Capítulo 1

Computación Estadística y Estadística Computacional. Evolución histórica

Extraído del discurso leído por el Prof. Dr. D. Andrés González Carmona en su entrada como académico de la Academia de Ciencias Matemáticas, Físico - Químicas y Naturales de Granada en 1997.

1.1. Introducción

La primera pregunta que cabe plantearse ante las palabras *Estadística Computacional* es ¿Qué es la Estadística Computacional? La respuesta es lo que pretendo dar a continuación, estudiando para ello un panorama histórico de este concepto. Al mismo tiempo ir, introduciendo algunas hipótesis que pretendo confirmar a lo largo del discurso.

En una primera intención podemos decir que la Estadística Computacional es la Estadística calculada, por medios electrónicos fundamentalmente, lo cual incluye prácticamente todos los campos de aplicación de la Estadística. De hecho muchos aspectos de la Estadística están influidos por la Estadística Computacional e intentaremos ponerlos de relieve.

He dicho por medios electrónicos fundamentalmente, porque hoy en día este es el medio de cálculo prácticamente único existente, ya que por su potencia supera al resto, aunque en épocas anteriores fueron otros los medios

de cálculo disponibles: manuales, mecánicos, gráficos, etc.

Al hacer hincapié en los medios electrónicos puede pensarse por tanto que la Estadística Computacional surge a partir de la Informática. Sin embargo, casi puede decirse que es la Informática la que surge a partir de la Estadística. En efecto, por imperativo legal, en Estados Unidos, es necesario realizar un censo de población en cada uno de los Estados, para asignar el número de congresistas en función de la cuota de población. Este censo debe realizarse regularmente y a finales del siglo pasado se producía ya el hecho de que debía realizarse un nuevo censo y aún no se había acabado el anterior. Ello condujo a uno de los empleados del censo, Herman Hollerith, a la aplicación a este problema de una técnica que ya se usaba en otros campos: el uso de fichas perforadas para codificar los datos y poder posteriormente realizar estadísticas de modo sencillo, obteniendo un gran éxito en la realización del censo de 1880, lo que difundió estas máquinas mundialmente. Diseñó para ello incluso un código que lleva su nombre y que fue utilizado hasta tiempos recientes, mediados de los 70, y fundó una empresa para la aplicación del método. Puesto que de aquí partió la fundación de IBM, es por lo que anteriormente he indicado que la Informática surge a partir de la Estadística.

Dos conclusiones pueden derivarse de este primer ejemplo: La primera, que *la evolución a lo largo del tiempo de los métodos de cálculo interacciona con las técnicas y condiciona las mismas*, lo cual sería un caso particular del aforismo de que el lenguaje condiciona el pensamiento, y una segunda que *cambios de tipo cuantitativo, si son lo suficientemente importantes, producen cambios de tipo cualitativo.*

Este tipo de conclusiones pueden obtenerse de una multitud de ejemplos de cómo algunas técnicas estadísticas han sido postergadas o priorizadas en función de las herramientas de cálculo disponibles en cada momento. A título de ejemplo incluyo los dos siguientes:

Media y Mediana El cálculo manual de la media, bastante largo, aunque sólo incluye operaciones elementales, propició la introducción de otro par metro de posición central, la mediana, que en la distribución normal coincide con el anterior y que, en ese contexto, es de cálculo prácticamente inmediato, ya que la ordenación de pocos datos es muy rápida.

El advenimiento de los ordenadores y la aplicación de cálculos a volúmenes superiores de datos produce una postergación de la mediana, ya que el problema de la ordenación de datos exige capacidad de almacenamiento y un tiempo que crece en un orden superior al lineal con el tamaño de datos a ordenar.

Cuando los ordenadores aumentan de potencia, se vuelve a recuperar la mediana, pero ahora con una finalidad distinta, como es lógico, ya que su cálculo sigue siendo gravoso. Su cálculo se realiza ahora debido a sus propiedades de robustez frente a la existencia de datos anómalos en la muestra.

Papel probabilístico normal El papel probabilístico normal surgió como una herramienta de cálculo nomográfico para la media y la varianza de unos datos (basada adicionalmente en la hipótesis de que sólo existía una distribución: la normal). En este tipo de papel, los datos correspondientes a una distribución normal se representan como una recta que depende de dos par metros, la media y la desviación típica. Bastar por tanto con representar los datos, trabajo que manualmente es trivial, y dibujar la recta que los representa (método de Henri) para poder leer los valores de ambos par metros con una precisión bastante aceptable, similar a la obtenida realizando las operaciones con una regla de cálculo. Cuando aparecen los medios de cálculo electrónicos que suministran ambos par metros con mayor precisión y rapidez, prácticamente desapareció este papel como ocurrió con muchas de las técnicas nomográficas. Al avanzar las posibilidades de cálculo, han reaparecido con un nuevo significado. Ahora, se representan los datos en el papel, pero no se ajusta una recta, sino que se representa aquella que corresponde a la distribución normal de media y varianza las calculadas por medios aritméticos, y se utiliza como *gráfico de diagnóstico*, esto es, para detectar causas por las que unos datos no han superado el contraste de normalidad: datos anómalos, mezclas de poblaciones, asimetrías en los datos, etc.. e incluso se han ampliado a otras distribuciones de inter,s con los *Q-plots* y *QQ-plots*, dando además lugar a una nueva familia de contrastes de normalidad, alejados de la metodología de Pearson, como son los contrastes de Shapiro–Wilks–Francia o de D'Agostino, y planteando una serie de nuevos problemas[22].

1.2. Evolución histórica de los medios de cálculo

Puesto que las técnicas estadísticas han sido postergadas o priorizadas en función de las herramientas de cálculo disponibles en cada momento, procede estudiar, brevemente, algunos hitos en la evolución histórica de los medios de cálculo en los que haré hincapié sólo en algunos aspectos interesantes.

1.2.1. Los primeros medios de cálculo en la Estadística

A principios de este siglo, K. Pearson editaba la revista *Tracts on computers*, que podría traducirse hoy en día como *Tratados sobre ordenadores*, pero si así lo hiciésemos estaríamos bien equivocados, ya que si se le da un vistazo podemos observar que en este caso los *computers* son los encargados de realizar los cálculos, esto es, personas y no máquinas, dando allí normas de cómo debían organizar los cálculos para realizarlos con eficacia y disminuyendo los errores. En esta revista aparecieron las primeras tablas de números aleatorios.

Suele indicarse que el principio de la Estadística Computacional en Europa tuvo lugar en el laboratorio de Rothamstead, U.K., donde se construyó un ordenador de memorias de mercurio, que fue utilizado en la construcción de tablas estadísticas.

Las tablas estadísticas fueron el primero de los trabajos de la Estadística Computacional. Su precedente fueron las tablas de tiro, necesarias para realizar los cálculos necesarios en artillería, en las que trabajó K. Pearson introduciendo las técnicas de diferencias finitas en Inglaterra. Son muy conocidas las tablas estadísticas de R. Fisher y F. Yates, realizadas en Rothamstead, especialmente por este último, cuya quinta y última edición, realizada en 1956, incluye ya tratamiento en ordenador. Precisamente el uso en los contrastes de hipótesis de niveles de significación, los conocidos 5 % y 1 % entre otros, proceden de un problema computacional, ya que antes de la existencia y difusión de ordenadores, era necesario utilizar tablas, que evitaban cálculos muy tediosos en las aplicaciones prácticas, y las tablas no incluían todos los valores posibles, sino que, como es lógico, sólo podían incluir un número finito (y pequeño además) de valores. Salvo en el caso de la distribución normal, cuya familia completa se reduce a una única función ya que los par metros son eliminables antes de utilizar las tablas, las distribuciones eran inversas, de manera que dado un valor, α, la tabla permitía encontrar el valor de la función que dejaba una cola (o dos) de ese tamaño α. Puesto que había que seleccionar valores de este parámetro, se incluyeron unos valores determinados, que han influido posteriormente al ser aceptados como algo *natural*.

En definitiva, esta necesidad de simplificar el cálculo, permitió un desarrollo amplio de la Estadística teórica, ya que el método de trabajo consistió en realizar un modelo, muy a menudo basado en la distribución normal, sobre el que se realizaban todos los cálculos sin haber observado los datos reales, de tal manera que estos se encajaban a posteriori en el modelo teórico, permitiendo

obtener conclusiones más o menos válidas según lo fuese el ajuste mencionado. Consecuencia de ello fue la introducción de una Estadística aproximada (asintótica) de tal modo que para muestras m s o menos grandes se tuviesen estos resultados. Una reacción contra este tipo de estadística, impuesto por necesidad y que sin embargo pareció *natural* durante mucho tiempo, era previsible que apareciese, incluso si no hubiese existido un cambio en la herramienta computacional. Así, por ejemplo, surgió la Estadística bayesiana, que propugna entre otras cosas una estadística basada en el dato, y también otros enfoques, basados en la existencia de medios computacionales, como el Análisis Exploratorio de Datos de Tukey o la Estadística sin modelo de la escuela francesa. Destaquemos como dato anecdótico, aunque no lo es tanto, que hace pocos años se presentó el programa de ordenador StatXact, para cálculos exactos de distribuciones y que analizando con él una gran cantidad de casos de la literatura, se demostró que la mayoría de las conclusiones obtenidas utilizando técnicas asintóticas eran falsas. Curiosamente este programa se desarrolló para ordenadores personales, no para grandes ordenadores, lo que nos da una idea de la velocidad de cambio en la capacidad de cálculo que tiene disponible un investigador.

La evolución de los ordenadores (en realidad de la microelectrónica) ha sido muy rápida. Hace cuarenta años, el 29 de Octubre de 1956, IBM presentó el primer disco duro para almacenamiento de datos. Su tamaño no era excesivo, para la época, como dos frigoríficos actuales, y a un precio *muy competitivo*, cincuenta mil dólares de aquel momento, hace cuarenta años. Con esos condicionantes, indiquemos que permitía almacenar de forma cómoda 5 Mb de datos. En la actualidad son habituales tamaños de almacenamiento 200 veces superiores en los ordenadores personales de gama baja con un precio 250 veces inferior (sin tener en cuenta la depreciación de la moneda).

En ese mismo año, en una revista dedicada al tema, se predecía la evolución de los ordenadores, indicándose entre otras, la de que a finales de siglo, para lo que aún falta un poco de tiempo, se llegarían a construir ordenadores con un peso inferior a la tonelada y media. Este es un ejemplo de una predicción totalmente correcta, ya que efectivamente ha sido esa una meta plenamente alcanzada.

1.2.2. Evolución de los programas

Para establecer la evolución de la Estadística Computacional, conviene estudiar brevemente la evolución de los programas de ordenador en general.

En el comienzo de los ordenadores, los programas son *Wired*, esto es, cableados. Quiere ello decir que en ese momento, un ordenador es una m quina concreta que realiza un cálculo concreto y para que realice otro cálculo, esto es, para modificar el programa, debe realizarse un circuito nuevo, o lo que es lo mismo, hay que construir una nueva m quina. El primer cambio, fundamental por cierto, es la aparición de ordenadores que se programan modificando los valores de su memoria, no modificando el circuito, lo que quiere decir que estamos ante una máquina universal, que es capaz de modificarse a sí misma y por tanto adaptarse a diferentes situaciones y problemas.

Para programar este tipo de máquinas se evoluciona rápidamente hacia la utilización de una serie de códigos mnemotécnicos que permitan comprender al usuario, de modo m s sencillo que un volcado de memoria, el significado de un programa y que se denomina *Assembler* o lenguaje ensamblador.

Un ensamblador es propio de la m quina para la que se ha escrito y por tanto no es utilizable, mejor sería decir que casi no lo es, en otra m quina distinta. Surgen así los lenguajes de alto nivel, como **Fortran** en 1955, que se pretende que sean independientes del ordenador concreto, de tal modo que un programa escrito en un ordenador puede ser utilizado en otro. Esta pretensión se divide entre dos tendencias: una la expresada y otra la de las casas comerciales que prefieren que el usuario dependa del ordenador que tiene en un momento dado[1]. Sin embargo, la demanda por parte de los usuarios hace que se implanten los lenguajes de alto nivel y en **Fortran**, por ejemplo, se escribe tal cantidad de software que hace que hoy en día siga siendo muy utilizado.

La evolución siguiente se produce con la introducción del concepto de *Rutina* como una parte de un programa que, por el hecho de repetirse varias veces en el mismo, se considera importante, de tal modo que adquiere vida propia, independientemente del programa en el que se utiliza. Es más, da lugar al concepto de programación estructurada, de tal modo que los programas se componen de partes que son depuradas independientemente, modificadas y reutilizadas en otros programas. Las rutinas se englobaron en estructuras m s complejas, denominadas *Bibliotecas* que permiten manejarlas de modo sencillo, incorporando con facilidad a un programa, solamente aquellas partes que son necesarias. Estos conceptos de rutina y biblioteca se incluyeron en **Fortran** y fueron uno de los hechos determinantes de su vigencia.

[1]El estándar de facto en ordenadores es el denominado PC, que fue puesto en el mercado por IBM con una estructura abierta y de hecho supuso enormes pérdidas para la compañía.

A grandes rasgos, el siguiente paso fue la aparición de los *Entornos de programación* cuyo paradigma fue *Unix*. Un entorno de programación es un conjunto de herramientas que facilitan el desarrollo de un trabajo de programación. En tanto que **Fortran** permitía la aplicación de un programa escrito en otro ordenador, las fases de escritura de un programa, depuración, compilación, etc. eran distintas de un ordenador a otro. Unix, por el contrario, intentaba suministrar herramientas que facilitasen, no ya el trasvase, sino el desarrollo de programas, de modo que el manejo de ficheros, la edición de los mismos, las fases de compilación y utilización, fuesen las mismas independientemente de la máquina sobre la que se trabajase.

1.2.3. Evolución de los programas estadísticos

Los primeros problemas estadísticos computacionales (electrónicos) están relacionados con rutinas, son meras traslaciones de las fórmulas originales. Durante bastante tiempo, este fue el principal problema del que se ocupó la Estadística Computacional. Y no es un problema trivial. Como recoge Thisted[31], pensemos, por ejemplo, en el cálculo de una cantidad propia de la estadística elemental: la varianza. La fórmula que la define es la conocida $\sigma^2 = \sum (x_i - \bar{x})^2/n$. Voy a incluir tres algoritmos, de los muchos posibles, para realizar dicho cálculo. Escribo los tres en **Fortran**, aunque no es necesario conocer este lenguaje para comprenderlos. El primero, que llamar, A, es

```
MEDIA=0
DO I=1,N
        MEDIA=MEDIA+X(I)
END DO
MEDIA=MEDIA/N
VARIA=0
DO I=1,N
        VARIA=VARIA+(X(I)-MEDIA)**2
END DO
VARIA=VARIA/N
```

El segundo, que llamar, B, es

```
MEDIA=0
VARIA=0
```

```
DO I=1,N
        MEDIA=MEDIA+X(I)
        VARIA=VARIA+X(I)**2
END DO
VARIA=VARIA/N-(MEDIA/N)**2
```

y por último, el tercero, que llamar, C, es

```
SORT(X)
MEDIA=0
DO I=1,N
        MEDIA=MEDIA+X(I)
END DO
MEDIA=MEDIA/N
DO I=1,N
        X(I)=(X(I)-MEDIA)**2
END DO
SORT(X)
VARIA=0
DO I=1,N
        VARIA=VARIA+X(I)
END DO
VARIA=VARIA/N
```

Cada uno de los tres algoritmos desarrolla un método distinto: El algoritmo A corresponde directamente a la fórmula que hemos dado de la varianza. El algoritmo B corresponde a la modificación deducida del teorema de K"ning, $\mu_2 = m_2 - m_1^2$. El C modifica al A exclusivamente en la inclusión de un algoritmo, que he denominado SORT, que ordena los valores del vector de datos de menor a mayor.

El algoritmo A es la transcripción directa de la fórmula. Para el cálculo manual este procedimiento es muy largo, por lo que se introdujo el segundo procedimiento, que lo mejora en un par de aspectos: el primero es que los datos se leen una sola vez y el segundo es que, puesto que la media no suele coincidir con ninguno de los valores observados, al hacer las diferencias de cada valor a la media, se introducen normalmente gran cantidad de decimales en el método A, en tanto que en B se trabaja directamente con los valores iniciales. Si recordamos además que las medidas observadas procederán de un

instrumento de medida que tiene una unidad de medida, es habitual que los valores expresados en función de la misma correspondan a números enteros, con lo que, ayudados de una tabla de cuadrados, el cálculo se simplifique bastante. Tanto es así que, cuando el que les habla estudiaba Estadística en la Facultad, este es el método que aprendía y este es el método que durante varios años enseñó a sus alumnos para el cálculo manual de media y varianza. El tercer procedimiento, en el cálculo manual, coincide con el primero, ya que es irrelevante la ordenación. Sin embargo, en el cálculo en ordenador, en el que no se conservan un número ilimitado de cifras en los cálculos, puede producir grandes diferencias, si existen valores tan distantes entre sí que el orden en que se suman sea relevante. Al mismo tiempo ser muy oneroso en tiempo de cálculo, ya que la ordenación consume mucho tiempo.

¿Por qué estos tres algoritmos? Ya hemos dicho que el segundo se introdujo cuando la herramienta del cálculo era el calculista (ayudado o no de medios mecánicos) pero en el cálculo en ordenador puede producir resultados totalmente erróneos, debido al orden de magnitud de x^2, que se calcula directamente. Este tipo de algoritmo se utilizó, por ejemplo, en **SPSS** en sus primeras versiones, obteniéndose en algunos casos varianzas negativas. El tercer método subsana estos problemas a costa de un consumo en tiempo de cálculo que, con tamaños moderadamente grandes, puede ser excesivo. Podemos casi decir que, en la mayoría de los casos, el primer método es el más eficiente en términos de tamaño de memoria, tiempo de cálculo y precisión, para los actuales ordenadores. Nos reafirmamos así en el hecho de que los métodos de cálculo influyen en los cálculos disponibles y un método desechado vuelve a ser útil posteriormente.

A título anecdótico diremos que, hoy en día, no debería utilizarse ninguno de los tres métodos, ya que el cálculo de estos momentos se realiza más satisfactoriamente mediante el llamado método de las medias provisionales[20], que es un método de un solo paso y muy estable numéricamente.

Tampoco cabe pensar que este sea un tema ocioso, ya que todavía est este problema activo. Por ejemplo, en octubre de 1996, aparecen en EDSTAT-L[11] referencias a problemas de este tipo; en primer lugar, surge el tema de que una hoja de cálculo muy acreditada, EXCEL de Microsoft, produce valores erróneos en el cálculo de la varianza de determinados datos. En la discusión del tema quedan claros varios aspectos acordes con lo que anteriormente hemos indicado pero sigue apareciendo algún concepto erróneo. Así, se prueba el algoritmo A y se indica con sorpresa que este sí funciona bien (ya he indicado antes que se recomendaba el B) aparece la explicación correspondiente

al uso de la fórmula de Köning, pero se deslizan errores, al afirmar que los algoritmos de un solo paso producen estos errores, desconociendo la existencia del método de medias provisionales, y se hace referencia a fórmulas como la del algoritmo de dos pasos con corrección, $\sum(x-\bar{x})^2-(1/N)(\sum(x-\bar{x}))^2$. Por otra parte se indica que, sin embargo, en el cálculo de la asimetría y el aplastamiento, correspondientes a momentos centrados de tercer y cuarto orden, los cálculos son correctos, lo que indica el uso de algoritmos adecuados. Ello se achaca a que en los libros no se indica la fórmula de un solo paso, ya que si no, se hubiese calculado esta. Ello no es totalmente cierto, en los libros habituales de Estadística descriptiva aparecen las fórmulas de obtención de los momentos centrados, μ_k, en función de los no centrados, m_k, y viceversa, lo que pasa es que no aparece en los libros sumamente elementales que deben manejar los responsables de estos programas. A raíz de la discusión se han revisado espontáneamente otros programas y así ha aparecido, por ejemplo, que Delphi, conocido compilador de **Pascal** de Borland, incluye en la biblioteca de funciones matemáticas, como método de cálculo de la varianza, el de Köning, con el peligro de que cualquier programa que con el mismo se realice, incluir el código erróneo. Efectivamente, en el código fuente de esta biblioteca se indica:

```
{ MeanAndStdDev calculates Mean and StdDev in one pass, which
is 2x faster than calculating them separately.  Less accurate
when the mean is very large (> 10e7) or the variance is very
small. }
procedure MeanAndStdDev(const Data: array of Double; var Mean,
StdDev: Extended)
```

En ese mismo código se indican las referencias bibliográficas de las que se toma el método y que son las siguientes, advirtiendo que son recientes:

```
References:

1) P.J. Plauger, "The Standard C Library", Prentice-Hall, 1992,
   Ch. 7.
2) W.J. Cody, Jr., and W. Waite, "Software Manual For the
   Elementary Functions", Prentice-Hall, 1980.
3) Namir Shammas, "C/C++ Mathematical Algorithms for Scientists
   and Engineers", McGraw-Hill, 1995, Ch 8.
```

4) H.T. Lau, "A Numerical Library in C for Scientists and Engineers", CRC Press, 1994, Ch. 6.
5) "Pentium(tm) Processor User's Manual, Volume 3: Architecture and Programming Manual", Intel, 1994

El 31 de octubre de 1996, David C. Howell, del Departamento de Psicología de la Universidad de Vermont, plantea en la lista EDSTAT-L, lo que él denomina *Un problema interesante con SPSS*, consistente en que, en este lenguaje, escribe el siguiente programa:

```
new file.
input program.
loop #i = 1 to 30.
        do repeat response = r1 to r10.
                compute response = rv.normal(0,1).
        end repeat.
        end case.
end loop.
end file.
end input program.
compute rho = .4.
save outfile = "Instructor:data222.spss"/keep r1 to r10.
CORRELATIONS
 /VARIABLES=r1 r2 r3 rho
 /PRINT=TWOTAIL SIG
 /MISSING=PAIRWISE .
```

con la intención de generar números aleatorios que posteriormente utilizar . La variable ρ es una constante que se genera para cada caso. Por azar (casi habría que decir que por error) el investigador incluyó esta variable en la lista de variables a estudiar. Por construcción, esta variable tiene varianza nula y por tanto su correlación con otras variables no puede ser calculada. Sin embargo, los resultados del programa dieron correlaciones de -.22 con $r1$, .0757 con $r2$, y -.1397 con $r3$. Incluso, est correlada, 1.00, consigo misma, lo que no debería tampoco ocurrir.

El investigador siguió investigando e introdujo nuevas variables, introduciendo X con valor 1 en todos los individuos y Z con valor .1 asimismo en

todos. Las correlaciones de X con $r1$, $r2$ y $r3$ son correctas, esto es, no aparecen. Sin embargo sí lo hacen las correlaciones con Z y coinciden con las dadas para la variable ρ.

El investigador no se explica estos resultados, que ,l obtiene con la versión 6.1.1 en un Mac, y otros investigadores participan en la discusión replicando los mismos resultados con las versiones 6.1.3. y 7.0 para Windows (al menos es reconfortante la consistencia entre las versiones).

Evidentemente, existe un error de programación, de tal modo que si una variable es entera se detecta correctamente que es degenerada, pero no si es real, y se procede a realizar un cálculo que es por lo demás erróneo. Y esto ocurre en un programa líder del mercado y en un algoritmo muy sencillo, por lo que nos reafirmamos en la tesis de que es necesario disponer de datos de contraste y probar con ellos los diferentes programas, además de investigar los algoritmos implementados, que a menudo no son públicos por ser considerados como propiedad de la compañía que realiza las programaciones.

Volviendo a la evolución histórica, se dedica atención a continuación, lógicamente, a un tipo de problemas similares a estos, pero asociados al cálculo matricial. De estos esfuerzos surgen las bibliotecas Linpack, Eispack, etc. que dejan resuelto este tipo de problemas y que se utilizan posteriormente en cualquier programa.

Cuando estos problemas se solventan, aparecen ya rutinas dedicadas específicamente a cálculos estadísticos. Así comienzan en la facultad de Medicina de UCLA las rutinas **BIMED**, que posteriormente evolucionaron hasta convertirse en un lenguaje estadístico, de parámetros, conocido como **BMDP**. El coordinador de este proyecto, Dixon, recibió recientemente la medalla de oro de la ASA por su contribución al avance de la estadística, precisamente por esta tarea. De modo paralelo, en la universidad de Chicago, surge **SPSS**, muy similar a **BMDP**, aunque su sintaxis parece distinta a primera vista, y que tuvo también una amplia difusión. Ambos programas, utilizadísimos, corresponden a un tipo de lenguaje caracterizado por su f cil utilización y por no ser programables ni extensibles en el sentido amplio de la palabra. La mayoría de los primeros paquetes estadísticos fueron desarrollados en USA, y en gran parte aún sigue siendo así, y en segundo lugar se encuentra UK, aunque hoy en día, con el desarrollo de las comunicaciones, comienza a expandirse el desarrollo de programas a través del mundo.

Nelder, en 1965, mientras se encontraba en Adelaida, comenzó el desarrollo de Genstat, basado en un trabajo previo de ,l mismo, Wilkinson y James, sobre an lisis de la varianza. Posteriormente se desarrolló en Rothamstead y

la primera versión operativa estuvo disponible en 1971. Posteriormente, con la inclusión de los Modelos Lineales Generalizados, se desarrolló GLIM con el patrocinio de la Royal Statistical Society y se incluyeron en NAG (Numerical Algorithm Group) que también distribuye una biblioteca numérica para uso en programación tradicional. Estos programas, Genstat y GLIM, han tenido menor difusión, especialmente fuera del ámbito estadístico, debido a que son m s difíciles de utilizar que **BMDP** o **SPSS**, pero se diferencian de ellos, fundamentalmente, en el hecho de que son programables en sentido amplio, lo que permite su utilización para el desarrollo de nuevas técnicas estadísticas.

La consecuencia lógica es la creación de los Entornos de programación estadística, en los cuales se disponga de un método uniforme de trabajo, independientemente de la plataforma de hardware, con herramientas de apoyo y con la posibilidad de que sean extensibles y programables, lo que permite integrar en los mismos nuevos desarrollos.

En definitiva, aparece claramente un proceso recurrente, que podemos resumir del siguiente modo:

- – Aparece una técnica estadística que debe desarrollarse para su aplicación
- – Se implementa esta técnica para su utilización de modo independiente, bien en rutinas, como un programa completo o, m s recientemente, en un entorno de programación estadística.
- – Cuando la técnica se difunde y se contrasta, pasa a formar parte del acervo estadístico, y si es suficientemente interesante se incluye en los lenguajes de uso común.

1.3. Algunos campos de aplicación e interés de la Estadística Computacional en la actualidad

Son muchos los campos de desarrollo en Estadística Computacional y dar un repaso a todos, obligaría a una brevedad tal que solamente podríamos citar el nombre de los mismos. Yo he seleccionado algunos, que considero más interesantes, por las implicaciones que tienen o pueden tener en el resto de la Estadística.

1.3.1. Distribuciones exactas

El problema de las distribuciones exactas en el muestreo (en lenguaje coloquial podríamos decir *con letras*) surge por un problema computacional: Dada la dificultad de cálculo, se intenta poder resolver el problema una sola vez, sin tener en cuenta los datos concretos, y luego aplicarlo varias veces a datos distintos cada vez. De este modo, se resuelve un problema m s complejo que el planteado, ya que es m s general, pero se simplifica la aplicación posterior, que puede realizarse prácticamente de modo automático, eso sí, encajando el problema real en el marco del modelo teórico, lo que produce muy a menudo chirridos. Además, debido a este tipo de enfoque, sólo pueden resolverse problemas relativamente sencillos. Por ello, si se desean estudiar problemas suficientemente complejos, el método no ser aplicable.

En 1958, John Tukey introdujo el llamado método *jackknife*, construido sobre una técnica de estimación de sesgos de Quenouille y abandonada en su día por problemas de cómputo. El método de Tukey consiste en que conocidos unos datos, $x = (x_1, x_2, \ldots, x_n)$, de los que se quiere estudiar el comportamiento de un estadístico, $\widehat{\theta}$, que estima a un par metro de interés, θ, si denominamos por $x_{(i)}$ al conjunto excepto el dato i-ésimo y reevaluamos el estadístico de inter,s sobre cada conjunto, $x_{(i)}$, se obtiene un estimador del error del estimador con la ventaja de que su expresión

$$\left(\frac{n-1}{n} \sum_{i=1}^{n} \left(\widehat{\theta}_{(i)} - \overline{\theta}_{\cdot} \right)^2 \right)^{\frac{1}{2}} \quad \text{siendo} \quad \overline{\theta}_{\cdot} = \frac{1}{n} \sum_{i=1}^{n} \widehat{\theta}_{(i)}$$

se obtiene independientemente de que la distribución sea m s o menos complicada. Lo único que se necesita es poder calcular el estimador n veces. La introducción de este método marcó un hito esencial en el desplazamiento hacia la Estadística Computacional, aún cuando quedó en desuso tanto por la introducción de otros métodos m s recientes como por el hecho de que no funciona bien si los datos son muy *irregulares*.

En el 79, Efron, introdujo el *Bootstrap* como un intento para comprender mejor el método de Tukey, dando un salto adelante hacia los métodos denominados ya de cálculo intensivo y que podemos denominar como remuestreo (Resampling) que ponen de relieve la noción de que a partir del remuestreo de los datos puede obtenerse una aproximación a las variaciones muestrales con las cuales se producen los datos. Este método supone una revolución en un sentido adicional: No sólo es posible aplicarlo a casos que escapan a

la Estadística tradicional, sino que aplicados a los casos tratados en la misma podemos decir que la unión del ordenador y el remuestreo producen la estadística elemental, sin necesidad de fórmulas especializadas ni tablas, y que si se le añaden los conceptos estadísticos tradicionales obtenemos lo que podríamos denominar estadísticos eficientes de segundo orden.

1.3.2. El algoritmo EM

El algoritmo EM es una técnica iterativa para calcular estimadores por máxima verosimilitud con datos incompletos. Dempster, Laird y Rubin le dieron el nombre en función de los dos pasos computacionales implicados: Expectation y Maximization. Con datos incompletos hacemos referencia a una amplia variedad de modelos, entre los que se incluyen mixturas, convoluciones, datos censurados, datos truncados y datos faltantes. La idea general en este método es representar el vector de datos observados, y, como la realización de un vector de datos, x, observado indirectamente o incompletamente. Por ejemplo, en el problema de datos faltantes, x consiste en el vector de datos observados y de datos faltantes, en tanto que y consiste en los valores observados y unos indicadores de que faltan datos en el resto. Es posible que los datos completos contengan variables que nunca serán observables como datos en el sentido habitual, como por ejemplo cuando x contiene los efectos aleatorios en un problema de componentes de varianza.

Un ejemplo de utilización ser el siguiente (Laird): Supongamos que disponemos de n_0 observaciones correctamente clasificadas en una distribución trinomial con probabilidades $\theta_1, \theta_2, \theta_3$, con $\sum \theta_i = 1$ y, por otra parte, disponemos de n_1 observaciones clasificadas incompletamente, en el sentido de que somos capaces de distinguir si pertenecen al grupo 3 o a los grupos 1 y 2, pero sin poder distinguir entre estos dos últimos. Sea $n = n_0 + n_1$ el total de casos. Sean y_1 e y_2 las frecuencias de casos de las primeras n_0 observaciones que corresponden respectivamente a las categorías 1 y 2, y sea y_3 el número de casos de los n_1 que pertenecen a las categorías 1 ó 2. Por tanto, en la categoría 3 encontramos $n - \sum y_i$ casos. El vector de datos observados es $(y_1, y_2, n_0 - y_1 - y_2, y_3, n_1 - y_3)$, que en un caso concreto puede ser, por ejemplo, $(21, 9, 20, 7, 8)$. Si hubiésemos observado x, el estimador de θ sería x/n. En el método EM, comenzamos con una estimación inicial, θ^0, e iteramos los dos pasos siguientes hasta obtener la convergencia: El primer paso, es el del cálculo de valores esperados (Expectation) y en el mismo, suponiendo que $\theta = \theta^0$, calculamos un valor esperado para x reasignando y_3 a las categorías 1

y 2 en proporción a sus probabilidades respectivas. El segundo paso es el de maximización y en él consideramos los valores esperados de x obtenidos en el paso anterior como si fuesen valores observados y a partir de ellos calculamos el nuevo θ^1 utilizando la fórmula. Estos dos pasos se repiten sucesivamente hasta obtener convergencia en el estimador obtenido. Adviértase que, en este ejemplo, y en general, no es necesario especificar la relación de y a partir de x, lo único necesario es especificar x e y, los estimadores de máxima verosimilitud de θ basados en los datos completos y la densidad condicional de x dado y.

1.3.3. Estadística bayesiana

La Estadística bayesiana es otro de los campos en que la Estadística Computacional es muy importante. Tradicionalmente, la necesidad de resolver los problemas derivados de la aplicación de esta metodología ha conducido a la utilización de técnicas de integración, tanto numérica como simbólica. Pero existen más formas en las que se aplica y de modo m s innovativo. Una de ellas corresponde al *Gibbs sampling*. El método da un camino para aproximar las distribuciones a posteriori en muchos modelos Bayesianos. En particular, ha sido muy utilizado para aproximar las densidades a posteriori de funciones univariantes del par metro. Este método fue desarrollado en primer lugar por Geman y Geman[13] en 1984 para simular distribuciones a posteriori en la reconstrucción de im genes, utilizando distribuciones de Gibbs, de donde le viene el nombre. El método es el siguiente: Supongamos que queremos simular un vector aleatorio $U = (U_1, U_2, \ldots, U_k)$ cuya función de distribución k-dimensional es $F(u)$, donde $F(u)$ es desconocida o muy complicada, pero que, para cada i, la distribución de U_i condicionada al resto de componentes es conocida y relativamente fácil de simular. En primer lugar, simulamos un valor inicial U^0 de cualquier distribución con soporte contenido en el soporte de $F(u)$ y sea $u^0 = (u_1^0, \ldots, u_k^0)$ este valor concreto. En el primer ciclo, actualizamos en primer lugar la primera componente U_1 simulando U_1^1 con valor observado u_1^1 a partir de la distribución condicional de $U_1|(U_2 = u_2^0, \ldots, U_k = u_k^0)$, en segundo lugar actualizamos la segunda componente U_2 a partir de la distribución condicional de $U_2|(U_1 = u_1^1, U_3 = u_3^0, \ldots, U_k = u_k^0)$, y continuamos del mismo modo hasta actualizar U_k a partir de la distribución condicional de $U_k|(U_1 = u_1^1, U_2 = u_2^1, \ldots, U_{k-1} = u_{k-1}^1)$, con lo que terminamos el primer ciclo del procedimiento y obtenemos el valor u^1. A continuación repetimos el ciclo a partir de este valor y obtenemos u^2 y así proseguimos sucesiva-

mente. Pues bien, se puede demostrar que, bajo condiciones muy generales, las distribuciones de los vectores U^i, de los cuales eran realizaciones los u^i, convergen a la de U, por lo que en el límite obtendremos un vector aleatorio cuya distribución es prácticamente la pedida.

Este tipo de problemas requieren de simulación de variables aleatorias para lo que es necesario la generación de números aleatorios de distribuciones predeterminadas. El primer problema que se plantea es precisamente el del nombre: números aleatorios. En los primeros tiempos de la simulación se intentó recurrir a procesos físicos a partir de los cuales se generasen números, que por definición serían aleatorios de la distribución, fuese esta la que fuese, correspondiente al fenómeno físico. Este método, que a priori parece bueno adolece de bastantes defectos: No se conoce la distribución real de los fenómenos físicos utilizados, el proceso es además lento y caro, y no siempre está disponible cuando se necesita. Por eso se recurrió a establecer tablas de números aleatorios a partir de fenómenos que fuesen uniformes a priori, como por ejemplo en España las tablas de Royo, basadas en los sorteos de la Lotería Nacional, en Inglaterra las de Fisher y Yates y en Norteamérica la conocida *A million of random digits*. Estas tablas fueron muy utilizadas, especialmente en muestreo y en diseño de experimentos, pero su utilización era lenta y, sobre todo, para casos de simulación se agotan rápidamente.

La consecuencia lógica de estos defectos fue la introducción de métodos de generación en *directo* en ordenador, los denominados en un principio números pseudoaleatorios, y a los que se pidió que fuesen uniformes a posteriori, esto es, que cumpliesen diferentes condiciones de homogeneidad. Estos generadores han pasado, desde una primera etapa en que se construían por intuición (y en que fallaban estrepitosamente) a la etapa actual en que se aplican técnicas algorítmicas y algebraicas muy sofisticadas.

Es posible encontrar, a través de Internet, programas que implementan generadores de números aleatorios, que ya no adolecen de los problemas de los primeros generadores. Uno de los m s conocidos, para PC, es el denominado ULTRA, de Arif Zaman y George Marsaglia, cuya versión 1.01 es del año 92, que entre otras propiedades tiene un período muy largo (superior a 10^{356}, lo que da unos 10^{270} números por cada tomo del universo, según se estima su tamaño actualmente[2] pero permite simulaciones que se comportan muy bien. Su componente principal es el generador SWB, *Subtract-with-Borrow* descrito en su artículo[33] de 1991 con una semilla de 148 bytes, que su-

[2]Lo que podría ser totalmente falso.

pera ampliamente los contrastes de aleatoriedad teóricos y experimentales. Además incluye un generador congruencial con multiplicador 69069 y base 2^{32}, muy conocido y estudiado. Luego, los resultados de ambos generadores se combinan mediante una operación XOR para obtener el número generado definitivo.

Para cada distribución concreta se plantea el problema de obtener observaciones simuladas de la misma a través del método general, basado en que la distribución de la función de distribución de una variable continua es uniforme y por tanto basta invertirla, y métodos específicos basados en caracterizaciones de la distribución, que consumirán menos tiempo, siendo por tanto un problema abierto el de obtener caracterizaciones apropiadas.

1.3.4. Gráficos estadísticos

Los programas estadísticos incluyeron desde el principio gráficos, muy a menudo de baja resolución, para ayudar a comprender los resultados de los análisis y para explorar los datos buscando estructuras en ellos. La aparición de ordenadores con medios gráficos impulsó la utilización de los métodos gráficos, que son en la actualidad uno de los campos m s florecientes de la Estadística Computacional. Además la potencia de cálculo permite la inclusión de gráficos din micos que facilitan esa exploración de las estructuras subyacentes a los datos, especialmente con datos multivariantes. Este punto de vista, opuesto y al mismo tiempo complementario al de la estadística de contrastes, sirve por tanto para formar hipótesis, m s que para contrastarlas con otras dadas, lo que suele ser m s a menudo el tema de interés en investigación, en la línea del análisis exploratorio de datos de Tukey[32].

Los métodos de visualización estadística multivariante están basados fundamentalmente en la potencia de reconocimiento de patrones de la visión humana[3] y en la potencia de cálculo gráfico de los ordenadores para ayudar al estadístico a buscar estructuras, y por tanto a crear hipótesis, sobre los datos multivariantes, especialmente en estructuras de alta dimensionalidad. Como la representación es en definitiva bidimensional, deben cumplirse una serie de condiciones respecto de:

- la geometría original,
- la percepción tridimensional del analista y

[3]El paradigma son las Caras de Chernoff.

- la bidimensionalidad de la pantalla del ordenador y de sus limitaciones de cálculo.

1.3.5. Programación Lineal

Otras ramas, de nuestra área, como la Investigación Operativa, también participan de este tipo de temas ya que tradicionalmente se han usado los ordenadores para la resolución de los problemas. Un ejemplo ilustrativo corresponde al problema de la programación lineal. Este problema consiste en

$$\text{minimizar } c^T x \text{ sujeto a } Ax = b, x \geq 0$$

donde A es una matriz $m \times m$ de rango m, y b y c son vectores de dimensiones apropiadas.

Con este tipo de restricciones se obtiene un hiperpoliedro y es inmediato obtener que cualquier solución del problema debe encontrarse en su superficie. Si esta solución es única, se encontrar en un vértice, y si existen varias, serán en una arista, cara, etc. Una forma de solución es por tanto la de explorar todos los vértices del hiperpoliedro, que son un número finito, e investigar en cual o cuales de ellos se encuentra el mínimo. Este método tiene el inconveniente de que son demasiados los vértices a explorar, volveremos con más detalle sobre este tema, y para obviarlo, se introdujo el método del *simplex* por Dantzig y otros. En este método partimos de un vértice cualquiera, que podría ser la solución, y saltamos hasta otro vértice, que también podría ser solución, utilizando el método del gradiente, y con la condición de que el nuevo vértice no es peor que el antiguo, en términos de la función a minimizar. En caso de que sea igual, el problema se denomina degenerado y hay que tomar precauciones porque puede conducir a un ciclo infinito. Si el valor decrece estrictamente en cada iteración, el algoritmo termina en $\binom{n}{m}$ pasos a lo sumo. Muy a menudo el número de iteraciones crece linealmente con el tamaño n, aunque pueden construirse casos, como el de Klee y Minty, en el que el número de iteraciones puede llegar a ser una función exponencial de n.

Históricamente, el método del simplex ha sido el principal algoritmo aplicado a los problemas de programación lineal, aunque cuando el número de restricciones es grande puede no terminar en un tiempo prudencial. Karmarkar ha introducido un nuevo método, del que me gustaría destacar un aspecto. El método de Karmarkar es un método de *punto interior* ya que las

iteraciones permanecen en el interior del conjunto aceptable, y se demuestra que en el peor de los casos, el *tiempo de cálculo* crece según una función polinomial de n y el *tamaño de almacenamiento* del problema en el ordenador crece linealmente. Este método, para valores grandes de n, es casi siempre más rápido que el método del simplex y ha producido una revolución en los métodos de la programación lineal. El aspecto que me gustaría destacar es precisamente el de que parte de un punto interior, que por tanto *no* es una solución aceptable para llegar a otro punto interior, que de nuevo *no* es una solución aceptable. Es por tanto un enfoque radicalmente distinto del correspondiente al método del simplex y por tanto abre una nueva forma de enfoque del problema, en que no llegamos a la solución exacta sino a una suficientemente próxima y que, a todos los efectos, es completamente válida. De todos modos, el método del simplex parte de soluciones aceptables solo en teoría, ya que su representación en ordenador no lo es, e incluso los datos reales a los que se aplica no serán conocidos de forma exacta.

He marcado unos aspectos en el problema anterior que nos llevan a considerar los tiempos de resolución de problemas, los datos y su almacenamiento.

1.3.6. Tiempos y tipos de problemas

Si pensamos en un problema clásico de nuestra disciplina, el llamado *problema del viajante*, podremos ilustrar este aspecto de la computación. Éste es un problema, que como muchos problemas clásicos, es de enunciado sencillo y solución compleja. El enunciado es: Dadas n ciudades, unidas dos a dos por carreteras, un viajante desea visitarlas todas y volver a la ciudad de partida, con la condición de que el recorrido total sea mínimo.

El problema tiene una solución trivial: Existen $n!$ recorridos posibles, calculemos la longitud de cada uno y luego busquemos el mínimo. El algoritmo consumirá por tanto al menos $n(n!)$ pasos para encontrar el recorrido óptimo. Por tanto, si n es moderadamente grande, digamos $n = 50$, el número de pasos ejecutados por el algoritmo es mayor de 10^{60}. Suponiendo que un ordenador rapidísimo ejecutase 10^{10} pasos por segundo, el tiempo necesario para resolver el problema sería superior a 10^{42} años, por lo que podemos decir que aunque hemos imaginado una solución, esta no es factible. Este tipo de aumentos en el tiempo de resolución se da en cualquier algoritmo que no sea polinomial, e incluso en los polinomiales, si el orden del mismo es grande.

Dado un problema, es deseable encontrar un algoritmo de solución eficiente, esto es, uno cuyo tiempo de solución sea lineal o cuadrático. Si ello

no es posible, habrá que conformarse con uno que lo haga en tiempo polinomial. Desafortunadamente, para muchos problemas importantes no se ha encontrado este tipo de soluciones, habiéndose conseguido simplemente la clasificación de algunos tipos de problemas en una clase tal que si un elemento de la misma tiene un algoritmo polinomial, también lo tendrán los demás.

Por tanto, los tiempos y tipos de problemas, como exponencial, polinomial (lineal, cuadrático, superior) resolubles o no, son un tema de interés fundamental cuando nos interesa, precisamente, la resolución de problemas concretos en ordenador.

1.3.7. Datos

En Estadística, como es evidente, no podemos olvidarnos de los datos y, en consecuencia, de su almacenamiento en soporte de ordenador.

Los datos no siempre se recogen exprofeso para un análisis, sino que, cada vez más, existen ya y se seleccionan para su utilización. El problema de conseguir los datos necesarios para la realización de un análisis ha pasado, en muchos casos, desde el plano de su recolección expresa al plano actual de su selección de entre la enorme cantidad de datos que existen, dado que a menudo se encuentran ya recogidos y almacenados en soporte informático. Ello ha dado lugar a una serie de técnicas conocidas como *data mining* para seleccionar los datos útiles de entre la maraña de datos disponibles y para analizar esos datos con un enfoque distinto al tradicional, en que el investigador realiza una hipótesis y toma unos datos para contrastarla, a un nuevo enfoque en que, para unos datos existentes, el investigador, con ayuda del ordenador, debe encontrar hipótesis plausibles. Técnicas como las redes neuronales son fundamentales en este campo.

Otro aspecto es el que corresponde al acceso a los datos. En el enfoque tradicional, citado en el párrafo anterior, se trataban todos los datos, luego el único acceso a los mismos era el secuencial. Ahora, al disponer además de grandes volúmenes de datos, se plantean problemas de ordenación y búsqueda [19] que, al consumir mucho tiempo de ordenador, plantean problemas importantes.

Muchos lenguajes han prestado atención a lo que tradicionalmente era el problema estadístico: El análisis, con técnicas sofisticadas que consumen mucho tiempo de ordenador, de pequeños volúmenes de datos. Por ello se plantean problemas importantes cuando el volumen de datos a tratar es grande,

aunque la técnica no sea compleja. Manejar el censo de población en Andalucía, por ejemplo, supone graves problemas con casi todos los lenguajes. En algunos casos el problema es muy grave, así **S-plus** tiene problemas de tiempo para el tratamiento de grandes volúmenes de datos que sin embargo con programas más modestos pueden ser tratados m s eficientemente.

En cuanto a la presentación de los datos, mediante representaciones gráficas, se ha pasado de unos gráficos rudimentarios, con una interpretación clara realizada por especialistas, a una multitud de gráficos, con una realización magnífica, y que se prestan a manipulaciones, conscientes o no, que afectan a su significación. Y ello desde gráficos tan elementales como los conocidos *de tarta*, los diagramas de sectores, que si se presentan en tres dimensiones pueden manipularse sin m s que efectuar una rotación. El aforismo de la geometría, es v lido aquí: y podemos decir que la Estadística, en este aspecto, es el arte de razonar bien sobre figuras mal hechas.

Sobre los datos se presenta adicionalmente el problema, paradójico, de la carencia del dato, debido a pérdida, error en el tratamiento, no recolección en su momento, etc... lo que conlleva problemas de imputación de los datos faltantes o anómalos, tema al que ya he hecho referencia al tratar del algoritmo EM.

Además, para el desarrollo de las diferentes técnicas es necesario disponer de Bases de datos reales y de Datos de contraste, esto es, de datos reales o artificiales, que pongan de relieve las carencias de los programas, al estilo de los conocidos datos de Longley y otros similares, datos con multicolinealidad que, en su momento, hicieron fallar a todos los programas estadísticos en el análisis de regresión para este tipo de datos y prueba que, hoy en día, pasan sin ningún problema todos los programas. Debe prestarse atención, al mismo tiempo, al conocido problema consecuencia de la existencia de pruebas para los programas y que queda de relieve, por ejemplo, en el lenguaje **C**, que se comprobaba mediante el cálculo de números primos, que hoy ya no se considera determinante, ya que los fabricantes de estos lenguajes incluían rutinas especialmente diseñadas para la prueba.

1.3.8. Lenguajes estadísticos

Cada programa de ordenador está definido por su sintaxis y su semántica, que describen la *forma* y la *esencia* del lenguaje. En este sentido podría pensarse que la sintaxis de un lenguaje es meramente superficial e irrelevante, ya que lo que el lenguaje puede hacer o no, está determinado por la semántica.

Sin embargo, la sintaxis es muy importante, ya que determina el modo en que los usuarios del lenguaje se expresan y expresan sus ideas. De hecho, los usuarios tienden a diferenciar los lenguajes basándose en la sintaxis y no en las diferencias semánticas. Así, se distingue entre **Fortran** y **Basic** como dos lenguajes muy distintos, cuando en realidad son muy parecidos. Esto ocurre también en otros niveles, así los lenguajes estadísticos **S** y **Scheme** poseen diferencias sintácticas evidentes, pero sin embargo son muy similares en sus estructuras básicas, de tal modo que la traducción de las expresiones en **S** en *s*-expresiones de **Scheme**, generalmente se realizan simplemente mediante una reordenación, y así en el lenguaje **R** se utiliza una semántica similar a la de **Scheme**, una sintaxis similar a la de **S** y un traductor de esta sintaxis a la del primero, todo ello motivado por considerar ventajas en este enfoque mixto. Del mismo modo, se diferencia entre los lenguajes **Fortran** y **C**. Sin embargo, desde 1990 en que Feldman escribió **f2c**, un traductor de **Fortran** a **C** es posible traducir del primero al segundo lenguaje sin ningún esfuerzo. Considerados globalmente, por ejemplo, pese a que en su mayor parte coinciden, los usuarios ven distinto un lenguaje tradicional, con parámetros como **BMDP**, y un lenguaje mediante menús, como **Statgraphics**, aunque no es esa su diferencia fundamental.

Además de esos dos conceptos, una tercera capacidad que influye son las herramientas de apoyo, que junto con lo anterior forman el *entorno de trabajo* que es lo que en definitiva el usuario identifica como el programa. Por ejemplo, el lenguaje **Pascal** definido por N. Wirth, fue diseñado para enseñar programación y como tal se utilizaba en ambientes académicos. Al poco tiempo de ponerse en el mercado los PC, P. Khan introdujo **Turbo Pascal**, concebido con unas herramientas de apoyo útiles (compilador integrado con un editor con conducción automática a los errores) que supuso la práctica desaparición de compiladores de la competencia, la utilización de **Turbo Pascal** como lenguaje de programación y no de aprendizaje, y cuyo modelo de entorno se ha adoptado en una gran cantidad de compiladores de otros lenguajes.

La siguiente característica deseable es la *extensibilidad* en el sentido de que el lenguaje sea capaz de admitir nuevos conjuntos de órdenes o estructuras, de modo que no quede anclado en una modelización concreta. No debe confundirse esta extensibilidad con la encontrada en **Systat** o **BMDP**, que han añadido en determinados momentos nuevos módulos, ya que no coincide totalmente con lo deseado. Más bien, debe entenderse como la capacidad de ser *programable*, de tal modo que los nuevos programas se incorporen al

conjunto del lenguaje. Si estos nuevos programas permanecen en disco se tendrá el inconveniente de los tiempos de acceso y si se incorporan en memoria, la necesidad de aumentarla, por lo que debe llegarse en general a una solución de compromiso.

Es necesaria además *portabilidad* entre máquinas, porque además así resistir el paso del tiempo, aunque ello conlleve no utilizar al máximo las capacidades de una m quina concreta. La tecnología del software se desplaza desde las aplicaciones auto-contenidas y programas hacia los componentes de software, es necesario diseñar módulos que permitan la reutilización del código fuente, definir tipos de datos abstractos adecuados y suministrar un inicio de tipos y objetos extensibles.

Otro de los intentos en esta dirección es el proyecto *Voyager* que se desarrolla sobre *Oberon* en el laboratorio de estadística de Heidelberg. El sistema operativo Oberon y el lenguaje Oberon se han desarrollado desde 1991 en ETH Zürich. El sistema operativo está escrito en el propio lenguaje y es un sistema monotarea. Aunque permite ejecutar tareas en segundo plano, los procesos usuales consisten en la ejecución de una sola orden y el retorno al nivel común al terminar la ejecución. El sistema básico no ocupa demasiada memoria, aunque al ser extensible, las extensiones pueden ocupar un tamaño arbitrario. Es posible implementar un sistema operativo Oberon sobre otro sistema operativo, como MS-DOS, aunque tambi,n puede utilizarse como sistema nativo en un ordenador compatible, también es posible utilizarlo como una emulación sobre muchísimos sistemas operativos y arquitecturas de ordenador. Voyager se instala sobre Oberon y su tarea es añadir cálculo estadístico al sistema, suministrando gestión de datos, cálculos estadísticos específicos y presentaciones gráficas. Incluye los algoritmos estadísticos usuales, como BLAS/LAPCK para álgebra lineal, los de Applied Statistics, etc. Cualquier salida de Voyager tiene una forma que permite ser utilizada a su vez como una entrada para Voyager, y como es lógico, sin necesidad de escribirla de nuevo, basta con utilizar el redireccionamiento o la inclusión en una orden mediante una marca de referencia. Esto permite una facilidad enorme para el análisis estadístico. Además, puesto que funciona sobre Oberon, se aprovecha de las posibilidades de este. Como ya he indicado, en Oberon se pasa de un esquema de programas y rutinas a uno de componentes de software, esto es, las bibliotecas tienen un nivel más avanzado. Estas componentes pueden cambiarse individualmente incluso en el momento de la ejecución, lo que permite una gran flexibilidad. Puesto que un módulo cargado en Oberon permanece residente y su información es accesible si es exportado y si no es

exportado no es accesible, es posible utilizarlos para extender el lenguaje, incluso sin conocer el código con el que ha sido escrito.

Otra condición que debemos tener en cuenta es que algunos programas son de uso público y otros de pago. Incluyo en el primer tipo una gran variedad de tipos, desde los totalmente públicos, incluido el diseño, hasta los que lo son con determinadas restricciones. Todo ellos tienen la desventaja de que no se mueven por interés monetario, como ocurre hasta hoy con los teoremas en Matemáticas, y al tiempo esa es su ventaja, ya que muchos investigadores pueden contribuir a su desarrollo y se garantiza su existencia más allá de avatares comerciales, simplemente por su utilidad. Desgraciadamente muchos programas que comienzan siendo de uso público, terminan en el otro lado. En Estadística tenemos muchos ejemplos: Por ejemplo, el lenguaje **S** comenzó así, pero cuando AT&T dejó de trabajar en él, una casa comercial lo amplió llamándolo S-Plus y ya no lo es. Afortunadamente existen otros muchos, como **R**, **LispStat**, etc. que siguen siendo públicos y por tanto se puede trabajar sobre ellos y con ellos con entera libertad, y entiendo que es nuestra obligación difundirlos para mantener vivo el espíritu de la Ciencia.

El estudio comparado de diferentes lenguajes pone de manifiesto sus capacidades y sus deficiencias, las cosas que tienen en común y las que los diferencian, y por tanto permiten estudiar cómo debe ser un lenguaje para el futuro, aunque seguramente no coincidirán con lo que hoy en día pensamos.

1.3.9. El impacto de las telecomunicaciones en la Estadística Computacional

Siendo este un tema de importancia relevante, sólo le dedicar, unas líneas, ya que su difusión en todo tipo de medios de comunicación, lo haría aquí y ahora, redundante.

Entiendo que la Ciencia está basada, desde hace bastante tiempo y en nuestra sociedad, en la difusión libre de las ideas. Aún la existencia de las patentes, discutida en muchos foros, no impide esta difusión aunque sí la retrasa. La difusión de las ideas, paradójicamente, estaba hasta hace poco tiempo retrasada por una feliz circunstancia: la enorme cantidad de las mismas que había que difundir, unida al hecho de que el medio de difusión era escrito. La aparición de redes de ordenadores (la conocida Internet) y la difusión del concepto que comenzó siendo conocido como *ftp anónimo* ha permitido que la difusión de ideas, acompañada de su plasmación en artículos,

datos, programas, etc. sea hoy en día un hecho que provoca una revolución ya que es posible trabajar en muchos temas casi de modo simultáneo en diferentes partes del mundo y acceder (e incluso poder encontrar fácilmente entre tantos temas aquel que nos interesa) a informaciones de interés relevante.

En temas relacionados con estadística es preciso citar Netlib y Statlib como dos fuentes de información fundamentales, junto con la que suministran, especialmente en cuanto a datos y también programas en determinados casos, los organismos estadísticos como el IEA y el INE en España y otros muchos en el resto del mundo, como por ejemplo la oficina del Censo de los E.E.U.U. que hacen que las estadísticas públicas sean además públicas, en el sentido de su conocimiento. Además los foros de discusión (newsgroup) en estadística como sci.stat, sci.stat.consult, etc... y las listas como EDSTAT-L, BMDP-L, etc.. permiten un avance importante en la estadística.

1.4. Temas de investigación propios

1.4.1. Multidimensional Scaling

Multidimensional Scaling (MDS) es una técnica cuyos orígenes se remontan al modelo métrico de Torgerson (1958) y al modelo no métrico de Shepard (1962). Posteriormente, se desarrollaron algunos modelos importantes que impulsaron la técnica como el modelo no métrico de Kruskal (1964) o el primer modelo métrico con diferencias individuales, INDSCAL, de Carroll et al. (1972).

Supongamos que se dispone de un mapa en el que est n representadas una serie de ciudades. La construcción de la matriz de distancias a partir del mapa es un problema puramente algebráico. Consideremos la situación inversa. Supongamos que se dispone de una matriz de distancias y se desea reconstruir el mapa. Este problema se resolvió por Schoenberg (1935) y Young y Houselholder (1938), los cuales ofrecen una solución al problema cuando la matriz de partida es una matriz de distancias euclídeas y en un espacio de configuración de dimensión $k = \text{rang}(B)$.

El problema estadístico en MDS surge cuando se pretende la representación en un espacio de dimensión k' menor que k, o bien cuando la matriz de partida no es una matriz de distancias euclídeas sino una matriz de pseudodistancias y concretamente una matriz de disimilaridades o similaridades. Por tanto, MDS puede definirse como un conjunto de procedimientos de for-

ma que, dada una matriz de disimilaridad, puede encontrarse un espacio de configuración en dimensión k, suficientemente pequeña, de forma que la matriz de distancias asociada sea lo suficientemente *parecida* a la matriz de disimilaridades de que se partía.

Según se entienda el concepto de *parecida* hablaremos de MDS métrico o no métrico, de forma que cuando la relación entre distancias y disimilaridades sea cuantitativa el MDS ser métrico mientras que cuando sea cualitativa, es decir, cuando lo que se pretenda sea encontrar una matriz de distancias que preserve el orden de las disimilaridades de partida, se hablará de MDS no métrico. Además, en MDS intervienen dos conjuntos de elementos claramente diferenciados: el conjunto de estímulos o elementos que se pretenden representar y el conjunto de individuos o elementos que emiten el juicio sobre los estímulos que se pretenden representar. Si el conjunto de los individuos está igualmente ponderado hablaremos de MDS sin diferencias individuales mientras que si no está igualmente ponderado hablaremos de MDS con diferencias individuales. En este último, se tratará de encontrar una matriz de configuración conjunta, X, con las coordenadas de los estímulos y una matriz de ponderaciones, W, con los pesos de los individuos, que reflejará la influencia que la opinión de cada individuo ejerce sobre el modelo.

Los modelos antes citados son mínimo cuadráticos, pero Ramsay, en el año 1977, formula el primer procedimiento probabilístico de MDS métrico. Históricamente MDS est ligado al desarrollo de las ciencias de la conducta y de ahí proviene la terminología empleada en MDS de disimilaridad o estímulo que utilizamos. En general, de cualquier aparato de medida son deseables una serie de propiedades y en particular lo son para una medida de distancia y para una disimilaridad. De entre esas propiedades nos vamos a centrar en dos que son especialmente interesantes en MDS: el origen y la positividad.

En cuanto al origen, est claro que una distancia tiene un origen definido y en cero, mientras que en el caso de una disimilaridad esta propiedad hay que interpretarla como que, dados dos estímulos idénticos, sería deseable que se les diese un valor de no disimilaridad.

Por otro lado, la positividad en una distancia está clara en el sentido de que dados dos puntos que no sean idénticos, tendrán una distancia estrictamente positiva. En cuanto a las disimilaridades esta propiedad hay que interpretarla como que, dados dos estímulos no idénticos, estos tendrán un valor de disimilaridad no nulo.

Bajo este punto de vista y en el entorno de la inferencia estadística, Ramsay, en 1977, propone considerar en el modelo de Torgerson los errores

con carácter aleatorio en lugar de determinístico, de forma que, mediante la distribución lognormal biparamétrica como distribución apropiada, resuelve el problema del MDS para datos de disimilaridad positivos y con origen en cero.

A nivel práctico en las dos hipótesis anteriores, surgen diversos problemas. En primer lugar, incluso cuando las disimilaridades son positivas se comprueba experimentalmente que cuando a un individuo se le presentan dos estímulos que son idénticos, este tiende a asignarles algún valor de disimilaridad. Ello se traduce en que, en estos casos, los valores de disimilaridad no tienen su origen en cero.

Por otro lado, como pone de manifiesto Heiser, en el caso en que los datos de disimilaridad son obtenidos a partir de escalas de tipo intervalo el origen ni siquiera est determinado. Además, en muchas ocasiones, los datos de los que se dispone no vienen dados en términos de disimilaridades sino de valores de similaridad. En estos casos, puede obtenerse una medida de disimilaridad sin más que cambiar de signo la similaridad, de manera que no solamente no existirá un origen definido sino que además los valores de disimilaridad pueden ser negativos.

Por último, en muchos casos, lo realmente interesante no estriba en la relación que existe entre distancia y disimilaridad sino entre distancia y una función de la disimilaridad que además debe estimarse simultáneamente en el modelo. En esos casos, los valores transformados de las disimilaridades son comparados con las distancias y puede ocurrir que esos valores, sin embargo, sean negativos.

Esta problemática se refleja inevitablemente en los algoritmos de MDS y en muchos de ellos, o no es tratable, como ocurre en el propio caso del modelo de Ramsay o se traduce en falta de convergencia del algoritmo. Para el tratamiento de este tipo de datos hemos desarrollado un modelo probabilístico de MDS que mediante la distribución lognormal triparamétrica y el criterio de máxima verosimilitud, resuelve el problema del análisis de datos de disimilaridad negativos y con origen indeterminado mediante máxima verosimilitud, independientemente del criterio del investigador.

El modelo está basado en la relación entre disimilaridades y distancias dada por la siguiente expresión:

$$\beta \log(\theta - \delta_{ij}) = \log(d^*_{ij}) + e_{ij}, \quad e_{ij} \sim N(0, \sigma^2)$$

donde β es el par metro de la transformación, σ el factor de variabilidad, θ el

parámetro umbral y d_{ij}^* la distancia euclídea entre cada par de puntos. Para el tratamiento computacional del modelo, los parámetros se han agrupado en bloques para su estimación, dentro de un ciclo iterativo que está formado por dos fases principales y por varias fases secundarias.

La primera fase principal está destinada a la estimación del parámetro umbral, θ, mediante un ciclo iterativo basado en el método del gradiente. En la segunda fase se estiman el resto de los par metros, supuesto fijo el valor del parámetro umbral. Esta fase a su vez está dividida en varias subfases para la estimación de los parámetros. En primer lugar se estima el par metro de la transformación, β. A continuación se estima el parámetro de variabilidad, σ, y finalmente se estima la configuración, X, dentro de un ciclo iterativo basado en el método del gradiente. El criterio de convergencia empleado es estadístico y se basa en que la diferencia entre dos valores consecutivos de la log-verosimilitud no exceda de una cantidad fijada.

Este modelo se extiende al caso de diferencias individuales sin más que considerar la relación:

$$\beta_r \log(\theta_r - \delta_{ijr}) + v_r = \log(d_{ijr}^*) + e_{ijr}, \quad e_{ij} \sim N(0, \sigma_r^2)$$

donde para cada individuo r, β_r y v_r son los parámetros de la transformación, θ_r el parámetro umbral para cada individuo, r, y d_{ijr}^* la distancia euclídea ponderada para cada individuo, r, donde w_r será el vector de pesos asociado a dicho individuo.

Los parámetros son estimados por máxima verosimilitud, dentro de un ciclo iterativo similar al anterior, solo que ahora en la primera fase principal no se estima un par metro umbral sino un vector de par metros asociados a cada individuo, r, y en la segunda fase principal, en primer lugar se estiman los vectores de par metros de la transformación $\beta = (\beta_1, \ldots, \beta_r)$ y $v = (v_1, \ldots, v_r)$ y además hay que añadir un nuevo ciclo iterativo asociado a la estimación de la matriz de ponderaciones W.

Uno de los aspectos m s importantes de los modelos probabilísticos de MDS consiste en el estudio de la variabilidad del modelo. La necesidad de descomponer la variabilidad se comprueba experimentalmente observando un diagrama de dispersión simple entre las disparidades o disimilaridades estimadas y las distancias. Por muy buena que sea la transformación elegida para estimar las disparidades de las disimilaridades, el ajuste nunca es perfecto. Por tanto, la descomposición de la variabilidad es importante por dos motivos:

- Mediante la descomposición de la variabilidad pueden obtenerse estimaciones más precisas.
- Mediante la descomposición de la variabilidad pueden interpretarse las causas por las cuales los residuos no son cero.

Nosotros hemos propuesto un modelo para la descomposición de la variabilidad asociada a cada par de estímulos mediante la expresión,

$$\gamma_{ij}^2 = (\alpha_i^2 \alpha_j^2)^{1/2}$$

donde cada factor, α_i, representa la variabilidad asociada al estímulo i y puede interpretarse como la influencia que dicho estímulo ejerce sobre el individuo encuestado. Así, un estímulo con gran factor de variabilidad indicar que ejerce gran influencia sobre las opiniones que dicho individuo efectúa sobre los pares de estímulos en los que interviene. Los par metros se estiman por máxima verosimilitud mediante un ciclo iterativo basado en el método del gradiente debido a la imposibilidad de encontrar estimadores exactos. Este ciclo sustituye al bloque referente a la estimación del factor de variabilidad del modelo en el algoritmo anterior.

Una de las hipótesis fundamentales del modelo de Ramsay se basa en que los datos han sido obtenidos en una escala de razón con al menos 7 categorías. No obstante se comprueba experimentalmente que los individuos tienden a expresar sus opiniones usando sólo 4 ó 5 categorías, por lo que en ese caso el modelo de Ramsay no puede utilizarse. Takane, en 1981, propone el primer modelo probabilístico no métrico de MDS. El modelo de Takane se basa en la hipótesis de que los datos son observaciones categóricas, siendo las categorías intervalos contiguos de la recta real, en principio desconocidos. Además, el modelo de Takane no considera los valores de los datos de disimilaridad, sino solamente las frecuencias de clasificación de las opiniones sobre cada par de estímulos en cada categoría, siendo un modelo homocedástico.

Este modelo lo extendimos al caso heteroscedástico mediante la relación entre disimilaridades y distancias dada a través de las siguientes expresiones, la primera mediante un modelo aditivo y la segunda mediante uno multiplicativo:

$$d_{ijr} = d^*_{ijr} + e_{ijr} \qquad d_{ijr} = d^*_{ijr} e_{ijr} \qquad e_{ijr} \quad \sim N(0, \sigma\alpha_i\alpha_j),$$

donde α_i se interpreta como en el caso anterior y σ es el factor de variabilidad referente al individuo encuestado.

El tratamiento computacional del modelo se basa en una aproximación de la función de distribución normal mediante la distribución logística y los par metros son estimados por máxima verosimilitud mediante un ciclo iterativo compuesto de tres fases. En la primera es estimada la configuración mediante un ciclo iterativo del gradiente con reinicios. En la segunda son estimados los factores de variabilidad y en la tercera son estimadas las cotas de las categorías. El criterio de convergencia empleado, nuevamente vuelve a ser que la diferencia entre dos valores consecutivos de la logverosimilitud no exceda un valor.

Finalmente, una de las hipótesis básicas del modelo de Ramsay consiste en suponer independencia entre los datos observados. Si suponemos T réplicas, esta hipótesis se traduce en suponer las réplicas independientes entre sí. Para controlar la independencia de las réplicas introdujimos un factor de variabilidad, τ_t, mediante la siguiente expresión:

$$\sigma_{ijt} = (\alpha_i^2 \alpha_j^2)^{1/2} \tau_t$$

donde α_i se interpreta como en el caso anterior y τ_t representa la variabilidad que introduce la réplica t en el modelo. Así, un valor alto de τ_t indicará, bien que el método para obtener los datos resulta extraño para el individuo encuestado, bien que los estímulos no son conocidos por el individuo o bien que han influido sobre la opinión del encuestado diversas situaciones externas al problema, con lo cual podría considerarse su eliminación.

Por el contrario, si el valor de τ_t es bajo, ello indica que el procedimiento para la obtención de dicha réplica se ha visto afectado por el problema del aprendizaje y por tanto dicha réplica viola la hipótesis de independencia del modelo, con lo que podría ser eliminada.

1.4.2. Regresión no paramétrica

La regresión no paramétrica se ha desarrollado rápidamente como alternativa a la regresión paramétrica que no se muestra adecuada para ajustar una curva a muchos de los conjuntos de datos que se manipulan en la práctica. Son muchas las monografías recientes sobre el tema que prueban que las técnicas de regresión no paramétrica tienen mucho que ofrecer. El caso multivariante resulta ser muy importante en la práctica.

Sean $(X_1, Y_1), \ldots, (X_n, Y_n)$ un conjunto de vectores aleatorios en $\mathbb{R}^{d+1}$ independientes e idénticamente distribuidos, donde Y_i son variables respuesta

escalares y X_i son predictores $\mathbb{R}^d$-valuados con densidad común f con soporte en $\mathbb{R}^d$. El problema de regresión no paramétrica multivariante consiste en estimar

$$m(x) = E(Y \mid X = x)$$

sobre un vector $x \in \text{sop}(f)$ sin el supuesto de que m pertenezca a ninguna familia paramétrica de funciones. En general se supone el modelo

$$Y_i = m(X_i) + v^{1/2}(X_i)\varepsilon_i$$

donde $v(x) = \text{Var}(Y \mid X = x)$ es finita y ε_i son variables aleatorias mutuamente independientes e idénticamente distribuidas con media cero y varianza uno, e independientes de los X_i.

Son muchos los estimadores propuestos que están siendo objeto de estudio. Los más simples de comprender intuitivamente, analizar matemáticamente e implementar computacionalmente, y por tanto, los que más atención est n recibiendo en los últimos años son los estimadores de regresión no paramétrica tipo núcleo. Para tales estimadores el parámetro de suavizamiento es conocido como *ancho de banda*, que en el caso multivariante resulta ser una matriz $H_{d\times d}$ simétrica definida positiva. La forma más general de un estimador núcleo con matriz ancho de banda global H es

$$\hat{m}(x, H) = n^{-1} \sum_{i=1}^{n} K_H(X_i - x)Y_i$$

con

$$K_H(u) = \mid H \mid^{-1/2} K(H^{1/2}u)$$

K es conocido como un núcleo d-variante, al que habitualmente se le exige ser acotado, con soporte compacto, y tal que $\int u^T u K(u)du = \mu_2(K)I_{d\times d}$ y todos los momentos de orden impar de K se anulan, esto es,

$$\int {u_1}^{l_1} \cdots {u_d}^{l_d} K(u)du = 0 \quad \forall l_1, \ldots, l_d \text{ tales que su suma sea impar.}$$

Una componente importante de todos los estimadores de regresión no paramétrica es la elección de un parámetro conocido como *parámetro de suavizamiento*, para lo cual, y sobre todo en el caso multivariante, es importante disponer de selectores automáticos a través de los datos.

Tradicionalmente los selectores más utilizados han sido los basados en el criterio de validación cruzada u otros métodos de penalización asintóticamente equivalentes. Sin embargo, tales técnicas están sujetas a un alto grado

de variabilidad muestral, y además, con frecuencia, son difíciles de implementar cuando se necesita seleccionar un vector o una matriz de parámetros de suavizamiento.

Los métodos alternativos m s prometedores pertenecen a un tipo conocido como reglas *plug-in*, puesto que implican estimaciones de funcionales desconocidos que intervienen en la fórmula para los par metros de suavizamiento óptimos asintóticamente con respecto al error cuadrático medio integrado (MISE). Además una aproximación *plug-in* requiere una estimación de la varianza residual.

El estudio de la consistencia de los estimadores es, por supuesto, una cuestión de gran importancia. El estudio de las propiedades asintóticas del sesgo y la varianza de cada estimador permitir determinar su razón de convergencia. Habitualmente el orden de magnitud del sesgo y la varianza no es el mismo cerca de la frontera del sop(f) que en el interior.

Asímismo es importante el estudio de la distribución asintótica de los estimadores que permita llevar a cabo estudios inferenciales, y establecer medidas de la bondad del ajuste.

1.4.3. Sistemas expertos. El sistema estadístico Jandstat

La complejidad de los temas estadísticos hace que, incluso existiendo herramientas computacionales, y a menudo precisamente por su existencia, se cometan errores groseros en la aplicación de las técnicas estadísticas. También, debido a la enorme amplitud de técnicas, muchas son desconocidas para muchos grupos de científicos. Por ello, uno de los problemas abiertos es la introducción de la Inteligencia artificial dentro de la Estadística Computacional para construcción de sistemas expertos estadísticos.

Supongamos la situación común en estadística aplicada de disponer de una matriz de datos X y un lenguaje estadístico $\mathcal{L}$ (o varios) con el que deseamos analizar los datos X. Este an lisis, enunciado en lenguaje no estadístico, plantea una serie de problemas que podemos resumir en los siguientes:

1. Es necesario conocer qué tipo de análisis estadístico es el adecuado para la realización del análisis.

2. Es necesario conocer el lenguaje para poder escribir el programa que realice el análisis deseado.

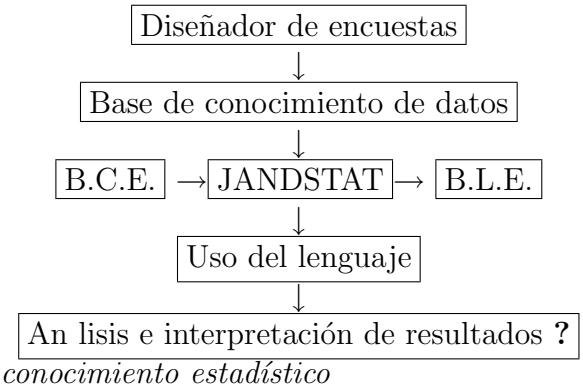
3. Es necesario saber interpretar los resultados que suministre el lenguaje.

El tipo de análisis a realizar depende de los objetivos buscados y también del *tipo* de datos que se poseen. Así, es habitual que, erróneamente, se apliquen técnicas preparadas para datos continuos a datos que son estrictamente nominales, debido a que los lenguajes presuponen una aplicación inteligente de los mismos por parte del usuario.

Por otra parte, ya que los lenguajes no contienen todas las técnicas es usual que la realización de un análisis concreto no se encuentre en el lenguaje que habitualmente se maneje, habiendo de recurrir a otro que ser posiblemente medio desconocido.

Estos son problemas que intentan resolver sistemas como **Blaise** para recogida de datos y los sistemas expertos, como **Designer Research** o **Statistical Navigator**, que intentan mejorar la selección del análisis de diferentes tipos de datos y la construcción de reglas para llevarlos a efecto.

Cuadro 1.1: Sistema JANDSTAT



B.C.E. *Base de conocimiento estadístico*
B.L.E. *Base de conocimiento de Lenguajes Estadísticos*

La solución que hemos dado –desarrollada bajo MS-DOS y Windows– consta de las siguientes partes:

1. Diseño de una base de conocimiento.

   Como mejora a la estructura de matriz de datos o incluso a la base de datos proponemos la construcción de una base de conocimiento que incluye no sólo los datos en si, sino el acceso a los mismos (propio de la base de datos) y la estructura interna de las variables. Para ello hemos realizado:

a) Un diseñador de estructura de variables, que se refleja externamente en un cuestionario para captura de datos, bien sobre papel bien interactivamente en ordenador o a partir de datos externos (ASCII, dBASE, LOTUS, etc..) ya que es común que los datos no se recojan directamente para el análisis.

b) Mantenimiento, mediante modificación de datos y estructuras, de una base de conocimiento.

2. Construcción y mantenimiento de una Base de conocimiento estadístico, que incluye tipos de análisis estadísticos, descripciones externas de los mismos (breves y largas) tipos de datos asociados y lenguajes que lo soportan.

3. Construcción y mantenimiento de una Base de Lenguajes estadísticos, en donde se recogen Lenguajes, Capacidades de los mismos y reglas para su programación.

Con estas condiciones el sistema es capaz de ofrecer sugerencias sobre el tipo de análisis apropiado a unos datos, con qué lenguaje puede analizarse y generar automáticamente un programa en el lenguaje apropiado con el que realizar el análisis seleccionado.

La fase de interpretación de resultados no es de posible implementación con los lenguajes convencionales, aunque sí en lenguajes como **S** y **Voyager**.

El sistema Jandstat viene a resolver los problemas de diseño, recogida y mantenimiento de datos; la selección de técnicas adecuadas y aplicación correcta de las mismas eligiendo el lenguaje estadístico adecuado y su manejo mediante escritura de programas.

1.5. Epílogo

Tras todo lo expuesto, parece evidente que la estadística es un campo vivo de investigación y que una parte muy importante de la misma es la Estadística Computacional, parte que, además, influye de modo decisivo en el propio concepto de la estadística. Tal vez el futuro de la estadística sea una estadística *sin modelo, exploratoria* de resumen de grandes masas de datos o gran cantidad de cálculo y de puesta en relieve de conceptos y estructuras ocultos en ellos.

Bibliografía

[1] R. Becker, J. Chambers, A. Wilks (1988), *The new S language*, Wadsworth & Brooks.

[2] Ed. D. Edwards and N.E. Raun (1988), *COMPSTAT, Proceedings in Computational Statistics, 8th Symposium held in Copenhagen* , Physica–Verlag Heidelberg.

[3] Ed. P. Dirschedl and R. Ostermann (1994), *Computational Statistics, 25th Conference on Statistical Computing*, Physica–Verlag Heidelberg.

[4] Ed. A. Prat (1996), *COMPSTAT, Proceedings in Computational Statistics, 12th Symposium held in Barcelona 1996*, Physica–Verlag Heidelberg.

[5] Francisco Cribari-Neto (1996), *Econometric Programming Environments*, cribari@ysidro.econ.uiuc.edu.

[6] T.F. Chan, G.H. Golub, R.J. LeVeque (1983), .Algorithms for computing the sample variance: Analysis and recommendations.en *American Statistician*, 37, 242–247.

[7] A.P. Dempster, N.M. Laird, D.B.Rubin (1977) "Maximum likelihood from incomplete data via the EM algorithm (with discussion).en *J.R.S.S.*, Series B, 39, 1–38.

[8] J. Dongarra, J.R. Bunch, C.B. Moler, G.W. Stewart (1978), *LINPACK Users Guide*, SIAM.

[9] B. Efron (1979), "Bootstrap methods: Another look at the jackknife.en *Annals of Statistics*, 7, 1–26.

[10] B. Efron, R. Tibshirani (1986), "Bootstrap Methods for Standard Errors, Confidence Intervals, and other Measures of Statistical Accuracy.[en] *Statistical Science*, 1, 54–77.

[11] EDSTAT-L, edstat-l@jse.stat.ncsu.edu, *Lista de discusión en Estadística.*

[12] S.I. Feldman, D.M. Gay, M.W. Maimone, N.L. Schryer (1990), *A Fortran-to-C Converter*, Computing Science Technical Report 149, AT&T Bell Laboratories.

[13] S. Geman, D. Geman (1984), "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images.[en] *IEEE Trans. Pattern Anal. Mach. Intelligence*, 6, 721–741.

[14] R. Ihaka, R. Gentleman (1996), R: A Language for Data Analysis and Graphics.[en] *JCGS*, 5, 299–314.

[15] E. Jantsch, O. Helmer, H. Kahn (1970) *Pronósticos del futuro*, Alianza Editorial.

[16] Journal of Computational and Graphical Statistics, A joint publication of ASA, IMS and IFNA.

[17] N. Karmarkar (1984), ."A new polynomial time algorithm for linear programming", *Combinatorica*, 4, 373–395.

[18] V. Klee, G. Minty (1972), "How good is the simplex algorithm?."[en] *Mathematical Programming: Recent Developements and Applications*, 263–282, Kluwer Academic Publishers

[19] D. E. Knuth (1981), *The Art of Computer Programming*, Addison–Wesley.

[20] L. Lebart, A. Morineau, J-P. F,nelon (1985), *Tratamiento estadístico de datos*, Marcombo.

[21] P. L'Ecuyer (1986), ."Efficient and Portable Combined Random Number Generators."[en] *Communications of the ACM*, 31, 742–749.

[22] E. Lozano (1995), *Aportaciones a las técnicas gr ficas para el estudio de normalidad y las causas de su p ,rdida*, Tesis doctoral, Universidad de Granada.

[23] Ramsay, J. O. (1982), *Some statistical aproaches to MDS data*, J.R. Statist. Soc. A, 145, Part 3, 285-312.

[24] Ed. C. R. Rao (1993), *Computational statistics, Handbook of statistics, v. 9*, North–Holland

[25] C. R. Rao (1994), *Estadística y Verdad aprovechando el azar*, PPU

[26] M. Reiser (1991), *The Oberon System*, Addison–Wesley

[27] J. R. Rice (1983), *Numerical methods, Software and analysis*, McGraw–Hill

[28] G.L. Steel, G.J. Sussman (1975), *Scheme: An Interpreter for the Extendended Lambda Calculus*, Memo 349, MIT Artificial Intelligence Laboratory.

[29] L. Tierney (1990), *Lisp-Stat*, John Wiley.

[30] L. Tierney (1996), Recent Developments and Future Directions in LispStat.[en] *JCGS*, 5, 250–262.

[31] R. Thisted (1988), *Elements of Statistical Computing. Numerical computation*, Chapman and Hall.

[32] J. W. Tukey (1977), *Exploratory Data Analysis*, Addison–Wesley.

[33] A. Zaman, G. Marsaglia (1991), .[A] New Class of Random Number Generators.[en] *Annals of Applied Probability*, 1, 462-480.