Ministerio de Salud Pública
Escuela Nacional de Salud Pública
"Carlos J. Finlay"

El Método de Remuestreo y su Aplicación en la Investigación Biomédica.

Autor: Dr. Aloysio Miranda Moles*.
Tutor Dr. Luis Carlos Silva Ayçaguer.**

Residente de tercer año en Bioestadística.
****Doctor en Ciencias, Investigador titular.***

Trabajo para optar por el título de
Especialista de 1er Grado en Bioestadística.

Ciudad de La Habana
2003

"El pensamiento estadístico
será un día tan necesario para el ciudadano eficiente
como la capacidad de leer y escribir."
H.G. Wells

AGRADECIMIENTO

El presente trabajo es la culminación de cuatro años de intenso bregar por el mundo maravilloso de la estadística. Durante esta “visita” he conocido a una gran cantidad de personas que me han ayudado en mi formación y han moldeado mi carácter, es por ello que saltando el formalismo clásico que esta sección ha demandado tradicionalmente, permítanme sólo mencionar a aquellas que son realmente artífice de este trabajo. Sé que todos aquellos que no menciono sabrán que les estoy eternamente agradecido.

A mis padres, sin su apoyo no estaría escribiendo estas líneas.

A mis hijos, por la deuda de amor que tengo con ellos y que nunca podré enmendar.

A mi novia Baby, por su apoyo incondicional en los momentos más difíciles de la realización de esta obra.

A mi profesor Luis Carlos, por su probidad y talento. Ha sido un verdadero honor trabajar a su lado.

RESUMEN

Desde finales de la década del 60 del pasado siglo comienza a desarrollarse un revolucionario método, conocido con el nombre "*resampling*" (remuestreo), para solucionar, por una parte, problemas en el marco de la teoría de probabilidades y la inferencia y, por otra, la desmotivación de los estudiantes durante los cursos de estadística. Dicho método se basa en el empleo de la simulación, la cual se verifica a través del uso intensivo de los recursos computacionales. Teniendo en cuenta el desconocimiento casi total por parte de la comunidad de estadísticos de nuestro país en relación con este recurso, así como los notables resultados que con él se pueden alcanzar, nos dimos a la tarea de realizar una revisión crítica de la bibliografía disponible y así desarrollar desde el punto de vista teórico-práctico las técnicas del *remuestreo*. En la presente tesis se examinan las etapas históricas por las que ha atravesado el método, y se expone el uso de este proceder para solucionar, tanto problemas clásicos (intervalos de confianza, pruebas de hipótesis, tamaño muestral), como tareas cuya solución analítica es difícil o imposible de obtener con el uso de las herramientas tradicionales (evaluación de procedimientos estadísticos, problemas de probabilidades, intervalos de confianza para estimadores de alta complejidad, Interim Analysis). Además, se ha realizado un esfuerzo orientado a facilitar el manejo del software *Resampling Stats* a través de la confección de una guía para la utilización eficiente del mismo. Como resultado de nuestro trabajo podemos concluir que el método de *remuestreo* a pesar de sus "bondades" no debe conducirnos a pensar, que es un método capaz de sustituir la inferencia clásica; más bien debe ser visto como una herramienta útil en situaciones donde ésta es inoperante o sumamente engorrosa.

ÍNDICE

A menudo la palabra “estadística” nos trae a la mente imágenes de números apilados en grandes arreglos y tablas, de volúmenes de cifras relativas a nacimientos, muertes, impuestos, poblaciones, ingresos, deudas, créditos y así sucesivamente (Huntsberger, 1983).

La Estadística como disciplina, sin embargo, es mucho más que sólo números apilados y gráficas bonitas. Es una ciencia con tanta antigüedad como la escritura; sus origines se remontan al antiguo Egipto, cuyos faraones lograron recopilar, hacia el año 3050 A.C, prolijos datos relativos a la población y a la riqueza del país.

Godofredo Achenwall (1719-1772) es quien la consolida y le da el nombre de "Estadística"[1], palabra que etimológicamente proviene del término latino "*status*"[2]; Von Scholer separó la teoría de la estadística de la aplicación práctica, y así pasó a ser la descripción cuantitativa de las cosas notables de un estado (Zupo, 2003).

No obstante, los orígenes de la teoría estadística moderna hay que verlos asociados a los juegos de azar. En 1654, Antoine Chevalier de Meré planteó a Blaise Pascal (1623-1662) y a Pierre de Ferrmat (1601-1665) un dilema. Se trataba de un juego consistente en lanzar un par de dados 24 veces. El problema consistía en decidir las condiciones bajo las cuales apostar a que al menos uno de los 24 resultados fuese un doble seis.

De las discusiones entre Pierre de Fermat y Blaise Pascal surgieron los principios fundamentales de la Teoría de Probabilidad (Lacourly, 2000).

Desde su mismo nacimiento, los teóricos de aquellos tiempos intuyeron el enorme poder de esta nueva rama de las matemáticas para lidiar con la incertidumbre y comprendieron que podían utilizarla para saber cuán exactas

[1] Hecho que se produjo en 1760
[2] Estado o condición de alguna cosa.

eran sus *estadísticas descriptivas*. Surgieron de esa forma los cimientos de lo que, con el decursar del tiempo, se convertiría en la herramienta más ampliamente utilizada y controversial de la ciencia estadística: la Inferencia (Simon y Bruce, 1991).

Sin embargo, a pesar del tiempo transcurrido y de la consistencia de los conceptos en que se sustenta, la teoría de probabilidades constituye una de las más frustrantes ramas del conocimiento humano por la frecuencia con que, tanto especialistas como principiantes en la materia, se equivocan al intentar resolver problemas relativamente “sencillos”.

En este sentido el filósofo Charles Santers Peirce expresaba: “En ninguna otra rama de las matemáticas es tan fácil para los expertos equivocarse como en la teoría de probabilidades” (Huff, 1959), observación especialmente válida cuando se pretende dar solución a problemas en que intervienen probabilidades condicionales. En efecto, un estudio procedente de la psicología cognitiva concluye: *“Cuando las personas tienen que enfrentarse con problemas en el ámbito de las probabilidades condicionales, con mucha frecuencia obtienen resultados erróneos*” (Piattelli, 1994).

En la segunda mitad del siglo XX, especialmente a partir de los años 70's, la Teoría de Probabilidad y su beneficiaria natural, la Estadística, se enseña por lo general en todas las carreras profesionales, y su conocimiento se exige prácticamente en todos los programas de postgrado de ingeniería y ciencias sociales. Sin embargo a los estudiantes no parece “pegársele” la materia; *“recuerdan el dolor pero no la esencia”* afirma Peter Ivars. Para muchos de ellos, pasar un curso de estadística constituye una dolorosa ceremonia de asistencia obligatoria en sus vidas estudiantiles, y la gran mayoría, incluso aquellos que lograron concluir satisfactoriamente sus cursos, la sacan inmediatamente de sus mentes para siempre.

Lo usual es, simplemente, memorizar ciertas reglas y aplicarlas o seleccionarlas ciegamente, sin entender nada o casi nada del por qué de tal elección. Enfrentarse con complejas formulaciones poco intuitivas, que en muchas ocasiones ni los propios profesores entienden (por ejemplo, cuál es el significado de e o Pi en la fórmula de la densidad Normal, distribución involucrada en la gran mayoría de las pruebas de hipótesis), resulta una tarea insuperable.

Por otro lado, las asunciones restrictivas propias de la inferencia estadística respecto a las distribuciones de las variables aleatorias, (p.ej. normalidad, independencia, homogeneidad de varianzas, etc.) que permiten la derivación analítica de las distribuciones muestrales de los estadísticos y la estimación de los parámetros que las caracterizan, en no pocas situaciones se incumplen. Como consecuencia, se obtienen resultados que pueden conducir a tomar decisiones erróneas o subóptimas.

La práctica habitual en la investigación ha sido obviar el análisis de los supuestos (Solanas y Sierra, 1992) y utilizar las pruebas estadísticas sin considerar la adecuación o no de las mismas, como si se tratara de un simple capricho de los matemáticos.

La comunidad de estadísticos hace ingentes esfuerzos por cambiar esta lamentable situación. Según el artículo de Ivars Peter, Barbara A. Bailar directora de la *Americam Statistical Association* planteaba: *"Muchas personas están ahora tratando de reformar el primer curso de Estadística."*

Por su parte, David S. Mooreof, autor de varios libros de estadística ampliamente usados en varias universidades, opina:

> *"Estamos tratando de reformar lo que hacemos, de modo que no estemos siempre impartiendo frías conferencias a estudiantes pasivamente sentados, tratando de transcribir lo que decimos a sus cuadernos de notas. La enseñanza de la estadística debe involucrar activamente a los*

estudiantes en el trabajo con los datos para que sean ellos mismos los que descubran los principios".

Una vía considerada para conseguir tal participación se abrió con el empleo de las computadoras y, especialmente, con el recurso de la simulación, arma poderosísima para resolver problemas complejos cuando la capacidad deductiva del hombre es incapaz de enfrentarse a ellos. El siguiente relato, presentado en un artículo elaborado por Julian Simon y titulado "The Philosophy and Pratice of Resampling Statistic", muestra con elocuencia el poderío de la simulación pero también el hecho de que se trata de un recurso de muy vieja data.

En el año 1615, jugadores italianos le llevaron al gran Galileo Galilei (1564-1642) un problema relacionado con los juegos del azar. Resulta que los teóricos de aquellos tiempos consideraban que la probabilidad de obtener en el lanzamiento de 3 dados un total de 9 era igual a la de que los tres números sumaran 10. Tal convicción se debía a que el número de vías para obtener esos totales (un 9 puede ser el resultado de obtener 126,135,144, 234, 225, 333) era el mismo que para obtener 10 (a través de 136,145, 523, 226, 244, 334). Sin embargo en el bregar del juego diario, habían encontrado que 10 salía con mayor frecuencia que el 9. Después que Galileo analizara el problema (para lo cual tuvo que inventar el artificio técnico del espacio muestral) determinó que efectivamente los jugadores tenían toda la razón: los matemáticos erróneamente emplearon las combinaciones en lugar de las permutaciones, que era lo correcto –para sumar un 9 existen 25 permutaciones, 2 menos que para obtener un 10-. Quedó demostrado así desde entonces el enorme poder de la simulación y su capacidad para arrojar resultados tan buenos como que los que se obtienen con los métodos teóricos basados en fórmulas.

Desgraciadamente, con el auge alcanzado en aquella época por la teoría de probabilidades, la simulación queda dormida para despertar con los truenos de la

segunda guerra mundial. Resulta que un grupo de físicos de la *Rand Corporation* (corporación donde se desarrolló uno de los más famosos dispositivos electrónicos para generar números pseudoaleatorios) comenzaron a usar sucesiones de números aleatorios para simular y estudiar procesos demasiados complejos para ser abordados mediante formulaciones teóricas, surgiendo de ésta forma la ampliamente conocida simulación “Monte Carlo”, así bautizada por Von Newman y Ulam en alusión a los juegos de azar de los casinos de la ciudad de Monte Carlo (Simon, 1997; Allen y colab, 1995; García, 1999).

Esta técnica se ha utilizado para que los estudiantes comprendan y entiendan teorías poco intuitivas, tales como la aproximación de la distribución binomial a la distribución normal y el teorema central del límite (Simon, 1997).

A pesar de que se ha usado con bastante frecuencia, sólo en las últimas décadas la técnica ha ganado el *status* de un método numérico completo (Allen, y col., 1995). Actualmente, la simulación Monte Carlo se usa rutinariamente en una gran diversidad de ramas de la investigación (Molinero, 2002).

En la primavera de 1967, el profesor Julian L. Simon de la Universidad de Illinois, retomando estas ideas, comienza a desarrollar un método revolucionario para enseñar y aplicar la estadística, ahora conocido como “*remuestreo*” (resampling). Sus bases descansan en la simulación Monte Carlo y su esencia consiste en usar el conjunto de datos observados o un conjunto de datos generados por un mecanismo -si estamos tratando un problema en el ámbito de las probabilidades- los cuales constituyen sin duda la mejor aproximación al modelo que queremos estudiar, para generar nuevas muestras hipotéticas, cuyas propiedades pueden ser fácilmente examinadas (Simon y Bruce, 1991). Es decir, permite trabajar directamente con el modelo físico simulándolo, en lugar de describirlo con procedimientos teóricos basados en supuestos que, en ocasiones, no tienen nada que ver con la realidad de la cual provienen los datos.

Esta misma lógica es aplicada tanto a asuntos propios de la teoría de probabilidades como a problemas de la estadística inferencial. La única diferencia radica en que cuando manejamos problemas de probabilidades el "modelo" es enteramente conocido de antemano.

El método fue formalmente enseñado por primera vez en 1967 y los resultados obtenidos fueron considerados "excelentes". Un año más tarde, bajo los auspicios del padre de la "nueva matemática", Max Beberman, Simon empleó este enfoque pedagógico con estudiantes de *high school* y luego constató que tal recurso había agradado a todos los que lo recibieron. Como consecuencia de tal acogida, se condujeron 3 ensayos controlados bajo la supervisión de Kenneth Travers (el primero en la Universidad de Illinois en 1973, el segundo en Polk Community College 1974, y un tercero en Olivet Nazarene College). Todos mostraron una marcada superioridad a favor del remuestreo sobre los métodos convencionales. Las principales conclusiones que sacaron los auspiciadores (Simon, Atkinson, y Shevokas, 1976) del estudio fueron las siguientes:

> *"Cuando el tiempo disponible es muy limitado o a los estudiantes les resulta difícil asirse firmemente a los métodos tradicionales, nosotros defendemos la enseñanza del método resampling. Donde hay más tiempo, y los estudiantes pueden aprender bien los métodos convencionales, nosotros abogamos por: (a) enseñar los procedimientos del método resampling desde el mismo principio del curso y luego enseñar los métodos tradicionales; de esta forma el estudiante es capaz de razonar y entender mucho mejor las complejas formulaciones; o (b) enseñar el método resampling después al método convencional como una alternativa al mismo problema, para que los estudiantes aprendan los métodos analíticos y brindarles una herramienta alternativa"*

En 1979, Bradley Efron (Efron,1979) desarrolla y publica el análisis formal del *Bootstrap* (uno de los dos procedimientos del *remuestreo*), término que procede

de la expresión inglesa *to pull oneself up by one's bootstrap* (que podría traducirse por: *levantarse mediante el propio esfuerzo*), tomada de una de las aventuras del Barón Munchausen, personaje de siglo dieciocho creado por el escritor Rudolph Erich Raspe, en la cual el barón había caído al fondo de un lago profundo y, cuando creía que todo estaba perdido, tuvo la idea de ir subiendo tirando hacia arriba de los cordones (*bootstrap*) de sus propias botas.

Es entonces cuando realmente el enfoque de Simon cobra una importante fuerza teórica y capta el interés de toda la comunidad de estadísticos, quienes comienzan a explorarlo y utilizarlo para solucionar una amplia gama de problemas en probabilidades e inferencia. Este proceder ha sido considerado por la *American Statistical Association* como "el único gran descubrimiento en estadística desde 1970" (Kotz and Johnson, 1992).

Para facilitar el proceso de simulación, en 1973 se ideó el software *Resampling Stats*, soporte técnico que impulsó el proceso de explotación del nuevo método. Aprovechando sus virtudes y haciendo uso del *remuestreo* se pueden resolver tanto problemas extremadamente sencillos, tales como determinar la probabilidad de obtener al menos una cara en el lanzamiento de dos monedas (erróneamente encarado por el gran matemático del siglo XVIII D'Alembert[3]), como problemas de una naturaleza o envergadura tal que no podrían ser resueltos con los métodos teóricos tradicionales. Por ejemplo, con la teoría formal no resulta nada fácil construir un intervalo de confianza para el alfa de Cronbach (la medida más ampliamente utilizada para aquilatar la llamada consistencia interna de un cuestionario, para cuya varianza hasta ahora nadie ha propuesto una solución analítica); el *resampling,* en cambio, resolvería el problema de manera inmediata.

[3] Citado por Simon en Resampling: The New Statistics.

Es tanto el poder de este proceder que se ha llegado a afirmar que la historia de la estadística sería otra si el desarrollo de la computación, imprescindible para aplicar el *remuestreo*, hubiera llegado antes (Ricketts y Berry, 1994).

Un gran número de artículos han probado la validez del *resampling*, o han proporcionado variantes válidas en situaciones como: Regresión (Efron, 1979; Wu 1986; Politis et al. 1997); datos censurados Efon y Tibshirani (1986); series temporales (Carlstein 1998; Bose 1990; Liu y Singh 1992; Kreiss y Franke 1992; Buhlmann 1997); ensayos clínicos (Berger 2002; Berger y Ivanova 2002)

En la actualidad los procedimientos de *remuestreo* son ampliamente conocidos y utilizados en diversos campos de la investigación. Se han ido incorporando como técnicas bastante habituales en el análisis e interpretación de cierto tipo de datos. La investigación médica contemporánea no escapa al empuje creciente con que estos métodos están compartiendo espacios con los convencionales. Una revisión realizada el 14 de Marzo del 2003 en 2 de las principales revistas médicas mundiales, *The Lancet* y *British Medical Journal* (ésta búsqueda sólo incluye de Enero de 1996 a Marzo del 2003), permitió constatar lo siguiente:

Número de Artículos encontrados que utilizan técnicas de simulación en el método.

Revistas	**Término de la referencia**		
	"Boostrap"	"Resampling"	"Monte Carlo Simulation"
The Lancet	30	5	280
B.M.J.	38	9	687

También efectuamos una búsqueda usando uno de los más potentes motores de búsqueda en el mundo ("*Google*"), y en la página web de la "*American Statistical Association*", la cual arrojó los siguientes resultados:

Número de referencias encontradas.

Sitios Web	**Término de la referencia**		
	"Boostrap"	"Resampling"	"Monte Carlo Simulation"
Google	654 000	159 000	130 000
A.S.A	260	172	195

En nuestro país, sin embargo, estos temas parecen ser virtualmente desconocidos. Así lo demuestra una revisión que efectuamos en la base de datos *CUMED* con todos y cada uno de los términos de referencias anteriores, la cual no arrojó ni un solo registro. No tenemos conocimiento de que se hayan usado por ningún investigador, y mucho menos que algunas de estas técnicas hayan sido empleadas en el análisis estadístico de investigaciones en el campo de la salud, a pesar de que, como hemos visto, su presencia es apreciable en el ámbito mundial.

Por otro lado, sabemos por nuestra propia experiencia, que cuando se imparte un curso de estadística al personal sanitario, al finalizarlo algunas personas "saben" pero casi nadie "entiende" verdaderamente la esencia conceptual de lo que han estudiado, y en el caso de quienes van a llevar a cabo sus tesis de terminación de residencia, los educandos salen a la caza de algún bioestadístico para que les ayude a desentrañar esa madeja de datos que en no pocas ocasiones los ahoga.

La investigación contemporánea exige una alta dosis de objetividad en el método que se utilice para el tratamiento de la información empírica, y una de las vías para tratar de no alejarnos de ese *desideratum* la ofrece el uso racional de las diferentes técnicas que brinda la estadística.

Se impone la búsqueda de alternativas capaces de romper con esta frustrante situación y a nuestro juicio el *remuestreo*, un procedimiento sumamente intuitivo, constituye una posibilidad atrayente para conseguirlo. Sus potencialidades son notables tanto para la enseñanza de la bioestadística, como para solucionar los problemas que se presentan en la investigación.

Consecuentemente con todo lo anterior, el presente trabajo procura facilitar el proceso de comprensión y explotación de los procedimientos bajo la rúbrica del *remuestreo*. Ello dejaría el dividendo, como mínimo, de incrementar la cultura científica de la comunidad de los profesionales sanitarios, y en particular la formación estadística de nuestros investigadores (especialmente, aquellos que se dedican a la bioestadística). Se procura asimismo favorecer que dichos profesionales puedan explorar por sí mismos las posibilidades de aplicar este recurso tanto en la docencia como en la práctica investigativa.

OBJETIVOS

1. Exponer los fundamentos teóricos básicos del método de *remuestreo*.

2. Confeccionar una guía para la utilización eficiente del software Resampling Stats.

3. Ilustrar la solución operativa de algunos problemas convencionales de la estadística y las probabilidades mediante los procedimientos de *remuestreo*.

4. Explorar e ilustrar las potencialidades del *remuestreo* en la solución de problemas estadísticos complejos.

MATERIAL Y MÉTODO

El trabajo se dividió en dos fases. En la primera, se realizó una revisión crítica de la bibliografía disponible relacionada con los siguientes temas:

- Método Resampling
- Prueba de Permutación
- Bootstrap
- Simulación Monte Carlo

La fuente principal para dicha revisión fue Internet, soporte informativo universal que, adecuadamente explotado, provee de una cobertura integral de temas como el que nos ocupa.

Se comenzó la revisión de la manera natural: examinando los artículos disponibles en el sitio web http://www.resample.com, página auspiciada por los promotores del método, lo que nos permitió alcanzar un primer nivel de conocimiento teórico. Dicho nivel se fue elevando en la medida que revisábamos otras fuentes. Ente otras (véase Bibliografía), nos nutrimos de dos libros de texto escritos por los "padres" del *resampling,* J. Simon y B. Efron: "Resampling: The New Statistics" y "An Introduction to the Bootstrap".

Una vez alcanzados los conocimientos suficientes era imprescindible contar con un software que nos facilitara desarrollar y poner en práctica dichos conocimientos. Para resolver este problema contactamos, vía correo electrónico, con los promotores y, luego de pactar un trato (debíamos mandar una copia de nuestro trabajo una vez terminado) recibimos gratuitamente el software "Resampling Stats", sin el cual nuestro esfuerzo hubiera resultado cuando menos estéril.

Contando con los recursos necesarios para desarrollar los objetivos propuestos, se pasó a la segunda fase, la cual estuvo caracterizada por la exploración en profundidad del programa y la aplicación práctica de los mencionados conocimientos teóricos.

El trabajo se dividió en Capítulos, y éstos en Secciones para lograr una mejor organización y estructuración de la tesis. Cada uno de dichos Capítulos está relacionado y le da salida a un objetivo específico. Así, en el capítulo número uno (Sección 1.1), se realiza un breve viaje a través del desarrollo histórico de la metodología *resampling*, tocando sus hitos históricos. Posteriormente, en la Sección 1.2 se desarrollan desde el punto de vista teórico las dos técnicas abarcadas por dicha metodología. El Capítulo finaliza con la exposición de las principales ventajas y desventajas del método (Sección 1.3).

El Capítulo número 2 está dedicado a la confección de una guía que facilita la utilización eficiente del Software *Resampling Stat* versión 5.0.2. La guía está confeccionada de forma tal que se van introduciendo paulatinamente los temas de manera que el lector no pierda el hilo conductor de la lectura. Las Secciones iniciales están dedicadas a la introducción del lector en el ambiente propio del *software*; mediante la utilización de gráficos se muestra cómo operar el programa.

En las Secciones subsiguientes se introducen los comandos disponibles en *Resampling Stats*. Dichos comandos se explican mediante ejemplos que se adecuan a cada situación. El modo elegido para dar tal explicación es el siguiente: en lugar de irlos introduciendo por orden alfabético, como es usual, con el fin de contribuir a su más rápida comprensión, hemos ido haciéndolo de manera que aquellos comandos que no necesiten de otros para su explicación sean los primeros, de forma tal que, llegado a un determinado nivel, no haya necesidad de saltar páginas con el fin de esclarecer el significado de esos otros comandos. Las Secciones 2.7 y 2.8 están dedicadas a ilustrar el manejo de los

archivos propios de *Resampling Stats* mediante la utilización de ejemplos. Este capítulo cierra con una exposición de las que, a nuestro juicio, son las principales virtudes y defectos del *software*.

Para dar cumplimiento al tercer objetivo específico nos planteamos prácticamente todos los problemas estadísticos más convencionales (construcción de intervalos de confianza y pruebas de hipótesis más usuales) y un buen número de preguntas relacionadas con la teoría de probabilidades. Cada uno de los problemas primero se resuelven mediante el uso de artificios manuales (lanzamiento de una moneda, selección de bolas de una urna, etc) para su más fácil comprensión y posteriormente se utiliza el software Resampling Stats para su solución formal. Los comandos utilizados en los programas se explican detalladamente de modo que el lector se va familiarizando con el manejo del software paralelamente al desarrollo de la exposición. En aquellos casos en que se contaba con la solución independiente (teoría de probabilidad o inferencia clásica) ésta se expuso de modo que se puede “palpar” la similitud de los resultados obtenidos entre ambos métodos.

Para realizar la exploración anunciada en el cuarto Capítulo, nos dimos a la tarea de buscar 4 ejemplos de naturaleza marcadamente distinta (el primero es un problema puro de probabilidad; luego se aborda una tarea de estimación; el tercer ejemplo ilustra cómo solucionar un problema metodológico de la estadística y el último concierne al tratamiento de datos experimentales) cuya derivación analítica fuese difícil o imposible mediante el uso de las herramientas habituales de la teoría de las probabilidades y la inferencia. Mediante una exposición detallada y siguiendo la estrategia desarrollada en el tercer Capítulo (en lo referente a la explicación de los comandos utilizados en la confección de los programas) ilustramos la solución de cada uno de ellos.

1.1 Breve reseña histórica.

La simulación y los métodos Monte Carlo (véase Introducción) han sido informalmente usados por siglos, aunque sólo se conocen evidencias explícitas de su utilización algo más formal desde mediados del siglo XVIII. Dos aplicaciones que se pueden resaltar son:

- El problema de Buffon (interesante problema en el campo de la geometría probabilística publicado en 1777 y cuyo resultado tiene especial atractivo, ya que se trata de un procedimiento que desemboca en un valor tan aproximado como se quiera del número π[4]).

- El uso del muestreo experimental en investigaciones relacionadas con la distribución t, llevado a cabo en 1908, por W.S. Gosset (Student).

Aun así, el desarrollo más sistemático y orgánico de la simulación y de los métodos Monte Carlo se remonta a la segunda guerra mundial, cuando Von Newman y Ulam, en sus investigaciones relacionadas con la bomba atómica, hicieron uso de la generación intensa de números aleatorios para resolver ecuaciones diferenciales y utilizaron métodos de reducción de la varianza para lograr una mayor eficiencia en este proceso (Aspuru-Guzik y Perusquía-Flores, 1999).

Posteriormente, alrededor de 1948, Fermi, Metrópolis y Ulam, obtienen estimadores Monte Carlo para los valores propios de la ecuación de Schrödinger, una de las fórmulas más utilizadas en la química cuántica y la física nuclear. (el lector interesado puede encontrar información detallada en Aspuru-Guzik y Lester, 2002).

[4] Una información detallada se puede encontrar en Schroeder (1974).

A finales de la década del 50 del pasado siglo, Dwass (1957), y Chung y Fraser (1958) publicaron, cada uno por separado, sendos artículos sobre la utilización de la prueba de permutación exacta de Fisher[5]. Ambos notaron que, con una muestra grande, dicha prueba no era factible de ejecutar debido a la dificultad de los cálculos (entonces no se contaba con las potentes computadoras de hoy). Sugirieron entonces que una sub-muestra aleatoriamente seleccionada de las posibles permutaciones podía proporcionar los beneficios de la prueba sin las excesivas demandas computacionales del procedimiento ortodoxo. Este artificio técnico sería la base sobre la que descansaría el *test de permutación estocástico,* una de las técnicas del *remuestreo*.

En este marco, el profesor Julian Simon, durante un curso de metodología de la investigación que impartía a 4 estudiantes de doctorado en 1967, se vio inclinado a utilizar la simulación Monte Carlo como recurso para evitar el empleo de formulaciones teóricas en la solución de problemas en el ámbito de las probabilidades, ya que dichas fórmulas sólo conseguían producir desánimo en los estudiantes. Este recurso se hizo rápidamente extensivo a la inferencia estadística surgiendo así los cimientos del *remuestreo*.

La idea básica y su mayor aplicabilidad consisten en abordar el problema al cual nos estemos enfrentando mediante simulación, para luego tomar decisiones desde el punto de vista probabilístico. Para ilustrar esta idea, supongamos que nos invitan a participar en el siguiente juego:

[5] Los detalles de esta técnica la podemos encontrar en Fisher (1935).

Sobre un tablero hay seis cuadrados marcados con los números 1, 2, 3, 4, 5, 6. Colocas tanto dinero como desees apostar en cualquiera de esos cuadrados. Se arrojan entonces tres dados. Si el número que se ha elegido aparece en un sólo dado, recuperas el dinero de la apuesta más una cantidad igual. Si el número aparece en dos de los dados, recuperas el dinero apostado más dos veces esa misma cantidad. Si el número aparece en tres dados, recuperas el dinero más tres veces la misma cantidad. Por supuesto, si el número no aparece en ninguno de los dados, el individuo que nos invita se queda con nuestro dinero. ¿Resulta ventajoso jugar en esas condiciones?

Antes de enrolarnos en tan singular "juego", seguramente desearíamos conocer cuál es el valor esperado de la ganancia a través de nuestras probabilidades de éxito de uno y otro (1, 2 ó 3 coincidencias). Aceptaríamos la propuesta si, al concluir, la diferencia entre lo recibido y lo pagado es positiva; es decir, si dicho valor esperado es positivo. Como se verá, esta solución exige conocimientos de teoría de distribuciones que están por encima de los que posee quien no esté avezado en estadística. Una persona en este caso podría razonar del modo siguiente[6] : la probabilidad de que mi número aparezca en un dado es de 1/6, pero como los dados son tres, la probabilidad debe ser 3/6 o 1/2, por lo tanto el juego es justo. Incluso dirías que tienes las de ganar, porque en una de cada dos partidas saldrá el número que has elegido y cubrirías pérdidas. Pero si además el número sale en dos dados comenzarás a ganar dinero. Por supuesto que esto es lo que el banquero desea que se suponga, pues la suposición es falaz.

[6] Razonamiento muy común; si no fuera así, no se explicaría la cantidad de individuos que se someten a una proposición, que como se verá, es leoninamente ventajosa para el banquero.

La solución probabilística correcta es la siguiente: Los resultados posibles son 6^3= 216. Supongamos que el jugador eligió el 4. Llamemos n_1 al número de lanzamientos que contienen un solo 4, n_2 al número de desenlaces que contienen dos veces el 4 y n_3 al de aquellos en que los tres dados producen un 4. las probabilidades respectivas son p_1, p_2 y p_3 donde $p_i = \frac{n_i}{216}$. No es difícil (aunque tampoco muy fácil para un lego) comprender que: n_1=75, n_2=15 y n_3=1.

Llamemos x al dinero apostado. Se pierde el dinero (x) si no se da ninguno de los tres casos (n_1, n_2, y n_3), se obtiene 2x si se da el primero de ellos (se gana x), se obtiene 3x en el segundo caso (se gana 2x), y se nos devuelve 4x en el tercero (en este caso, se gana 3x). Llamemos Y a la ganancia y C al número de coincidencias. La situación se sintetiza así:

C_i	Y	Prob(C=i)
0	-x	$125/216$
1	2x	$75/216$
2	3x	$15/216$
3	4x	$1/216$

Por lo tanto la ganancia esperada es:

$$E(y) = \frac{(125)\bullet(-x)+(75)\bullet(x)+(15)\bullet(2x)+(1)\bullet(3x)}{216} = -\frac{17}{216}x = -0.0787$$

O sea, la ganancia esperada en negativa. Lo que quiere decir que a la larga, por cada peso jugado, independientemente de la cantidad de dinero apostado, usted solo recuperará 92 centavos aproximadamente (más exactamente, perderá 7.87 centavos).

Otra estrategia, mucho más simple, y que casi nunca conduce a cometer errores, consiste en lanzar los dados muchas veces, y computar la cantidad de dinero que se recupera en cada caso, promediar dicha cantidad y examinar si ese número es menor o mayor de lo apostado. Esta segunda estrategia puede ser desarrollada sin acudir a la experiencia física mediante las técnicas de *remuestreo* (en el Anexo 1, se puede encontrar la solución al problema por esta vía)

Como recurso pedagógico, el *remuestreo* fue bien acogido por parte de los estudiantes; sin embargo, no fue hasta una década mas tarde, con la publicación formal del *bootstrap* (Efron, 1979), que empieza a ceder la resistencia encontrada en la comunidad de estadísticos. A partir de entonces, coincidiendo con la universalización de las PC, los procedimientos de simulación basados en *remuestreo* se comienzan a utilizar para solucionar una amplia gama de problemas en distintas ramas del saber.

1.2 Introducción al *remuestreo.*

El término *remuestreo* (resampling), según palabras del propio Simon, es aplicado a aquellas técnicas de simulación empleadas en la teoría de probabilidades y la inferencia estadística que, a partir de los datos observados (es decir, a partir de un modelo del proceso que deseamos estudiar), generan nuevas muestras simuladas de igual tamaño que la muestra original (a las que llamaremos *remuestras* para subrayar el hecho, de que son obtenidas a partir de la(s) muestra(s) observada(s)) con el propósito de examinar los resultados obtenidos en esas *remuestras*. En principio sólo han sido consideradas dos técnicas: la *Prueba de permutación estocástica* y *el bootstrap*.

1.2.1 Prueba de Permutación Estocástica.

La principal aplicación de la Prueba de Permutación Estocástica (PPE) es en el contexto de las pruebas de hipótesis. Su fundamento teórico se sustenta en la Prueba Exacta de Ficher (PEF), la cual fue introducida en 1930 para resolver un caso particular de la inferencia estadística: análisis de independencia en tablas de 2x2.

Recordemos que mediante la PEF podemos calcular, utilizando la distribución hipergeométrica, la probabilidad de obtener una distribución tan o más extrema como la observada en una situación particular. Ello nos coloca en una posición ventajosa a la hora de pronunciarnos sobre la veracidad de cierta hipótesis.

Para fijar ideas supongamos que estamos interesados en evaluar la eficacia de un nuevo medicamento para curar el cáncer, al cual denominaremos C. Para ello seleccionamos 40 pacientes con cáncer y asignamos 20 aleatoriamente al grupo que recibirá la droga C (n_1); los restantes 20 se asignan a un grupo (n_2) que recibirá como medida terapéutica un placebo. Supongamos además que luego de realizar dicha experiencia se obtienen los siguientes resultados:

Tabla No 1.1: Resultados del experimento.

Estatus del paciente	Grupo experimental	Grupo Control	Total
Fallecido	5	13	18
Curado	15	7	22
Total	20	20	40

Es fácil percatarse que la proporción de fallecidos en el grupo experimental (p_1) fue de 0.25 y en el grupo control (p_2) ascendió a 0.65. La diferencia de proporciones ($\hat{q}$)es p_2-p_1= 0.4. Llegado a este punto nos interesa saber si esa diferencia encontrada puede ser debida al azar o se debe a que el nuevo tratamiento realmente es más eficaz que un placebo en la cura del cáncer.

Para dirimir este asunto, lo usual es recurrir a las populares y no menos controversiales pruebas de hipótesis, a través de las cuales tratamos de pronunciarnos sobre la veracidad de cierta hipótesis prefijada de antemano como supuestamente cierta, conocida como hipótesis de nulidad: H_0.

En el ejemplo, se tendría que asumir primero que la proporción de fallecidos en los dos grupos es la misma, para luego poder determinar la probabilidad de obtener una diferencia de proporciones tan grande o mayor que la observada en las muestras ($\hat{q}$ =p_2-p_1≥0.4). Recuérdese que habiendo observado $\hat{q}$, el nivel de significación alcanzado de la prueba (NSA) se define como la probabilidad de observar al menos un valor tan alto como el alcanzado por $\hat{q}$ cuando la hipótesis de la no diferencia (H_0) es verdadera (Efron y Tibshirani, 1993).

NSA= Prob($\hat{q}$ *>=$\hat{q}$ / H_0)

Donde: $\hat{q}$ es el estimador de interés, y $\hat{q}$ * es una variable aleatoria que representa los valores posibles de$\hat{q}$.

Mientras menor sea el valor del NSA, mayor será la evidencia en contra de la hipótesis nula; se ha hecho muy popular plantear que si el NSA de la prueba es menor que 0.05, entonces H_0 ha de ser rechazada.

Una vía para calcular el NSA en situaciones como la del ejemplo desarrollado es utilizando la PEF. Mediante dicha prueba se obtiene la probabilidad de observar por azar una distribución de frecuencias tan o más extremas como la alcanzada en una situación particular, considerando los cuatro márgenes de la tabla de contingencia fijos. En este ejemplo se tendría que calcular:

$$p = \frac{\binom{18}{5}\bullet\binom{22}{15}+\binom{18}{4}\bullet\binom{22}{16}+\mathbf{K}\binom{18}{0}\bullet\binom{22}{20}}{\binom{40}{20}} = 0.0124$$

Nótese que lo que se ha hecho no es más que calcular analíticamente la frecuencia relativa del número de veces que se puede obtener una distribución como la observada en el experimento o más extrema (5 fallecidos o menos y 15 curados o más) si consideráramos seleccionar 20 individuos de un total de 40 (18 fallecidos y 22 curados).

Antes de la era de las computadoras personales (PC), especialmente cuando el tamaño de la muestra es grande, el cálculo derivado de la PEF resultaba cuando menos engorroso de ahí la propuesta hecha por Dawass y Chung y Fraser a finales de la década del 50 del pasado siglo (véase sección 1.1).

Sobre la base de esta propuesta y haciendo uso del desarrollo alcanzado por las PC es que se erige y desarrolla la PPE. Dicha prueba consiste en seleccionar mediante simulación una muestra aleatoria de las posibles permutaciones y examinar en cada una de ellas si el estadístico de interés $\hat{q}^*$ es mayor o igual que el valor observado de $\hat{q}$.

Siguiendo este razonamiento, podemos combinar las n_1+n_2 observaciones en un sólo grupo de tamaño N* (en nuestro ejemplo quedará conformado por 40 individuos, 18 fallecidos y 22 curados), tomar una remuestra sin reemplazo de tamaño n_1 (los restantes valores formarían la remuestra n_2), computar $\hat{q}*$ (la diferencia entre las proporciones en las remuestras), repetir el proceso B veces y generar una distribución de B valores de $\hat{q}*$, la cual es conocida como distribución de permutación. Calcular entonces la proporción de experiencias en las que el valor de $\hat{q}*$ fue mayor o igual a $\hat{q}$ (forma análoga de determinar en cuántas experiencias se obtuvo una distribución tan o más extrema que la observada en el experimento), si este valor es menor que 0.05 rechazamos la hipótesis nula con un nivel de confianza del 95% (Efron y Tibshirani, 1993; Simon, 1997).

Para efectuar una simulación coherente con el fundamento anterior, simplemente podemos seguir los siguientes pasos:

1. Colocamos en una urna 5+13=18 bolas rojas (representan a los fallecidos de cáncer) y 15+7=22 bolas blancas (pacientes no fallecidos).

2. Seleccionamos aleatoriamente, sin reemplazo, 20 bolas (grupo experimental, n_1).

3. Calculamos la proporción (p_1) de bolas rojas (proporción de fallecidos en n_1)

4. Consideramos las restantes 20 bolas como representantes del grupo control, (n_2).

5. Calculamos la proporción (p_2) de bolas rojas (proporción de fallecidos en n_2).

6. Calculamos la diferencia p_2-p_1.
7. Si la diferencia es igual o mayor a 0.4, registramos de algún modo este resultado como “positivo”.
8. Repetimos cien veces los pasos previos.
9. Calculamos la proporción de resultados “positivos” registrados (estimamos el NSA).

Al ejecutar este proceso (véase Tabla No. 1.2), en 2 ocasiones obtuvimos una diferencia entre las proporciones de fallecidos en los dos grupos al menos igual que la observada en el estudio original (0.4). Lo que quiere decir, que la probabilidad de obtener una diferencia como la observada producto del azar es tan baja (p= 0.02) que podemos rechazar la casualidad como origen de la diferencia encontrada (nótese la extrema similitud con el resultado obtenido mediante la PEF). Concluyendo, podemos afirmar (si estamos dispuesto a correr el riesgo que significa haber realizado la simulación solo 100 veces) que existe evidencia muestral suficiente para considerar la droga C eficaz en la cura del cáncer.

Tabla 1.2: Distribución de la diferencia de proporciones originada mediante simulación.

B	p2-p1	B	p2-p1	B	p2-p1	B	p2-p1
1	0	26	0	51	0.2	76	0.2
2	0.2	27	0.1	52	0.1	77	0.2
3	0.2	28	0.1	53	0.1	78	**0.4***
4	0	29	0	54	0.1	79	0.1
5	0.3	30	0	55	0	80	0.1
6	0.3	31	**0.4***	56	0.2	81	0.0
7	0.1	32	0.1	57	0.1	82	0.0
8	0.2	33	0.2	58	0.2	83	0.2
9	0.1	34	0	59	0.2	84	0.1
10	0.1	35	0.3	60	0.3	85	0.3
11	0	36	0.1	61	0.1	86	0.3
12	0.1	37	0.2	62	0	87	0.1
13	0.3	38	0	63	0.1	88	0.2
14	0	39	0.1	64	0.2	89	0.2
15	0.2	40	0.2	65	0.1	90	0.1
16	0.2	41	0.1	66	0	91	0.1
17	0.1	42	0.3	67	0	92	0.1
18	0.1	43	0.2	68	0.1	93	0.0
19	0	44	0.2	69	0.1	94	0.1
20	0.2	45	0.3	70	0.1	95	0.3
21	0.1	46	0.2	71	0.2	96	0.0
22	0.2	47	0.1	72	0.1	97	0.0
23	0.2	48	0	73	0.0	98	0.2
24	0	49	0.1	74	0.1	99	0.1
25	0.1	50	0.1	75	0.0	100	0.2

Este pensamiento puede ser extensivo a la situación en la cual el estadístico de interés es la diferencia de medias en muestras independientes (véase sección 3.4).

1.2.2 El Bootstrap

El bootstrap constituye la técnica más versátil y conocida dentro del método de *remuestreo.* En esencia es bastante parecida a la PPE pero con una diferencia que le imprime un sello especial: la selección de las *remuestras* se hace **con reemplazo**.

La idea básica, en síntesis, es tratar la(s) muestra(s) como si fuera la población, (debido a la analogía entre muestra y población) y a partir de ella extraer **con reposición** un gran número de *remuestras* de tamaño n. Así, aunque cada *remuestra* tendrá el mismo número de elementos que la muestra original, mediante el remuestreo con reposición cada una podría incluir algunos de los datos originales más de una vez. Como resultado, cada *remuestra* será, muy probablemente, algo diferente de la muestra original; con lo cual, un estadístico $\hat{q}^*$, calculado a partir de una de esas *remuestras* tomará un valor diferente del que produce otra *remuestra* y del $\hat{q}$ observado. La afirmación fundamental del *bootstrap* es que una distribución de frecuencias de esos $\hat{q}^*$ calculados a partir de las *remuestras* es una estimación de la distribución muestral de $\hat{q}$ (Mooney y Duval, 1993).

Más formalmente los pasos básicos en la estimación bootstrap son los siguientes (HincKley, 1988) (Lunneborg, 2001):

1. Construir una distribución de probabilidad empírica, $\hat{F}$(x), a partir de la muestra, asignando una probabilidad de 1/n a cada punto, x_1, x_2 …, x_n. Esta es la función de distribución empírica (FDE) de x, la cual es el estimador no paramétrico de máxima verosimilitud de la función de distribución de la población, F(x).

2. A partir de la FDE, $\hat{F}$ (x), se extrae una muestra aleatoria simple de tamaño n con reposición.

3. Se calcula el estadístico de interés $\hat{q}$, a partir de esa "*remuestra*"; llamémosle $\hat{q}^*_i$ al resultado.

4. Se repiten los pasos 2 y 3 en B ocasiones, donde B es un número "grande"[7].

5. Construir una distribución de probabilidad a partir de los B valores $\hat{q}^*_i$, asignando una probabilidad de 1/B a cada punto, $\hat{q}^*_1$, $\hat{q}^*_2$,…, $\hat{q}^*_B$. Esta distribución es la estimación *bootstrap* de la distribución muestral de $\hat{q}\,[\tilde{F}^*(\tilde{q}^*)]$.

Como resultado de este proceso se pueden derivar al menos 3 aplicaciones prácticas:

§ valorar el sesgo y el error muestral de un estadístico calculado a partir de una muestra.

§ establecer un intervalo de confianza para un parámetro estimado.

§ realizar pruebas de hipótesis respecto a uno o más parámetros poblacionales.

[7] Teóricamente, la magnitud de B en la práctica depende de las pruebas que se van a aplicar a los datos. Se ha afirmado que, B debería ser de entre 50 a 200 para estimar el error típico de $\hat{q}$, y de al menos de 1000 para estimar intervalos de confianza alrededor de $\hat{q}$ por el método del percentil (Efron y Tibshirani, 1986, 1993). Sin embargo, esto tiene reducida importancia en la actualidad, pues las computadoras personales son tan rápidas que no tiene sentido tener un afán especial en trabajar con valores bajos de B y, por otra parte, nunca es pernicioso que B sea demasiado grande. Por lo general, con B=1000 se suelen conseguir buenos resultados y valores de B superiores a 5000 yo no agregan ninguna ventaja .

Estimación Bootstrap del error típico.

El bootstrap fue introducido como un método basado en cálculos intensivos mediante ordenador para estimar el error muestral de un estadístico. Tiene la ventaja sobre los métodos tradicionales de no requerir formulaciones teóricas y poder emplearse para cualquier estimador, por complejo que éste sea (véase un ejemplo donde se aplica a un estadígrafo muy complicado en la sección 4.2).

Explícitamente, la estimación *bootstrap* del error muestral de un estadístico es como sigue

- Se extraen B *remuestras bootstrap* independientes de la FDE(x).
- Se computa el estadístico de interés en cada una de las B *remuestras*, obteniendo $\hat{q}^*_b$.
- Se estima el error muestral de $\hat{q}$ mediante la desviación estándar de la función de distribución obtenida a través de los B $\hat{q}^*_i$, es decir a través de $\tilde{F}^*(\tilde{q}^*)$.

Intervalos de confianza Bootstrap

Existen 3 métodos a través de los cuales se pueden construir intervalos de confianza *bootstrap*:

1. Método de aproximación normal
2. Método de los percentiles
3. Método de los percentiles corregidos.

El primero de ellos utiliza la misma estructura de los procedimientos paramétricos en la construcción de intervalos de confianza. Si es posible asumir que el estadístico se distribuye según la curva normal pero el cálculo del error típico resulta analíticamente difícil o no existe fórmula para su cálculo, entonces podemos emplear la distribución muestral *bootstrap* para estimar el error típico e insertarlo en la correspondiente expresión del IC paramétrico.

El método del percentil hace uso literal de la idea básica del *bootstrap*, es decir $\tilde{F}^*(\tilde{q}^*)$ se aproxima a $\tilde{F}(\tilde{q})$. La idea es muy simple: un intervalo con un nivel de confianza 1-α incluye todos los valores de $\tilde{q}^*$ entre los percentiles $\frac{a}{2}$ *y* $(1-\frac{a}{2})$ de la distribución de $\tilde{F}^*(\tilde{q}^*)$.

El método del percentil conserva la esencia no-paramétrica del enfoque *bootstrap* y libera al usuario de las asunciones de la estadística paramétrica (véase ejemplos en los Capítulos 3 y 4).

El tercer método, es similar al procedimiento anterior; lo único que cambia es el modo de calcular los percentiles para obtener el intervalo. Según Efron y Tibshirani (1993), donde se explica en detalle cómo se computan los percentiles corregidos, este método es el más adecuado, ya que corrige la asimetría que pudiera presentar la distribución muestral del estadístico.

Pruebas de hipótesis *bootstrap*.

El algoritmo de trabajo para resolver un problema en el marco de las pruebas de significación estadística empleando la técnica *bootstrap* (Figura 1), es bastante similar al descrito anteriormente para la PPE, aunque el hecho de seleccionar las remuestras **con reemplazo** implica una diferencia teórica básica:

- En la *prueba de permutación* el NSA es un valor exacto, ya que la estimación de p no depende de ningún parámetro desconocido (recuérdese que es un resultado de la distribución hipergeométrica); sin embargo si se efectúa un muestreo con reposición entonces la probabilidad buscada depende de la distribución binomial cuya función de distribución viene dada por la siguiente expresión:

$$p(x) = \binom{n}{x} p^x \bullet q^{n-x}$$

Donde n: es el número de realizaciones de la variable aleatoria x, p es la probabilidad de éxito (en el caso que nos ocupa p no es otra cosa que el valor del estadístico de interés $\hat{q}$) y q= (1-p).

Como se puede observar el NSA depende del parámetro desconocido p. Por lo que es necesario estimarlo a través de las muestras seleccionadas. Por este motivo es que se dice que el NSA, en este caso, es una aproximación y la *PPE* tiene mayor potencia (una información detallada se puede encontrar en Corcoran y Mehta (2002)). Sin embargo hay que decir que este escollo teórico no tiene mayor implicación práctica ya que a la larga se obtienen casi idénticos resultados con ambos métodos (Simon, 1997).

Figura 1: Estimación del NSA mediante el *bootstrap*.

1. Seleccionar aleatoriamente B remuestras de tamaño n1+n2 **con reemplazo** de la muestra original. Las primeras n1 observaciones las llamaremos n^*_{1i}; las restantes observaciones las llamaremos n^*_{2i}, para i=1,2,…B.
2. Calcular el estadístico de interés $\tilde{q}$ en cada remuestra, obteniendo así B $\tilde{q}^*_b$.
3. Estimar el NSA_{boot} mediante:

$$\overline{NSA}_{boot} = \frac{n}{B}$$

donde n es el número de veces para las que ocurre $\tilde{q}^*_i \geq \tilde{q}_{obs}$,

1.2.3 Nacimiento de Resampling Stats.

El principal problema que enfrentaron en una primera etapa los seguidores de *resampling* era lo soporífero que resultaba simular el problema de interés con el uso de artificios manuales (selección de bolas de una urna, utilización de tablas de números aleatorios, lanzamiento de monedas, etc.). Ello no solamente exigía mucho tiempo, sino que también hacía mucho menos atractivo el procedimiento.

Para resolver esta problemática y aprovechando el auge que ya por entonces empezaba a tener el desarrollo de las computadoras personales, Werdenfeld y Simon desarrollaron un *software* denominado RESAMPLING STATS[8] (inicialmente denominado SIMPLE STATS). Este programa utiliza una serie de comandos que imitan la ejecución de acciones manuales tales como las anteriormente mencionadas, lo cual agiliza drásticamente los cómputos[9]. Por ejemplo, con la orden **URN 4#1 6#0 A**, se le comunica al programa que "construya" un vector llamado A, el cual contiene cuatro veces el número uno y seis veces el número cero; luego, si escribimos la secuencia **SHUFFLE A A**, se ejecuta la acción de reordenar los valores de A y colocar el resultado en el propio vector A; **TAKE A 1,20 B**, le ordena a la PC que seleccione los valores que ocupan en A las posiciones de la uno a la veinte y los coloque en un vector llamado B. Así sucesivamente, vamos construyendo un programa que automáticamente simula la situación en la cual estamos interesados. En el **Anexo No. 2** el lector podrá hallar un programa para resolver el problema planteado en la Sección 1.2.1.

1.3 Principales ventajas y desventajas del método de *remuestreo* en relación con los procedimientos convencionales.

Ventajas:

1. La principal ventaja es la ausencia de formulaciones teóricas a los efectos de ser aplicado, algo que en muchas ocasiones, en vez de motivar, desalienta a los estudiantes.

2. Es un procedimiento que puede vanagloriarse por su sencillez en relación con la complejidad propia de la alternativa clásica.

[8] Disponible una versión DEMO en http://www.resample.com
[9] Una *PPE* como la ilustrada se ejecuta en 0.1 segundo.

3. Resulta sumamente intuitivo.

4. Como el usuario tiene que “manipular” intensamente los datos, aprende mucho más de ellos y se motiva por descubrir él mismo la estructura que tienen.

5. Puede ser aplicable en situaciones donde los procedimientos formales son inviables.

6. Como sólo se tiene que simular el evento de interés, casi siempre se obtiene un resultado adecuado.

7. Siempre puede obtenerse el error estándar de cualquier estimador, algo impensable con la estadística frecuentista.

Desventajas:

1. En determinadas situaciones problémicas puede ser un método más laborioso que el correspondiente a los procedimientos convencionales.

2. Ocasionalmente resulta difícil simular el proceso físico en el cual estamos interesados.

3. Cuando se cumplen los supuestos que rigen la inferencia frecuentista, el *remuestreo* pudiera ser menos eficiente.

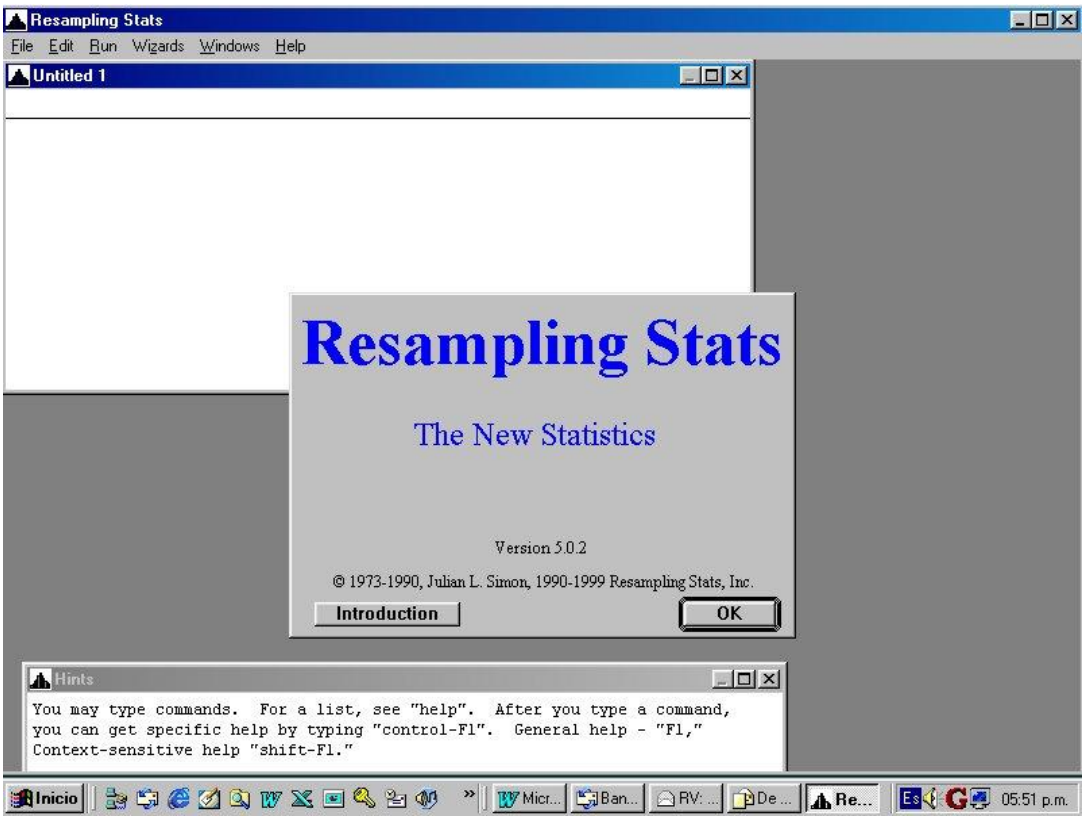
4. La probabilidad calculada es aproximada.

El software **Resampling Stats** (versión 5.0.2) no es el único programa disponible en el mercado que resulta útil para resolver problemas en el ambiente del *remuestreo; S y* S-*Plus*, por ejemplo, son paquetes estadísticos que tienen incluidos módulos de *remuestreo* (Becker 1988); además, estos procedimientos pueden ser programados con cualquier lenguaje de programación.

Sin embargo, a nuestro juicio, el recurso ideado por Werdenfeld y Simon es extremadamente sencillo y fácilmente comprensible (sin que ello implique pérdida de eficacia), razón por la cual creemos que posee excelentes cualidades para ser explotadas.

2.1 Arranque del programa

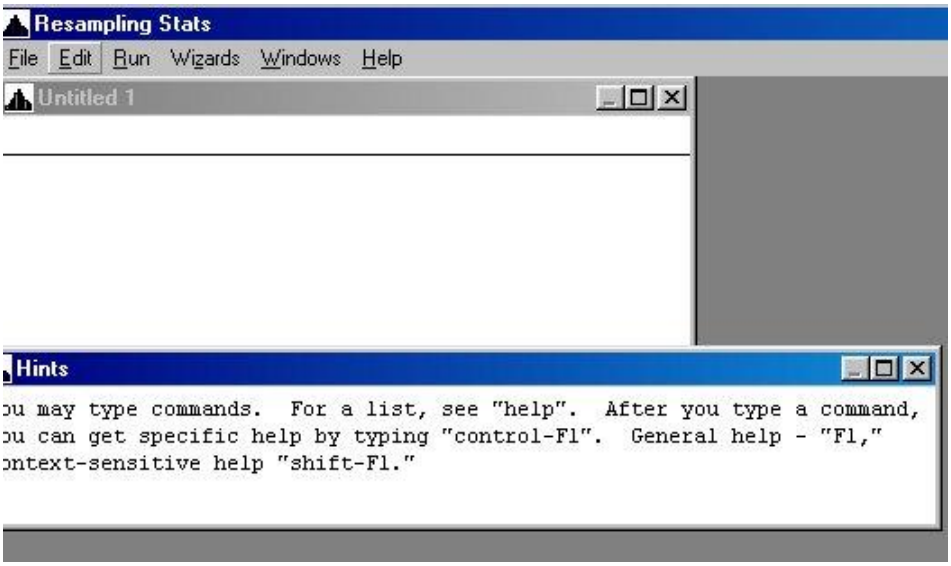
Una vez invocado el programa *Resampling,* aparece una ventana con el indicativo de que estamos dentro de él.



Hacemos click en la pestaña “ok” y ya estamos listos para empezar a trabajar con el software.

2.2 Entorno de trabajo

El entorno de trabajo de **Resampling Stats** es completamente gráfico y sigue las reglas típicas de las aplicaciones para Windows. Una vez abierto Resampling, aparecerá una ventana en la que podemos apreciar el aspecto general del entorno de trabajo.



En la parte superior aparece la barra de menús. Cada menú da acceso a distintas acciones posibles de ejecutar. Cada una de ellas guarda estrecha relación con las operaciones propias de un tema. Los menús que aparecen de izquierda a derecha son los siguientes.

File: Contiene las operaciones básicas que se pueden realizar con los archivos propios de Resampling (abrir, cerrar, guardar, etc). Estos ficheros tienen extensión .STA, y son tipo texto (véase una explicación *in extenso* en la Sección 2.6)

Edit: Proporciona todas las opciones disponibles para editar programas; copiar, cortar, pegar, seleccionarlo todo, son algunas de las acciones que podemos ejecutar.

Run: Brinda todas las funciones para ejecutar un programa que hayamos elaborado: chequear, ejecutar, detener la ejecución, limpiar una ventana de resultados.

Wizards: El uso de este menú simplifica la tarea de escribir un programa; con esta opción el usuario no tiene que escribir completamente el programa, sólo la esencia y Resampling se encarga de lo demás. Escribir la esencia, chequear el proceder, ayuda, repetir el proceso, se encuentran entre las tareas básicas bajo “Wizards”.

Windows: Permite configurar la forma de visualización de las distintas ventanas que aparecen en el entorno de trabajo.

Help: Permite el acceso directo a la ayuda.

2.3 Reglas para el uso de Resampling Stats

Extensión de un programa: la extensión de un programa no debe exceder las 32k. Para averiguar cuántas líneas de programación puede tener un programa, basta dividir 32000 entre el número promedio de caracteres por línea. Por ejemplo un programa que tenga 40 caracteres por línea puede contener 800 líneas de programación, un número ciertamente grande.

Nombres de programas: Para nombrar un programa, el usuario podrá utilizar cualquier cadena de letras o números; aunque admite más de 60 caracteres, se aconseja no exceder los 31.

Vector: Se puede decir que los vectores son la célula básica en la programación de un problema utilizando **Resampling Stats**. Se trata de un arreglo ordenado de números. La capacidad de un vector por “default” es de 1000 coordenadas, pero esta cifra puede variar utilizando el comando **MAXSIZE**, el cual asigna espacio adicional al vector.

Nombres de vectores: Los nombres de los vectores son definidos por los usuarios, no se admiten más de 8 caracteres y pueden comenzar con una letra, el signo $ o con el caracter _.

Variable: Llamamos así por lo general a un vector de una sola coordenada. Usualmente es el vector resultante de una operación previa, por ejemplo:

DATA (7 2 3) A	se crea el vector A cuyo contenido esta formado por los valores 7 2 3
SUM A C	suma los valores que integran el vector A y coloca el resultad en el vector C; en este caso, suele también llamarse *variable* C a dicho vector para subrayar que se trata de un vector que solo contiene un número

Palabras prohibidas: Algunas palabras no pueden ser usadas para nombrar vectores o variables.

Todos los comandos son palabras prohibidas, así como:

- **append**
- **between**
- **file**
- **memberof**
- **missing**

Test: Es definido como el resultado de una comparación entre un operador y un operando. Los operadores son <, >, <=, >=, <>, =, memberof y between. Por ejemplo, considere la variable Z en la cual se halla el número 10; el *test* Z>8, puede resultar verdadero para el operando IF, 0 para COUNT, etc.

2.4 ¿Cómo escribir un programa en Resampling Stats?

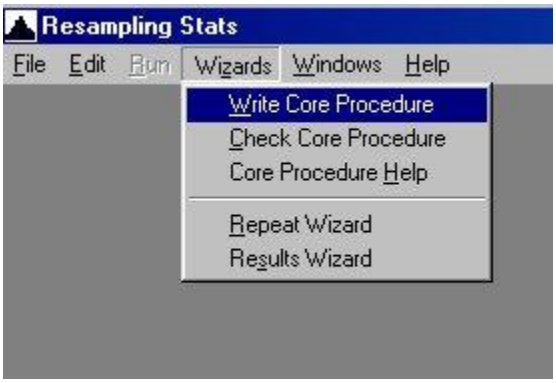
Consideremos el siguiente problema:

Problema 2.1: ¿Cuál es la probabilidad de obtener cinco o más veces el número 3, en el lanzamiento de 10 dados?

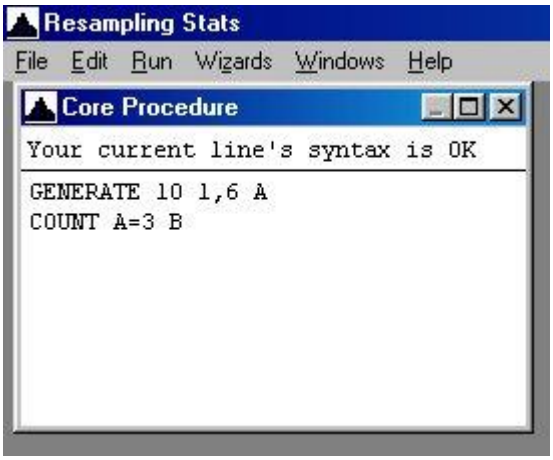
Existen 2 maneras de escribir un programa con Resampling Stats: utilizando Wizards o escribiendo completamente el programa.

Con el primer procedimiento, Resampling le ayuda en la elaboración del programa, veamos más fielmente como hacerlo:

1. Desde el menú Wizards, seleccione “Write Core Procedure”.



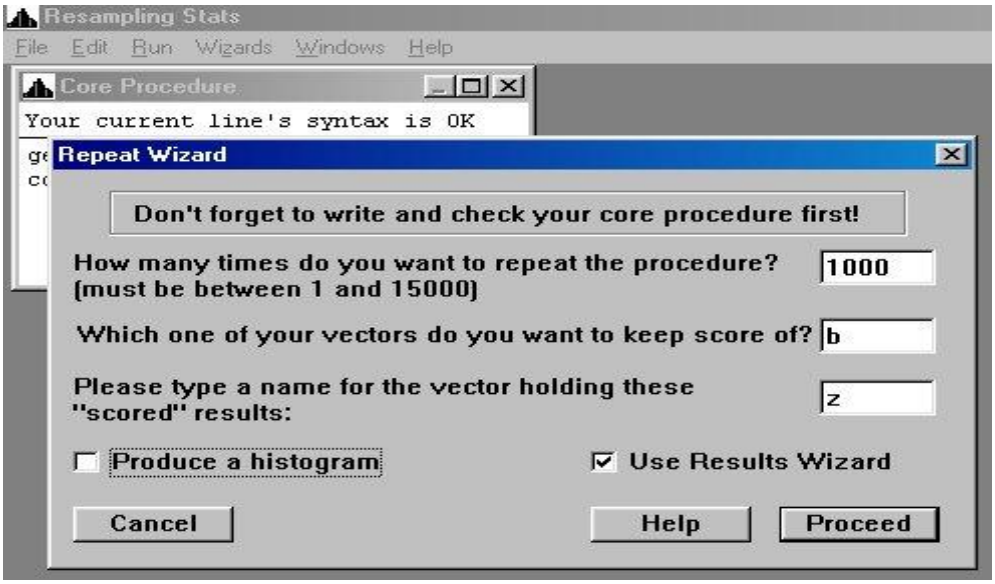
2. Inmediatamente después se abre una ventana llamada “Core Procedure”, haga click sobre ella para hacerla activa y escriba la esencia del problema.



La orden **GENERATE 10 1,6 A**, genera 10 números aleatorios entre 1 y 6 (lanzamiento de un dado 10 veces) y coloca el resultado en el vector **A;** luego **COUNT A=3 B** cuenta el número de veces que aparece el número 3 entre las coordenadas de vector **A** y ubica dicho resultado en la variable **B**.

3. Desde el menú “Wizards”, selecciona la opción “Check Core Procedure” para verificar que has escrito bien los comandos.

4. Desde el mismo menú, selecciona “Repeat Wizards”, y responde las preguntas que se muestran en la siguiente caja de diálogo.

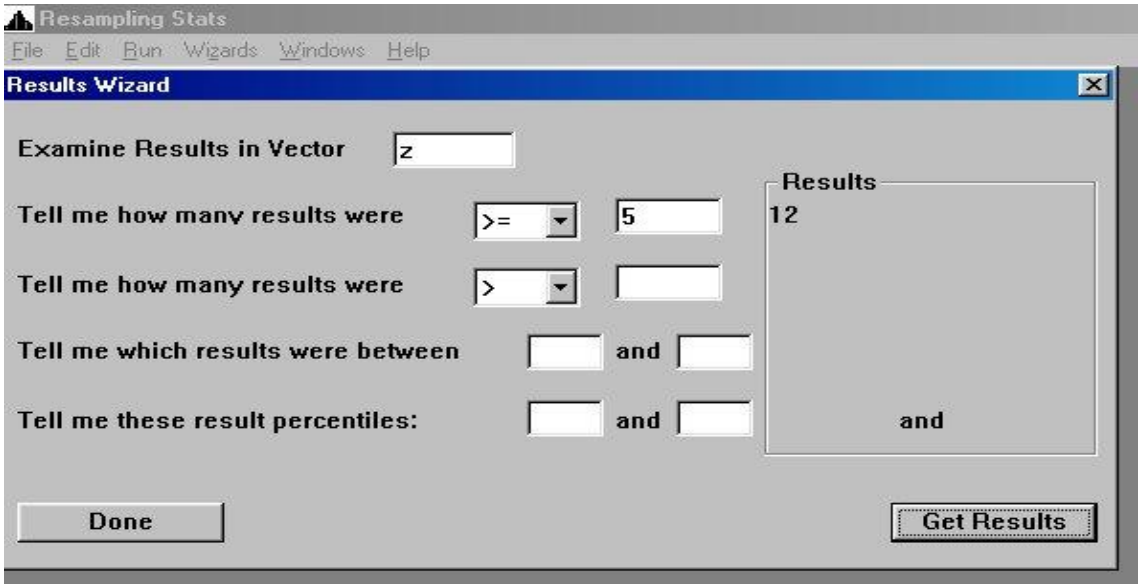


Lo primero que se pregunta es cuántas veces deseas que se repita el experimento, por *default* (defecto) el programa se ejecuta 1000 veces, pero uno puede escoger otro.

La segunda pregunta que debemos responder es, cuál de las variables deseamos registrar (“almacenar”), ya que cada vez que Resampling ejecuta una simulación, el contenido de las variables es borrado para prepararlas con vistas al nuevo experimento.

En la tercera ventanilla, se escribe el nombre de la variable resultado (**Z**), en la cual se colocará el valor de la variable a registrar; si se quiere obtener un histograma de Z, sólo hay que marcar la casilla “Produce a histogram”; haciendo click sobre “Proceed” se ejecuta la simulación.

5. Luego de haber concluido las 1000 simulaciones ordenadas, se abre un cuadro de diálogo llamado “Result Wisards”, y ya estamos listos para analizar los resultados.

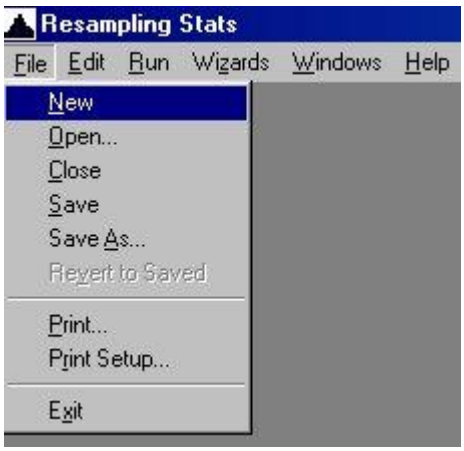


Conclusión: Como se puede apreciar en el cuadro anterior, en sólo 12 ocasiones se obtuvieron 5 o más resultados igual a 3 en 1000 experimentos, lo que representa una probabilidad estimada de 0.012. Desde la ventana precedente se pueden solicitar otros resultados.

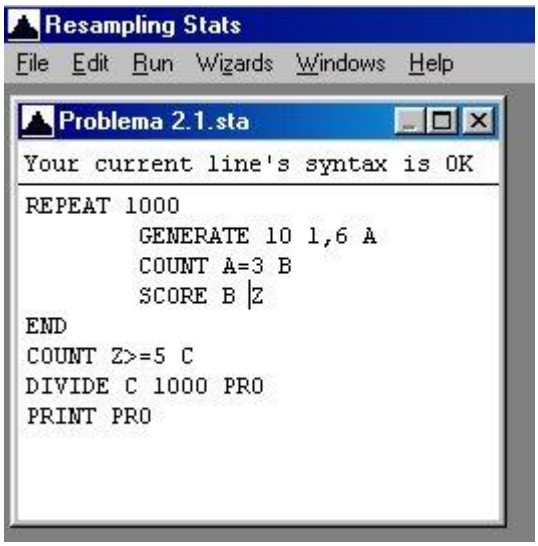
Escribir completamente el programa.

Este problema también se puede solucionar, escribiendo todos los comandos sin la ayuda de Resampling Stats. Cabe aconsejar a los principiantes que comiencen con el método anteriormente expuesto y, sólo cuando hayan adquirido cierta habilidad, entonces adoptar este procedimiento. Veamos cómo hacerlo:

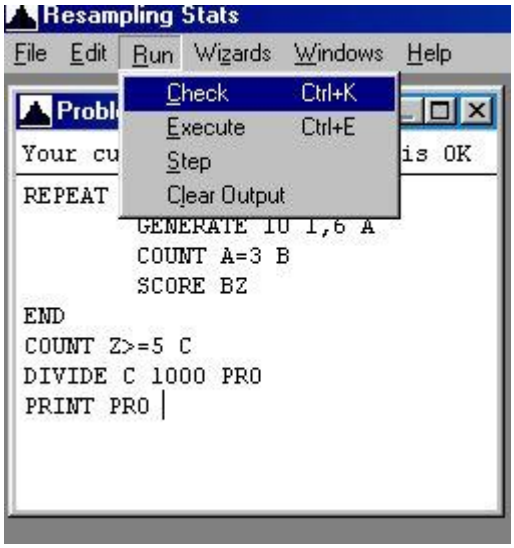
1. Desde el menú “File” seleccione “New”.



2. Sobre la ventana que se activará escriba todos los comandos.

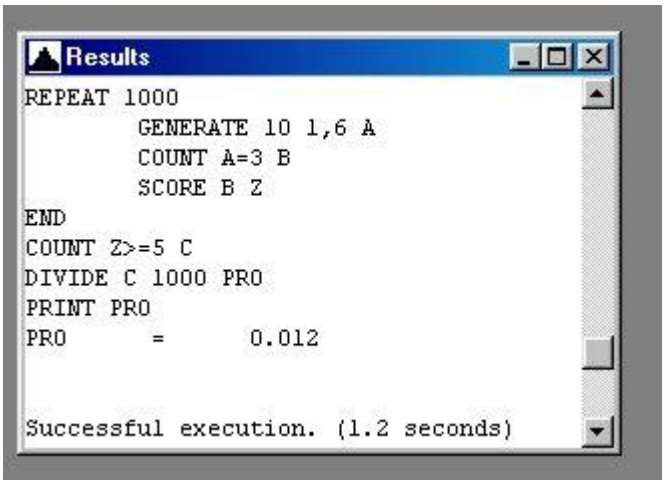


3. Sobre el menú “Run”, marcar “Check”, y Resampling Stats chequeará el programa escrito.



4. Si no hay ningún error en el programa, marcar “Execute” y se ejecutará el programa. Si lo hubiere, emergerá la información consignando en qué punto se halla y en qué consiste probablemente dicho error.

5. Luego se abrirá una ventana llamada “Resultados” que muestra el resultado.



2.5 Comandos básicos utilizados en Resampling Stats y sus sintaxis.

A continuación se irán explicando los diferentes comandos que pueden emplearse en el programa.

- **COPY, DATA y NUMBERS**

Preferimos unificar estos tres comandos para su explicación, ya que los tres tienen exactamente la misma función: colocan un conjunto de datos o una secuencia de números en un vector, además de ser utilizables para copiar el contenido de un vector ya existente en otro.

Sólo ejemplificaremos uno de ellos, pues todos tienen la misma sintaxis.

Sintaxis: COPY <vector de entrada> <vector resultado>

Ejemplos:

1. Para colocar datos en un vector

COPY (2 4 5) A.	genera el vector A, cuyo contenido es: 2 4 5.

2. Para entrar una secuencia de datos:

DATA 1,5 B	genera el vector B, en este caso formado por la secuencia de números del 1 al 5

3. Para copiar una variable:

NUMBERS A C	el vector C contiene ahora la información que está en A, en este caso estará formado por los valores 2 4 5

- **URN**

Coloca en un vector una cantidad determinada de números definida por el usuario. Con este comando se crea un vector que simula una urna clásica en la que hay bolas de diferentes colores (obviamente, de ahí su nombre).

Sintaxis: URN <cantidad># <valor><cantidad># <valor>…<vector resultado>

Ejemplo:

URN 10#1 20#5 60#6 A	"construye" una urna (vector A con 90 coordenadas) la cual contiene diez veces el número uno, veinte veces el cinco, y sesenta el seis

- **SAMPLE**

Selecciona aleatoriamente una muestra, con reemplazo, de los valores de un vector, y crea otro vector con el resultado; esta muestra puede ser tan grande como queramos.

Sintaxis: SAMPLE <tamaño de la muestra> <vector de entrada> <vector resultado>

Ejemplo:

DATA (1 2 3 4 5 6 7) A	crea el vector A formado por los números 1 2 3 4 5 6 7
SAMPLE 7 A B	selecciona una muestra de tamaño 7 con reemplazo del vector A y la sitúa en el vector B; el resultado puede ser, por ejemplo, B=1 1 2 2 3 4 5

Nótese que, debido a que la selección es con reemplazo, la muestra resultante puede contener un mismo elemento más de una vez.

- **TAKE**

Selecciona los elementos de un vector que ocupan una posición determinada. Este comando es usado frecuentemente para seleccionar muestras sin reemplazo de un vector.

Sintaxis: TAKE <vector de entrada> <posición a seleccionar> <vector resultado>

Ejemplo:

DATA (1 2 8 4 2 2 7 8 9) A	crea el vector A cuyo contenido es: 1 2 8 4 2 2 7 8 9
TAKE A 1,4 B	selecciona del vector A, los valores ubicados en las posiciones de la 1 a la 4, y coloca el resultado en el vector B; en este caso, el resultado será: 1 2 8 4

- **RANDOM y GENERATE**

Los comandos **RANDOM** y **GENERATE**, generan números aleatoriamente seleccionados dentro de un recorrido especificado, incluyendo los valores extremos de dicho recorrido.

Sintaxis: RANDOM <tamaño> <recorrido> < vector resultado>

Sintaxis: GENERATE <tamaño> <rango> <vector resultado>

Ejemplo:

RANDOM 20 1,9 A	genera 20 números aleatorios entre uno y nueve; coloca el resultado en el vector A; éste podría ser A = 7 2 4 9 7 6 2 1 7 6 8 1 7 4 3 8 1 5 2 2
GENERATE 100 20, 30 A	genera 100 números aleatorios entre 20 y 30; coloca el resultado en el vector A

- **PRINT**

Muestra en pantalla el valor de los vectores especificados por el usuario.

Sintaxis: PRINT <vector de entrada> <vector de entrada>…<vector de entrada>

Ejemplo:

GENERATE 4 1,10 A	genera 4 números aleatorios entre uno y diez y los coloca en el vector A
PRINT A	muestra en pantalla el contenido del vector A; por ejemplo, A podría ser: 1 3 5 9

- **ADD**

Suma los elementos de un vector con los correspondientes elementos de otro u otros vectores, y coloca el resultado en otro vector. Los diferentes vectores a sumar no necesariamente tienen que tener la misma longitud; si fueran diferentes las longitudes, **Resampling Stats** replicará el último valor de aquellos vectores cuya longitud sea menor hasta igualar la longitud del vector mayor.

Sintaxis: ADD <vector de entrada> <vector de entrada>…<vector resultado>.

Ejemplo:

DATA (2 4 6) A	genera el vector A cuyo contenido es: 2 4 6
DATA (3 5 7 1 2) B	genera el vector B cuyo contenido es: 3 5 7 1 2
ADD A B C	suma cada elemento del vector A, con los correspondientes elementos del vector B y coloca el resultado en el vector C
PRINT C	muestra en la pantalla el contenido del vector C; en este caso C quedará formado por: 5 9 13 7 8 (nótese que, como el vector A tiene dos coordenadas menos que B, a los efectos de la suma, se replica el último número de A, el 6, dos veces)

- **SUM**

Suma los elementos de un vector.

Sintaxis: SUM <vector de entrada> <variable resultado>

Ejemplo:

DATA (10 12.1 14.7) A	genera el vector A con los números 10 12.1 14.7
SUM A B	suma los valores del vector A y coloca el resultado en la variable B
PRINT B	muestra el valor de la variable B; en este caso vale 36,8

Nótese la diferencia con ADD

- **SUBTRACT**

Resta los elementos de un vector de los correspondientes elementos de otro vector (u otros vectores). Opera de forma similar al comando ADD: si los vectores no son de igual amplitud, entonces Resampling replica el último número de aquellos vectores cuya amplitud sea menor, tantas veces como sea necesario para igualar la dimensión del vector de mayor longitud.

Sintaxis: SUBTRACT <vector 1> <vector 2>… <vector resultado>

Ejemplo:

DATA (10 12 14) A	genera el vector A cuyo contenido es: 10 12 14
DATA (8 7) B	genera el vector B cuyo contenido es: 8 7
SUBTRACT A B C	calcula la diferencia entre cada elemento del vector A y su correspondiente valor en el vector B; coloca el resultado en el vector C; en este caso el resultado resulta ser: 2 5 7

- **MULTIPLY**

Multiplica dos o más vectores, Si los vectores no son de igual longitud, entonces **Resampling Stats** replica el último número de aquellos vectores cuya longitud sea menor, tantas veces como sea necesario para igualar la dimensión del vector de mayor longitud.

Sintaxis: MULTIPLY <vector de entrada> <vector de entrada>…<vector resultado>

Ejemplo:

DATA (2 3 9) A	genera el vector de datos A cuyo contenido es: 2 3 9
DATA (4 5) B	genera el vector de datos B cuyo contenido es: 4 5
MULTIPLY A B C	multiplica los elementos de los vectores A y B y coloca el resultado en el vector C; como en este caso, el vector B tiene una longitud menor que A, Resampling replica el número 5 del vector B una vez y se obtiene el resultado: 8 15 45.

- **DIVIDE**

Divide los correspondientes valores contenidos entre dos o más vectores. Si los vectores involucrados en la operación no tienen la misma longitud, **Resampling Stats** automáticamente clona el último valor de aquellos vectores de menor longitud tantas veces como sea necesario con el objetivo de igualarlos.

Sintaxis: DIVIDE <vector de entrada> <vector de entrada>… <vector resultado>

Ejemplo:

DATA (8 12 16) A	genera el vector A cuyo contenido es: 8 12 16
DATA (4 6 8) B	genera el vector de datos B en este caso su contenido es: 4 6 8
DATA (2 2) C	genera el vector de datos C igual a 2 2 (éste vector no tiene la misma longitud que los anteriores, Resampling copiará el último 2 una ves)
DIVIDE A B C D	produce el vector D formado por los valores: 1 1 1, resultante de dividir 8 entre 4, y ese resultado entre 2 (para la primera coordenada); 12 entre 6 y el cociente entre 2 y, nuevamente, 16 entre 8 y este cociente entre el 2 que agregó a C

- **MEAN**

Calcula la media de un vector, los valores faltantes no son incluidos en el análisis.

Sintaxis: MEAN <vector de entrada> <variable resultado>

Ejemplo:

DATA (2 3 4 5 6 7 8) A	genera el vector de datos A igual a: 2 3 4 5 6 7 8
MEAN A B	calcula la media del vector A y coloca el resultado en la variable B (en este caso B es igual a 5)

- **COUNT**

Cuenta el número de veces que determinado número (o los elementos de cierto recorrido de números) se encuentra(n) en un vector.

Sintaxis: COUNT <vector de entrada> <test> <variable resultado>

Ejemplos:

COUNT A=5 B	cuenta el número de veces que aparece el cinco en el vector A y coloca el resultado en la variable B
COUNT A>=5 B	cuenta los valores mayores o iguales que cinco que hay en el vector A y coloca el resultado en la variable B

COUNT A<5 B cuenta los valores menores de 5 que hay en el vector A y coloca el resultado en la variable B

- **CONCAT**

Combina los elementos de un vector con los elementos de otro u otros vectores, colocándolos en un vector resultado.

Sintaxis: CONCAT <vector 1> <vector 2>…<vector resultado>

Ejemplo:

DATA (2 4 6) A genera el vector de datos A igual a: 2 4 6
DATA (3 5 7) B genera el vector de datos B igual a: 3 5 7
CONCAT A B C combina los elementos de los vectores A y B; coloca el resultado en C; en este ejemplo dicho vector será: 2 4 6 3 5 7

- **SHUFFLE**

Reordena aleatoriamente los elementos contenidos en un vector.

Sintaxis: SHUFFLE <vector de entrada> <vector resultado>

Ejemplo:

NUMBERS (2 3 4 5 6 7 8) A coloca los números 2 3 4 5 6 7 8 en el vector A
SHUFFLE A B reordena aleatoriamente los valores del vector A y los coloca en el vector B (en este caso, este pasa a ser: 2 7 5 3 8 4 6)

- **MULTIPLES**

Este comando encuentra cuántas veces un mismo número aparece repetido dos, tres, o más veces en un vector especificado.

Sintaxis: MULTIPLES <vector de entrada> <test> <variable resultado>

El test involucra la utilización de los operadores vistos anteriormente.

Ejemplo:

DATA (2 4 7 7 5 2 2) A genera el vector de datos A igual a: 2 4 7 7 5 2 2

MULTIPLES A>1 B cuenta cuántos números aparecen repetidos más de una ves en el vector de entrada A, y coloca el resultado en la variable B, en este caso el resultado es 2 ya que 2 y 7 aparecen más de una vez

- **RUNS**

Comando utilizado, cuando se quiere determinar el número de secuencias con una amplitud especificada de un número en un vector.

Sintaxis: RUNS <vector de entrada> <test> <variable resultado>

Ejemplo:

Suponga que el vector A esta formado por los siguientes números: { 1 1 2 3 2 4 4 5 5 5 6 2}, si nosotros deseamos conocer cuantas secuencias de amplitud dos hay en este vector utilizamos el comando RUNS:

RUNS A=2 B cuenta cuántas secuencias de amplitud dos hay en el vector A, y coloca el resultado en la variable B. En este caso es igual a 2 (las del uno y las del cuatro)

Nótese que este comando es diferente a MULTIPLES.

- **SORT**

Ordena los elementos de un vector y coloca el resultado en un vector resultado.

Sintaxis: SORT {descendente} <vector de entrada> <vector resultado>

Ejemplo:

GENERATE 20 1,50 A genera veinte números aleatorios del uno al cincuenta y los coloca en el vector A

SORT A B ordena los valores del vector A en forma ascendente y los coloca en el vector B

SORT descending A C ordena los valores del vector A en forma descendente y los coloca en el vector C

- **DEDUP**

Elimina aquellos valores que se encuentran repetidos en un vector manteniendo el primero que aparece; coloca todos los valores diferentes en otro vector.

Sintaxis: DEDUP <vector de entrada> <vector resultado>

Ejemplo:

DATA (5 2 3 4 2 5) A	genera el vector de datos A igual a: 5 2 3 4 2 5
DEDUP A B	elimina los valores que se encuentran en las posiciones 5 y 6 dentro del vector A ya que estos se encuentran repetidos; coloca el resultados en el vector B cuyo contenido en este caso es 5 2 3 4

- **WEED**

El comando WEED elimina los valores de un vector, que cumplen con cierta condición (test) definida por el usuario.

Sintaxis: WEED <vector de entrada> <test> <vector resultado>

Ejemplo:

GENERATE 20 1,8 A	genera 20 números aleatorios entre 1 y 8; coloca el resultado en el vector A
WEED A>=5 B	elimina los valores del vector A que cumplen con la condición especificada (>=5); coloca el resultado en el vector B; En este caso dicho vector estará solamente constituido por valores entre 1 y 4.

- **END**

Este comando le informa a **Resampling Stats** que ha alcanzado la última línea de una expresión lógica iniciada con IF o de un bloque de comandos **REPEAT** o **WHILE**.

Sintaxis: END

Ejemplo: (lo explicaremos con el siguiente comando, para su mejor comprensión)

- **REPEAT**

Ejecuta repetidamente un conjunto de órdenes definidas entre el comando **REPEAT** y el comando **END**; a esta operación se le denomina "***looping***". Es importante percatarse que cada ves que se utilice este comando se debe concluir con **END**.

Sintaxis: REPEAT<número de entrada>

Ejemplo:

REPEAT 1000	repite 1000 veces la secuencia de comandos entre **RANDOM** y **SCORE**
RANDOM 1 1,2 A	genera un número aleatorio entre uno y dos y lo coloca en el vector A
SCORE A Z	agrega el resultado del vector A en el vector Z
END	finaliza la secuencia y retorna al comando REPEAT hasta que se han completado los mil ciclos

- **IF**

Es un comando útil para crear expresiones lógicas; es decir, cuando figura este comando, **Resampling Stats** ejecuta sólo aquellas acciones que cumplen con cierta condición especificada.

Sintaxis: IF <número de entrada> <test>

Ejemplo: (lo explicaremos con el siguiente comando en el contexto de un problema concreto)

- **SCORE**

Agrega o "almacena" el resultado de un vector en uno o más vectores. Cada vez que el programa repite un ciclo definido por el comando **REPEAT** se borra el contenido del (los) vector(s) usado(s); con el comando **SCORE** podemos "almacenar" el resultado de cada experimento para ser usado en la fase de análisis.

Sintaxis: SCORE <vector de entrada> <variable resultado>…

Ejemplo:

En un lote de 12 vacunas hay 4 vencidas. Si se escogen 2 vacunas sin reemplazo del lote. ¿Cuál es la probabilidad de que las dos estén vencidas?

URN 4#1 8#0 A	coloca en una urna (vector A) 4 unos y 8 ceros
REPEAT 1000	repite el experimento 1000 veces
SHUFFLE A B	reordena aleatoriamente los doce números contenidos en el vector A y coloca el resultado en el vector B
TAKE B 1,2 C	selecciona los números que ocupan los lugares 1 y 2 dentro del vector B (equivalente a tomar de la urna 2 números sin reemplazo) y coloca el resultado en C
SUM C T	suma el resultado del vector C y coloca el resultado en la variable T
IF T = 2	si el valor de la variable T es igual a 2 entonces
SCORE 1 Z	agrega un 1 en el vector Z
END	finaliza la condición
END	finaliza el ciclo y regresa a la línea de programación definida por el comando REPEAT hasta completar todos los ciclos (1000)
SIZE Z P	computa el tamaño del vector Z y coloca el resultado en la variable P (cuenta el número de unos que hay en Z)
DIVIDE P 1000 PRO	divide el contenido de la variable P entre 1000 (estima la probabilidad deseada) y coloca el resultado en la variable PRO
PRINT PRO	muestra el resultado en pantalla de la variable PRO; en este caso es igual a 0.09

- **PERCENTILE**

Tabula los valores de un vector ordenándolos de menor a mayor, y luego calcula el valor correspondiente al percentil (o percentiles) especificado(s).

Sintaxis: PERCENTILE <vector de entrada> <percentiles> < variable resultado>

Ejemplo:

PERCENTILE A (2.5 97.5) B calcula los percentiles 2.5 y 97.5 de la distribución de valores que se encuentran en el vector B

- **HISTOGRAM**

Dibuja un histograma a partir de una distribución de frecuencias contenidas en un vector.

Sintaxis: HISTOGRAM {porcentaje} {escala del eje Y<número de entrada> <vector de entrada> <vector de entrada>…

Ejemplos:

1. Sin ninguna opción especificada en la sintaxis:

HISTOGRAM Z Resampling Stats escala automáticamente los valores de los ejes cartesianos del histograma; el eje Y representará frecuencias absolutas (Gráfico No. 1).

2. Con la opción PERCENT:

HISTOGRAM PERCENT Z el eje Y en este caso representa porcentajes (Gráfico No. 2

3. Con la opción YSCALE

HISTOGRAM YSCALE 100 Z el máximo valor que asume el eje Y es 100 (Gráfico No.3)

Grafico No. 1: Forma de escalar el Histograma cuando no se especifica ninguna opción en la sintaxis.

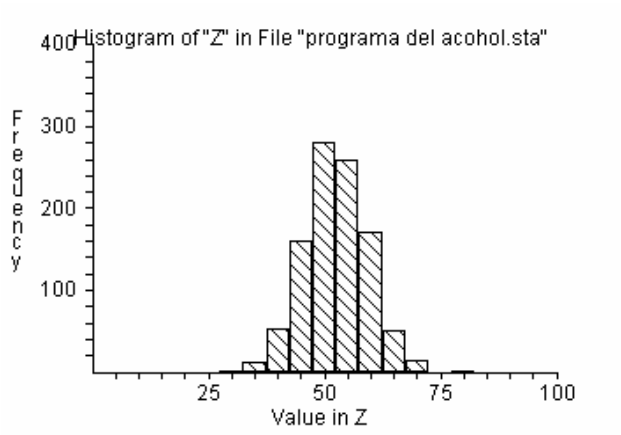


Gráfico No. 2: Forma en que queda escalado el eje Y al utilizar en la sintaxis el comando PERCENT

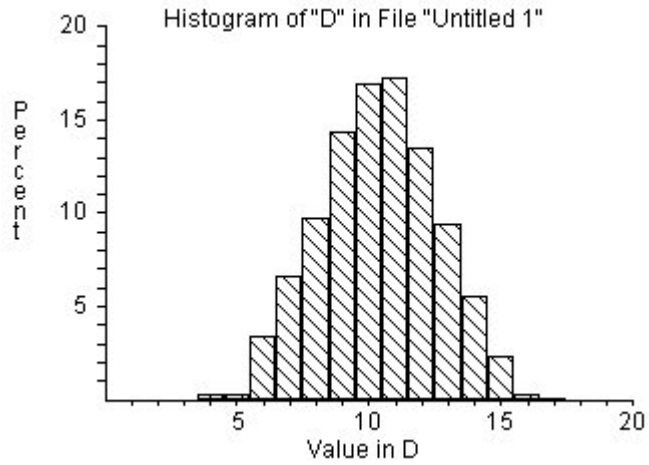
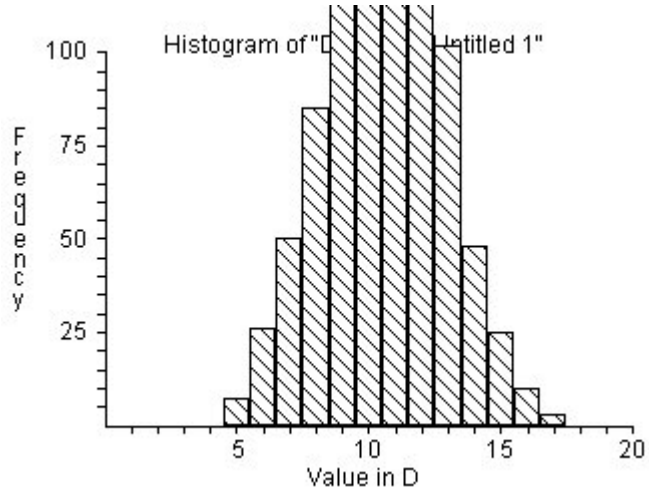


Gráfico No.3: Forma en que queda escalado el eje Y al utilizar en la sintaxis la orden YSCALE 100.



2.6 Comandos relacionados con funciones matemáticas y estadísticas.

2.6.1 Matemáticas:

- **ABS**

Convierte los valores observados en un vector en sus respectivos valores absolutos y los coloca en otro vector.

Sintaxis: ABS <vector de entrada><vector resultado>

Ejemplo:

DATA (-23 45 -34.5 67.8 –54) A	genera un vector de datos A igual a: -23 45 -34.5 67.8 –54
ABS A B	transforma los valores observados en el vector A en sus correspondientes valores absolutos y coloca el resultados en el vector B; en este ejemplo éste pasa a ser 23, 45, 34.5, 67.8, 54

- **POWER**

Eleva los valores de un vector a la potencia indicada por los elementos de un segundo vector.

Sintaxis: POWER <vector de entrada> <vector de entrada><variable resultado>

Ejemplo:

DATA (5 7 9) A	genera el vector de datos A igual a: 5 7 9
DATA (2 3 4) B	genera el vector de datos B igual a: 2 3 4
POWER A B C	eleva el 5 al cuadrado, el 7 a la tres y el 9 a la cuatro, así se obtiene el vector C formado por los valores 25, 343 y 6561

- **PRODUCT**

Multiplica entre sí los elementos de un vector.

Sintaxis: PRODUCT <vector de entrada> <variable resultado>

Ejemplo:

DATA (2 10 67.4) A	genera el vector de datos A igual a: 2 10 67.4
PRODUCT A B	multiplica entre sí los elementos del vector A y coloca el resultado en la variable B; en este caso pasa a ser 1348

- **LOG**

Calcula el logaritmo natural de cada elemento de un vector.

Sintaxis: LOG <vector de entrada> <variable resultado>

- **MAX y MIN**

Identifican el máximo y el mínimo valor respectivamente, entre los elementos de un vector.

Sintaxis: MAX <vector de entrada> <variable resultado>

MIN <vector de entrada> <variable resultado>

- **RANKS**

Asigna un recorrido a los valores de un vector y coloca dichos recorrido en otro vector.

Sintaxis: RANKS {DESCENDING} <vector de entrada> <vector resultado

Ejemplo:

DATA (1 2 2 3 9 8 0) A	genera el vector de datos A igual a: 1 2 2 3 9 8 0
RANKS A B	computa el recorrido en orden ascendente que le corresponden a cada uno de los valores del vector A, y coloca el resultado en el vector B
PRINT B	muestra el resultado de B; en este caso pasa a ser 2 3.5 3.5 5 7 6 1

- **ROUND**

Redondea los valores de un vector transformándolos en números enteros

Sintaxis: ROUND <vector de entrada> <variable resultado>

Ejemplo:

DATA (2.3 5.5 23.09 45.6) A	genera el vector de datos A igual a: 2.3 5.5 23.09 45.6
ROUND A B	redondea los valores del vector A y coloca el resultado en el vector B, en este caso éste pasa a ser: 2 6 23 46

- **SQRT**

Calcula la raíz cuadrada de cada elemento de un vector y coloca el resultado en una variable definida por el usuario.

Sintaxis: SQRT<vector de entrada> <variable resultado>

Ejemplo:

DATA (16 64 81) J	genera el vector de datos J formado por los valores: 16 64 81
SQRT J Y	calcula la raíz cuadrada de cada uno de los valores contenidos en el vector J y coloca el resultado en el vector Y, que en este caso pasa a ser: 4 8 9

- **SQUARE**

Eleva al cuadrado cada elemento de un vector colocando el resultado en una variable definida por el usuario.

Sintaxis: SQUARE <vector de entrada> <variable resultado>

Ejemplo:

DATA (3 56 23) A	genera el vector de datos A formado por los valores 3, 56 y 23
SQUARE A H	eleva al cuadrado cada uno de los valores contenidos en el vector A y coloca el resultado en el vector H, en este caso pasa a ser: 9, 3136, 529

2.6.2 Comandos de uso en estadística.

Comandos útiles para generar distribuciones teóricas:

- **EXPONENTIAL**
- **LOGNORMAL**
- **NORMAL**
- **PARETO**
- **POISSON**
- **WEIBULL**
- **UNIFORM**

Todos estos comandos tienen la misma estructura; es por ello que sólo expondremos uno a manera de ejemplo.

Sintaxis: NORMAL <tamaño> <media> <desviación estándar> <variable resultado>

Ejemplo:

NORMAL 100 25 2 A	genera 100 valores de una distribución Normal con media 25 y desviación estándar 2; coloca el resultado en el vector A

Medidas de tendencia central:

- **MEDIAN** (mediana)
- **MEAN** (media)
- **MODE** (moda)

Los tres comandos tienen la misma sintaxis:

Sintaxis: COMANDO <ventor de entrada> <variable resultado>

Calculan la mediana, la media y la moda respectivamente de los elementos de un vector.

Ejemplo:

DATA (23 45 65 76 89 43) M	genera el vector de datos M cuyo contenido está formado por los números: 23 45 65 76 89 43
MEAN M X	calcula la media de los valores contenidos en el vector M y coloca el resultado en la variable X, en éste ejemplo dicha variable pasa a ser: 56.833

Medidas de dispersión:

- **STDEV y VARIANCE**

Calculan la desviación estándar y la varianza respectivamente de un vector. Por tener la misma sintaxis explicaremos solo uno de ellos.

Sintaxis: STDEV {divn} <vector de entrada> <variable resultado>

El programa por defecto calcula la desviación estándar utilizando como denominador n-1, aunque se puede calcular también para n.

Ejemplo:

DATA (23 34 54 67.8 45.6 32.5) A	genera el vector de datos A igual a: 23 34 54 67.8 45.6 32.5
STDEV divn A B	calcula la desviación estándar del vector A utilizando como denominador n, ya que en la sintaxis se utilizó la orden divn (de no haberse utilizado, entonces el denominador hubiera sido n-1); coloca el resultado en B
PRINT B	muestra el resultado en pantalla de la variable B; en el ejemplo ésta equivale a 14.91

- **SUMABSDEV**

Calcula la suma de las diferencias absolutas entre dos vectores.

Sintaxis: SUMABSDEV <vector de entrada> <vector de entrada> <variable resultado>

Ejemplo:

DATA (23 45 5 6 9) A	genera el vector de entrada A cuyo contenido es: 23 45 65 76 89
DATA (25 40 7 3 14) B	genera el vector de entrada B cuyo contenido es: 87 908 65 23 14
SUMABSDEV A B C	calcula la suma de las diferencias absolutas entre las coordenadas de los vectores A y B; coloca el resultado en la variable C. En este caso éste pasa a ser: 17.

Si los vectores no fueran de la misma longitud Resampling Stats replica el número que ocupa la última posición del vector de menor longitud.

- **SUMSQRDEV**

Calcula la suma de las diferencias cuadradas entre dos vectores.

Sintaxis: SUMSQRDEV <vector de entrada><vector de entrada><variable resultado>

Ejemplo:

DATA (23 45 65 76 89) A	genera el vector de entrada A cuyo contenido es: 23 45 65 76 89
DATA (87 908 65 23 14) B	genera el vector de entrada B cuyo contenido es: 87 908 65 23 14
SUMSQRDEV A B C	Calcula la suma de las diferencias cuadradas entre las coordenadas de los vectores A y B; coloca el resultado en la variable C. En este caso éste pasa a ser: 7.573e+005

- **CORR**

Calcula el coeficiente de correlación entre dos vectores.

Sintaxis: CORR <vector de entrada><vector de entrada><variable resultado>

DATA (23 12 45 65 23) A	genera el vector de datos A igual a: 23 12 45 65 23
DATA (10 5 20 60 21) B	genera el vector de datos B igual a: 10 5 20 60 21
CORR A B C	calcula el coeficiente de correlación entre los vectores A y B; coloca el resultado en la variable C (en este caso pasa a ser 0.9102)

Algunos comandos adicionales.

- **CLEAR**

Borra el contenido de un vector.

Sintaxis: CLEAR <vector de entrada>

- **MAXSIZE**

Recuérdese que la capacidad máxima de un vector por defecto es 1000; si utilizamos el comando **MAXSIZE,** podemos aumentar esta capacidad hasta 1 000 000 valores.

Sintaxis: MAXSIZE {default <tamaño>} {<variable><tamaño>…<variable> <tamaño>}

Ejemplo:

MAXSIZE A 3300 B 4000	aumenta la capacidad del vector A hasta 3300 valores, de forma que el vector A ahora puede contener 3300 números y el B 4000.
GENERATE 3300 1,80 A	genera 3300 números aleatorios entre uno y ochenta y los coloca en el vector A
COPY A B	copia las coordenadas del vector A en el vector B

- **RECODE**

Recodifica los valores contenidos en un vector que cumplan con cierta condición; estos valores asumirán el especificado por el usuario.

Sintaxis: RECODE <vector de entrada> <test> <número de entrada> <variable resultado>

Ejemplo:

GENERATE 100 1,80 A	genera 100 números entre uno y ochenta y los coloca en el vector A
RECODE A>=50 0 B	los valores de A mayores o iguales que 50 se recodifican, a partir de este momento serán ceros, el nuevo vector es colocado en B

- **SIZE**

Determina el tamaño de un vector.

Sintaxis: SIZE <vector de entrada> <variable resultado>

Ejemplo:

DATA (23 45 65 42 12 34 56 78 90) A	genera el vector de datos A igual a: 23 45 65 42 12 34 56 78 90
SIZE A B	calcula las dimensiones del vector A y coloca el resultado en la variable B; en este caso equivale a 9

- **WHILE**

Con este comando se pueden repetir un conjunto de órdenes mientras se cumpla con una condición especificada por el usuario dentro de un *loop*.

Sintaxis: WHILE <número> <test>

.

.

END

Ejemplo:

DATA (2 3 4 5 6 7 8) A	genera el vector de datos A igual a: 2 3 4 5 6 7 8
COPY (1) B	genera el vector de datos B cuyo contenido es un número 1
WHILE B<=7	mientras el valor del vector B sea menor o igual a 7
TAKE A B C	selecciona dentro del vector A el número que ocupa la posición contenida en la variable B
SUM 1 B	adiciona 1 al valor de la variable B; en este caso, ahora la variable B contiene un 2, y así en cada ciclo este vector irá aumentando en una unidad él valor de su contenido, de forma tal que este proceso se repetirá 7 veces, ya que la orden WHILE está definida de forma tal que el proceso se repite a menos que la variable B haya alcanzado el valor 8
END	termina la condicional

Esta secuencia de comandos selecciona de uno en uno cada elemento del vector A.

2.7 Exportar datos.

El usuario de Resampling puede exportar archivos tipo texto utilizando para ello el comando **WRITE.**

Sintaxis: WRITE {FILE "nombre del archivo"} {MISSING <{número>} {APPEND} <vector de entrada> {<formato>}…

Ejemplo:

GENERATE 100 1,10 A	genera cien números aleatorios entre uno y diez y coloca el resultado en el vector A

WRITE FILE " Ejemplo" A	crea un archivo, llamado "Ejemplo" a partir del vector A.

Como se aprecia en la sintaxis, también se pueden guardar en el archivo varios vectores. Por ejemplo, WRITE FILE "Ejemplo" A B C D E, guardaría en el archivo "Ejemplo" los vectores A, B, C, D y E de forma tal que cada uno de ellos sería una columna de números, como se muestra a continuación.

48	54	74	10	25
21	20	23	26	41
74	54	68	85	74

También se puede especificar el estilo y la precisión con que el usuario desee que se guarde la información, ejemplo:

WRITE FILE "Ejemplo" A% 2.2f

El archivo "Ejemplo" se guardará con un formato tal que cada valor tendrá dos decimales fijos (f).

La opción APPEND se utiliza cuando queremos añadir un vector a un archivo ya existente. Por ejemplo:

DATA (1 2 3 4 5 6) A	genera el vector de datos A cuyo contenido en este caso es: 1 2 3 4 5 6
WRITE FILE "Ejemplo" APPEND A	añade al archivo "Ejemplo" (ya existente) el vector A al final del último número.

Si no utilizamos esta opción, Resampling borrará automáticamente el contenido del archivo "Ejemplo" para colocar en su lugar el nuevo vector.

2.8 Importar datos:

Cuando manejamos información con el software Resampling Stats, ésta puede ser entrada directamente con los comandas **DATA**, **COPY y NUMBER** (ver sección 2.5) o bien introducida a partir de un archivo ya existente con extensión tipo texto (*.txt) o con extensión .STA (los archivos generados por **Resampling Stats**), utilizando para ello el comando **READ**.

Sintaxis: READ {FILE "nombre del archivo"} {valores missing <número>} [<variable resultado> <formato>]

Ejemplo:

Supongamos que el archivo “ic-alcohol”.txt contiene valores relacionados con el consumo diario de alcohol expresado en (g/l) de 142 pacientes con psoriasis (esta información está contenida en forma de columna numérica) y 157 controles (segunda columna numérica). Deseamos emplear ésta información para calcular la ingestión media de alcohol tanto en los pacientes con psoriasis como en los controles.

READ FILE “ic-alcohol” B C	toma los valores de la primera columna del archivo especificado, y los coloca en el vector B; los valores de la segunda columna los coloca en C
MEAN B x1	calcula la media del vector B y coloca el resultado en la variable x1
MEAN C x2	calcula la media del vector C y coloca el resultado en la variable x2

Si el archivo a leer, contiene alguna variable que exceda la máxima capacidad que admite por *default* el programa (1000), necesariamente hay que utilizar el comando **MAXSIZE**. Por ejemplo, si el archivo de datos “Max” contiene información de, digamos, 9827 unidades de observación entonces para poder leer dicha información podríamos utilizar la secuencia de comandos siguientes:

MAXSIZE B 9827	aumenta la capacidad del vector B hasta 9827 coordenadas.
READ FILE “Max” B	lee el archivo “Max” y coloca los valores en el vector B

Con la opción Format, se le puede comunicar a **Resampling Stats** que lea solo los valores alternos dentro de un archivo, por ejemplo, supongamos que tenemos un archivo llamado “Ejemplo” con la siguiente información:

22
58
76
45
52
87
26

Si escribimos la siguiente orden, entonces solo se leerán los valores alternos:

READ FILE "Ejemplo" A FORMAT el vector A sólo contendrá los valores 22, 76, 52 y 26

Podemos conseguir saltar una columna de datos que no se desee utilizar asignando la misma a una variable ficticia (dummy).

Ejemplo:
El archivo "variables_dummy" contiene información de los resultados académicos obtenidos, durante el primer semestre, por 10 alumnos. Se desea calcular la nota promedio alcanzada en los siguientes módulos:

- Ensayos Clínicos (primera columna).
- Factores Pronósticos (tercera columna).
- Investigación Epidemiológica (cuarta columna).

Contenido del archivo "variables_dummy".

98	100	90	95
92	85	97	90
90	93	87	95
94	90	97	79
78	87	92	97
91	98	89	96
80	90	95	92
70	85	82	90
89	98	90	75
84	87	90	95

READ FILE "variables_dummy" C dummy P E

	con esta orden, Resampling lee la primera columna y la asigna al vector C, la segunda columna es ignorada, la tercera y la cuarta son asignadas a los vectores P y E respectivamente
MEAN C X1	calcula la media del vector C y coloca el resultado en la variable X1
MEAN P X2	calcula la media del vector P y coloca el resultado en la variable X2
MEAN E X3	calcula la media del vector E y coloca el resultado en la variable X3
PRINT X1 X2 X3	muestra el resultado en pantalla de las variables indicadas. En el ejemplo: X1=86.6, X2=90.9 y X3=90.4

2. 9 Principales ventajas y desventajas del software Resampling Stats.

Ventajas:

- Software de fácil manejo
- Rápido de aprender
- Los números pseudoaleatorios generados son de extrema calidad.
- La ejecución del programa es rápida y eficiente.
- La ayuda siempre está visible y lista para ser invocada.

Desventajas:

- El hecho de trabajar sólo con ficheros tipo texto, es una verdadera privación para el usuario (sería bueno que pudiera leer, por ejemplo, bases de datos en Excel).
- Los programas se gestan sobre la base de vectores.
- Puede ser realmente engorroso programar determinados problemas.

3.1 Cálculo de probabilidades con técnicas de simulación.

El cálculo de probabilidades se ha convertido en un hecho casi rutinario para la gran mayoría de las personas. Constantemente, aunque casi siempre de forma inconsciente, estimamos la probabilidad subjetiva de que ocurra un suceso. Por ejemplo: antes de salir de casa miramos al cielo y estimamos la probabilidad de que llueva; un médico en el ejercicio de su profesión estima la probabilidad de que un paciente tenga la enfermedad x; un gerente estima la probabilidad de que determinada estrategia comercial tenga éxito, etc.

Para calcular más formalmente (o de forma objetiva) la probabilidad de ocurrencia de un determinado suceso muchas veces podemos seguir una de tres estrategias básicas, a saber:

1. Utilizar la teoría formal de las probabilidades.
2. Realizar la experiencia física (el experimento en el cual estamos interesados) un gran número de veces y calcular la frecuencia relativa con que se produce el evento de interés.
3. Simular la experiencia física (por lo general, usando computadoras).

La primera de ellas, implica un conocimiento teórico de las reglas de la teoría de probabilidades y su mayor dificultad radica en que, en determinadas situaciones resulta verdaderamente complejo hallar la solución analítica del problema. Tal estrategia puede entrañar la solución de integrales sumamente complicadas o intrincados problemas de combinatoria o ambos. En otros casos, no basta; tal es el caso de la probabilidad que corresponde a cada cara de un poliedro irregular que ha de ser lanzado.

La segunda, no requieren de tal cuerpo teórico; sin embargo, realizar ciertos experimentos un gran número de veces resulta en ocasiones una tarea engorrosa y tediosa. Aunque, en ocasiones, es la única manera posible de estimar una probabilidad (aparte de la estimación subjetiva). Ese es el caso, por poner un ejemplo aun más simple, de una moneda que ha sido doblada o arqueada con una pinza. No hay otro modo de estimar la probabilidad de que salga cada una de las caras como no sea lanzándola un gran número de veces.

La última posibilidad, la de aplicar técnicas de simulación mediante el ordenador, se ha convertido en una estrategia sumamente atractiva e intuitiva, ya que, por un lado, no se requiere del conocimiento teórico de la teoría de probabilidades y por otro, agiliza el proceso de realizar el experimento. Su principal dificultad radica en que simular un experimento exige conocer las leyes físicas que lo rigen. Eso puede dimanar de supuestos teóricos que se atribuyen al modelo físico en el cual estamos interesados.

Para estimar empíricamente la probabilidad de un suceso, el recurso de la simulación se apoya en la definición frecuencial de probabilidad:

Si algún experimento u observación se repite un gran número de veces (B), y si un suceso A ocurre k veces, la frecuencia relativa de A; k/B, será aproximadamente igual a la probabilidad de A, esto es: $P(A) \approx k/B$.

Basándose en esta definición, las probabilidades se calculan después de realizarse la observación o experimento; de ahí que también suele conocérsele con la expresión *probabilidad a posteriori*. Veamos con un simple ejemplo cómo opera la definición anterior:

Supongamos que deseamos calcular la probabilidad de obtener cara en el lanzamiento de una moneda. Para ello simularemos el lanzamiento de una moneda B veces y calcularemos la frecuencia relativa del suceso "*se obtiene cara*". Si se fuera a aplicar el procedimiento 2, no hay que hacer supuesto alguno; simplemente se podría repetir la experiencia muchas veces. Supongamos que, al ejecutar esta estrategia obtenemos lo siguiente.

Tabla 3.1: Número de caras obtenidas al lanzar una moneda B veces.

B	No. De caras	Proporción de caras
100	48	0.480
200	102	0.510
400	196	0.490
800	397	0.496
1000	509	0.509

Así, cada uno de los valores de la última columna es una aproximación de la probabilidad que se desea conocer. Cabe esperar que, a medida que B aumenta, la aproximación se vuelva más confiable. Se define el valor exacto de la probabilidad como el límite cuando B tiende a infinito.

En otras palabras, si se repite un experimento en condiciones uniformes, para valores crecientes de B la frecuencia relativa de un suceso fijo A asociado al experimento exhibe una marcada tendencia a permanecer constante.

Los siguientes ejemplos ilustran el poder de la simulación como herramienta utilizada para resolver problemas en el marco de las probabilidades.

Problema 3.1.1

Una familia tiene 2 hijos, uno de ellos se sabe que es varón, ¿cuál es la probabilidad de que el otro hijo sea igualmente varón?

Para realizar una valoración informal, enviamos este sencillo problema por correo electrónico a 25 residentes de Epidemiología y Bioestadística. El 100% respondió erróneamente que la probabilidad deseada era 0.5^{10}. El error dimana de no comprender el carácter condicional de la probabilidad solicitada. Veamos la solución correcta de este problema.

Si una familia tiene dos hijos, los posibles resultados son:

1. varón –hembra
2. hembra - varón
3. varón – varón
4. hembra – hembra

Como vemos hay tres eventos que son útiles a los efectos del problema (al menos sabemos que uno de los hijos es varón) y de ellos sólo uno corresponde al hecho “los dos son varones”, por tanto la probabilidad deseada en 1/3. Si utilizamos adecuadamente la simulación, seguramente nunca cometeremos el error antes mencionado.

Los siguientes pasos reproducen exactamente el problema al que nos estamos enfrentando.

1. Lanzamos 2 monedas que no estén trucadas, simulando de esta forma la familia con los 2 hijos.

2. Si cae una cara (al menos tienen un hijo varón) registramos este evento(1); si obtenemos 2 caras también lo registramos (2) (el otro hijo también es varón); si salen dos escudos, el experimento se desecha.

3. Repetimos los pasos 1 y 2 un buen número de veces.

4. Computamos la proporción de familias con 2 hijos varones en relación con el total de familias que al menos tienen un varón (observe que este número es la suma de los eventos 1 y 2).

[10] Hemos adoptado la probabilidad de nacer varón igual a 0.5, sólo para abreviar los cálculos y de esa forma mejorar la comprensión de la idea.

Para ahorrarnos el trabajo de tener que lanzar las monedas, acudimos al software Resampling Stats.

REPEAT 1000	repite el experimento 1000 veces
GENERATE 2 0,1 A	genera 2 números aleatorios, 1 ó 0 y los coloca en el vector A (que representa las familias con 2 hijos)
COUNT A=1 B	cuenta cuántos 1 hay en el vector A (si tienen o no hijos varones), y coloca el resultado en la variable B
IF B=1	si el valor de la variable B es igual a 1 (hay exactamente un varón)
SCORE 1 C	agrega un uno al vector C
END	finaliza la condicional
IF B=2	si el valor de la variable B es igual a 2 (los 2 hijos son varones)
SCORE 2 C	agrega un 2 al vector C
END	finaliza la condicional
END	finaliza el experimento
COUNT C=1 V1	cuenta cuántos unos hay en el vector C (familias con 1 varón) y coloca el resultado en la variable V1
COUNT C=2 V2	cuenta cuántos dos hay en el vector C (familias con dos varones) y coloca el resultado en la variable V2
LET P=V2+V1	suma los valores de las variables V1 y V2 y coloca el resultado en la variable P (el número de familias con al menos un varón.
DIVIDE V2 P PRO	divide el valor de la variable V2 entre el de la variable P y coloca el resultado en la variable PRO (calcula la proporción de familias con 2 varones entre aquellas que al menos tienen un hijo varón)
PRINT PRO	muestra en pantalla el resultado de la variable PRO

Luego de correr el programa, la probabilidad deseada ascendió a 0.336. Como vemos, se trata de un resultado enteramente coherente con el resultado teórico correcto del problema.

Problema 3.1.2

¿Cuál es la probabilidad de que 2 o más personas, entre un total de 25 que comenzarán el primer año de bioestadística en la Escuela Nacional de Salud Pública, cumplan años el mismo día?

La solución analítica de este viejo y sorprendente problema, aunque no es de las más difíciles, sí entraña cierta complejidad, sobre todo desde el punto de vista de los cómputos. Veamos como procederíamos por esta vía:

Llamémosle A al suceso "al menos dos personas celebran sus cumpleaños a la vez" y A^c al evento complementario "no hay dos personas para las que coincida la fecha de nacimiento"

Suponiendo un año de 365 días, el número de casos posibles de celebración de cumpleaños es 365^{25} (primera complicación, ya que este número es enorme). El número de casos favorables a que no existan dos personas que hayan nacido el mismo día se puede obtener de la siguiente forma: 365 × 364 × 363 × …, 341 (segunda complicación, aunque no es tan grande como el primero, es imposible de obtener con una calculadora de bolsillo) ya que la primera de las personas puede haber nacido uno de los 365 días del año, la siguiente unos de los 364 días restantes y así sucesivamente, por lo que la probabilidad de que no hayan dos personas que cumplan años el mismo día viene dado por:

$$p(A^c) = \frac{365\times364\times363,...\times341}{365^{25}} = 0.4313$$

así la probabilidad que se buscaba es:

$p(A) = 1 - p(A^c) = 1 - 4313 = 0,5687$

Mediante simulación, la solución es extremadamente sencilla si se compara con el procedimiento desarrollado anteriormente. Los siguientes pasos pueden ayudarnos a resolver el problema intuitivamente:

1. Con una tabla de números aleatorios examinamos los primeros 25 números que se encuentran en el recorrido de 001 hasta 365 (los días del año).

2. Si los 25 números son diferentes, se registra 0; en caso contrario anotamos 1.

3. Repetimos los pasos anteriores digamos 1000 veces.

4. Computamos la proporción de unos entre el número de veces que repetimos el experimento.

Con Resampling Stats es sencilla la solución.

REPEAT 1000	repite el experimento 1000 veces
GENERATE 25 1,365 A	genera 25 números aleatorios entre 1 y 365 y los coloca en el vector A
MULTIPLES A>1 B	cuenta cuántas veces un mismo número en el vector A aparece más de una vez y coloca el resultado en la variable B
SCORE B Z	agrega el resultado de la variable B en el vector Z
END	finaliza el experimento
COUNT Z>0 P	cuenta cuántos valores contenidos en el vector Z son mayores que 0 y coloca el resultado en la variable P (individuos que cumplieron años el mismo día)
DIVIDE P 1000 PRO	divide el contenido de la variable P entre 1000 y coloca el resultado en la variable PRO (computa la p deseada)
PRINT PRO	muestra en pantalla el contenido de la variable PRO

Al correr el programa obtuvimos una probabilidad de 0.5688, un número muy próximo al obtenido teóricamente, ahora prescindiendo de las complicaciones que implica dicho procedimiento**.**

Problema 3.1.3

Para concluir esta sección, proponemos resolver el siguiente problema: Se lanzan 23 dados; ¿cuál es la probabilidad de que la suma sea un múltiplo de 7?

Este problema resulta realmente complicado de resolver mediante la teoría formal de las probabilidades, ya que implicaría tener que determinar de cuántas maneras distintas se puede obtener un múltiplo de 7 (el número de formas en que puede, por ejemplo, obtenerse 84 es muy difícil de establecer) y dividirlo por el astronómico número 6^{23} ó estimarla mediante la aproximación a la distribución normal. Sin embargo, mediante la simulación la respuesta es casi inmediata. El siguiente programa permite estimar la probabilidad de interés:

DATA(28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133) M7
Genera el vector de datos M7 cuyo contenido son todos los múltiplos de 7 que se pueden obtener al lanzar 23 dados

REPEAT 1000	repite el experimento 1000 veces
GENERATE 23 1,6 DADOS	genera 23 números aleatorios ente 1 y 6 (simula el lanzamiento de 23 dados); coloca el resultado en el vector DADOS
SUM DADOS TOTAL	suma los valores contenidos en el vector DADOS y coloca el resultado en la variable TOTAL
IF TOTAL MEMBEROF M7	si el contenido de la variable TOTAL es "miembro" del vector M7 (si la suma de los números obtenidos mediante el lanzamiento de los 23 dados es uno de los 16 múltiplos de 7 que se encuentran en el vector M7)
SCORE 1 Z	agrega un 1 en el vector Z
END	finaliza la condicional
END	finaliza el primer ciclo y se repite el proceso hasta completadas las mil experiencias
SIZE Z R	calcula el tamaño del vector Z (en cuántas experiencias la suma de los 23 dados resultó ser un múltiplo de 7) y coloca el resultado en la variable R
DIVIDE R 1000 PRO	divide el contenido de la variable R entre 1000 y coloca el resultado en la variable PRO (estima la probabilidad deseada)
PRINT PRO	muestra en pantalla el resultado de la variable PRO

Luego de correr éste programa, obtuvimos una probabilidad estimada de 0.143.

Como se ha ilustrado con estos ejemplos, resolver casi cualquier problema en el marco de la teoría de las probabilidades usando el poder de la simulación, es una tarea sencilla, resulte o no sencilla la solución con los métodos formales basados en fórmulas (siempre, claro está, que nos hallemos en el marco de una formación experimental como las que se han ilustrado).

3.2 Intervalos de confianza *bootstrap*.

Cuando conducimos una investigación, casi siempre estamos interesados en conocer el valor de algún parámetro en la población objeto de estudio. Por razones de eficiencia, nos vemos por lo general obligados a seleccionar una muestra de esa población, a través de la cual tratamos de conocer aproximadamente el valor de interés. A este proceso se le conoce como "estimación".

Para enfrentarse a la incertidumbre que supone trabajar con una muestra, se ideó un procedimiento conocido como estimación por intervalos de confianza, que da lugar a un recorrido de posibles valores entre los cuales estamos confiados que esté el verdadero valor que tratamos de conocer.

Cuando los supuestos sobre los que descansa la estadística frecuentista no se cumplen y, sobre todo, cuando no se conocen los recursos matemáticos imprescindibles para abordar este asunto, el *bootstrap* se ha convertido en una alternativa sumamente atrayente y muy utilizada en la actualidad. Los siguientes ejemplos demuestran la sencillez del método.

3.2.1 Intervalos de confianza *Bootstrap* para la media.

Problema 3.2.1 Algunos científicos han sugerido que la psoriasis es más frecuente entre los individuos que son bebedores intensos de alcohol que entre bebedores moderados y abstemios. Con vistas a la planificación de un estudio analítico a gran escala un grupo de investigadores decidieron seleccionar una muestra aleatoria de 50 pacientes con psoriasis y estimar a partir de esa muestra el promedio del consumo de alcohol expresado en ml/día de la población de pacientes con psoriasis. Los sujetos seleccionados fueron examinados acerca de sus hábitos de consumo de alcohol (Tabla 3.2.1). Calcular un intervalo de confianza del 95% para el valor de la ingestión media de alcohol a partir de estos datos.

Tabla 3.2.1: Consumo diario de alcohol en 50 pacientes con psoriasis*.

Indi.	Consumo alcohol (ml/día)	Indi.	Consumo alcohol (ml/día)	Indi.	Consumo alcohol (ml/día)	Indi.	Consumo alcohol (ml/día)	Indi.	Consumo alcohol (ml/día)
1	2.8	11	75.2	21	38	31	9.4	41	105.2
2	63	12	42.5	22	29.8	32	112.5	42	1.6
3	65.9	13	46	23	66.8	33	193.2	43	102.1
4	71.1	14	155	24	162.9	34	59.7	44	28.7
5	65.4	15	40.4	25	7.7	35	147	45	144.6
6	166.4	16	65.4	26	151.4	36	8	46	92.8
7	23	17	49.5	27	112.9	37	155.3	47	31.9
8	14.2	18	99.6	28	98.2	38	8.6	48	146.2
9	139.2	19	132.7	29	23.6	39	97.2	49	212.6
10	179.9	20	95.2	30	98.5	40	3	50	163.6

*datos hipotéticos.

¿Cómo resolver el problema mediante el *bootstrap?*

Lo primero que tenemos que hacer es construir un universo hipotético de valores de consumo de alcohol en pacientes con psoriasis y de éste seleccionar un gran número de *remuestras* con reemplazo, calcular el estadístico de interés para cada *remuestra* (en el ejemplo que nos ocupa la media del consumo de alcohol), y finalmente calcular los percentiles correspondientes al nivel de confianza deseado.

Los siguientes pasos nos conducen operativamente a solucionar el problema.

1. Anotar los 50 valores del consumo de alcohol de los pacientes en tarjetas independientes, y colocarlas en una urna.

2. Seleccionar aleatoriamente y **con reemplazo** 50 tarjetas de la urna (de esta forma se genera un universo infinito de valores).

3. Calcular la media del consumo de alcohol de esa *remuestra*.

4. Repetir 1000 veces los pasos 2 y 3.

5. Estimar los percentiles 2.5 y 97.5 de la distribución de la media a partir de los 1000 valores de medias obtenidos mediante el procedimiento descrito. Estos serían los límites de un intervalo de confianza *bootstrap* del 95%.

Resolución del problema a través del Resampling Stats.

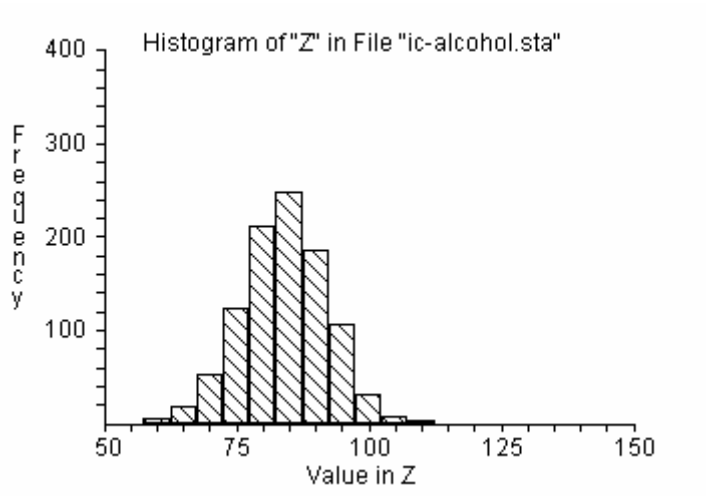
Con los datos de la Tabla No. 3.2.1 creamos un fichero tipo texto llamado “ic-alcohol” con el propósito de optimizar el proceso de introducción de datos.

READ FILE “ic-alcohol” A	“lee” el fichero “ic-alcohol” donde se encuentran los 50 valores del consumo de alcohol (ml/día) de los pacientes y los ubica en el vector A.
REPEAT 1000	repite el experimento 1000 veces
SAMPLE 50 A B	selecciona una muestra de tamaño 50 con reemplazo del vector A y coloca el resultado en el vector B
MEAN B MEDIA	calcula la media de los valores contenidos en el vector B y coloca el resultado en la variable Media (calcula la media del consumo de alcohol en la muestra elegida)

SCORE MEDIA Z	agrega el contenido de la variable MEDIA en el vector Z
END	finaliza el primer ciclo y se repite el procedimiento hasta completar los mil experimentos)
PERCENTILE Z (2.5 97.5) P	calcula los percentiles 2.5 y 97.5 de la distribución de valores contenidos en el vector Z, y coloca el resultado en el vector P
PRINT P	muestra en pantalla el valor de la variable P (intervalo de confianza al 95%)
HISTOGRAN PERCENT Z	dibuja un histograma de la distribución de medias contenidas en el vector Z

Al correr el programa obtuvimos lo siguiente:

P= 69.24 100.76



Con lo cual podemos afirmar: tenemos confianza en que el verdadero valor del consumo de alcohol promedio en la población de pacientes con psoriasis está entre 69.24 y 100.76 ml/día, ya que este intervalo se obtuvo por un método que el 95% de las veces logra incluir el verdadero valor del parámetro estimado.

Si construimos un intervalo de confianza para la media del consumo de alcohol de estos paciente por los métodos tradicionales obtenemos un resultado notablemente parecido: IC= 67.52 a 100.68.

3.2.2 Intervalos de confianza para la diferencia entre medias independientes.

Problema 3.2.2

Aunque se conoce que las anomalías ocasionadas por la deformidad de la caja torácica intervienen en la disfunción ventilatoria, pocas veces se ha estudiado la relación de la función muscular inspiratoria con el desarrollo de insuficiencia respiratoria. Un grupo de investigadores deseaba valorar la función de los músculos inspiratorios y relacionarla con incapacidad ventilatoria e insuficiencia respiratoria. Se estudiaron 9 adultos con disfunción de los músculos que participan en el proceso respiratorio (DMR), junto con un grupo control de 6 miembros normales del personal de laboratorio. La presión inspiratoria máxima (PIMAX) es una medición que refleja la fuerza combinada de todos los músculos respiratorios, de manera que valores bajos de ésta indican una capacidad inspiratoria disminuida. En el Tabla 3.2.2 se encuentran los datos obtenidos de PIMAX en ambos grupos de individuos. ¿Se puede concluir sobre la base de estos resultados que la media de PIMAX en el grupo de sujetos con disnea de esfuerzo es significativamente diferente de la de los sujetos del grupo control? Utilice intervalos de confianza para la diferencia de medias.

Tabla 3.2.2: Presión inspiratoria (PIMAX) en los grupos de pacientes estudiados.

Pacientes	**Controles PIMAX (cm de H2O)**	**Casos (DMR) PIMAX (cm de H2O)**
1	60.45	54.8
2	78.4	62.0
3	90.5	63.3
4	100.05	44.2
5	81.4	40.3
6	85.3	36.3
7		19.3
8		24.6
9		26.6
Media	**82.68**	**41.26**
Diferencia	**41.42**	

En este caso, lo que estamos tratando de esclarecer es si la diferencia observada realmente es debida a la disfunción de los músculos respiratorios o pudiera explicarse solo por el azar. Un procedimiento de trabajo sería examinar la distribución de diferencias de medias, entre un gran número de muestras seleccionadas a partir de las dos poblaciones en estudio; otro (el usual) sería estimar un intervalo de confianza basado en los datos disponibles y operar con él de modo que si contiene el cero, entonces no se podría descartar que la diferencia observada fuese debida a la variabilidad aleatoria; si este hecho no se produce, concluiríamos que la diferencia observada se debe a que los pacientes con disfunción de los músculos respiratorios tienen una capacidad pulmonar disminuida con respecto a los individuos normales.

Una tercera alternativa sería crear, sobre la base de los datos muestrales, universos hipotéticos formados por replicaciones de los valores empíricos, y seleccionar un número grande de remuestras para estimar la distribución poblacional de las diferencias de medias entre estos dos grupos de individuos. Los siguientes pasos ayudan a la comprensión del procedimiento*:*

1. Crear un universo hipotético para los pacientes con disfunción de los músculos inspiratorios (casos) y otro para los individuos normales (controles).

2. Seleccionar aleatoriamente y **con reemplazo** (con lo cual estaríamos replicando infinidad de veces los valores de las unidades de observación) 9 pacientes del universo de los casos y 6 individuos del universo de controles

3. Computar la media en ambas muestras y luego la diferencia entre ellas.

4. Repetir los pasos 2 – 3 un buen número de veces, digamos 1000.

5. Calcular los percentiles 2.5 y 97.5 de la distribución de las diferencias de medias originadas mediante el bootstraping. El intervalo así formado contiene, con una confianza del 95%, el verdadero valor de la diferencia de medias.

Resolución del problema con Resampling Stats

DATA (54.8 62.0 63.3 44.2 50.3 36.3 19.3 24.6 26.6) CASO

Genera un vector de datos nombrado “CASO” con los valores obtenidos a partir de las mediciones de PIMAX (población hipotética de casos)

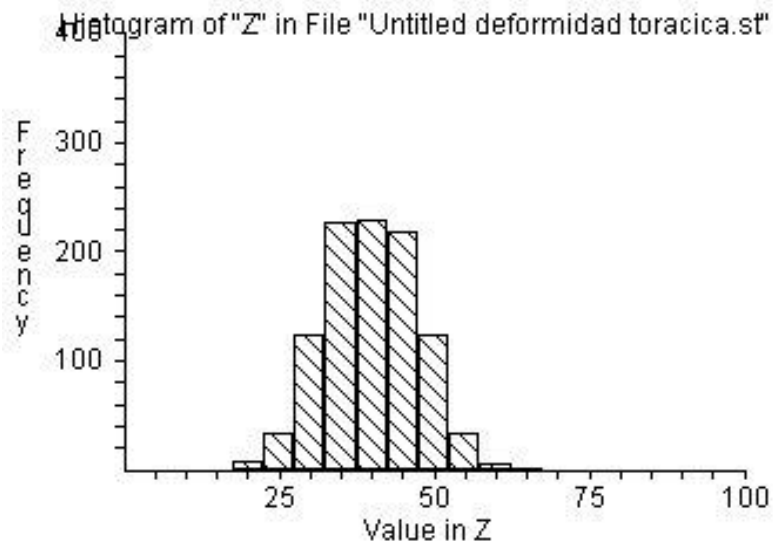
DATA (60.45 78.4 90.5 100.05 81.4 85.3) CONTROL

Genera un vector de datos nombrado “CONTROL” con los valores obtenidos a partir de las mediciones de PIMAX (población hipotética de controles)

REPEAT 1000	repite los pasos subsiguientes 1000 veces
SAMPLE 9 CASO X1	selecciona una muestra aleatoria de tamaño 9 con reemplazo del vector CASO y coloca dichos valores en el vector X1
SAMPLE 6 CONTROL X2	selecciona una muestra de tamaño 6 con reemplazo del vector CONTROL y coloca dichos valores en el vector X2
MEAN X1 MEDIA1	calcula la media de los valores del vector X1 y coloca el resultado en la variable MEDIA1 (valor medio de PIMAX en los casos)
MEAN X2 MEDIA2	calcula la media de los valores del vector X2 y coloca el resultado en la variable MEDIA2 (valor medio de PIMAX en los controles)
SUBTRACT MEDIA2 MEDIA1 DIFF	calcula la diferencia entre los valores de las variables MEDIA2 y MEDIA1 y coloca el resultado en la variable DIFF (computa la diferencia de medias obtenidas mediante la simulación)
SCORE DIFF Z	agrega el resultado de la variable DIFF en el vector Z
END	finaliza el ciclo SAMPLE-SCORE hasta completar las mil repeticiones
HISTOGRAM Z	construye un histograma con los valores contenidos en el vector Z (dibuja un histograma utilizando la distribución de diferencias de medias obtenidas mediante simulación)
PERCENTILE Z (2.5 97.5) P	calcula los percentiles 2.5 y 97.5 de los valores contenidos en el vector Z y coloca el resultado en la variable P (intervalo de confianza del 95% para la diferencia de medias entre los casos y los controles)
PRINT P	muestra el resultado de la variable P

Estos son los resultados que obtuvimos:

P= 25.574 a 54.069



Este resultado es compatible con la hipótesis propuesta: la función de los músculos inspiratorios en adultos está asociada con incapacidad ventilatoria e insuficiencia respiratoria, ya que, como vemos, el intervalo de confianza no incluye al 0, hecho indicativo de que las poblaciones de las cuales se extrajeron las muestras son diferentes en cuanto al parámetro estudiado. El intervalo de confianza paramétrico en este caso es de: 24.76 a 58.07cm H_2O, resultado similar al obtenido mediante el *remuestreo*.

3.3 Pruebas de Hipótesis para la diferencia de 2 proporciones.

Problema 3.3: Baumgartner y colaboradores (1985)[11] publicaron un estudio doble ciego, aleatorio, sobre la eficacia del antisuero J5 en la prevención de shock séptico en pacientes quirúrgicos graves con infecciones por gérmenes Gram negativos. Estudiaron 262 pacientes internados en la unidad de cuidados intensivos quirúrgicos; 126 pacientes recibieron el antisuero J5 y 136 recibieron plasma control. Los resultados se muestran en la Tabla 3.3.1. Sobre la base de estos datos, ¿se puede concluir que el antisuero J5 es eficaz en la prevención de shock séptico en este tipo de pacientes?

[11] Tomado de Saunters y Trapp (1998)

Tabla 3.3.1. Episodios de shock séptico en pacientes quirúrgicos graves tratados con el antisuero J5 en comparación con controles.

Shock	**Grupo experimental**		**Grupo control**		**Total**
	No.	**%**	**No.**	**%**	
Sí	6	4.76	15	11.02	21
No	120	95.24	121	88.98	241
Total	126	100	136	100	262

Cuando aplicamos los procedimientos de *remuestreo* para darle solución a algún problema en el ámbito de la inferencia estadística, lo esencial es reconocer y entender cabalmente la situación a la cual nos estamos enfrentando.

En el caso que nos ocupa, la pregunta científica es si el antisuero J5 es eficaz en la prevención de shock séptico en pacientes quirúrgicos con sepsis a gérmenes Gram negativos. Para responderla, como vimos en la sección 1.2.1, podemos combinar las dos muestras y formar un universo hipotético constituido por (6+15) pacientes que sufrieron una complicación grave y (120+121) que no se complicaron. Luego se calcula la probabilidad que tiene tal universo de originar muestras con diferencias de tasas de shock tanto o más extremas que la observada en el estudio. Sólo si ésta probabilidad es menor que 0.05, rechazamos la hipótesis nula H_0: $P_1 = P_{2,} = 0.5$.

Un paso crucial es identificar si nos encontramos frente a una situación en la cual debemos ejecutar una prueba de una o dos colas. Si contamos con información *a priori* y existen fuertes evidencias para pensar que la diferencia, si se da, es en una dirección dada (p.ej. el antisuero j5 no puede ser peor que un placebo en la prevención de shock séptico), entonces examinamos el punto derecho de la *distribución permutada* (los resultados tanto o más extremos que los observados solo del lado derecho). Si, por el contrario, no tenemos opinión a priori sobre la dirección de la posible diferencia, entonces nos interesa una diferencia en cualquier dirección y examinamos ambos extremos.

Para atacar el problema pudiéramos seguir los siguientes pasos:

1. Llenamos una urna con 126+136= 262 bolas, 21 de ellas rojas (el paciente tuvo una complicación séptica), y las restantes blancas (el paciente no tuvo ninguna complicación).

2. Seleccionamos aleatoriamente y sin reemplazo 126 bolas (grupo experimental) de la urna y computamos la cantidad de bolas rojas allí ubicadas.

3. Hallamos la proporción de bolas rojas en la muestra del grupo experimental (proporción de pacientes que se complicaron p_1).

4. Seleccionamos una segunda muestra de la urna de la misma forma que la descrita en el paso 2, pero de tamaño 136 (grupo control).

5. Hallamos la proporción de bolas rojas en la muestra del grupo control (proporción de pacientes que se complicaron en el grupo control p_2).

6. Calculamos la diferencia de proporciones de bolas rojas entre las dos muestras seleccionadas (p_2-p_1).

7. Repetimos los pasos 2 al 6 numerosas veces

8. Si la diferencia es mayor o igual que 0.0626 (la diferencia observada empíricamente) anotamos 1 (de esta forma abordamos el problema con una prueba de 1 cola)[12].

9. Hallamos la proporción de 1's registrados en relación con la cantidad de veces que efectuamos el experimento y de esa forma estimamos el valor p

Si ese valor es menor que 0.05, rechazamos la hipótesis de nulidad.

Nótese que al unir y mezclar los supuestos sujetos experimentales con los supuestos controles y dividir ese conjunto aleatoriamente en subconjuntos de tamaños 126 y 136, se esta asumiendo la validez de H_0 (nada distingue a unos de otros) y que lo que se computa es la probabilidad empírica de obtener diferencias mayores que 0.0626 bajo dicho supuesto.

Solución con Resampling Stats

URN 21#1 241#0 UNIVERSO	**"**construye" un vector llamado Universo el cual contiene 21 unos y 241 ceros (pacientes con y sin shock)
REPEAT 1000	repite el experimento 1000 veces
SHUFFLE UNIVERSO UNIVERSO	reordena aleatoriamente el vector "Universo"
TAKE UNIVERSO 1,126 X1	toma los valores del vector universo que ocupan las posiciones de la 1 a la 126 (divide aleatoriamente el universo en dos conjuntos. Con los primeros 126 números del vector UNIVERSO forma el primer grupo -grupo experimental-) y coloca el resultado en el vector X1
TAKE UNIVERSO 127, 262 X2	toma los valores ubicados en las posiciones de la 127 a la 262 dentro del vector UNIVERSO y los coloca en el vector X2 (grupo control)

[12] Si el problema fuera coherente con una prueba de 2 colas, tendríamos que computar las valores absolutos de la diferencia.

COUNT X1=1 P1	cuenta el número de unos que hay en el vector X1 (pacientes con shock en la muestra experimental) y coloca el resultado en la variable P1
DIVIDE P1 126 PRO1	divide el valor de la variable P1 entre 126 y coloca el resultado en la variable PRO1 (proporción de pacientes con shock en el grupo experimental)
COUNT X2=1 P2	cuenta cuántos unos hay en el vector X2 (pacientes con shock en la muestra control) y coloca el resultado en la variable P2
DIVIDE P2 136 PRO2	divide el valor de la variable P2 entre 136 (proporción de pacientes con shock en el grupo control) y coloca el resultado en la variable PRO2
SUBTRACT PRO2PRO1 DIFF	calcula la diferencia entre los valores de las variables PRO2 y PRO1 (computa la diferencia de proporciones entre las muestras) y coloca el resultado en la variable DIFF
IF DIFF >= 0.0626	si el valor contenido en la variable DIFF es mayor o igual que 0.0626 (si la diferencia es mayor o igual a la observada en el ensayo clínico realizado)
SCORE 1 Z	agrega un 1 en el vector Z
END	finaliza la condicional
END	finaliza el experimento y regresa a la línea de programación definida por el comando SHUFFLE hasta terminar los mil ciclos
SIZE Z D	calcula el tamaño del vector Z y ubica el resultado en la variable D (en cuántas remuestras se encontró una diferencia al menos igual o mayor que la observada)
DIVIDE D 1000 PRO	divide el valor de la variable D entre 1000 y coloca el resultado en la variable PRO (estima la probabilidad deseada)
PRINT PRO	muestra el resultado en pantalla de la variable PRO

El resultado que obtuvimos al correr el programa fue 0.03; si comparamos este resultado con el obtenido mediante un procedimiento paramétrico vemos que es casi exactamente igual (p=0.0311; el lector lo puede comprobar efectuando una prueba de hipótesis para la igualdad de dos proporciones). Teniendo en cuenta el resultado de la prueba concluimos que:

Hay suficiente evidencia muestral como para afirmar que el antisuero J5 es eficaz en la prevención de shock séptico en los pacientes quirúrgicos con alto riesgo de sepsis a gérmenes Gram negativos, con un nivel de confianza del 95%.

3.4 Pruebas de hipótesis para la diferencia de medias en muestras independientes.

Problema 3.4.1

Supongamos que, un grupo de investigadores del "Centro de Ingeniería Genética y Biotecnología" está desarrollando un nuevo medicamento para prolongar la vida de los pacientes que padecen SIDA. Para comenzar a transitar por todas las etapas necesarias que demuestren la presunta eficacia del nuevo medicamento, han propuesto desarrollar un ensayo clínico (fase preclínica) en el cual asignarán 7 simios previamente infectados con el virus a un grupo que recibirá el nuevo medicamento (grupo experimental), y 9 simios (también infectados) a un grupo que recibirá como tratamiento un placebo (grupo control). Luego de ejecutar este procedimiento obtuvieron los resultados de la Tabla 3.4.1. ¿Es posible concluir que el nuevo tratamiento prolonga el tiempo de supervivencia entre los simios infectados?

Tabla 3.4.1: Días de supervivencia para los simios en los dos grupos estudiados.

Grupos	**Días de supervivencia**	**Media**
Experimental	94, 38, 23, 197, 99, 16, 141	86.86
Control	52, 10, 40, 104, 50, 27, 146, 31, 46	56.22
Diferencia observadas de medias		30.64

Cuando efectuamos una prueba de significación para probar hipótesis como la expresada en el ejemplo, contamos con dos muestras aleatorias independientes $z=(z_1, z_{2,\ldots,}z_n)$ y $y=(y_1,y_2,\ldots y_n)$ seleccionadas posiblemente de diferentes distribuciones de probabilidades F y G.

$F \longrightarrow z=(z_1, z_2,\ldots,z_n)$

$G \longrightarrow y =(y_1, y_2, \ldots, y_m)$

Habiendo observado *z* y *y*, deseamos valorar la hipótesis H_0: F=G; si no podemos rechazarla mediante una prueba de significación, entonces se podría pensar que no hay diferencia entre el comportamiento probabilístico del vector z y el vector y, dicho de otra forma, que la diferencia observada empíricamente es atribuible a la variabilidad aleatoria y no a diferencias intrínsecas entre los grupos estudiados.

Siendo coherente con el pensamiento estadístico desarrollado en el Capítulo 1, podemos valorar la hipótesis H_0: F=G mediante el uso de una *PPE*. Recuérdese que esta prueba es útil para realizar contrastes de hipótesis que expresan la igualdad de dos distribuciones.

Los siguiente pasos pueden seguirse para resolver el problema:

1. Tomamos un conjunto de tarjetas sobre las cuales escribimos los 16 valores del tiempo de supervivencia obtenidos en los simios estudiados.
2. Seleccionamos aleatoriamente **sin reemplazo** 7 de ellos (grupo experimental) y calculamos la media $\bar{z}^*$.
3. Las restantes 9 tarjetas formarán la muestra del grupo control; con los datos que ellas registran calculamos la media $\bar{y}^*$.
4. Efectuamos la diferencia de medias ($\bar{q}^*_{1=} \bar{z}^* - \bar{y}^*$).
5. Si la diferencia es mayor o igual que 30.64 (el valor de la diferencia observada en el ensayo) registramos este resultado.
6. Repetimos los pasos 2 al 5 B veces.
7. Hallamos la proporción de 1's en relación con la cantidad de veces que efectuamos el experimento y de esa forma estimamos el valor p

8. Si p es menor que 0.05 rechazamos la hipótesis de nulidad.

Solución con Resampling Stats:

DATA (94 38 23 197 99 16 141)A	genera un vector de datos A cuyo contenido es igual a los valores del tiempo de supervivencia en la muestra del grupo experimental
DATA (52 10 40 104 50 27 146 31 46)B	genera un vector de datos B cuyo contenido es igual a los valores del tiempo de supervivencia en la muestra del grupo control
CONCAT A B UNI	combina los valores de los vectores A y B, y coloca el resultado en el vector UNI (forma un universo hipotético)
REPEAT 1000	repite 1000 veces los pasos que siguen hasta hallar el END correspondiente
SHUFFLE UNI G	reordena aleatoriamente los valores del vector UNI y coloca este nuevo ordenamiento en el vector G (en cada ciclo se obtendrá un nuevo reordenamiento del vector UNI)
TAKE G 1,7 Z	toma los valores del vector G que se encuentran entre la posición 1 y la posición 7 y los coloca en el vector Z (selecciona una muestra aleatoria sin reemplazo: grupo experimental)
TAKE G 8,16 Y	toma los valores del vector G que se encuentran entre la posición 8 y la posición 16 y los coloca en el vector Y (selecciona una muestra aleatoria sin reemplazo: grupo control)
MEAN Z Z1	calcula la media de los valores contenidos en el vector Z, y coloca el resultado en la variable Z1 (media de la muestra del grupo experimental)
MEAN Y Y1	calcula la media de los valores contenidos en el vector Y y coloca el resultado en la variable Y1 (media de la muestra del grupo control)
SUBTRACT Z1 Y1 DIFF	calcula la diferencia entre los valores de las variables Z1 y Y1, coloca el resultado en la variable DIFF (calcula el estadístico de interés $\bar{q}^*$)
IF DIFF>=30.63	si el valor de la variable DIFF es mayor o igual a 30.63 (mayor que la diferencia observada empíricamente)

SCORE 1 T	consigna el hecho agregando un 1 en el vector T (registra éste resultado)
END	finaliza la condicional
END	finaliza el primer ciclo y repite el proceso hasta que se complete 1000 veces
SIZE T W	calcula el tamaño del vector T y coloca el resultado en la variable W (en cuántas remuestras la diferencia fue al menos igual o mayor a la observada)
DIVIDE W 1000 PRO	divide el valor de la variable W entre 1000 y coloca el resultado en la variable PRO (estima el valor de la probabilidad deseada)
PRINT PRO	muestra en pantalla el resultado de la variable PRO

Luego de correr el programa, en 133 ocasiones el valor para la diferencia de medias fue tan grande o mayor que la observada en el estudio original; ello representa una probabilidad de p= 0.133 (con la *prueba t de students* el valor que se obtiene es 0.141).

Sobre la base de la prueba realizada podemos afirmar con un a =0.05 que no existen diferencias entre las medias: no hay evidencia muestral suficiente como para afirmar que el tratamiento es eficaz en la prolongación del tiempo de supervivencia de los simios.

3.5 Correlación y regresión lineal simple.

En numerosas ocasiones la investigación clínica o epidemiológica nos coloca en situaciones en las que se requiere analizar la relación lineal entre dos variables cuantitativas. Los dos objetivos fundamentales de este análisis serán: por un lado, determinar si dichas variables están asociadas y en qué sentido se da dicha asociación (es decir, si los valores de una de las variables tienden a aumentar –o a disminuir- al aumentar los valores de la otra) y, por otro, estudiar si los valores de una variable pueden ser utilizados para predecir el valor de la otra.

La forma correcta de abordar el primer problema es recurriendo al uso del coeficiente de correlación de Pearson. Sin embargo, el estudio de la correlación es insuficiente para obtener una respuesta a la segunda cuestión: se limita a indicar la fuerza y la dirección de la asociación mediante un único número, mientras que nosotros estaríamos interesados en modelar dicha relación y usar una de las variables (variable independiente) para "explicar" la otra (variable dependiente). Para tal propósito, lo usual es recurrir a la técnica de regresión, mediante la cual se busca la función matemática que mejor refleja la relación entre las variables; en el caso que nos ocupa, esta función es una línea recta, cuyos parámetros (intercepto y pendiente) pueden ser estimados mediante el método de los mínimos cuadrados.

Construir intervalos de confianza (IC) tanto para el coeficiente de correlación como para los parámetros de la línea recta mediante el uso de los métodos tradicionales no es tan sencillo. Por ejemplo, para construir un IC para el coeficiente de correlación, primero hay que realizar la trasformación Z de Fisher para normalizar la distribución muestral de dicho coeficiente, pues esta distribución está sesgada como consecuencia del llamado "efecto techo" (Saunters y Trapp, 1997) y luego hacer la transformación inversa, (antilogaritmo), con lo cual finalmente podemos obtener el IC deseado. Con los procedimientos de *remuestreo* esta tarea es sencilla, como se ilustra en el siguiente ejemplo.

Problema 3.5.1: Un grupo de asesores académicos de la facultad de medicina "Mariana Grajales" de Holguín, sospecha que los resultados obtenidos por los estudiantes en la prueba de ingreso a la universidad en la asignatura de biología influye en la nota promedio al concluir el primer año. Si sus sospechas son válidas, construirían un modelo predictivo mediante el cual podrían anticiparse a un desenlace no exitoso y realizar un trabajo diferencial con estos estudiantes. Para corroborar su hipótesis, decidieron realizar un estudio con 40 estudiantes a los cuales se les registró la nota alcanzada en biología y los respectivos resultados obtenidos al final del primer año. Los resultados se muestran en la Tabla 3.5.1. ¿Hay asociación entre los resultados alcanzados por los estudiantes en la prueba de biología y el promedio obtenido al finalizar el primer año? ¿Es posible predecir las calificaciones obtenidas por los estudiantes al terminar el primer año sobre la base del conocimiento de la nota de biología?

Tabla 3.5.1. Resultados alcanzados por 40 estudiantes en la prueba de biología y el promedio obtenido al finalizar el primer año de la carrera de medicina.

Estudiante	Nota Biolog.	Promedio al Año	Estudiante	Nota Biolog.	Promedio al Año
1	77.92	3.90	21	41.00	2.89
2	69.99	3.50	22	81.25	4.06
3	81.39	4.07	23	85.97	4.30
4	63.97	4.50	24	95.32	4.77
5	52.19	3.00	25	92.07	4.60
6	95.71	4.79	26	66.63	3.33
7	67.38	3.37	27	65.32	3.27
8	90.43	4.52	28	50.50	2.00
9	97.66	4.88	29	97.21	4.86
10	60.36	2.50	30	87.21	4.36
11	86.03	4.30	31	91.88	4.59
12	77.37	3.87	32	83.81	4.19
13	94.22	4.71	33	75.34	3.77
14	80.13	4.01	34	68.33	3.42
15	60.82	3.04	35	69.00	2.31
16	85.00	4.25	36	85.38	4.27
17	94.16	4.71	37	75.33	3.77
18	68.00	3.40	38	80.59	4.03
19	80.09	4.00	39	61.25	2.00
20	100.00	5.00	40	75.40	3.77

Para responder a la primera pregunta, construiremos un intervalo de confianza *bootstrap* para el coeficiente de correlación lineal de Pearson (R), y así determinar la fuerza de esa asociación. De existir asociación lineal, entonces trataríamos de responder a la segunda pregunta mediante la regresión. Los siguientes pasos pueden ayudar a resolver el problema:

1. Construir un universo hipotético de las variables estudiadas. Nótese que esta situación difiere un tanto de otras, ya que las medidas que obtenemos de las unidades de observación vienen en parejas. Es decir, el valor de la prueba de ingreso para un estudiante forma pareja con el promedio obtenido por el mismo estudiante al terminar el primer año de la carrera.

2. Seleccionar aleatoriamente con reemplazo cuarenta pares de valores de dicho universo.

3. Calcular el coeficiente de correlación en esa *remuestra* y guardar el resultado obtenido.

4. Repetir los pasos 2 y 3, digamos, 1000 veces.

5. Calcular los percentiles 2.5 y 97.5 de la distribución muestral bootstrap del coeficiente de correlación.

Siguiendo estos pasos obtenemos un IC del 95% de confianza para el coeficiente de correlación.

Con el software Resampling Stats procederíamos de la siguiente manera.

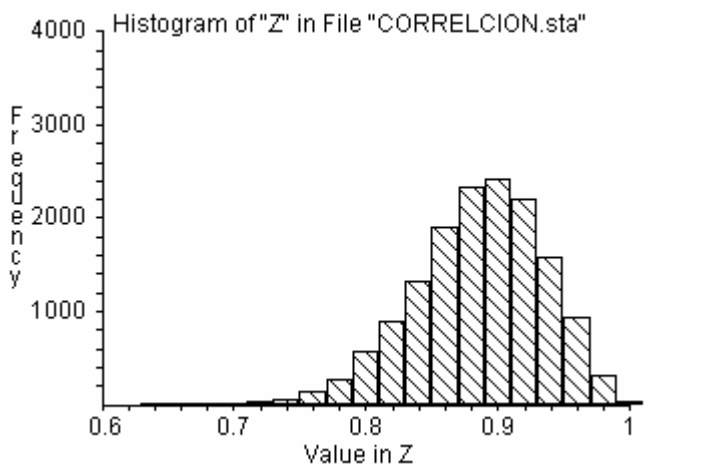
Con el objetivo de optimizar el proceso de introducción de datos construimos un fichero tipo texto nombrado “CORRELACION” con la información contenida en la Tabla No. 3.5.1.

READ FILE “CORRELACION” PB PA	lee el fichero “CORRELACION” donde se encuentran los 40 pares de valores para los estudiantes y los coloca en los vectores PB y PA respectivamente
REPEAT 1000	repite el experimento 1000 veces

GENERATE 40 1,40 A	genera 40 números aleatorios entre 1 y 40; coloca el resultado en el vector A
TAKE PB A XI	selecciona del vector PB los valores que ocupan las posiciones contenidas en el vector A (de esta forma podemos seleccionar aleatoriamente con reemplazo 40 valores indicativos de los resultados de los estudiantes en la prueba de ingreso) y coloca el resultado en el vector XI
TAKE PA A YI	selecciona del vector PA los valores que ocupan las posiciones contenidas en el vector A (de esta forma podemos seleccionar aleatoriamente con reemplazo 40 valores indicativos de los resultados de los estudiantes al finalizar el año, nótese que mediante esta estrategia podemos conseguir que los pares de valores no se aleatoricen) y coloca el resultado en el vector YI
CORR XI YI R	calcula el coeficiente de correlación entre los valores contenidos en los vectores XI y YI; coloca el resultado en la variable R
SCORE R Z	registra el valor de la variable R en el vector Z (construye la distribución bootstrap del coeficiente de correlación)
END	finaliza el primer ciclo y repite los pasos anteriores hasta completadas las 1000 simulaciones
PERCENTILE Z (2.5 97.5) P	calcula los percentiles 2.5 y 97.5 a partir de los valores contenidos en el vector Z (intervalo de confianza del 95%) y coloca el resultado en la variable P
HISTOGRAM Z	dibuja un histograma a partir de los valores contenidos en el vector Z
PRINT P	muestra en pantalla el valor contenido en la variable P

Luego de correr el programa anterior obtuvimos lo siguiente:

P= 0.78 a 0.96



Con estos resultados podemos concluir, que existe una correlación lineal intensa entre la nota obtenida por los estudiantes en la prueba de ingreso (biología) y el promedio alcanzado al concluir el primer año de la carrera de medina, ya que hemos obtenido un IC para el coeficiente de correlación de 0.78 a 0.96 mediante un método, que el 95% de las veces logra incluir el verdadero valor del parámetro estimado. Este resultado es muy similar al que se obtiene mediante los procedimientos convencionales (0.78 a 0.94).

Como comprobamos que existe una asociación lineal intensa, podemos pensar en modelar la asociación que existe entre las dos variables estudiadas, con el fin de construir un modelo matemático que nos permita predecir el promedio que alcanzará un estudiante de medicina al concluir el primer año, a partir del valor de la nota obtenida en la prueba de biología.

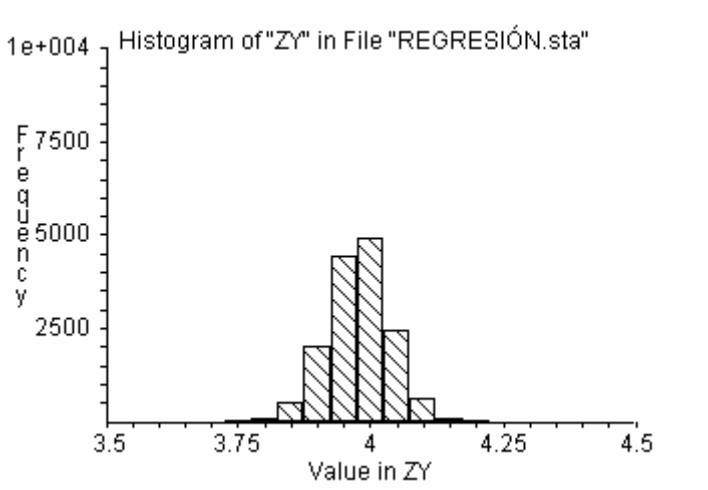
Mediante el método de los mínimos cuadrados, estimaremos los parámetros (intercepto y pendiente) de la recta y utilizando las virtudes del *remuestreo* y la sencillez del lenguaje de programación del software Resampling Stats, construiremos un programa, que automáticamente nos brindará un intervalo de confianza para el valor promedio que esperamos de un estudiante de medicina al finalizar su primer año, si sabemos, por ejemplo, que obtuvo 80 puntos en la prueba de ingreso en biología.

READ FILE "CORRELACIÓN" PB PA	lee el fichero "CORRELACIÓN" donde se encuentran los resultados alcanzados por los estudiantes en la prueba de biología y el promedio al concluir el primer año; coloca los valores en los vectores PB PA.
COPY (80) X	genera el vector X cuyo contenido será el valor de la prueba de biología para el estudiante x sobre el cual pretendemos predecir su promedio al finalizar el primer año. En este caso será 80
REPEAT 1000	repite el experimento 1000
GENERATE 40 1,40 A	genera 40 números aleatorios entre 1 y 40 y los coloca en el vector A
TAKE PB A XI	toma del vector PB los valores que ocupan las posiciones definidas por el vector A y coloca el resultado en el vector XI
TAKE PA A YI	toma del vector PA los valores que ocupan las posiciones definidas por el vector A y coloca el resultado en YI
MEAN XI MEDIAXI	calcula la media de los valores contenidos en el vector XI y coloca el resultado en la variable MEDIAXI
MEAN YI MEDIAYI	calcula la media de los valores contenidos en el vector YI y coloca el resultado en la variable MEDIAYI
CORR XI YI R	calcula el coeficiente de correlación entre los vectores XI y YI; coloca el resultado en la variable R
STDEV PI SX	calcula la desviación estándar de los valores contenidos en la variable PI y coloca el resultado en la variable SX
STDEV PA SY	calcula la desviación estándar de los valores contenidos en la variable PA y coloca el resultado en la variable SY
LET Bi=R*(SY/SX)	calcula el valor de Bi (coeficiente de regresión, nótese que con cada remuestra se obtendrá un valor ligeramente distinto de este coeficiente)
LET B_0=MEDIAYI-(Bi*MEDIAXI)	calcula el valor de B_0 (intercepto, nótese que con cada remuestra se obtendrá también un valor ligeramente distinto)
LET Y=Bo+Bi*X	calcula el valor de Y (calcula el valor puntual de la predicción)
SCORE Y ZY	"almacena" el valor de Y en el vector ZY
END	finaliza el primer ciclo y repite el procedimiento hasta completado las 1000 simulaciones

HISTOGRAM ZY	dibuja un histograma con los valores contenidos en el vector ZY
PERCENTILE ZY (2.5 97.5) P	calcula los percentiles 2.5 y 97.5 de los valores contenidos en el vector ZY y coloca el resultado en la variable P (IC para la predicción)
PRINT P	muestra en pantalla el resultado de la variable P

Nótese que para realizar una predicción por intervalo de confianza del promedio que obtendrá un alumno al concluir el primer año de la carrera de medicina conociendo su nota de biología, a partir de este sencillo programa sólo tenemos que cambiar el valor del vector X. Al correr este programa para un estudiante con una nota de 80 puntos en la prueba de biología obtuvimos el siguiente intervalo de confianza para su promedio esperado:

P = 3.8659 4.0908



A partir de estos resultados cabe esperar que un alumno que obtenga 80 puntos en la prueba de biología alcance al terminar el primer año de medicina un promedio que podemos “confiar” se halle entre 3.87 y 4.09 puntos. Al comparar éste intervalo con el que se obtiene mediante un procedimiento paramétrico (para ello tuvimos que utilizar el software JK Curve Fit, un programa especialmente diseñado para obtener todos los parámetros de la regresión) pudimos comprobar que es virtualmente el mismo (3.85 a 4.10). Queda demostrado una vez más la utilidad de este procedimiento.

3.6 Tamaño de muestra para una proporción

El cálculo del tamaño muestral es un tema que se torna en extremo escabroso para muchos investigadores. La teoría formal suele resultar inextricable, debido principalmente a la complejidad de las fórmulas que utiliza. En esta sección resolveremos un problema en el cual calcularemos el tamaño de muestra, utilizando para ello las técnicas de *remuestreo*, una alternativa que resulta atractiva por su sencillez y transparencia.

Supongamos que estamos interesados en estimar la prevalencia de pacientes hipertensos de cierta fábrica que cuenta con 1000 trabajadores. Para conseguir los propósitos planteados se ha propuesto la estrategia de seleccionar una muestra simple aleatoria cuyo tamaño pretendemos determinar. Los investigadores tienen razones para suponer que la prevalencia de hipertensos en la fábrica pudiera ser aproximadamente 20%. Una estrategia podría ser, la de elegir un número razonado de efectivos y, aprovechando las virtudes del *remuestreo*, calcular el error de muestreo en que podríamos incurrir con dicho tamaño muestral si nuestras suposiciones fueran válidas.

Por ejemplo, si nuestra experiencia y sentido común nos orientan a pensar que con 400 individuos podemos estimar adecuadamente la verdadera prevalencia, podríamos generar varias veces 400 números aleatorios entre 1 y 100 (los números del uno al veinte representan a los individuos hipertensos), computar en cada caso el porcentaje de hipertensos, y calcular en cada una de ellas el error de muestreo: la diferencia entre la supuesta prevalencia (en el caso que nos ocupa 20%) y la estimada mediante la simulación. Si el error muestral es suficientemente pequeño en relación con nuestro propósito, entonces aceptamos el tamaño muestral escogido.

Realizamos una simulación a modo de ejemplo en 10 ocasiones utilizando el software Resampling Stats:

REPEAT 10	repite la simulación 10 veces
GENERATE 400 1,100 A	genera 400 números entre 1 y 100 (selección de una muestra simple aleatoria de 400 individuos) y los coloca en el vector A
COUNT A <=20 B	cuenta cuántos números del 1 al 20 hay en el vector A (cuántos pacientes resultaron ser hipertensos en la muestra simulada) y coloca el resultado en la variable B
SUBTRACT 80 B C	calcula la diferencia entre el resultado obtenido en B y 80 (calcula la diferencia producto del azar entre el número de hipertensos obtenidos mediante simulación y el número de hipertensos que tendrían que haber ocurrido si el verdadero porcentaje es del 20%) y coloca el resultado en la variable C
ABS C D	calcula el valor absoluto de la variable C y coloca el resultado en la variable D (el valor absoluto de la diferencia)
DIVIDE D 400 E	divide el valor de la variable D entre 400 (calcula la PROPORCIÓN de la diferencia aleatoria), coloca el resultado en la variable E
SCORE E Z	agrega el valor de la variable E en el vector Z (“almacena” el valor de la diferencia aleatoria)
END	termina el ciclo y repite la secuencia de comandos anteriores hasta que se hayan completado los diez ciclos
MEAN Z K	calcula la media del vector Z (calcula el error aleatorio promedio) y coloca el resultado en K
PRINT K	muestra el resultado de la variable K

Estos son los resultados que obtuvimos al hacerlo:

Tabla 3.5.1: Error aleatorio en que se incurre al estimar la prevalencia de HTA con una muestra de 400 individuos.

Muestras simuladas	Número de pacientes con HTA	Error aleatorio (%)
1	72	2.00
2	79	0.25
3	86	1.50
4	84	1.00
5	79	0.25
6	76	1.00
7	81	0.25
8	82	0.50
9	88	2.00
10	71	2.25
Media		**1.10**

Esto quiere decir que, en promedio, obtendremos una estimación con un error de muestreo de un 1,1% alrededor del verdadero porcentaje de pacientes hipertensos. Queda en manos del investigador decidir si este error aleatorio es o no admisible teniendo en cuenta los intereses de la investigación.

Durante el complejo proceso de investigación científica surgen disímiles problemas, muchos más de las que solemos pensar, cuyo abordaje mediante la teoría convencional de la estadística se torna muy difícil o, incluso, imposible. Los procedimientos de simulación se erigen entonces como armas alternativas sumamente valiosas para intentar darles solución.

A continuación se exponen detalladamente cuatro ejemplos que ilustran con elocuencia tal afirmación.

4.1 Magia y probabilidad

Supongamos que un "mago" comunica que es capaz de adivinar una carta solo conocida por nosotros. La única demanda del "mago" es que sigamos mentalmente una regla que se inicia con la elección de una carta. La regla en cuestión es la siguiente:

> *Se cogen 3 juegos de cartas, y se barajan. Pensamos un número entre uno y diez (supongamos que fue el 3). El "mago" comienza a pasar una a una todas las cartas del mazo. Según su indicación debemos identificar mentalmente la carta que ocupa la posición correspondiente al número pensado (la tercera posición en el ejemplo). Si dicha carta es, por ejemplo, un trece (no importa el signo o palo) entonces debemos contar hacia delante a partir de esa posición, 13 lugares más y observar la carta que ocupa esa posición (el lugar 16 en este ejemplo). Supongamos que el número que lleva esa otra carta es un cinco. Tomamos nota mental de la carta que está situada cinco lugares más adelante. El proceso se repite hasta que el "mago" haya terminado de pasar todas las cartas. Al concluir, él comunica la última carta observada por nosotros sin que en momento alguno le hayamos dado información de lo que hemos venido haciendo mentalmente.*

Luego de realizar la experiencia, efectivamente, el supuesto mago adivina la última de las cartas observadas. El hecho es desconcertante, como siempre, hasta que nos enteramos de su *modus operandi*.

La estrategia del "mago" es muy simple: consiste en realizar mentalmente la misma operación que nos ha indicado, comenzando con su propio número secreto entre 1 y 10, y luego nos comunica la última carta de su serie.

Para entender mejor el proceso, supongamos que hemos elegido el número tres, y luego de barajar los tres mazos de cartas, éstas quedan del siguiente modo (de izquierda a derecha y de arriba hacia abajo):

4♠	7♣	**13♥**	8♦	1♥	8♦	9♣	13♠	9♣	6♦	2♥	13♠
6♠	5♣	6♥	**5♦**	3♥	5♦	7♣	1♠	**3♣**	4♦	3♥	**12♠**
10♣	4♦	3♥	12♠	11♠	2♣	1♥	7♦	13♥	2♦	10♣	**2♠**
10♥	**2♦**	10♣	**5♠**	11♦	4♥	12♣	2♠	**11♣**	8♦	10♥	4♠
6♥	5♣	3♠	1♦	4♠	1♣	6♦	**7♥**	6♣	9♦	6♥	6♠
13♠	2♣	**1♥**	**3♦**	8♠	10♣	**13♦**	3♥	6♣	7♦	12♥	11♠
2♠	7♥	7♦	1♣	2♥	10♦	7♣	**6♠**	9♦	11♥	11♣	8♠
5♥	**8♦**	11♣	4♠	3♠	8♥	5♦	6♣	2♠	**9♣**	2♥	7♦
8♠	11♥	1♦	12♣	8♥	10♦	**13♣**	3♠	12♦	1♥	7♣	4♠
10♠	11♣	11♥	12♦	9♠	11♣	1♦	**3♥**	12♣	8♦	**8♥**	7♠
9♠	6♠	9♦	13♥	9♣	4♥	**5♦**	11♦	10♣	3♠	2♥	**13♣**
9♣	5♥	13♠	1♣	10♦	9♦	9♥	4♣	12♦	7♥	8♦	1♠
12♠	13♣	12♥	5♦	10♠	13♥	12♦	5♣	4♠	3♣	4♦	5♥

Es fácil convencerse de que la serie de nuestras cartas secretas resulta ser la conformada por las celdas sombreadas en la matriz:

13♥- 5♦- 3♣- 12♠- 2♦- 5♠- 11♣- 7♥- 1♥- 3♦- 13♦- 6♠- 8♦- 9♣- 13♣- 3♥- 8♥- 5♦- 13♣ y 12♠

Ahora, supongamos que el supuesto mago para llevar adelante su propio proceso, eligió el número uno. Siguiendo la misma regla establecida a partir de "su" número secreto sobre la misma configuración, obtiene la serie sombreada:

4♠	7♣	13♥	8♦	**1♥**	**8♦**	9♣	13♠	9♣	6♦	2♥	13♠
6♠	**5♣**	6♥	5♦	3♥	5♦	**7♣**	1♠	3♣	4♦	3♥	12♠
10♣	4♦	3♥	12♠	11♠	**2♣**	1♥	**7♦**	13♥	2♦	10♣	2♠
10♥	2♦	**10♣**	5♠	11♦	4♥	12♣	2♠	11♣	8♦	10♥	4♠
6♥	5♣	3♠	1♦	4♠	1♣	**6♦**	7♥	6♣	9♦	6♥	6♠
13♠	2♣	1♥	3♦	8♠	10♣	13♦	3♥	6♣	7♦	12♥	11♠
2♠	**7♥**	7♦	1♣	2♥	10♦	7♣	6♠	**9♦**	11♥	11♣	8♠
5♥	8♦	11♣	4♠	3♠	**8♥**	5♦	6♣	2♠	9♣	2♥	7♦
8♠	**11♥**	1♦	12♣	8♥	10♦	13♣	3♠	12♦	1♥	7♣	4♠
10♠	11♣	11♥	12♦	9♠	11♣	1♦	3♥	12♣	8♦	**8♥**	7♠
9♠	6♠	9♦	13♥	9♣	4♥	**5♦**	11♦	10♣	3♠	2♥	**13♣**
9♣	5♥	13♠	1♣	10♦	9♦	9♥	4♣	12♦	7♥	8♦	1♠
12♠	13♣	12♥	5♦	10♠	13♥	12♦	5♣	4♠	3♣	4♦	5♥

El comunica “su” último número (**12♠**), que coincide con el “nuestro”.

Al examinar las dos series, tenemos:

	13♥	5♦	3♣	12♠	2♦	5♠	11♣	7♥	1♥	3♦	13♦	6♠	8♦	9♣	13♣	3♥	**8♥**	**5♦**	**13♣**	**12♠**
4♠	1♥	8♦	5♣	7♣	4♦	2♣	7♦	10♣	6♥	6♦	13♠	7♥	9♦	8♥	11♥	10♠	**8♥**	**5♦**	**13♣**	**12♠**

Como se puede apreciar, a partir de cierto punto (último 8♥), las series convergen de modo que al final, como es lógico, desembocan en la misma carta. Cabe preguntarse ahora si la convergencia observada en el ejemplo tiene que producirse necesariamente, con independencia de los valores iniciales del “mago” y de su “víctima”. Es obvio que si ambos eligieran el mismo número inicial (evento que ocurre con probabilidad igual a 0.1), las dos series serán idénticas. También lo es que ellas pueden concluir en la misma carta aunque los números iniciales difieran, como acabamos de ver. Pero no es imposible encontrar ejemplos en que las dos series no convergen en punto alguno.

Lo mágico radica en que, siguiendo la misma regla 2 veces, empezando con respectivos números aleatorios entre 1 y 10, la probabilidad de que las series coincidan en algún momento es sumamente alta.

Llegado a este punto la pregunta que se impone es: ¿cuál es concretamente esa probabilidad?

Desde el punto de vista teórico, el cálculo de esa probabilidad es sumamente complejo; sin embargo mediante la simulación la respuesta se obtiene de manera casi inmediata. Veamos a continuación cómo proceder:

- **Solución vía Resampling Stats:**

URN 12#1 12#2 12#3 12#4 12#5 12#6 12#7 12#8 12#9 12#10 12#11 12#12 12#13 A
genera el vector A con 156 coordenadas: 12 números 1, 12 números 2,…,12 números 13 (simula 3 juegos de cartas)

MAXSIZE DEFAULT 10000	incrementa las capacidades de los vectores a 10 000 caracteres.
REPEAT 10000	repite el experimento 10 000 veces.
SHUFFLE A J	aleatoriza el orden en que se ubican las coordenadas del vector A y coloca el resultado en el vector J (se barajan las cartas)
REPEAT 2	repite 2 veces la siguiente secuencia de comandos
GENERATE 1 1,10 B	genera un número aleatorio entre 1 y 10; coloca el resultado en la variable B (se piensa un número entre uno y diez)
TAKE J B C	selecciona del vector J el elemento ubicado en la posición que indica el valor de la variable B y coloca el resultado en la variable C (nos fijamos en el número de la carta que ocupa la posición del número pensado)
ADD B C D	suma los valores de las variables B y C; coloca el resultado en la variable D (suma el número pensado con el número de la carta que ocupa la posición de ese número)

WHILE D<=156	las próximas órdenes se repiten mientras el valor de la variable D sea menor o igual a 156
TAKE J D C	selecciona del vector J el número indicado por el valor de la variable D y coloca el resultado en la variable C (de esa forma se van seleccionando las cartas según la regla)
ADD D C D	suma los valores de las variables D y C; coloca el resultado en la variable D
END	finaliza la condicional
SCORE C Z	agrega el valor de la variable C en el vector Z (registra la última de las cartas observadas)
END	finaliza el primer ciclo y repite las órdenes anteriores nuevamente (el mago sigue la misma regla)
MULTIPLES Z>1 ADIVINO	cuenta cuántas veces un mismo número aparece repetido más de una vez en el vector Z y coloca el resultado en la variable ADIVINO (se determina si las dos cartas coinciden)
SCORE ADIVINO P	agrega el valor de la variable ADIVINO en el vector P
CLEAR Z	limpia el contenido del vector Z (de manera que al repetir el nuevo experimento éste no contenga ningún valor)
END	termina el primer experimento y repite el ciclo anterior hasta que se completen las 10 000 simulaciones
COUNT P=1 T	cuenta cuántos unos hay en el vector P y coloca el resultado en la variable T (se computa en cuántos de los experimentos las series convergieron)
DIVIDE T 10000 PRO	divide el valor de la variable T entre 10 000 y coloca el resultado en la variable PRO (se estima la probabilidad deseada)

PRINT PRO muestra en pantalla el contenido de la variable PRO

En este proceso, simulando 10 000 veces, las series convergieron 9718 veces (solo en 82 ocasiones el mago y su "víctima" concluyen con cartas diferentes). Ello representa una probabilidad estimada de 0.97.

Con el programa es muy fácil corroborar que la probabilidad de éxito es aproximadamente igual a 0.70 si se usa un solo mazo, a 0.91 si se emplean dos y a 0.98 si se usaran 4. Así, pueden manejarse fácilmente otras variantes; por ejemplo, si el número inicial se elige entre 1 y 6 con tres mazos de cartas la probabilidad de éxito del "mago" se modifica imperceptiblemente (aproximadamente 0.98).

4.2 Intervalos de confianza para un indicador que no tiene fórmula analítica para calcular su error muestral.

En muy variadas situaciones, nos vemos obligados a construir indicadores para aquilatar cierta realidad; así, por ejemplo, se construyen indicadores que miden eficiencia de los servicios hospitalarios, calidad de vida, diferencia de género etc. Cuando trabajamos con una muestra, más que la estimación puntual que nos aporta ese nuevo indicador creado, nos interesa construir intervalos de confianza para estimar el parámetro en cuestión.

En limitadas situaciones el cálculo del error muestral de esos nuevos indicadores, mediante el empleo de formulaciones teóricas, es sencillo o posible. Lo habitual es que ocurra todo lo contrario. En éstas situaciones el método de *remuestreo*, particularmente el *bootstrap*, no tiene parangón en cuanto a sencillez y posibilidades, el siguiente ejemplo demuestra tal afirmación.

Supongamos que en el servicio de quemados del hospital docente "V.I. Lenin" ingresaron 458 pacientes en el año 2002 y 350 en el 2003. Supongamos además que se quiere determinar si la mortalidad, en dicho

servicio, se ha comportado de forma similar en ambos años; para ello, se seleccionan aleatoriamente 40 pacientes ingresados en ambos años (20 para el año 2002 y 20 para el 2003). Llamaremos *y* a la variable dicotómica:

$$y = \begin{cases} 1 \; si \; egresó \; muerto \\ 0 \; si \; egresó \; vivo \end{cases}$$

y llamemos *p* a la que mide la probabilidad de morir en el momento del ingreso, valor que sale de un modelo validado de regresión logística. Dicha probabilidad puede interpretarse obviamente como una medida de la gravedad con que ingresa un paciente. Los resultados pueden verse en la Tabla 4.2.

Tabla 4.2: Estado de los pacientes ingresados según su situación al egreso y gravedad al ingreso. Hospital docente "V.I. Lenin". Años 2002-2003.

Pacientes	**AÑO 2002**		**AÑO 2003**	
	y	**p**	**y**	**p**
1	1	0.97	1	0.6
2	0	0.03	0	0.3
3	1	0.9	1	0.77
4	0	0.06	0	0.56
5	1	0.88	0	0.43
6	0	0.09	0	0.2
7	1	0.92	1	0.03
8	1	0.98	0	0.88
9	1	0.66	1	0.77
10	1	0.77	1	0.9
11	0	0.02	0	0.8
12	0	0.4	0	0.43
13	0	0.22	0	0.67
14	1	0.91	1	0.03
15	0	0.04	0	0.9
16	1	0.55	1	0.05
17	1	0.78	1	0.98
18	0	0.1	1	0.65
19	0	0.12	1	0.23
20	0	0.1	0	0.4

es fácil percatarse que la tasa de mortalidad en el servicio de quemados es igual en ambos años:

$$TM = \frac{1}{20}\sum_{i=1}^{n} y_i \bullet 100 = 50\% \qquad [1]$$

ya que en ambos años murieron 10 de los 20 pacientes. Sin embargo, afirmar que la mortalidad es idéntica puede ser considerado poco aceptable, pues en esa afirmación no se tiene en cuenta la gravedad de los pacientes al ingresar (si en determinado año ingresan pacientes más graves, cabe esperar que la mortalidad sea mayor aún cuando las condiciones de calidad en la atención no se hayan modificado). Para ello se ha propuesto el siguiente indicador (Silva, 1995):

$$TMAG = \frac{\sum_{i \in m} \frac{1}{p_i}}{\sum_{i \in t} \frac{1}{p_i}} \qquad [2]$$

donde:
m: es el subconjunto de pacientes fallecidos
t: es la muestra total.

Es fácil ver que:

[1] = [2] si p_i fuera constante para los n pacientes.

Si ahora utilizamos TMAG en lugar de TM obtenemos: 7,3% para el año 2002 y 80,6% para el 2003. Lo que indicaría que en el año 2002 se alcanzó una mortalidad (ajustada por gravedad) mucho menor que en el 2003, cuando se contempla la gravedad al ingreso de los pacientes. Pero, ¿podemos estar seguros que ese resultado es realmente así o es producto de la variabilidad aleatoria?

Para responder a esta pregunta, podríamos construir un intervalo de confianza para la diferencia de las TMAG entre ambos años, es decir:

$$DIFF_{TMAG} = TMAG_{2002} - TMAG_{2003}$$

si el intervalo de confianza así construido contiene al cero, no podríamos descartar la variabilidad aleatoria como responsable de la diferencia encontrada.

Analíticamente es muy difícil encontrar una expresión que resuelva esta interrogante, el *bootstrap*, en contraposición, nos devuelve la respuesta inmediatamente. El siguiente programa así lo demuestra:

Con fines de optimizar el proceso de programación, con los datos contenidos en la Tabla 4.2 se construyó un fichero tipo texto nombrado IC-TMAG

READ FILE "IC-TMAG" A B C D	Lee el fichero IC-TMAG donde se encuentra indicado el status de los pacientes al egreso (vivo o fallecido), así como la gravedad de los mismos; coloca los valores en los vectores A B C y D de forma tal que en los vectores A y C se encuentra el status de los pacientes al egreso y en los vectores B y D la gravedad expresada en términos probabilísticos.
GENERATE 20 1 INV	genera 20 números 1 y los coloca en el vector INV
DIVIDE INV B INVERSO1	divide los valores contenidos en el vector INV entre sus correspondientes valores del vector B; coloca el resultado en el vector INVERSO1. (calcula en inverso de la probabilidad de morir para los pacientes ingresados en el año 2002)
DIVIDE INV D INVERSO2	divide los valores contenidos en el vector INV entre sus correspondientes valores del vector D; coloca el resultado en el vector INVERSO2. (calcula en inverso de la probabilidad de morir para los pacientes ingresados en el año 2003)

REPEAT 1000	repite 1000 veces la secuencia de comandos subsiguientes
REPEAT 20	repite 20 veces la secuencia de comandos subsiguientes hasta el penúltimo comando END
GENERATE 1 1,20 T	genera un número aleatorio entre uno y 20; coloca el resultado en la variable T
TAKE A T PAR1	selecciona del vector A, el valor contenido en la posición indicada por la variable T y coloca el resultado en la variable PART1 (selecciona aleatoriamente un paciente ingresado en el año 2002)
TAKE C T PAR2	selecciona del vector C el valor contenido en la posición indicada por la variable T y coloca el resultado en la variable PART2 (selecciona aleatoriamente un paciente ingresado en el año 2003)
IF PAR1=1	si el valor contenido en la variable PAR1 es igual a 1(si el paciente seleccionado en el año 2002 falleció)
TAKE INVERSO1 T PROINV1	selecciona del vector INVERSO1 el número que se encuentra ubicado en la posición indicada por la variable T y coloca el resultado en la variable PROINV1 (selecciona el inverso de la probabilidad de muerte del paciente fallecido)
SCORE PROINV1 Z	agrega el valor de la variable PROINV1 en el vector Z (de esta forma se crea un vector que contendrá el inverso de la probabilidad de muerte de aquellos pacientes que fallecieron en el 2002)
END	finaliza la condicional
IF PAR2=1	si el valor contenido en la variable PAR2 es igual a 1 (si el paciente seleccionado en el año 2003 falleció)
TAKE INVERSO2 T PROINV2	selecciona del vector INVERSO2 el número que se encuentra ubicado en la posición indicada por la variable T y coloca el resultado en la variable PROINV2 (selecciona el inverso de la probabilidad de muerte del paciente fallecido)
SCORE PROINV2 Z2	agrega el valor de la variable PROINV2 en el vector Z2 (de esta forma se crea

	un vector que contiene el inverso de la probabilidad de muerte de aquellos pacientes que fallecieron en el año 2003)
END	finaliza la condicional
TAKE INVERSO1 T DISTRI1	selecciona del vector INVERSO1 el valor ubicado en la posición indicada por la variable T y coloca el resultado en la variable DISTRI1 (selecciona el inverso de la probabilidad de muerte de los pacientes incluidos en la muestra del año 2002)
SCORE DISTRI1 DISTRIB1	agrega el valor de la variable DISTRI1 en el vector DISTRIB1 (se crea un vector que contiene todos los inversos de las probabilidades de muerte de los pacientes seleccionados en la muestra del año 2002, ya sean egresados vivos o fallecidos)
TAKE INVERSO2 T DISTRI2	selecciona del vector INVERSO2 el valor ubicado en la posición indicada por la variable T y coloca el resultado en la variable DISTRI2 (selecciona el inverso de la probabilidad de muerte de los pacientes incluidos en la muestra del año 2003)
SCORE DISTRI2 DISTRIB2	agrega el valor de la variable DISTRI2 en el vector DISTRIB2 (se crea un vector que contiene todos los inversos de las probabilidades de muerte de los pacientes seleccionados en la muestra del año 2003, ya sean egresados vivos o fallecidos)
END	finaliza el ciclo definido por el comando REPEAT 20
SUM DISTRIB1 TOTALDI1	suma las coordenadas del vector DISTRIB1 y coloca el resultado en la variable TOTALDI1(suma el inverso de la probabilidad de muerte para todos los pacientes incluidos en la remuestra para el año 2002)
SUM Z SUMAPIN1	suma las coordenadas del vector Z y coloca el resultado en la variable SUMAPIN1 (suma el inverso de la probabilidad de muerte sólo para los pacientes fallecidos que se incluyeron en la remuestra para el año 2002)

DIVIDE SUMAPIN1 TOTALDI1 TMAG1 divide el valor de la variable SUMAPIN1 entre el valor de la variable TOTALDI1 y coloca el resultado en la variable TMAG1 (calcula el indicador de interés para el año 2002)

SUM DISTRIB2 TOTALDI2 suma las coordenadas del vector DISTRIB2 y coloca el resultado en la variable TOTALDI2 (suma el inverso de la probabilidad de muerte para todos los pacientes incluidos en la remuestra para el año 2003)

SUM Z2 SUMAPIN2 suma las coordenadas del vector Z2 y coloca el resultado en la variable SUMAPIN2 (suma el inverso de la probabilidad de muerte sólo para los pacientes fallecidos que se incluyeron en la remuestra para el año 2003)

DIVIDE SUMAPIN2 TOTALDI2 TMAG2 divide el valor de la variable SUMAPIN2 entre el valor de la variable TOTALDI2 y coloca el resultado en la variable TMAG2 (calcula el indicador de interés para el año 2003)

SUBTRACT TMAG1 TMAG2 DIFF resta los valores contenidos en las variables TMAG1 y TMAG2; coloca el resultado en DIFF (calcula la diferencia entre las tasas ajustadas para los dos años)

SCORE DIFF IC agrega el valor de la variable DIFF al vector IC (se genera una distribución simulada de diferencias entre las tasas ajustadas)

CLEAR Z limpia el contenido del vector Z

CLEAR DISTRIB1 limpia el contenido del vector DISTRIB1

CLEAR Z2 limpia el contenido del vector Z2

CLEAR DISTRIB2 limpia el contenido del vector DISTRIB2

END finaliza el ciclo definido por el comando REPEAT 1000

PERCENTILE IC (2.5 97.5) P calcula los percentiles 2.5 y 97.5 de la distribución de valores contenidos en el vector IC y coloca el resultado en la variable P (se estima un IC para la diferencia $TMAG_{2002} - TMAG_{2003}$)

PRINT P muestra en pantalla el resultado de la variable P

Luego de correr éste programa obtuvimos un intervalo de confianza definido por los límites -0.84969 y -0.37447; ahora sí podemos concluir que: tenemos una confianza del 95% que la diferencia entre las tasas de mortalidad ajustadas según la gravedad de los pacientes al ingreso ($TMAG_{2002} - TMAG_{2003}$) es distinta de 0, ya que éste intervalo se obtuvo por un método que el 95% de las veces logra incluir el verdadero valor del parámetro estimado.

Si nuestro objetivo se hubiese centrado, por ejemplo, en calcular un intervalo de confianza del 95% para la estimación de la $TMAG_{2002}$, entonces con un programa similar (Anexo 3) podríamos determinar que el verdadero valor de la $TMAG_{2002}$ está ubicada, con un 95% de confianza, entre un 2.44% y un 18.31%.

4.3 Uso del *remuestreo* para valorar el desempeño de procedimientos estadísticos alternativos.

Una situación que se presenta con frecuencia en la investigación, especialmente en el ámbito metodológico, es aquella en la cual se tiene que escoger, entre varios, el mejor procedimiento para abordar un problema.

Un ejemplo en el marco de la estadística es el siguiente: si deseáramos estimar un parámetro, y hubiera varias alternativas, tendríamos que escoger el estadígrafo que en términos generales nos devuelva el valor más cercano a dicho parámetro. Para tal efecto, podemos seguir la estrategia de calcular el llamado Error Cuadrático Medio (ECM), que no es más que el promedio de las diferencias al cuadrado entre las estimaciones y el verdadero valor del parámetro estimado; de manera que aquel estimador que exhiba el menor ECM será el que mejores resultados nos brinde.

Más formalmente, si q es cierto parámetro y $\tilde{q}$ un estimador, tenemos:

$$ECM(\tilde{q}) = E(\tilde{q} - q)^2$$

El cómputo analítico de ese indicador es, ocasionalmente, posible y sencillo (por ejemplo, cuando q es una media poblacional y $\tilde{q}$ es la media muestral); pero por lo general, ese no es el caso. Por tal motivo, el empleo de la simulación es imprescindible en este tipo de situaciones. Veamos con un ejemplo, deliberadamente construido de forma tal que sabemos de antemano cuál es el mejor estimador (la media muestral), la validez de tal procedimiento.

Supongamos que un investigador desea conocer la media de una población a través de una muestra de tamaño 100 seleccionada aleatoriamente, y que para ello cuenta con las siguientes seis medidas de tendencia central, candidatas todas a ser usadas como estimador de la media:

1. Media muestral
2. Media geométrica
3. Media armónica
4. Mediana
5. El promedio entre el 25 y el 75 percentil (promedio intercuartil)
6. El promedio entre el máximo y el mínimo valor de la muestra

El problema consiste en determinar cuál de los posibles procedimientos de estimación tiene un menor ECM. La solución analítica exacta para los estimadores del 2 al 6 es extremadamente compleja.

Para resolver ésta incógnita, por conducto de la simulación, generamos una población con parámetros conocidos y seleccionamos n muestras. En cada una de ellas, estimamos el parámetro de interés con los 6 estadígrafos. Así obtendremos n estimaciones de la media poblacional con cada uno de los estimadores; con esos datos podemos estimar entonces los correspondientes ECM.

La solución del problema puede depender de las características de la población. El siguiente programa, confeccionado mediante Resampling Stats, es un ejemplo de cómo procederíamos con los 6 estimadores en el caso particular en que fijamos que la distribución es Normal con media 25 y desviación estándar igual a 4, que el número de simulaciones es 1000, en tanto que el tamaño muestral es 100.

REPEAT 1000	repite el experimento 1000 veces
NORMAL 100 25 4 A	genera una distribución Normal de tamaño 100 con media 25 y desviación estándar 4; coloca el resultado en el vector A. (simula una muestra de tamaño 100 de una población infinita con esos parámetros)
MEAN A X	calcula la media del vector A y coloca el resultado en la variable X
MEDIAN A MEDIANA	calcula la mediana del vector A y coloca el resultado en la variable MEDIANA
MAX A MAXIMO	calcula el máximo valor contenido en el vector A y coloca el resultado en la variable MAXIMO
MIN A MINIMO	calcula el valor mínimo de los valores contenidos en el vector A y coloca el resultado en la variable MINIMO
ADD MAXIMO MINIMO SUMA	suma los valores contenidos en los vectores MAXIMO Y MINIMO y coloca el resultado en la variable SUMA
DIVIDE SUMA 2 PMAXMIN	divide el valor de la variable SUMA entre 2 y coloca el resultado en la variable PMAXMIN

PERCENTILE A (25 75) P	calcula los percentiles 25 y 75 de la distribución de valores del vector A y coloca el resultado en la variable P
SUM P SUMAP	suma los valores contenidos en la variable P y coloca el resultado en la variable SUMAP
DIVIDE SUMAP 2 PIC	divide la variable SUMAP entre 2 y coloca el resultado en la variable PIC
GENERATE 100 1 INV	crea el vector INV de dimensión 100 cuyas coordenadas son iguales a la unidad
DIVIDE INV A INVERSO	divide cada valor del vector INV entre los valores del vector A y coloca el resultado en el vector INVERSO (calcula el inverso de la muestra seleccionada)
SUM INVERSO SUMA	suma el contenido del vector INVERSO y coloca el resultado en la variable SUMA (suma de los inversos)
DIVIDE SUMA 100 J	divide el contenido de la variable SUMA entre 100 y coloca el resultado en la variable J (calcula en promedio de los inversos)
DATA (1) R	coloca un 1 en el vector R
DIVIDE R J MARMONI	divide el valor del vector R entre el contenido de la variable J y coloca el resultado en MARMONI (inverso del promedio de los inversos; de esa forma se obtiene la media armónica
LOG A LOGXI	calcula el logaritmo de cada valor contenido en el vector A y coloca el resultado en el vector LOGXI
SUM LOGXI F	suma el contenido del vector LOGXI y coloca el resultado en la variable F
DIVIDE F 100 K	divide la coordenada de la variable F entre 100 y coloca el resultado en la variable K
EXP K MGEOMET	calcula la exponencial del valor de la variable K y coloca el resultado en la variable MGEOMET (se obtiene la media geométrica)
SCORE X Z	agrega el valor de la variable X en el vector Z (de forma tal que se originará, al finalizar las 1000 simulaciones, una distribución de medias estimadas)

SCORE MEDIANA Q	agrega el valor de la variable MEDIANA en el vector Q (de forma tal que se originará, al finalizar las 1000 simulaciones, una distribución de las medianas estimadas)
SCORE PMAXMIN W	agrega el valor de la variable PMAXMIN en el vector W (de forma tal que se originará, al finalizar las 1000 simulaciones, una distribución del promedio estimado entre el valor máximo y mínimo)
SCORE PIC M	agrega el valor de la variable PIC en el vector M (de forma tal que se originará, al finalizar las 1000 simulaciones, una distribución de la estimación del promedio Inter. Cuartil)
SCORE MARMONI PO	agrega el valor de la variable MARMONI en el vector PO (de forma tal que se originará, al finalizar las 1000 simulaciones, una distribución de las medias armónicas estimadas)
SCORE MGEOMET TO	agrega el valor de la variable MGEOMET en el vector Q (de forma tal que se originará, al finalizar las 1000 simulaciones, una distribución de las medias geométricas estimadas)
END	finaliza el primer ciclo y se repite el proceso hasta completadas las mil simulaciones
GENERATE 1000 25 B	crea el vector B de dimensión 1000 cuyas coordenadas son iguales a 25 (el verdadero valor del parámetro estimado)
SUBTRACT Z B DIFF	resta las coordenadas de los vectores Z y B; coloca el resultado en el vector DIFF (desviaciones de los valores de las medias estimadas respecto al verdadero valor del parámetro)
SQUARE DIFF CUADA	eleva al cuadrado cada valor contenido en el vector DIFF y coloca el resultado en el vector CUADA (cuadrado de las diferencias)
SUM CUADA CUADA	suma los valores contenidos en el vector CUADA y coloca el resultado en la variable CUADA (suma las diferencias al cuadrado)

DIVIDE CUADA 1000 ECMX	divide el contenido de la variable CUADA entre 1000 y coloca el resultado en la variable ECMX (promedio de las diferencias al cuadrado; se obtiene el ECM para la media
SUBTRACT Q B DIFF	resta las coordenadas de los vectores Q y B, coloca el resultado en el vector DIFF (desviaciones de los valores de las medianas estimadas respecto al verdadero valor del parámetro)
SQUARE DIFF CUADA	eleva al cuadrado cada valor contenido en el vector DIFF y coloca el resultado en el vector CUADA (cuadrado de las diferencias)
SUM CUADA CUADA	suma los valores contenidos en el vector CUADA y coloca el resultado en la variable CUADA (suma las diferencias al cuadrado)
DIVIDE CUADA 1000 ECMMEDIA	divide el contenido de la variable CUADA entre 1000 y coloca el resultado en la variable ECMMEDIA (promedio de las diferencias al cuadrado; se obtiene el ECM para la mediana)
SUBTRACT W B DIFF	resta las coordenadas de los vectores W y B; coloca el resultado en el vector DIFF (desviaciones de los valores estimados para el promedio entre el máximo y el mínimo respecto al verdadero valor del parámetro)
SQUARE DIFF CUADA	eleva al cuadrado cada valor contenido en el vector DIFF y coloca el resultado en el vector CUADA (cuadrado de las diferencias)
SUM CUADA CUADA	suma los valores contenidos en el vector CUADA y coloca el resultado en la variable CUADA (suma las diferencias al cuadrado)
DIVIDE CUADA 1000 ECMPMAXI	divide el contenido de la variable CUADA entre 1000 y coloca el resultado en la variable ECMPMAXI (promedio de las diferencias al cuadrado; se obtiene el ECM para el promedio entre el máximo y el mínimo)

SUBTRACT M B DIFF	resta las coordenadas de los vectores M y B; coloca el resultado en el vector DIFF (desviaciones de los valores del promedio Inter cuartil estimado respecto al verdadero valor del parámetro)
SQUARE DIFF CUADA	eleva al cuadrado cada valor contenido en el vector DIFF y coloca el resultado en el vector CUADA (cuadrado de las diferencias)
SUM CUADA CUADA	suma los valores contenidos en el vector CUADA y coloca el resultado en la variable CUADA (suma las diferencias al cuadrado)
DIVIDE CUADA 1000 ECMPIC	divide el contenido de la variable CUADA entre 1000 y coloca el resultado en la variable ECMPIC (promedio de las diferencias al cuadrado; se obtiene el ECM para el PIC)
SUBTRACT PO B DIFF	resta las coordenadas de los vectores PO y B; coloca el resultado en el vector DIFF (desviaciones de los valores de la media armónica estimada respecto al verdadero valor del parámetro)
SQUARE DIFF CUADA	eleva al cuadrado cada valor contenido en el vector DIFF y coloca el resultado en el vector CUADA (cuadrado de las diferencias)
SUM CUADA CUADA	suma los valores contenidos en el vector CUADA y coloca el resultado en la variable CUADA (suma las diferencias al cuadrado)
DIVIDE CUADA 1000 ECMARMON	divide el contenido de la variable CUADA entre 1000 y coloca el resultado en la variable ECMARMON (promedio de las diferencias al cuadrado; se obtiene el ECM para la media amónica)
SUBTRACT TO B DIFF	resta las coordenadas de los vectores TO y B; coloca el resultado en el vector DIFF (desviaciones de los valores de las medias geométricas estimadas respecto al verdadero valor del parámetro)

SQUARE DIFF CUADA	eleva al cuadrado cada valor contenido en el vector DIFF y coloca el resultado en el vector CUADA (cuadrado de las diferencias)
SUM CUADA CUADA	suma los valores contenidos en el vector CUADA y coloca el resultado en la variable CUADA (suma las diferencias al cuadrado)
DIVIDE CUADA 1000 ECMGEOME	divide el contenido de la variable CUADA entre 1000 y coloca el resultado en la variable ECMGEOME (promedio de las diferencias al cuadrado; se obtiene el ECM para la media geométrica)

PRINT ECMX ECMMEDIA ECMPMAXI ECMPIC ECMARMON ECMGEOME

Muestra en pantalla los valores respectivos para el ECM de todos los estimadores.

Al correr este programa, obtuvimos los siguientes resultados:

Estimadores	**ECM**
• Media muestral.	0.16
• Promedio entre el 25 y el 75 percentil	0.19
• Mediana	0.25
• Media geométrica	0.27
• Media armónica	0.63
• Promedio entre el máximo y el mínimo valor de la muestra	1.51

Con este resultado, podemos concluir no sólo que el mejor estimador en esta situación es la media muestral, ya que su ECM es el mínimo, sino que podemos ordenar a los seis candidatos de mejor a peor.

4.4 Interim Analysis

Cuando se decide conducir un ensayo clínico, es conveniente evitar que se incluyan más pacientes de los estrictamente necesarios. Al menos, dos tipos de razones justifican tal punto de vista:

- Razones económicas
- Razones éticas

Lo económico tiene importancia obvia, pues el gasto innecesario de recursos, tanto materiales como de tiempo, hace ineficiente el proceso de investigación bastante encarecido ya en la actualidad. Lo ético, por su parte, tiene gran trascendencia desde el punto de vista social y humano. En efecto, en algunos ensayos clínicos que se ejecutan, la eficacia del tratamiento experimental se hace evidente mucho antes de completar el proceso previsto por el protocolo. Continuar tratando, bajo estas condiciones, con placebo (u otra droga menos eficaz) a personas que podrían beneficiarse de las bondades del nuevo tratamiento es, cuando menos, muy cuestionable. Sin embargo, la violación de un protocolo es una práctica muy conflictiva por razones obvias (se confecciona exactamente para pautar lo que se debe de hacer)

Es por ello que resultan especialmente atractivos aquellos diseños que de antemano, en la etapa de planificación, determinan puntos intermedios en los que el investigador puede pronunciarse a favor (o en contra) de la eficacia del medicamento experimental.

Así, están adquiriendo carta de ciudadanía en la metodología de los ensayos clínicos los diseños en dos o más etapas, en cada una de las cuales el investigador tiene la posibilidad de parar el experimento si se alcanza cierto número de éxitos a favor del tratamiento experimental (o decididamente contrarios a él). Es decir, de lo que se trata es de realizar análisis intermedios (Interim analysis) con el propósito de adoptar de manera ágil una decisión respecto a la eficacia del producto experimental.

Sin embargo, este tipo de diseño tiene una gravísima dificultad: el cómputo de la famosa p con que operan las pruebas estadísticas es en extremo difícil y, como si fuera poco, varía en correspondencia al diseño planteado (Berger y Berry, 1998) tornándose cada vez más complejos en la medida que las etapas van incrementándose. El recurso de la simulación brinda excelentes resultados cuando es utilizado para este fin. Ilustrémoslo con un simple ejemplo.

Supongamos que un investigador desea conducir un ensayo clínico con el objetivo de evaluar la eficacia de una nueva droga en el tratamiento de la psoriasis y, luego de un análisis de factibilidad, determina seguir la siguiente estrategia:

En una primera etapa, se observarán 60 pacientes con psoriasis en ambos brazos. Para cada uno de ellos, se seleccionará, aleatoriamente, en cuál de los brazos se le aplicará el nuevo medicamento; en el otro se aplicará un placebo. El investigador piensa que si se obtienen 36 éxitos ó más en favor del tratamiento experimental se debe detener el ensayo y otorgarle licencia al nuevo medicamento. Si, por el contrario, el número de éxitos no supera a 35, entonces se pasaría a una segunda etapa en la cual se seleccionarán 50 nuevos pacientes que se manejarán de igual forma que en la primera etapa. Si tras esta nueva etapa se llegara a obtener 70 ó más éxitos (sumando los de la primera etapa con los de la segunda), entonces se declarará eficiente el nuevo tratamiento. De lo contrario, el estudio concluye.

La pregunta que de inmediato procede responder es ¿cuál es la probabilidad de declarar exitoso el nuevo tratamiento (36 ó más éxitos en la primera etapa ó 70 ó más en caso de que se haya efectuado la segunda) bajo el supuesto de que no existen diferencias entre ellos? Esto no es más que determinar la probabilidad de cometer el error de primer tipo que le corresponda a esta estrategia (es decir, determinar el valor a que de hecho se está asignando a la regla de decisión establecida). Por vía analítica, el cálculo de esta probabilidad es sumamente complejo.

Mediante simulación podemos proceder de la siguiente manera:

MAXSIZE DEFAULT 10000	aumenta las capacidades de los vectores hasta 10 000 caracteres
REPEAT 10000	repite 10 000 veces el experimento

GENERATE 60 1,2 A	genera 60 números aleatorios entre 1 y 2 y coloca el resultado en el vector A (simula con probabilidad ½ los posibles éxitos del nuevo tratamiento)
COUNT A=1 B	cuenta cuántos unos hay en el vector A y coloca el resultado en la variable B (en cuántos casos el nuevo tratamiento fue superior al placebo)
IF B>=36	si el valor contenido en la variable B es mayor o igual a 36 (si el número de éxitos es 36 o más)
SCORE 1 Z	agrega un uno en el vector Z
END	finaliza la condicional
IF B<36	si el valor contenido en la variable B es menor que 36 (sino se alcanzó 36 éxitos con la nueva droga)
GENERATE 50 1,2 D	genera 50 números aleatorios entre uno y dos y coloca el resultado en el vector D
COUNT D=1 B2	cuenta cuántos unos hay en el vector D y coloca el resultado en la variable B2 (éxitos del nuevo tratamiento en la segunda etapa)
ADD B B2 TOTAL	suma las coordenadas de las variables B y B2; coloca el resultado en la variable TOTAL (suma los éxitos obtenidos con el nuevo medicamento en las dos etapas)
IF TOTAL >=70	si el valor de la variable TOTAL es mayor o igual a 70 (si se obtuvieron 70 o más éxitos sumando las dos etapas)
SCORE 1 Z	agrega un uno en el vector Z (el nuevo medicamento se considera eficaz)
END	finaliza la primera condicional
END	finaliza la segunda condicional
END	finaliza el experimento

COUNT Z=1 T	cuenta cuántos unos hay en el vector Z y coloca el resultado en la variable T (cuenta en cuántos experimentos se declaró exitoso el nuevo medicamento)
DIVIDE T 10000 P	divide el valor de la variable T entre 10 000 y coloca el resultado en la variable P
PRINT P	muestra el valor de la variable P en pantalla

Luego de correr este programa, la probabilidad de declarar "exitoso" el nuevo medicamento bajo el supuesto de que H_0 sea verdadera resultó ser 0.077. Ello significa que el umbral para declarar significación en este diseño es a = 0.077.

Nótese que lo que se ha hecho es fijar una región crítica y encontrar luego la probabilidad asociada. Pero lo que, generalmente, se hace en la investigación es lo contrario: se fija un valor a (típicamente 0.05) y se encuentra la región crítica. Un programa que nos indique automáticamente cuál es el número de éxitos (región crítica) tal que al calcular el nivel de significación alcanzado en la prueba éste sea menor o igual a el a prefijado por el investigador, teniendo en cuenta cierto *modus operandi*, sería de gran utilidad a los investigadores. Veamos cómo proceder entonces apoyándonos en el ejemplo anterior.

Recordemos que de lo que se trataba era de pronunciarnos sobre la eficacia de una nueva droga para el tratamiento de la psoriasis; pero ahora no tenemos idea de cual podría ser el número de éxitos necesarios para tal pronunciamiento. En esta situación la estrategia consiste en fijar un umbral, digamos, 0.05 y un *modus operandi*, que podría ser el siguiente:

Se seleccionan 50 pacientes y se declara eficaz la nueva droga si se producen A éxitos ó más en favor de la misma. En caso contrario, se seleccionan otros 50 pacientes y si el número de éxitos, sumando los que se obtuvieron en la primera etapa y en la segunda, es 2A ó más se declara eficaz el nuevo medicamento.

La interrogante a responder es: ¿Cuál es el número mínimo de éxitos (A) a fijar para que la probabilidad de rechazar bajo H_0 sea 0.05?

Para ello, hemos desarrollado un programa que a partir de un valor inicial A_1 de posible éxito, encuentra automáticamente la región crítica mediante una estrategia muy simple: calcula el valor de p con A_1 éxitos. Si p es mayor que 0.05, incrementa el valor de A_1 en una unidad y se obtiene A_2; calcula nuevamente la p, sin no se alcanza un valor menor o igual a 0.05 se repite la estrategia hasta alcanzar el primero que logre dicho objetivo.

MAXSIZE DEFAULT 15000	aumenta las capacidades de los vectores hasta 15 000 caracteres
COPY (25) T	genera la variable T formada por una única coordenada: el número 25 (valor inicial del número de éxitos A_1, fijado por el investigador)
COPY (1) P	genera la variable P formada por una única coordenada: el número 1 (valor inicial de a)
WHILE P>0.05	mientras el valor de la variable P sea mayor que 0.05 (mientras que el valor de a sea mayor que 0.05)
MULTIPLY T 2 T2	multiplica el valor de la variable T por 2 y coloca el resultado en la variable T2 (de esta forma indicamos que el número de éxitos, para considerar la nueva droga eficaz, en la segunda etapa, es el doble que en la primera (2A))

REPEAT 15000	repite 15 000 veces el experimento
GENERATE 50 1,2 A	genera 50 números aleatorios entre 1 y 2 y coloca el resultado en el vector A (simula con probabilidad ½ los posibles éxitos del nuevo tratamiento)
COUNT A=1 B	cuenta cuántos unos hay en el vector A y coloca el resultado en la variable B (en cuántos casos el nuevo tratamiento fue superior al placebo)
IF B>=T	si el valor contenido en la variable B es mayor o igual al valor de la variable T (si el número de éxitos es mayor o igual a Ai)
SCORE 1 Z	agrega un uno al vector Z
END	finaliza la condicional
IF B<T	si el valor contenido en la variable B es menor que el contenido en la variable T (sino se alcanzó Ai éxitos con la nueva droga en la primera etapa)
GENERATE 50 1,2 D	genera 50 números aleatorios entre uno y dos y coloca el resultado en el vector D (se seleccionan 50 nuevos pacientes y se obtiene el resultado de los tratamientos)
COUNT D=1 B2	cuenta cuántos unos hay en el vector D y coloca el resultado en la variable B2 (éxitos del nuevo tratamiento en la segunda etapa)
ADD B B2 TOTAL	suma las coordenadas de las variables B y B2; coloca el resultado en la variable TOTAL (suma los éxitos obtenidos con el nuevo medicamento en las dos etapas)
IF TOTAL >=T2	si el valor de la variable TOTAL es mayor o igual al de la variable T2 (si se obtuvieron 2A, o más éxitos sumando las dos etapas)

SCORE 1 Z	agrega un uno en el vector Z (el nuevo medicamento se considera eficaz)
END	finaliza la primera condicional
END	finaliza la segunda condicional
END	finaliza el experimento
COUNT Z=1 Q	cuenta cuántos unos hay en el vector Z y coloca el resultado en la variable Q (cuenta en cuántos experimentos se declaró exitoso el nuevo medicamento).
DIVIDE Q 15000 P	divide el valor de la variable Q entre 15000 y coloca el resultado en la variable P.
SCORE P RESULTA	agrega el valor de la variable P en el vector RESULTA (este vector estará formado por los valores calculados de a)
CLEAR Z	limpia el vector Z, de forma que no este ocupado por ningún valor de ser necesario otro ciclo.
ADD 1 T T	suma un uno a la variable T y coloca el resultado en la propia variable T (se incrementa en una unidad el valor inicial de A).
END	finaliza la condicional
PRINT RESULTA	muestra en pantalla el contenido de la variable RESULTA

Luego de correr el programa obtuvimos lo siguiente:

A	25	26	27	28	29	30	31	**32**
p	0.6724	0.5422	0.4027	0.2819	0.1802	0.1098	0.064	**0.032**

Por tanto, el valor de A en este caso particular es 32, ya que, como se observa, es el primer valor que produce una p menor o igual a 0.05. Ello implica que un investigador que se planté un diseño como el anterior para un nivel de significación de 0.05, tendría que fijar A en 32 si quiere realizar un ensayo eficiente en relación con el número mínimo de sujetos a incluir.

Como hemos podido comprobar el *remuestreo*, de manera sencilla y muy intuitivamente, ofrece enormes ventajas al investigador ávido de soluciones a no pocos problemas que surgen antes y durante el proceso de investigación, ayudando incluso a romper la rigidez metodológica que impera en la investigación actual.

CONSIDERACIONES FINALES

Luego de haber realizado una incursión al método de *remuestreo*, procede detenerse a reflexionar sobre algunos aspectos que a nuestro juicio resultan vitales tener presente.

Como resultado del manejo transparente de los datos, donde no intervienen resultados teóricos ajenos a los datos mismos, **el *remuestreo* resulta ser un método sencillo y altamente intuitivo**. Este hecho facilita la solución de problemas en el marco de la teoría de probabilidades y ayuda a la comprensión de la inferencia facilitando el proceso de aprendizaje de la misma.

Durante la realización de este trabajo hemos podido comprobar que **los resultados obtenidos mediante el uso del *remuestreo* en la solución de problemas clásicos, tanto de inferencia como de probabilidades, son virtualmente idénticos a los obtenidos mediante los métodos tradicionales.** En principio, tal resultado parece sorprendente. De hecho, históricamente el *remuestreo* había sido motivo de suspicacias, ya que se pensaba que se trataba de un "*recocinado*" de datos. Sin embargo, al meditar detenidamente sobre el asunto, podemos percatarnos de que ambos métodos (remuestreo – métodos tradicionales) operan con exactamente la misma información (aquella contenida en la muestra observada) solo que manejada con enfoques radicalmente diferentes: el *remuestreo* hace uso literal de las ideas básicas de la estadística (por ejemplo, el error muestral de un estadístico se obtiene calculando la desviación estándar de una distribución simulada de dicho estadístico) y los métodos tradicionales se apoyan en supuestos teóricos que facilitan la derivación analítica de dichas distribuciones usando la información muestral en combinación con el teorema central del límite; en definitiva se trata de procesos diferentes (uno no paramétrico y otro paramétrico) aplicados a la misma información empírica.

Este hecho no debe conducirnos a pensar que el remuestreo es un método surgido para sustituir los procedimientos habituales de la estadística, más bien **debe ser visto como una útil alternativa complementaria, especialmente en situaciones donde dichos procedimientos son inaplicables** (situaciones no tan infrecuentes como se pudiera pensar).

Contar con un software como Resampling Stats es un verdadero privilegio para cualquier estadístico. Por su sencillez, se puede llegar a alcanzar, en un lapso breve, un pleno dominio del mismo, **lo cual habilita al usuario de una herramienta poderosísima para resolver, mediante el método de *remuestreo*, problemas tanto en el contexto de la inferencia estadística, como en el marco de la teoría de probabilidades cuyas derivaciones analíticas sean difíciles ó imposibles mediante los recursos tradicionales.**

El desarrollo y prestigio alcanzado en materia de investigación y el alto nivel técnico de nuestros profesionales, son poderosas motivaciones para realizar **recomendaciones** que deberían ser tenidas en cuenta en relación con el *remuestreo* en nuestro país:

- **ü** Que el aparato formativo de los bioestadísticos incorpore acciones orientadas a divulgar el conocimiento de las técnicas de *remuestreo*.

- **ü** Incluir un módulo en la residencia de bioestadística que trate éste método. Además de brindarle al estudiante una herramienta útil y en muchas situaciones la única disponible para resolver diversos problemas, lo proveerá de un recurso muy útil para comprender los procedimientos convencionales y las nociones relacionadas con la teoría de las probabilidades.

BIBLIOGRAFÍA

Allen RC, Bottcher C, Bording P, et all. (1995). Introduction to Monte Carlos Methods. Computational Science Education Project. URL disponible en http://csep1.phy.ornl.gov/mc/mc.htm

Arcos CA, et all (2002). Intervalos de confianza alternativos para los cuantiles de una población finita. Estadística Española. 44 (149): 69-88.

Aspuru GA, Lester WA (2002). Quantum Monte Carlo methods for the solution of the Shroedinger equation for molecular systems. Handbook of Numerical Analysis. l7.URL disponible en http://alan.aspuru.com/publications.php

Aspuru GA, Perusquía-Flores RA (1999). Monte Carlo Cuántico: Desarrollo de un paquete de software educativo y de investigación. [Tesis Doctoral]. UNAM. México. URL Disponible en http://alan.aspuru.com/publications.php

Becker R, Chambers J, Wilks A (1988). The S language. Wadsworth: Belimont CA.

Berger JO, Berry DA (1998). Statistical Analysis and the Illusion of Objectivity. American Scientist. 76: 159-165.

Berger VW (2002). Improving the Information Content of Endpoints in Clinical Trials. Controlled Clinical Trials. 23: 1-13.

Berger VW, Ivanova A (2001). Chapter 14. Permutation Tests for S-PLUS for Phase III Clinical Trials. In Millard SP, Krause A (eds). Applied Statistics in the Pharmaceutical Industry with Case Studies Using S-PLUS. New York: Springer-Verlag. 349-374.

Bose A (1990). Bootstrap in Moving Average Models. Annals of the Institute of Statistical Mathematics. 42: 753-768.

Buhlmann P (1997). Sieve Bootstrap for Time Series. Bernoulli. 3: 123-48.

Carlstein EDO, Hall P, Hesterberg T y Kunsch HR (1998). Matched-block Bootstrap for Dependent Data. Bernoulli. 4: 305-328.

Corcoran CD, Mehta CR (2002). Exact level and power of permutation, bootstrap and asymptotic tests of trend. Journal of Modern Applied Statistical Methods. 1: 42-51.

Cung JH, Fraser DAS (1958). Randomization Tests for Two-Sample Problem. Journal of the American Statistical Association. 53: 29-35.

Dawass M (1957). Modified Randomization Test for Nonparametric Hypotheses. Annals of Mathematical Statistics. 29: 181-187.

Efron B, Tibshirani RJ (1986). Bootstrap methods for standards error, confidence intervals, and other measures of statistical accuracy. Statistical Science. 1(1): 54-77.

Efron B, Tibshirani RJ (1993). An Introduction to the Bootstrap. New York: Chapman & Hall.

Efron, B (1979). Bootstrap Methods: Another Look at the Jackknife. The Annals of Statistics. 7: 15-30.

Fisher NI, Hall P (1990). On bootstrap hypothesis testing. Austral. J. Statistist. 32: 177-190.

Fisher RA (1935). The Design of Experiments. London: Oliver and Boyd.

García RLM, Sánchez PEA, Martines GF (1999). Dos ejemplos de introducción de los métodos de simulación Monte Carlo en los primeros cursos de las carreras técnicas. Lecturas Matemáticas. 20: 39-60.

Garfield J (1991). Reforming the Introductory Statistics Course. American Educational Research. Association Annual Meeting. Chicago.

Good, P. (2001). Resampling methods. (2nd ed.). Boston: Birkhauser.

Hinkley DW (1988). Bootstrap Methods. Journal of tie Royal Statistical Society, Series B. 50: 321-337.

Huff D (1959). How to take a Chance: New York: W.W. Norton.

Huntsberger DV (1983). Elementos de Estadística Inferencial. Mexico: Continental. 404.

Kotz S, Johnson N L (1992). Breakthroughs in Statistics. Volumes I and II. New York: Springer-Verlag.

Kreiss JP y Franke J (1992). Bootstrapping Stationary Autoregressive Moving Average Models. Journal of Time Series Analysis. 13: 297-317.

Lacourly N (2000). Una Pequeña Historia de la Estadística. Departamento de Ingeniería Matemática. Facultad de Ciencias Físicas y Matemáticas. Universidad de Chile. URL disponible en http://www.dim.uchile.cl

Liu RY y Singh K (1992). Moving Blocks Jackknife and Bootstrap Capture Weak Dependence. In: Exploring the Limits of Bootstrap, R. Lepage y L. Billard eds. New York: Wiley. 225-248

Lunneborg CE (1987). Bootstrap Application for the Behavioral Sciences. Seattle. University of Washington.

Lunneborg CE (2000). Data analysis by resampling: Concepts and applications. Pacific Grove. CA. Duxbury.

Molinero LM (2002). Métodos Autosuficientes de estimación y contraste de hipótesis. Utilización de la simulación y el método de Monte Carlo en Bioestadística. URL disponible en http:www.seh-lelha.org/stat1.htm

Mooney CZA, Duval RD (1993). Bootstrapping: A Nonparametric Approach to Statistical Inference. Newbury Park, CA: Sage.

Peter I (1991). Pick a Sample. Science News. (2): 23-35. URL disponible en: http://www.resample.com

Piattelli PM (1994). Inevitable Illusions. New York: Wiley.

Politis DN, Romano JF y Wolf M (1997). Subsampling for Heteroskedatic Time Series. Journal of Econometrics. 81: 281-317.

Rao JNK., Wu CFJ, Yue K (1992). Some recent work on resampling methods for complex surveys. Survey Methodology.18: 209-217.

Ricketts C, Berry J (1994). "Teaching Statistics Through Resampling". Teaching Statistics.6 (2), Summer: 41-44.

Romano JP (1989). Bootstrap an randomization test of some non-parametric hypothesis. Annals of Statistics. 17: 141-159.

Saunters DB, Trapp RG (1998). Bioestadística Médica. México: El Manual Moderno. 99-117.

Schroeder L (1974). Buffon's needle problem: An exciting application of many mathematical concepts. Mathematics Teacher. 67 (2): 183-186.

Simon J, Bruce P (1991). Resampling: A Tool for Everyday Statistical Work. Chance. 4 (1): 22-32.

Simon JL, Atkinson DT, y Shevokas C (1976). "Probability and Statistics: Experimental Results of a Radically Different Teaching Method". American Mathematical Monthly. 83 (9): 733-739.

Simon JV (1997). Resampling: The New Statistics. [Libro Electrónico]. 2ª ed. Arlington: Resampling Stats, Inc. URL disponible en http://www.resample.com.htm

Silva LC (1995). Excursión a la regresión logística en ciencias de la salud. Madrid: Díaz de Santos.

Solanas A, Sierra V (1992). Bootstrap: Fundamentos e Introducción a sus Aplicaciones. Anuario de Psicología. Nueva York. 55: 143-154.

Wu CFJ (1986). Jackknife, Bootstrap and Others Resampling Methods in Regression Analysis. The Annals of Statistics.14: 1261-1350.

Zupo ZL (2002). Historia de la Estadística. Universidad Continental de Ciencias e Ingeniería [Electrónico] 5 (26). URL disponible en http://www.continental.edu.pe/revista/articulos/historia_estadistica

ANEXOS

Anexo No 1: Problema de los tres dados.

MAXSIZE DEFAULT 15000	aumenta la capacidad de los vectores a 15000 plazas
REPEAT 15000	repite el experimento 15000 veces
GENERATE 1 1,6 JUGADOR	genera un número aleatorio entre uno y seis y coloca el resultado en la variable JUGADOR (el jugador apuesta a uno de los números)
GENERATE 3 1,6 DADOS	genera tres números aleatorios entre 1 y seis y coloca el resultado en el vector DADOS (se lanzan tres dados)
TAKE DADOS 1 A	toma del vector DADOS el número que ocupa la primera posición y lo coloca en la variable A
TAKE DADOS 2 B	toma del vector DADOS el número que ocupa la segunda posición y lo coloca en la variable B
TAKE DADOS 3 C	toma del vector DADOS el número que ocupa la tercera posición y lo coloca en la variable C
IF A=JUGADOR	si el número de la variable A coincide con el número contenido en la variable JUGADOR
SCORE 1 RESULT	agrega un 1 en el vector RESULT (el jugador acierta el número del primer dado)
END	finaliza
IF B=JUGADOR	si el número de contenido en la variable B coincide con el de la variable JUGADOR
SCORE 1 RESULT	agrega un 1 en el vector RESULT(el jugador acierta el número del segundo dado)
END	finaliza la condicional
IF C=JUGADOR	si el número de la variable C coincide con el de la variable JUGADOR
SCORE 1 RESULT	agrega un 1 en el vector RESULT (el jugador acierta el número del tercer dado)
END	finaliza la condicional
SIZE RESULT TOTAL	calcula el tamaño del vector RESULT y coloca el resultado en la variable TOTAL (en cuántas ocasiones el jugador acertó el número apostado)
SCORE TOTAL Z	agrega el resultado de la variable TOTAL en el vector Z

CLEAR RESULT	borra el contenido de la variable RESULT (para que este vacía al iniciar el otro ciclo)
END	termina el ciclo
COUNT Z=0 PERDI	cuenta cuántos 0 hay en el vector Z (simulaciones en las que el jugador perdió su dinero) y coloca el resultado en la variable PERDI
COUNT Z=1 GANA1	cuenta cuántos 1 hay en el vector Z (simulaciones en las que el jugador ganó la misma cantidad jugada) y coloca el resultado en la variable GANA1
COUNT Z=2 GANA2	cuenta cuántos 2 hay en el vector Z (número de simulaciones en las que el jugador adivinó dos veces el número jugado) y coloca el resultado en la variable GANA2
MULTIPLY GANA2 2 GANA2	multiplica el contenido de la variable GANA2 por 2 (calcula ha cuánto asciende el monto de la ganancia al adivinar dos números) y coloca el resultado en la propia variable
COUNT Z=3 GANA3	cuenta cuántos 3 hay en el vector Z (número de simulaciones en las que el jugador adivinó las tres veces el número jugado) y coloca el resultado en la variable GANA3
MULTIPLY GANA3 3 GANA3	multiplica el contenido de la variable GANA3 por tres (calcula a cuánto asciende el monto de la ganancia al adivinar los tres números) y coloca el resultado en la propia variable
ADD GANA1 GANA2 GANA3 GANANCIA	suma el contenido de las variables GANA1, GANA2 y GANA3; ubica el resultado en la variable GANANCIA (calcula el total de dinero ganado
SUBTRACT GANANCIA PERDÍ E	calcula la diferencia entre los valores de las variables GANANCIA y PERDÍ (diferencia entre lo ganado y lo perdido) coloca el resultado en la variable E
DIVIDE E 15000 V(E)	divide el contenido de la variable E entre 15000 (calcula la “ganancia” promedio); coloca el resultado en la variable V(E)
PRINT V(E)	muestra el resultado de la variable “V(E)” (el valor esperado de la “ganancia”).

RESULTADOS DE LA SIMULACIÓN

VECTOR NO. 1: Z

BIN CENTER	FREQ	PCT	CUM PCT
0	8687	57.9	57.9
1	5159	34.4	92.3
2	1077	7.2	99.5
3	77	0.5	100.0

NOTE: EACH BIN COVERS ALL VALUES WITHIN 0.1 OF ITS CENTER.

V(E)= -0.0762

Anexo 2: Prueba de Permutación.

URN 18#1 22#0 A	"construye" un vector llamado A, cuyo contenido esta formado por: 18 unos y 22 ceros (18 pacientes fallecidos y 22 sobrevivientes)
REPEAT 100	indica que se repita cien veces la secuencia de órdenes que subsigue
SHUFFLE A A	reordena aleatoriamente las ordenadas del vector A (en cada repetición se tendrá un nuevo ordenamiento)
TAKE A 1,20 N1	toma del vector A los valores que ocupan las posiciones de la uno a la veinte (muestra sin reemplazo del grupo experimental) y los coloca en el vector N1
TAKE A 21,40 N2	toma los restantes valores (muestra del grupo control) y los coloca en el vector N2
COUNT N1=1 FALLE_N1	cuenta el número de unos que hay en el vector n1 (los fallecidos en la remuestra del GE) y ubica el resultado en la variable FALLE_N1
COUNT N2=1 FALLE_N2	cuenta el número de unos que hay en el vector N2 (los fallecidos en la remuestra del GC) y ubica el resultado en la variable FALLE_N2
DIVIDE FALLE_N1 20 P1	calcula la proporción de unos contenidos en la variable FALLE_N1 y coloca el resultado en la variable P1 (proporción de fallecidos en la remuestra del GE)
DIVIDE FALLE_N2 20 P2	calcula la proporción de unos en la variable FALLE_N2 (proporción de fallecidos en la remuestra del GC); coloca el resultado en la variable P2
SUBTRACT P2 P1 DIF	calcula la diferencia entre la variable P2 y P1 y coloca el resultado en la variable DIF (diferencia de proporciones entre las dos remuestras)
SCORE DIF Z	registra el valor de la variable DIF en el vector Z (registra el valor de la diferencia P2-P1)
END	indica que se repita el proceso salvo que se hayan producido los cien ciclos indicados por REPEAT

COUNT Z>=0.4 T	cuenta cuántos valores mayores o iguales que 0.4 hay en Z, y coloca el resultado en la variable T (en cuántas realizaciones se encontró una diferencia al menos tan grande como la observada en el experimento original)
DIVIDE T 100 P	divide el valor de la variable T entre 100 y coloca el resultado en la variable P (calcula la famosa p)
PRINT P	muestra en pantalla el valor de la variable P

En este caso obtuvimos una probabilidad igual a 0.02.

Anexo 3: Intervalo de confianza para la $TMAG_{2002}$.

READ FILE "IC-TMAG2002" A B	lee el fichero "IC-TMAG2002" (previamente confeccionado en forma de texto) donde se encuentra el status de los pacientes al egreso y la gravedad de los mismos en términos probabilísticos; coloca dichos valores en los vectores A y B respectivamente
GENERATE 20 1 INV	genera 20 números 1 y los coloca en el vector INV
DIVIDE INV B INVERSO	divide los valores contenidos en el vector INV entre sus correspondientes valores del vector B; coloca el resultado en el vector INVERSO. (calcula en inverso de la probabilidad de morir para los pacientes ingresados en el año 2002)
REPEAT 1000	repite 1000 veces la secuencia de comandos subsiguientes
REPEAT 20	repite 20 veces la secuencia de comandos subsiguientes hasta el penúltimo comando END
GENERATE 1 1,20 T	genera un número aleatorio entre uno y 20; coloca el resultado en la variable T
TAKE A T PAR	selecciona del vector A el valor contenido en la posición indicada por la variable T y coloca el resultado en la variable PAR (selecciona aleatoriamente un paciente ingresado en el año 2002)
IF PAR=1	si el valor contenido en la variable PAR es igual a 1(si el paciente seleccionado en el año 2002 falleció)
TAKE INVERSO T PROINV	selecciona del vector INVERSO el número que se encuentra ubicado en la posición indicada por la variable T y coloca el resultado en la variable PROINV (selecciona el inverso de la probabilidad de muerte del paciente fallecido)
SCORE PROINV Z	agrega el valor de la variable PROINV en el vector Z (de esta forma se crea un vector que contendrá el inverso de la probabilidad de muerte de aquellos pacientes, de la remuestra, que fallecieron en el 2002)
END	finaliza la condicional

TAKE INVERSO T DISTRI	selecciona del vector INVERSO el valor ubicado en la posición indicada por la variable T y coloca el resultado en la variable DISTRI (selecciona el inverso de la probabilidad de muerte de los pacientes incluidos en la muestra del año 2002).
SCORE DISTRI DISTRIBU	agrega el valor de la variable DISTRI en el vector DISTRIBU (se crea un vector que contiene todos los inversos de las probabilidades de muerte de los pacientes seleccionados en la remuestra del año 2002, ya sean egresados vivos o fallecidos)
END	finaliza el ciclo definido por el comando REPEAT 20.
SUM DISTRIBU TOTALDIS	suma las coordenadas del vector DISTRIBU y coloca el resultado en la variable TOTALDIS(suma el inverso de la probabilidad de muerte para todos los pacientes incluidos en la remuestra para el año 2002)
SUM Z SUMAPIN	suma las coordenadas del vector Z y coloca el resultado en la variable SUMAPIN (suma el inverso de la probabilidad de muerte sólo para los pacientes fallecidos que se incluyeron en la remuestra para el año 2002).
DIVIDE SUMAPIN TOTALDIS TMAG	divide el valor de la variable SUMAPIN entre el valor de la variable TOTALDIS y coloca el resultado en la variable TMAG (calcula el indicador de interés para el año 2002)
SCORE TMAG IC	agrega el contenido de la variable TMAG al vector IC (se genera un vector con la distribución simulada de las TMAG estimadas para el año 2002)
CLEAR Z	limpia el contenido del vector Z.
CLEAR DISTRIBU	limpia el contenido del vector DISTRIBU
END	finaliza el ciclo definido por el comando REPEAT 1000
PERCENTILE IC (2.5 97.5) P	calcula los percentiles 2.5 y 97.5 de la distribución de valores contenidos en el vector IC y coloca el resultado en la variable P
PRINT P	muestra en pantalla el valore de la variable P

P= 0.0244- 0.1831