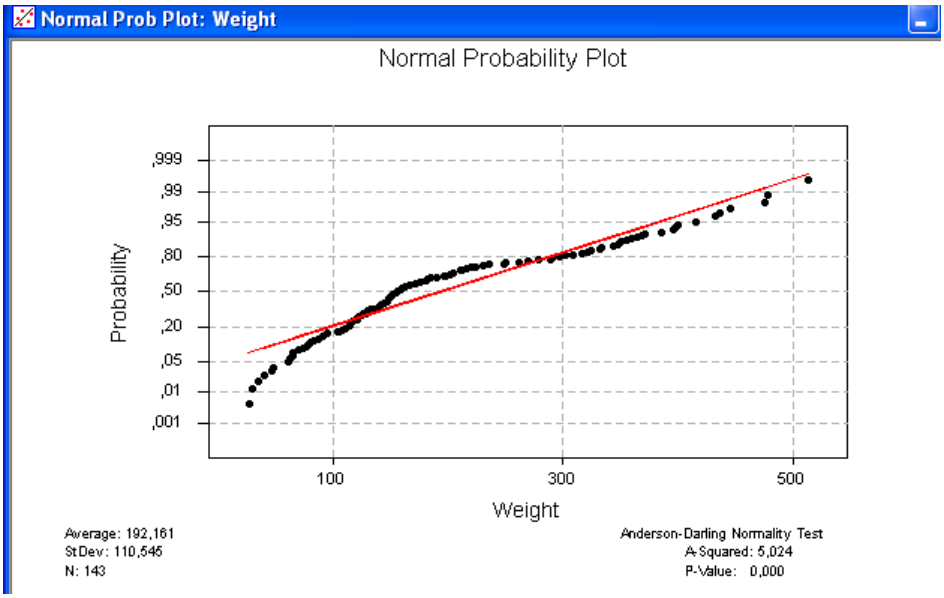
```
box.cox.powers(dist)
dist.t<-box.cox(dist,0.495)
hist(dist.t)
qqnorm(dist.t)
```

Los gráficos resultantes representan la capacidad del proceso de transformación

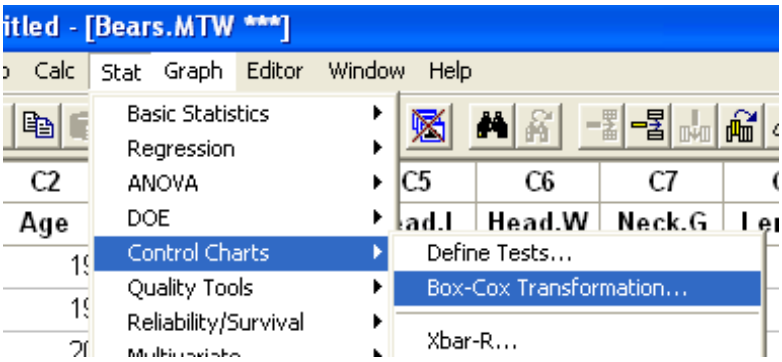


Ejemplo con MINITAB

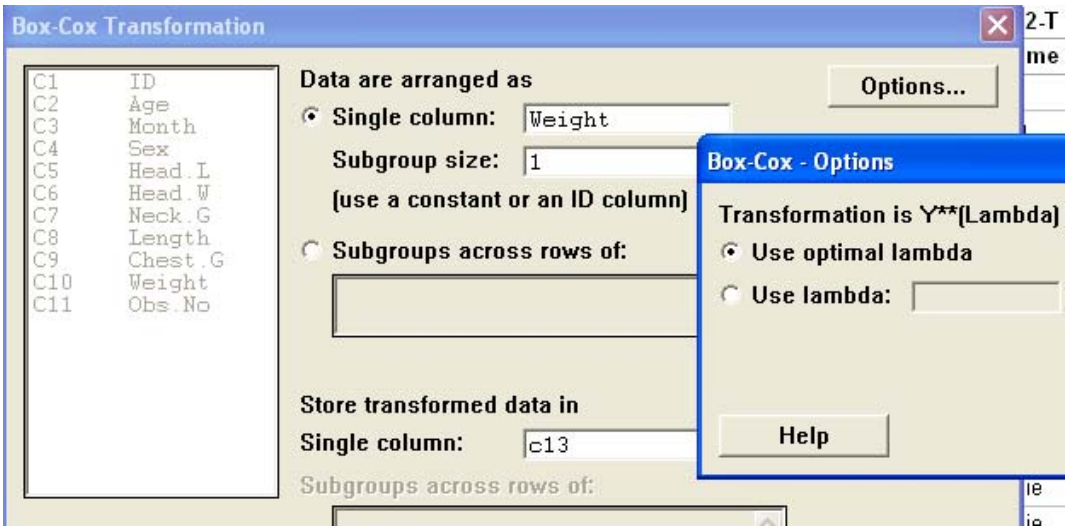
En los datos "Bears.MTW", la variable "Weight" no presenta un patron de normalidad

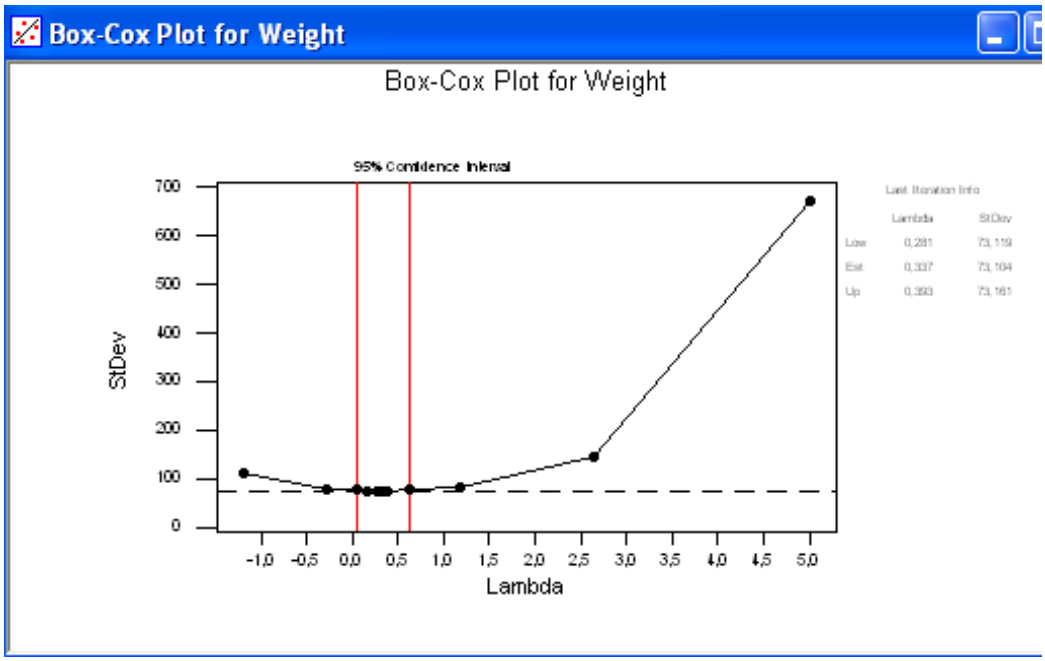


si se abre la persiana **Stat**

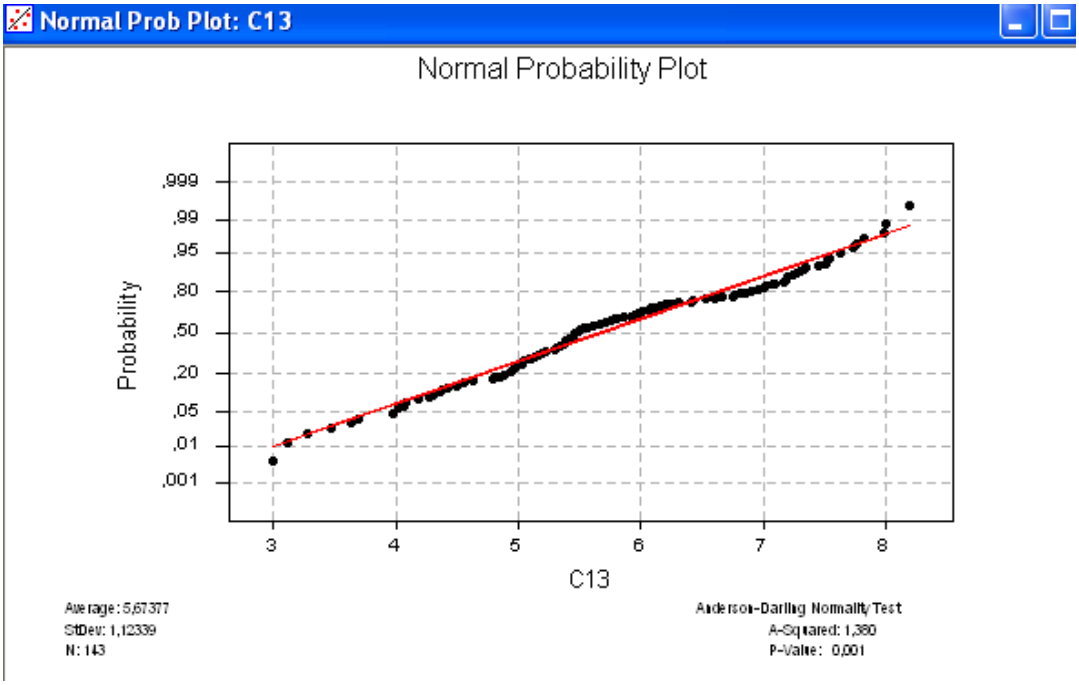


se abre una ventana diálogo que nos permite imponer o calcular el valor de λ

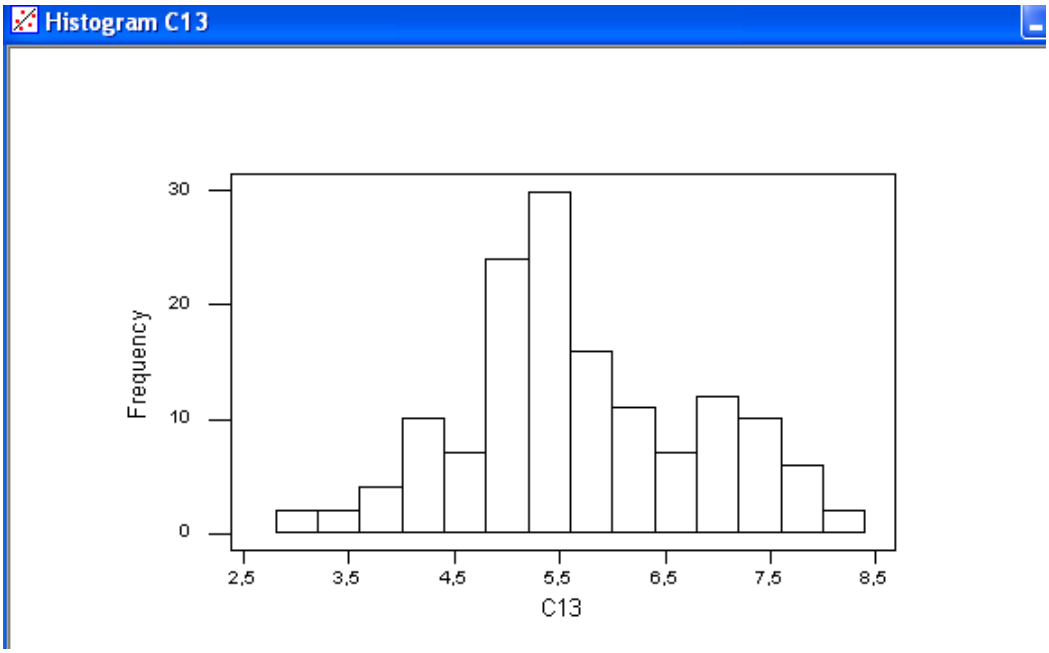




de forma que obteniendo el QQ-plot para los datos transformados se obiene



que junto con su histograma



refleja los efectos de la transformación.

9. DATOS ATÍPICOS Y AUSENTES

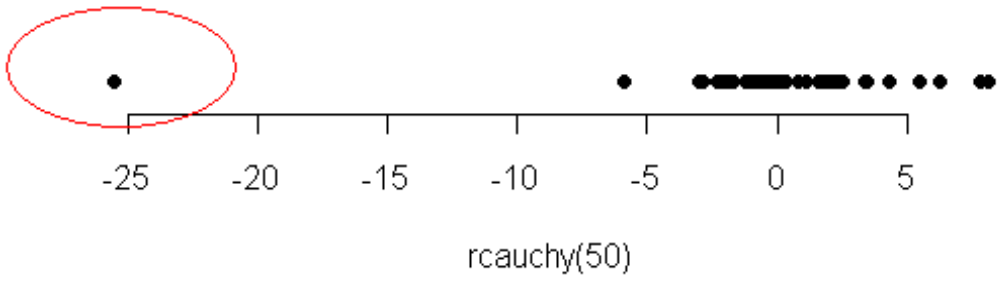
Son observaciones que, de algún modo afectan a las conclusiones estadísticas que se pueden extraer de una base de datos. Las definiciones pueden considerarse:

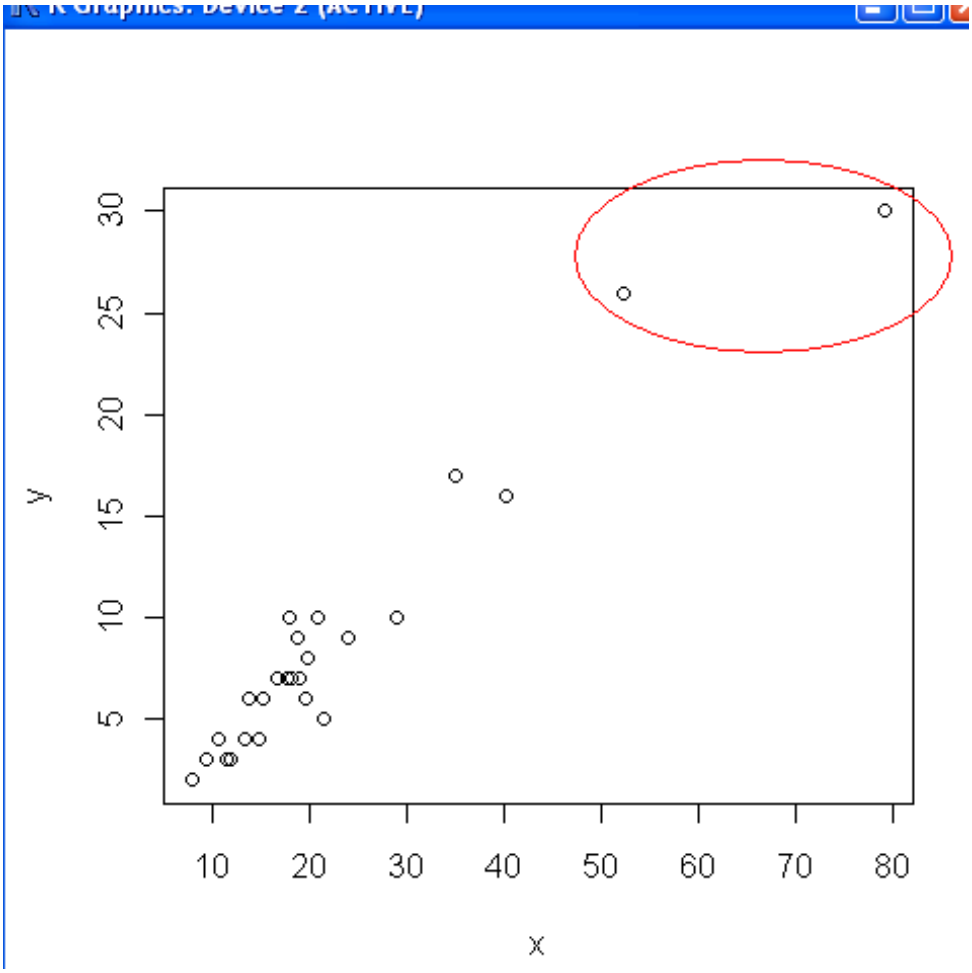
Datos atípicos (outliers): Observaciones separadas del bloque general de los valores correspondientes en la base de datos.

Datos ausentes (missing): Son los huecos de la base de datos, es decir aquéllas observaciones que, por alguna razón, no se han anotado.

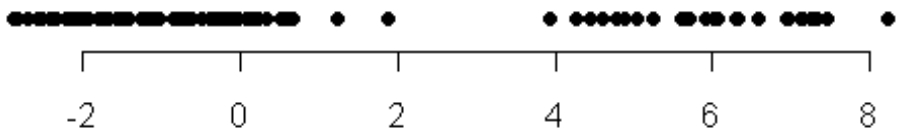
9.1. Tratamiento de datos atípicos

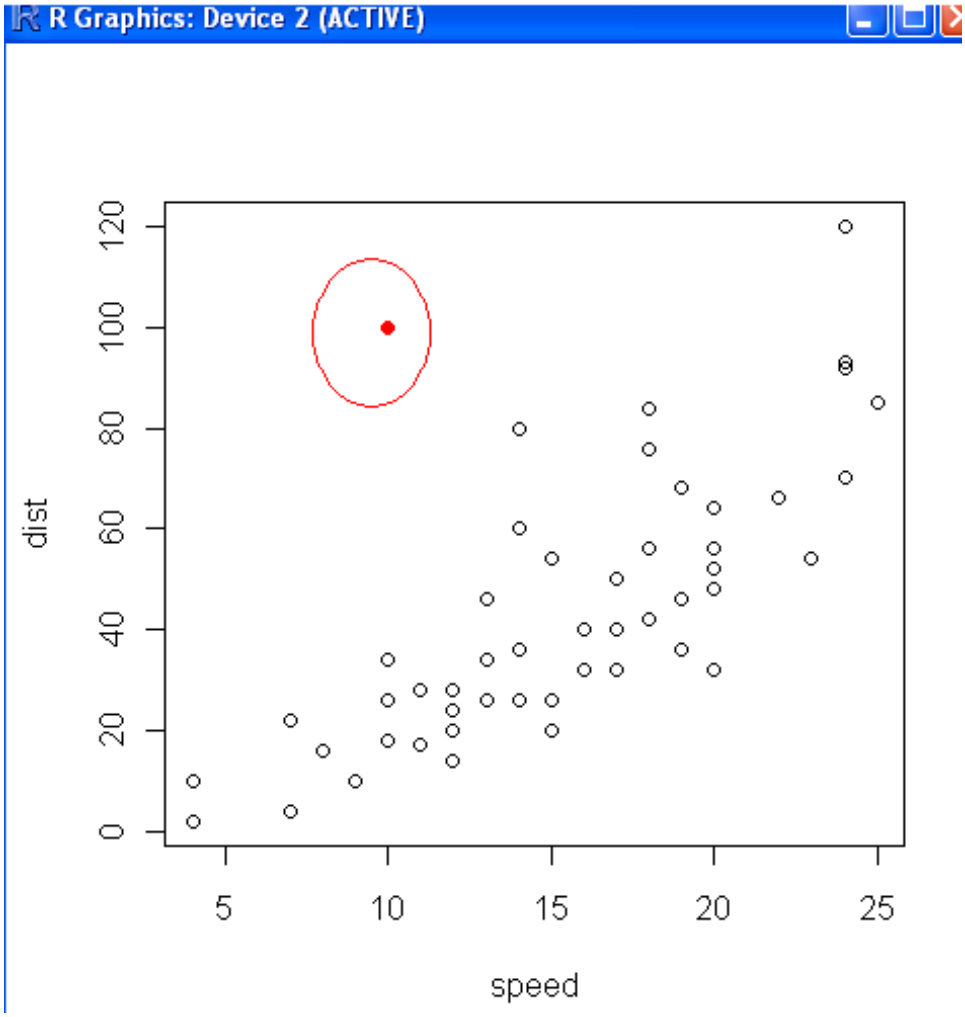
Los datos atípicos pueden ser extremos





o por el contrario





pero sí pueden causar problemas en la determinación de las conclusiones finales.

Las causas pueden ser desde errores fortuitos hasta la existencia de otros modelos subyacentes mucho más interesantes, en algunas ocasiones, que el resto del estudio. En todo caso es recomendable el anásis de los mismos en lugar de la eliminación directa, como ha sido la forma de tratarlos en el pasado.

Los procedimientos son muchos y a veces muy específicos del problema a tratar, pero pueden englobarse en dos grandes bloques: Métodos de detección y Procedimientos de acomodación.

9.1.1. Métodos de detección

Gráficos: Diagramas de cajas, gráficos de puntos o nubes de puntos.

Contrastes de discordancia: Basados en estadísticos que utilizan las diferencias entre las observaciones ordenadas, tipo los de **Dixon**

$$\frac{x_{s:n} - x_{r:n}}{x_{q:n} - x_{p:n}}$$

o los de **Grubbs**

$$\frac{\sum_{i=1}^{n-2}\left(x_{i:n}-\bar{x}_{n,n-1}\right)^2}{\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2}$$

donde $\bar{x}_{n,n-1}=\frac{\sum_{i=1}^{n-2}x_{i:n}}{n-2}$. En muchos de estos contrastes puede darse el "efecto de enmascaramiento" que puede ocultar la presencia de un dato extremo, por el efecto de una observación cercana y también extrema.

9.1.2. Procedimientos de acomodación

Se aplican cuando se considera que los datos atípicos deben incorporarse en la obtención de las conclusiones finales, pero estando preparados para minimizar sus efectos adversos. Por tanto aplicando métodos que se denominan **robustos** frente a observaciones anómalas.

Así, por ejemplo, para la estimación de la localización en observaciones unvariantes, se utilizarán:

L-estimadores: Combinacnes lineales ponderadas de las observaciones ordenadas.

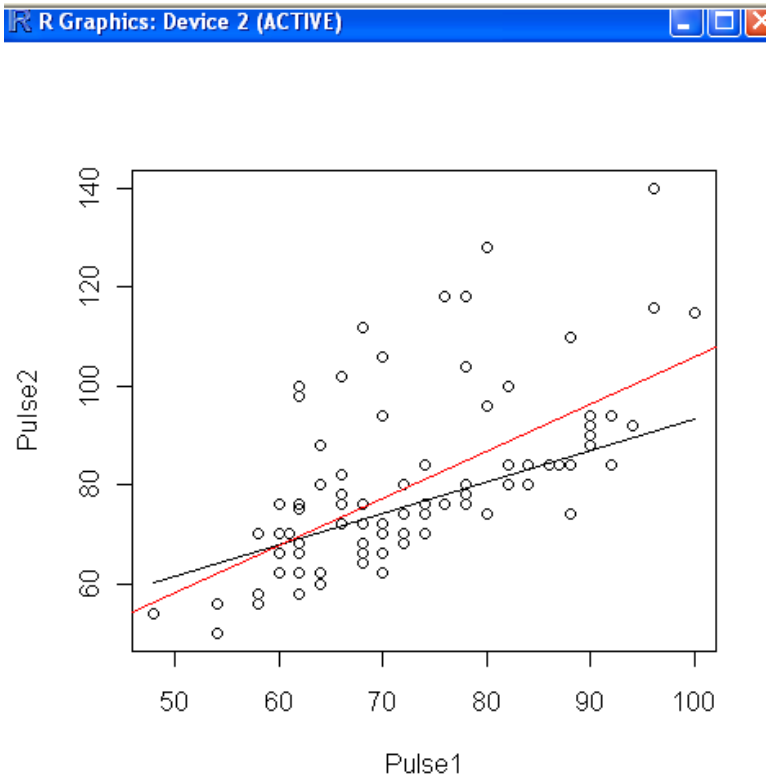
M-estimadores: Generalización del método de la Máxima Verosimilitud.

R-estimadores: Basados en los rangos de las observaciones.

En los problemas de regresión lineal se suele recurrir a la determinación de la **Recta Resistente** (Resistant Line), que mediante un proceso iterativo construye una recta de regresión, reduciendo el efecto de los posibles datos atípicos.

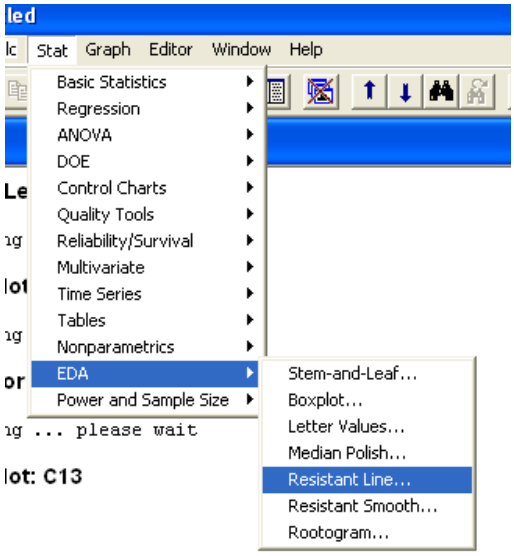
Ejemplo con R

```
library(LearnEDA)
help(rline)
peru<-
read.table("http://www.mat.ucm.es/~palomam/aed/datos/pulse.txt"
,header=T)
attach(peru)
fit=rline(Pulse1,Pulse2)
plot(Pulse1,Pulse2)
curve(fit$a+fit$b*(x-fit$xC),add=TRUE)
abline(lm(Pulse2~Pulse1),col="red")#recta de minimos cuadrados
```



Ejemplo con MINITAB

Abriendo la persiana **Stat**



y se lo aplicamos a las variables Pulse1 y Pulse2 de los datos Peru.MTW de MINITAB.

Results for: Pulse.MTW

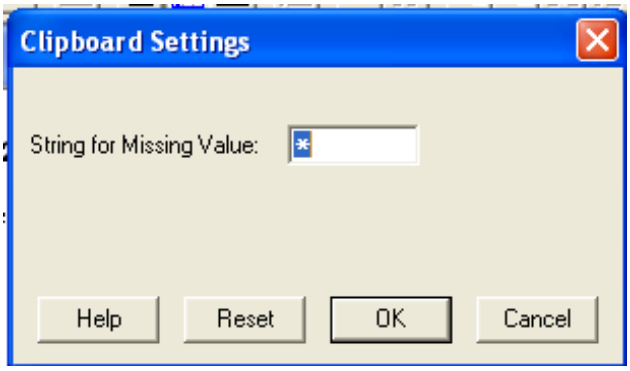
Resistant Line Fit: Pulse2 versus Pulse1

```
Slope    =       0,7500 Level =      20,5000 Half-slope ratio =    1,250
```

9.2. Tratamiento de datos ausentes

Este tipo de observaciones aparecen con distintos tipos de símbolos:

MINITAB se representan por defecto como *, pero puede cambiarse si se abre **Editor → Clipboard Settings**



En R es NA (Not Available).

Los mecanismos que han producido dichas observaciones suelen denominarse:

MCAR (Missing Completely at Random) debidos totalmente al azar.
MAR (Missing at Random) según los tipos de datos.
MNAR (Missing not at Random) no debido a azar.

En la mayoría de los procedimientos se suele poder seleccionar el tratamiento que se les quiere dar.que va desde el rechazo hasta la imputación de valores que podrían haber sido los observados en esos huecos De todas formas en un

análisis inicial de los datos habrá que estudiarlos y en su caso omitirlos o sustituirlos por valores "posibles".

Dentro de R se pueden utilizar, para empezar, las siguientes funciones:

NA y *missing* en el paquete **base** para detectarlos y manipularlos.

knn en el paquete **EMV** que estima la mejor forma de rellenar los huecos en una matriz.

Ejercicio 1

```
row.names(airquality[is.na(airquality[,1]),])
is.na(airquality[,1])
z1<-na.omit(airquality[,1])
na.action(z1)
z2<-na.pass(airquality[,1])
```

cuya salida es

```
> row.names(airquality[is.na(airquality[,1]),])
[1] "5"   "10"  "25"  "26"  "27"  "32"  "33"  "34"  "35"  "36"
  "37"  "39"
[13] "42"  "43"  "45"  "46"  "52"  "53"  "54"  "55"  "56"  "57"
  "58"  "59"
[25] "60"  "61"  "65"  "72"  "75"  "83"  "84"  "102" "103" "107"
 "115" "119"
[37] "150"
> is.na(airquality[,1])
```

[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE

[13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[25] TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE

[37] TRUE FALSE TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE FALSE FALSE

[49] FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[61] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE

[73] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE

[85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[97] FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE FALSE

[109] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE

[121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

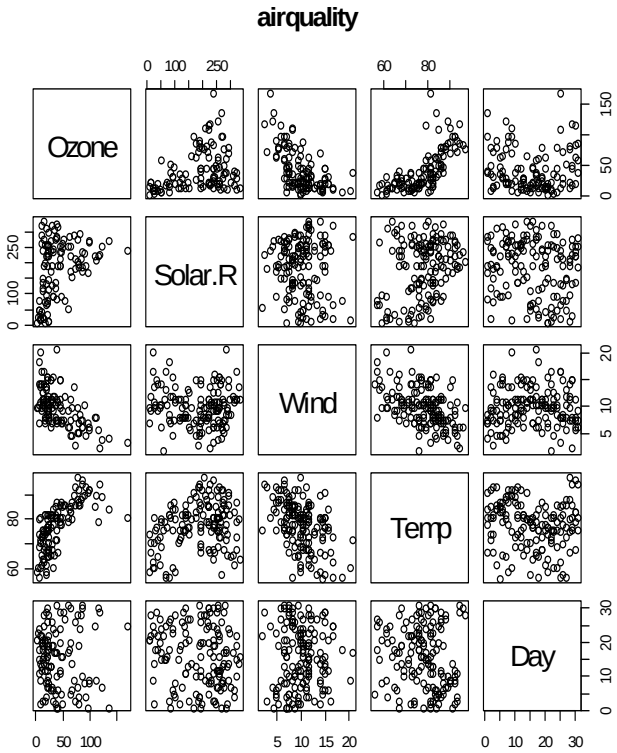
[145] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE

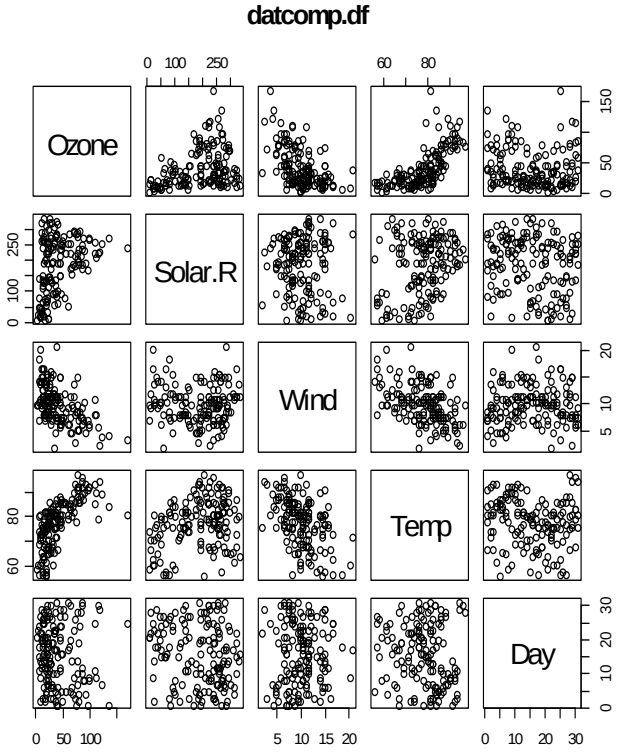
```
> z1<-na.omit(airquality[,1])
> na.action(z1)
 [1] 5 10 25 26 27 32 33 34 35 36 37 39 42 43 45 46 52 53 54
[20] 55 56 57 58 59 60 61 65 72 75 83 84 102 103 107 115 119 150
attr(,"class")
[1] "omit"
> z2<-na.pass(airquality[,1])
>
> z2
 [1] 41 36 12 18 NA 28 23 19 8 NA 7 16 11 14 18 14 34 6
 [19] 30 11 1 11 4 32 NA NA NA 23 45 115 37 NA NA NA NA NA
 [37] NA 29 NA 71 39 NA NA 23 NA NA 21 37 20 12 13 NA NA NA
 [55] NA NA NA NA NA NA NA 135 49 32 NA 64 40 77 97 97 85 NA
 [73] 10 27 NA 7 48 35 61 79 63 16 NA NA 80 108 20 52 82 50
 [91] 64 59 39 9 16 78 35 66 122 89 110 NA NA 44 28 65 NA 22
[109] 59 23 31 44 21 9 NA 45 168 73 NA 76 118 84 85 96 78 73
[127] 91 47 32 20 23 21 24 44 21 28 9 13 46 18 13 24 16 13
[145] 23 36 7 14 30 NA 14 18 20
```

Ejercicio 2

```
library(EMV)
mairquality<-as.matrix(airquality)
knn(mairquality)
datcomp<-knn(mairquality)$data
datcomp.df<-data.frame(datcomp)
names(datcomp.df)<-names(airquality)
pairs(airquality[,-5],main=‘‘airquality’’)
pairs(datcomp.df[,-5],main=‘‘datcomp.df’’)
```

El efecto del “rellenado” se puede observar en los siguientes gráficos


airquality
Ozone
Solar.R
Wind
Temp
Day


datcomp.df
Ozone
Solar.R
Wind
Temp
Day