Sección de Matemáticas
Universidad de La Laguna

Shaina Daryanani Hassani

Contrastes de aleatoriedad: detección de patrones no aleatorios en muestras de datos

Randomness tests: detection of non-random patterns in data samples

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Junio de 2023

Dirigido por
Carlos González Alcón

Carlos González Alcón
Departamento de Matemáticas,
Estadística e Investigación
Operativa
Universidad de La Laguna
38200 La Laguna, Tenerife

Agradecimientos

Quisiera aprovechar esta oportunidad para expresar mi más sincero agradecimiento a todos aquellos que me han apoyado en la elaboración de mi TFG.

A mi tutor Carlos, quien me ha brindado su tiempo, experiencia, conocimientos y apoyo constante durante todo el proceso. Su asesoramiento ha sido fundamental para la realización de este trabajo. Agradezco su paciencia y dedicación para guiarme en este proyecto.

Quiero extender mi agradecimiento a mi familia y amigos, quienes han sido mi principal pilar de apoyo y motivación. Gracias por su amor, paciencia y confianza en mí.

Shaina Daryanani Hassani
La Laguna, 22 de mayo de 2023

Resumen · Abstract

Resumen

Este Trabajo de Fin de Grado se centra en el estudio de los contrastes de aleatoriedad y su aplicación en diferentes contextos. A través de un experimento de lanzamiento de monedas y el análisis de elecciones del Euromillones, se investiga si los eventos aparentemente aleatorios exhiben patrones ocultos o son verdaderamente impredecibles. Se adaptaron contrastes existentes y se desarrollaron nuevas metodologías para evaluar la aleatoriedad en situaciones de no equiprobabilidad y bidimensionalidad. La implementación en el lenguaje R permitió realizar análisis rigurosos y obtener conclusiones sobre la aleatoriedad de los eventos estudiados. En resumen, este trabajo contribuye a comprender la esencia de la aleatoriedad y detectar patrones en eventos que inicialmente parecen aleatorios.

Palabras clave: *Aleatoriedad – Contraste estadístico– Secuencia aleatoria – p-valor – Lenguaje R.*

Abstract

This Final Degree Project focuses on the study of randomness contrasts and their application in different contexts. Through a coin-tossing experiment and Euromillions election analysis, it investigates whether seemingly random events exhibit hidden patterns or are truly unpredictable. Existing contrasts were adapted and new methodologies were developed to assess randomness in situations of non-equiprobability and two-dimensionality. The implementation in the R language allowed rigorous analysis and conclusions about the randomness of the events studied. In summary, this work contributes to understanding the essence of randomness and detecting patterns in events that initially seem random.

Keywords: *Randomness – Statistical contrast – Random sequence – Implementation – p-value.*

Contenido

Introducción

El presente Trabajo de Fin de Grado tiene como objetivo abordar el tema de los contrastes de aleatoriedad, investigando la respuesta a la pregunta de si los eventos que consideramos aleatorios realmente exhiben este comportamiento impredecible o si, en cambio, se esconden patrones en resultados aparentemente aleatorios. Para ilustrar este dilema, se llevó a cabo un experimento en el que se solicitó a un grupo de alumnos que lanzaran diez veces una moneda. Luego, se les pidió que simularan el mentalmente 100 lanzamientos más y los escribieran en un formulario web que se les facilitó.

Lanzamiento moneda 10 veces	Simulación de 100 lanzamientos
cxxccccxcx	xcxcxcxccxxxxxcxcxcxcxxxcxxxcxxcxcxcx cxccxcxcxcxcxcxcxcxcxcxcxcxcxcxxccxcxc xcxcxcxcxcxcxcxcxcxcxcxcx

Figura 0.1. Lanzamientos de monedas (reales y simuladas) del alumno 1.

Ante este experimento, nos planteamos cómo de buenos somos simulando el azar, es decir, si seremos capaces de distinguir lanzamientos de monedas "físicos"de los mentales generados por los estudiantes. A partir de esta inquietud, surge la necesidad de utilizar los contrastes de aleatoriedad como una herramienta para analizar y evaluar la presencia de patrones en los lanzamientos simulados.

Es interesante recordar una anécdota que nos muestra cómo incluso en el mundo de la tecnología, la aparente aleatoriedad puede ser manipulada. Steve Jobs, el célebre cofundador de Apple, mencionó en una ocasión durante la presentación del iPod Shuffle: "Hemos hecho el shuffle del iPod menos aleatorio para que parezca más aleatorio". Esta declaración se da en respuesta a las quejas de algunos usuarios que experimentaron ocasiones en las que les salía una canción dos veces seguidas o canciones del mismo grupo muy frecuentemente.

En otra ocasión, planteamos una actividad en la que se les solicitó a los alumnos que completaran un boleto de Euromillones. El objetivo principal de esta actividad será analizar si las elecciones realizadas por los participantes son verdaderamente aleatorias o si revelan simetrías o patrones predecibles.

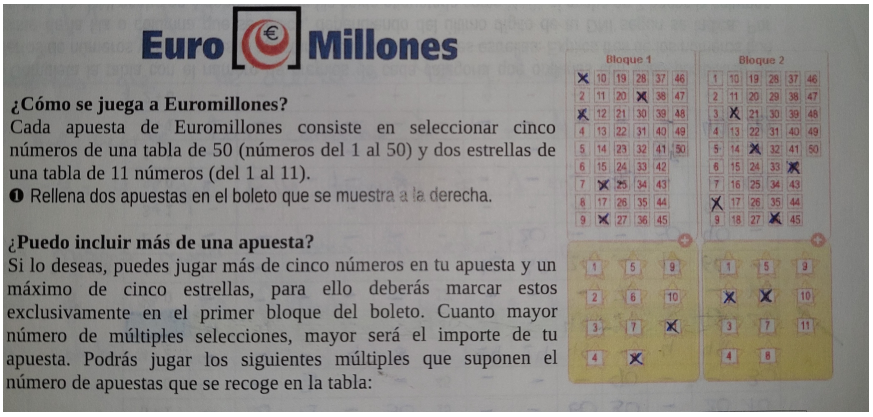

Figura 0.2. Actividad de Euromillones realizada al alumnado de 2º curso del Grado en Matemáticas. Ejercicio del alumno 13.

Mediante la aplicación de los contrastes, se busca explorar y examinar los datos recopilados. Los tests más clásicos que analizamos en los primeros capítulos están diseñados para estudiar datos unidimensionales, por lo que son adecuados para los lanzamientos de monedas de nuestro primer experimento. Sin embargo, cuando se trata de estudiar datos más complejos como el Euromillones, los contrastes tradicionales no son directamente aplicables debido a su naturaleza bidimensional.

En este contexto, además de proponer nuevas pruebas, se plantea la necesidad de adaptar algunos de los contrastes existentes para que funcionen en situaciones de no equiprobabilidad y bidimensionalidad, como las elecciones de los números en dicho juego de azar. Esta modificación nos permitirá evaluar la presencia de simetrías, la tendencia a esparcir lo números, la formación de estructuras en los números elegidos por los participantes, y determinar si los resultados son verdaderamente aleatorios o no. El ajuste de las pruebas para estas características es un desafío interesante que nos permitirá ampliar el alcance de estas herramientas y su aplicación en el análisis de eventos más complejos.

Este trabajo se divide en tres capítulos. En el capítulo 1, se exploran pruebas estadísticas para secuencias de números enteros. En el segundo, se abordan contrastes para evaluar la aleatoriedad en secuencias de números reales. Finalmente, en el tercer capítulo, se aplican estos métodos al ejemplo de Euromillones, adaptándolos a situaciones no equiprobables y bidimensionales.

1

Pruebas de aleatoriedad para secuencias de números enteros

Las secuencias aleatorias tienen un papel esencial en muchas áreas de la ciencia y la tecnología, ya que son utilizadas para simular eventos y procesos que ocurren en la realidad.

Por ejemplo, en el ámbito de la física, las secuencias aleatorias desempeñan un papel crucial en varios aspectos de la investigación y el análisis. Según Johnson en su libro 'Randomness in Physics: A Modern Perspective', las secuencias aleatorias son utilizadas en numerosos experimentos y simulaciones para introducir elementos de incertidumbre y reproducir fenómenos complejos y estocásticos (Johnson, 2018). Además, destaca que las secuencias aleatorias son particularmente relevantes en áreas como la mecánica estadística (modelación del comportamiento de sistemas físicos con múltiples partículas) y la teoría del caos (generación de trayectorias caóticas).

En el campo de la criptografía, las secuencias aleatorias juegan un papel fundamental para garantizar la seguridad de los sistemas de cifrado. Según Smith en su libro 'Cryptographic Foundations: Principles and Applications', las secuencias aleatorias son indispensables en el proceso de generación de claves criptográficas. En esta área, estas secuencias no solo se utilizan para generar claves seguras, sino que también son fundamentales para garantizar la resistencia frente a diversos ataques criptográficos.

El objetivo de los tests de aleatoriedad es asegurarse de que los números generados por diferentes procesos son realmente aleatorios y no tiendan a seguir cierto patrón. Estos tests generalmente se basan en la idea de que una secuencia aleatoria debería tener dos características, la uniformidad y la independencia entre sus elementos. Esto es, se debería poder considerar como la realización de una secuencia de variables aleatorias independientes que se distribuyen de manera uniforme en cierto conjunto (discreto o continuo).

Destacamos que se puede conseguir una secuencia lo suficientemente larga antes de comenzar a repetirse para que no sea rechazada por las pruebas. Por tanto, en nuestro caso, vamos a considerar como secuencia aleatoria a aquella

que ha conseguido superar los tests. Este criterio está basado en los que marcó el matemático Derrick Henry Lehmer:

1. Cualquier individuo que no conozca el proceso por el que se ha obtenido la sucesión debe ser incapaz de predecir cuál va a ser el siguiente elemento.
2. La sucesión debe pasar con éxito todas las pruebas de aleatoriedad mediante las que sea estudiada.

1.1. Prueba de bondad de ajuste: chi-cuadrado

La prueba chi-cuadrado comprueba si una muestra de datos se ajusta a una distribución discreta dada, teniendo en cuenta que la distribución muestral toma una cantidad discreta de posibles valores (o categorías) cada una con probabilidad p. Tendremos asociado el contraste de hipótesis

$$\begin{cases} H_0 \equiv & \text{Las frecuencias observadas se ajustan a la distribución esperada} \\ H_1 \equiv & \text{Otro caso.} \end{cases}$$

Para verificar que los datos se ajustan, comparamos para cada categoría el número de observaciones que pertenecen a ella, Y_k, con el número de observaciones esperado $n \cdot p_k$. Esto lo haremos tomando el estadístico

$$V = (Y_1 - np_1)^2 + \cdots + (Y_k - np_k)^2.$$

Sin embargo, nos fijamos que en el estadístico anterior estamos considerando que todas las categorías tienen el mismo peso, sin embargo, existen experimentos donde los diferentes valores tienen probabilidades diferentes de ser tomados.

Por ello, para poder contemplar los diferentes pesos, tendremos que modificar el estadístico anterior, quedando de la siguiente forma:

$$V = \frac{(Y_1 - np_1)^2}{np_1} + \cdots + \frac{(Y_k - np_k)^2}{np_k}.$$

Este estadístico, como demostró Karl Pearson en el año 1900, se distribuye como una chi-cuadrado de $k - 1$ grados de libertad. Una vez calculado su valor, nos iremos a la tabla de la distribución y buscaremos el valor crítico para dichos grados de libertad con la significación que nosotros decidamos. Finalmente, compararemos el valor del estadístico y el valor crítico para determinar si los datos se ajustan o no a esa distribución discreta.

Sin embargo, esta prueba tiene una limitación fundamental en cuanto al número mínimo de observaciones necesarias para producir resultados precisos y fiables. En particular, se suele considerar que chi-cuadrado no es apropiada

cuando el número esperado de observaciones en alguna de las categorías es menor que 5. Esto se debe a que su distribución es un resultado asintótico, por lo que se basa en la suposición de que cada categoría tiene un número suficientemente grande de observaciones para que la distribución sea aproximadamente normal. En el caso de que no se cumpla este requisito para alguna categoría, realizaremos un agrupamiento de categorías, es decir, se junta con la categoría adyacente o la más cercana, de manera que la frecuencia esperada resultante sea mayor o igual a 5.

Ejemplo. Se tiene una caja que contiene 6 bolas, entre las que encontramos dos rosas (R), una blanca (B) y tres negras (N). El experimento va a consistir en sacar 30 bolas con reemplazamiento. Nuestro objetivo será comprobar si nuestro resultado se ajusta a lo esperado.

En primer lugar, tenemos que obtener las frecuencias teóricas. Estas van a venir dadas de la siguiente forma:

- Rosas: $30 \cdot \frac{2}{6} = 10$
- Blancas: $30 \cdot \frac{1}{6} = 5$
- Negras: $30 \cdot \frac{3}{6} = 15$

Nótese que la frecuencia esperada para cada categoría es mayor o igual que cinco, por lo que es viable aplicar chi-cuadrado considerando las tres categorías. Una vez realizado el experimento, hemos obtenido el siguiente resultado:

$$BNNNRRNNBBNBRRNNBNRNNBNNNRNRNRN$$

Si observamos la secuencia resultante, nos fijamos que las frecuencias observadas de cada color son:

- Rosas: 8 bolas.
- Blancas: 6 bolas.
- Negras: 16 bolas.

Para poder aplicar chi-cuadrado, necesitamos calcular el valor del estadístico V. En nuestro caso, obtenemos que su valor es

$$V = \frac{(8-10)^2}{10} + \frac{(6-5)^2}{5} + \frac{(16-15)^2}{15} = \frac{4}{10} + \frac{1}{5} + \frac{1}{15} = \frac{2}{3} = 0.6667$$

A continuación, debemos comparar con el valor crítico correspondiente de la distribución con $k = 3 - 1 = 2$ grados de libertad. En nuestro caso, se tiene que dicho valor es 0.103 (para $\alpha = 0.05$).

Como el valor del estadístico V es mayor que valor crítico, concluimos que tenemos que rechazar la hipótesis nula de lo que los resultados del experimento se ajustan a la distribución esperada.

1.2. Tests de aleatoriedad

Las pruebas de aleatoriedad son contrastes estadísticos usados para determinar si cierta muestra o conjunto de datos sigue un patrón o, por el contrario, puede calificarse de aleatoria.

En este apartado estudiaremos seis tests diferentes. Las secuencias sobre las que vamos a aplicar las diferentes pruebas están formadas por números enteros:

$$\langle Y_n \rangle = Y_0, Y_1, Y_2, \dots$$

Queremos verificar que sus elementos están distribuidos de manera independiente y uniformemente entre 0 y $d-1$.

1.2.1. Test Chi-Cuadrado (Chi-Cuadrado Test)

Una primera prueba que podemos aplicar a nuestra secuencia es la prueba chi-cuadrado, para ver si podemos considerar que su distribución se ajusta a una uniforme. Con ella no estamos comprobando la independencia.

Ejemplo. Dada la secuencia formada por los 100 primeros decimales del número π,

3141592653589793238462643383279502884197169399375105820974944592307816406286208998628034825342117067

vamos a considerar que todos los dígitos tienen iguales probabilidades de aparecer.

Como estamos asumiendo que todos los elementos tienen la misma probabilidad, tendremos 10 categorías (números entre 0 y 9) con probabilidad $\frac{1}{10}$. Por tanto, tenemos que la frecuencia esperada para cada categoría es $100 \cdot \frac{1}{10} = 10 > 5$.

A continuación, contaremos las frecuencias observadas para cada número.

Elemento	0	1	2	3	4	5	6	7	8	9
Frec. obs.	8	8	12	12	10	8	9	8	12	13

Por último, usando las frecuencias anteriores, obtenemos que el valor del estadístico chi-cuadrado será

$$V = \frac{(8-10)^2}{10} + \frac{(8-10)^2}{10} + \frac{(12-10)^2}{10} + \frac{(12-10)^2}{10} + \frac{(10-10)^2}{10} +$$

$$+ \frac{(8-10)^2}{10} + \frac{(9-10)^2}{10} + \frac{(8-10)^2}{10} + \frac{(12-10)^2}{10} + \frac{(13-10)^2}{10} = \frac{19}{5} = 3.8.$$

Si comparamos este valor con 3.325 (valor crítico para nueve grados de libertad y $\alpha = 0.05$), vemos que el estadístico es mayor. Por ello, rechazamos la hipótesis nula concluyendo que la secuencia no es aleatoria.

En lo que se refiere a su implementación computacional, utilizaremos la función ya implementada en el paquete estadístico de R.

```
> chisq.test(table(secuencia_pi(1000))

Chi-squared test for given probabilities

X-squared = 3.8, df = 9, p-value = 0.9241
```

1.2.2. Test de Series (Serial Test)

Este test consiste en coger todos los posibles pares que podemos formar con los elementos que contiene la secuencia y calcular la frecuencia con la que aparecen en la misma.

Posteriormente, aplicaremos chi-cuadrado con $k = d^2$ y $p_s = \frac{1}{d^2}$, pues d^2 es número de parejas diferentes que se pueden formar, teniendo en cuenta, que dentro de la misma pueden repetirse los elementos y que influye el orden (variaciones con repetición de d elementos tomados de 2 en 2).

No obstante, es necesario destacar que no se debe hacer este estudio sobre los pares

$$(Y_j, Y_{j+1}) = (Y_0, Y_1), (Y_1, Y_2), (Y_2, Y_3), \ldots,$$

ya que no estaríamos teniendo en cuenta factores que pueden determinar algún patrón dentro de la secuencia. Por ejemplo, vamos a suponer que tenemos una secuencia formada por ceros y unos, si dividimos la secuencia por pares tomados de la manera anterior, no estaríamos estudiando si existe un patrón en lo que se refiere a la posición de los números. Es decir, si el cero aparece más veces en posición derecha de la pareja y el uno más veces en la izquierda o viceversa se tendrían patrones que determinarían que la secuencia no es aleatoria.

Por ello, lo recomendable es realizar el test para dos subsecuencias de la principal. Primero lo aplicamos en

$$(Y_0, Y_1), (Y_2, Y_3), (Y_4, Y_5), \ldots$$

y después en

$$(Y_1, Y_2), (Y_3, Y_4), (Y_5, Y_6), \ldots$$

(Recordando siempre que existe dependencia entre ambos).

Para poder afirmar que la secuencia principal es aleatoria, es necesario que el test determine que ambas subsecuencias lo son.

Toda la teoría anterior puede ser aplicada para grupos de longitudes mayores. Es decir, supongamos que vamos a dividir nuestra secuencia en grupos de longitud t. En este caso, el razonamiento será exactamente el mismo con las ciertas variaciones a la hora de aplicar el test chi-cuadrado, ya que ahora tendremos $k = d^t$ categorías, cada una con probabilidad $p_k = 1/d^t$. Además se tendrá que analizar la aleatoriedad de las siguientes t subsecuencias:

- $(Y_0, Y_1, \ldots, Y_{t-1}), (Y_t, Y_{t+1}, \ldots, Y_{2t-1}), \ldots$
- $(Y_1, Y_2, \ldots, Y_t), (Y_{t+1}, Y_{t+2}, \ldots, Y_{2t}), \ldots$

$\vdots$

- $(Y_{t-1}, Y_t, \ldots, Y_{2t-2}), (Y_{2t-1}, Y_{2t}, \cdots, Y_{3t-2}), \ldots$

Al igual que en el caso de las parejas, para determinar si la secuencia original es aleatoria será necesario que todas las subsecuencias anteriores pasen el test satisfactoriamente.

Ejemplo. Si consideramos el número pi expresado en base 3, vamos a tomar la secuencia formada por las 100 primeras cifras. Si consideramos que vamos a analizar su aleatoriedad por parejas, tendremos las siguientes subsecuencias con sus respectivas frecuencias:

10·02·02·12·22·12·01·20·12·11·12·21·10·22·10·20·20·10·02·00·21·00·01·20·10·

11·21·12·02·21·00·22·10·02·12·10·20·21·10·21·20·12·22·22·00·10·22·01·00·20·10

Pareja	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
Frec. obs.	5	3	5	10	2	7	7	6	6

00·20·21·22·21·20·12·01·21·11·22·11·02·21·02·02·01·00·20·02·10·00·12·01·

01·12·11·20·22·10·02·21·00·21·21·02·02·11·02·12·01·22·22·20·01·02·20·10·02·01

Pareja	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
Frec. obs.	4	7	10	3	4	4	6	7	5

Por último, se calcularán los estadístico de chi-cuadrado sobre las frecuencias de cada una de las subsecuencias.

$$V_1 = \frac{(5-5.6667)^2}{5.6667} + \frac{(3-5.6667)^2}{5.6667} + \frac{(5-5.6667)^2}{5.6667} + \frac{(10-5.6667)^2}{5.6667} + \frac{(2-5.6667)^2}{5.6667} +$$
$$+ \frac{(7-5.6667)^2}{5.6667} + \frac{(7-5.6667)^2}{5.6667} + \frac{(6-5.6667)^2}{5.6667} + \frac{(6-5.6667)^2}{5.6667} = 7.7647.$$

$$V_2 = \frac{(4-5.5556)^2}{5.5556} + \frac{(7-5.5556)^2}{5.5556} + \cdots + \frac{(5-5.5556)^2}{5.5556} = 6.8799.$$

Teniendo en cuenta que 2.733 es el valor crítico para ocho grados de libertad y $\alpha = 0.05$, se tiene que ambos estadísticos toman valores mayores. Por tanto, se concluye que la secuencia no es aleatoria.

A continuación, se presenta la implementación de la prueba utilizando *R*.

```r
test_series <- function(secuencia, t, soloprimero=F){
#t es la longitud de los grupos que voy a estudiar
d <- length(unique(secuencia))
resultados <- list()
if (soloprimero) numtests=1  else numtests=t
for (i in 1:numtests){
  grupos <- base_d(secuencia[i:length(secuencia)], t, d)
  gruposfac <- factor(grupos, levels=0:(d^t-1))
  test <- chisq.test(table(gruposfac))
  resultados[[i]] <- (list(estadistico = test$statistic,
  p.valor = test$p.value))}
return(resultados)}
```

1.2.3. Test del Póker (Poker Test)

Para explicar este test pensamos en el juego del póker en el que nos dan cinco naipes con los que podemos formar distintas "figuras": pareja, trío, dobles parejas, full, ... según se repitan los números de los naipes. Dividiremos nuestra secuencia en subsecuencias de longitud cinco:

$$(Y_{5j}, Y_{5j+1}, Y_{5j+2}, Y_{5j+3}, Y_{5j+4}), 0 \leq j \leq k$$

Estos grupos van a ser clasificados según el patrón al que se ajusten.

- Todos distintos $\equiv \{a, b, c, d, e\}$
- Una pareja y tres distintos $\equiv \{a, a, b, c, d\}$
- Dos parejas y otro distinto $\equiv \{a, a, b, b, c\}$
- Un trío y dos distintos $\equiv \{a, a, a, b, c\}$
- Un trío y una pareja $\equiv \{a, a, a, b, b\}$
- Un cuarteto y otro distinto $\equiv \{a, a, a, a, b\}$
- Todos iguales $\equiv \{a, a, a, a, a\}$

Sin embargo, el test nos pide versionar estos patrones para tratar de simplificar la clasificación. Por ello, los nuevos serán los siguientes:

- 5 valores distintos $\equiv$ todos distintos
- 4 valores distintos $\equiv$ pareja
- 3 valores distintos $\equiv$ dobles parejas o trío.
- 2 valores distintos $\equiv$ full o póker.
- 1 valor distinto $\equiv$ repóker.

Nótese que la clasificación será más fácil, puesto que, entre otras cosas, hemos reducido a cinco las diferentes categorías.

En general, este test se puede aplicar sobre grupos de longitud k. En este caso, se tendrán k grupos diferentes, que irán desde aquellos con un solo valor distinto hasta aquellos que contengan k valores distintos.

Asumiendo que todos los elementos tienen la misma probabilidad de ser elegidos, llegamos a que la probabilidad de tener r valores distintos en un grupo de k elementos viene dada por:

$$p_r = \frac{d(d-1)\cdots(d-r+1)}{d^k} \left\{ {k \atop r} \right\}$$

donde $\left\{ {k \atop r} \right\}$, representa el número de Stirling de segunda especie.

Las categorías más extremas (r pequeños o grandes) si su probabilidad es demasiado pequeña se combinan con otras.

Estas probabilidades p_r son las que se utilizarán para aplicar el test chi-cuadrado a las categorías que se consideren.

Vamos a demostrar la expresión dada anteriormente para las probabilidades p_r.

Demostración. En primer lugar, vamos a definir:

- d = número de posibles elementos distintos en la secuencia.
- k = longitud de los grupos en que dividimos la secuencia.
- r = número de elementos distintos que tiene cada grupo.

Recordamos que en un contexto de equiprobabilidad, la probabilidad de un suceso A se puede calcular mediante el cociente

$$P(A) = \frac{\text{Número de casos favorables a } A}{\text{Número de posibles casos totales}}.$$

Los casos totales son el número de grupos de longitud k distinos que se pueden formar. En dichos grupos influye el orden, y además, es evidente que se permiten las repeticiones. Por tanto, viene dado por las d^k variaciones con repetición de d elementos tomados de k en k.

A continuación, obtendremos los casos favorables.

- La cantidad de formas distintas de escoger los r elementos diferentes que va a tener cada subconjunto viene dada por las $\frac{d!}{(d-r)!}$ variaciones sin repetición de d elementos tomados de r en r.
- Para obtener las diversas formas de escoger los $k-r$ elementos que quedan para completar el subconjunto tendremos que tener en cuenta que, como se mencionó anteriormente, solo pueden haber r elementos distintos (que ya han sido fijados). Por lo que los que nos quedan, deben ser iguales a alguno de esos r. (Si escogemos uno distinto, el grupo tendría $r+1$ diferentes en vez de r.)

Vamos a utilizar el número de Stirling de segunda clase

$$\left\{ k \atop r \right\},$$

que cuenta el número de formas de dividir un conjunto de k elementos en r partes. Es decir, el número de formas de ordenar el grupo de k elementos con r diferentes.

Por consiguiente, llegamos a

$$p_r = \frac{d(d-1)\cdots(d-r+1)}{d^k} \left\{ k \atop r \right\}.$$

■

Ejemplo. Tenemos la secuencia formada por las 200 primeras cifras de pi. A continuación, procederemos a dividir de 5 en 5 los elementos de la secuencia.

31415 · 92653 · 58979 · 32384 · 62643 · 38327 · 95028 · 84197 · 16939 · 93751·
05820 · 97494 · 45923 · 07816 · 40628 · 62089 · 98628 · 03482 · 53421 · 17067·
98214 · 80865 · 13282 · 30664 · 70938 · 44609 · 55058 · 22317 · 25359 · 40812·
84811 · 17450 · 28410 · 27019 · 38521 · 10555 · 96446 · 22948 · 95493 · 03819

Si contamos el número de elementos diferentes que contiene cada uno de los grupos, obtenemos las frecuencias presentadas en la tabla. Nótese que las frecuencias esperadas de los grupos con uno y dos elementos diferentes son menores que 5, por lo que éstas se incluirán en el conteo de la categoría de 3 elementos diferentes. De esta manera, esta categoría representará al total de grupos que contienen, como máximo, tres elementos distintos.

Elem. dif	3	4	5
Frec. esp.	8.14	20.16	12.91
Frec. obs.	5	17	18

Finalmente, se obtendrá el estadístico de la prueba chi-cuadrado para comparar las frecuencias observadas con las esperadas.

$$V = \frac{(5-8.14)^2}{8.14} + \frac{(17-20.16)^2}{20.16} + \frac{(18-12.906)^2}{12.906} = 3.7172$$

Como el estadístico es mayor que el valor crítico (0.103) para dos grados de libertad y $\alpha = 0.05$, concluimos que la secuencia no es aleatoria.

En el siguiente cuadro se muestra una función realizada en R para la ejecución del contraste.

```r
test_poker <- function(secuencia, k){
#k es la longitud que va a tener cada grupo
d <- length(unique(secuencia))
num_manos <- length(secuencia) %/% k
#número de elementos distintos en cada grupo.
distintos <- numeric(num_manos)
for (i in 1:num_manos){
  distintos[i] <- length(unique(secuencia[(k*i-(k-1)):(k*i)]))}
#Calculamos probabilidades
sapply(1:k, FUN=function(x){stirling_2(k,x)}) -> st
cumprod(d:(d-k+1)) * st / (d^k) -> p_r
#Juntar por abajo(t elementos diferentes)
tmin <- 1
sum1 <- 0
for (i in p_r){
  if (i*num_manos < 5){sum1 <- sum1 + i
  tmin <- tmin + 1}
  if (i*num_manos >= 5){break()}}
p_r[tmin] <- sum1 + p_r[tmin]
#Juntar por arriba (t elementos diferentes)
tmax <- k
sum2 <- 0
for (i in rev(p_r)){
  if (i*num_manos < 5){sum2 <- sum2 + i
  tmax <- tmax - 1}
  if (i*num_manos >= 5){break()}}
p_r[tmax] <- p_r[tmax] + sum2
p_r <- p_r[tmin:tmax]
#Grupos de longitud > t, se cuentan junto las de longitud t
 for (j in 1:length(distintos)) {
   distintos[j] <- if(tmin < distintos[j] &&
   distintos[j] < tmax) distintos[j]
   else if (distintos[j] <=tmin) tmin else tmax}
distfac <- factor(distintos, levels=tmin:tmax)
test <- chisq.test(table(distfac), p_r)
return(list(estadistico = test$statistic, p.valor =
test$p.value, cat.min.dif = tmin, cat.max.dif = tmax))}
```

1.2.4. Test del Coleccionista (Coupon Collector's Test)

El test del Coleccionista parte de un conjunto de n números enteros que se pueden tomar. En primer lugar, definiremos como subsecuencia completa a aquella subsecuencia en la que cada entero aparece, al menos, una vez. Dada una secuencia, vamos a contar el número de subsecuencias de cada longitud que hay en ella
Tenemos que el procedimiento para conseguir las secuencias completas va a ser:

1. Comenzamos a recorrer la secuencia hasta que nos encontremos con que tenemos al menos un elemento de cada (la colección completa). De esta manera, obtendríamos una subsecuencia completa. Calculamos su longitud.
2. Continuamos recorriendo la secuencia desde donde nos quedamos en el paso 1 hasta volver a conseguir una subsecuencia completa. Posteriormente, procederemos a contar su longitud.
   Repetimos el procedimiento, hasta que se nos acaba la secuencia. Si la última subsecuencia no es completa, ésta no se incluirá en el conteo.

Una vez calculada la longitud de cada subsecuencia completa, procederemos a contar las frecuencias de cada longitud. Posteriormente, aplicaremos la prueba chi-cuadrado considerando como categorías las longitudes calculadas anteriormente. Es evidente que la longitud mínima será n (si tiene longitud menor no puede tener al menos un elemento de cada). Finalmente, calculamos la probabilidad de cada categoría:

$$p_r = \frac{n!}{n^r} \cdot \left\{ {r-1 \atop n-1} \right\}, \quad n \leq r < t \quad \text{y} \quad p_t = 1 - \frac{n!}{n^{t-1}} \cdot \left\{ {t-1 \atop n} \right\}$$

Para concluir este contraste, vamos a demostrar la fórmula de la probabilidad p_r.

Demostración. Llamamos n al número de posibles elementos distintos que tenemos y r a la longitud de la secuencia.

Los casos totales, es decir, el número de total de secuencias de longitud r que se pueden formar serán las n^r variaciones con repetición de n elementos tomados de r en r.

A continuación, obtendremos los casos favorables.

- Como estamos considerando subsecuencias completas, sabemos que contienen, al menos, un elemento de cada uno de los n diferentes. Si tenemos en cuenta que según el orden en el que aparezcan estos n elementos se obtiene una subsecuencia u otra, tendremos que las $n!$ permutaciones sin repetición nos dan el número de órdenes diferentes en los que vamos a ir haciendo la colección.

- Por otro lado, para asegurarnos que la secuencia es de longitud r, tenemos que el último elemento de la misma debe ser uno que no haya salido anteriormente. Por lo que vamos a fijar que el elemento que ocupa la posición r va a ser el elemento n.
  Los $r-1$ elementos restantes, podrán tomar solo los $n-1$ primeros valores, ya que recordamos que el elemento n lo hemos reservado para la última posición. Para considerar todas las secuencias posibles, haremos uso de los números de Stirling de segunda clase:
  $$\left\{ r-1 \atop n-1 \right\},$$
  ya que nos devuelve el número de formas de dividir un conjunto de $r-1$ elementos en $n-1$ partes. Es decir, el número de formas de ordenar la secuencia de $r-1$ elementos restantes en $n-1$ subsecuencias diferentes.

  Por consiguiente, llegamos a
  $$p_r = \frac{n!}{n^r} \cdot \left\{ r-1 \atop n-1 \right\}, \text{ con } r \leq n$$

■

No obstante, se tiene que el test del coleccionista podría considerarse muy restrictivo, por lo que se plantea la posibilidad de considerar colecciones de menor longitud, es decir, para que una subsecuencia sea completa tendrá que contener t elementos diferentes en ella. De esta forma, no todas las colecciones estarían formadas necesariamente por los mismos elementos. En este caso,

$$p_r = \binom{n}{t} \frac{t!}{n^r} \left\{ r-1 \atop t-1 \right\}, \text{ con } r = t, t+1, ...$$

será la probabilidad de que una colección tenga longitud r.

Ejemplo. Dada la secuencia cuyos elementos son las cifras de pi. Vamos a considerar que una subsecuencia está completa cuando tenemos 4 elementos diferentes.
31415 · 9265 · 3589 · 7932 · 3846 · 2643 · 38327 · 9502 · 88419 · 7169 · 399375 · 1058 · 2097 · 4944592 · 3078 · 1640 · 628620 · 899862 · 8034 · 8253 · 42117 · 0679 · 8214 · 80865 · 1328 · 2306 · 6470 · 9384 · 4609 · 550582 · 2317 · 25359 · 4081 · 28481 · 11745 · 0284 · 1027 · 0193 · 8521 · 5105559.

Una vez obtenidas las subsecuencias completas, contamos el número que hay cada cada longitud. Todas las subsecuencias de longitud mayor o igual que 6, se contarán dentro de la categoría de longitud 6. Esto se debe, a que chi-cuadrado exige que las frecuencias esperadas en cada categoría sea mayor o igual que 5 $40 * p_7 = 40 \cdot 0.04536 = 1.8144 < 5$. En este caso, tenemos

Longitud	4	5	6
Frec. esp.	20.160	12.096	7.744
Frec. obs.	26	8	6

Tras haber obtenido las frecuencias, procederemos a compararlas mediante el estadístico V, que se distribuye según una chi-cuadrado de 3 grados de libertad.

$$V = \frac{(26-20.16)^2}{20.16} + \frac{(8-12.096)^2}{12.096} + \frac{(6-7.744)^2}{7.744} = 3.471$$

Finalmente, hemos llegado a un estadístico mayor que 0.352 (valor crítico para dos grados de libertad y α=0.05). Por tanto, concluimos que la secuencia no es aleatoria.

A continuación, se presenta la implementación de la prueba utilizando el lenguaje de programación *R*.

```
test_coleccionista <- function(secuencia, t){
# t son los elementos diferentes para secuencia completa
n <- length(unique(secuencia))
long_subsecuencias_completas <- c()
i <-  1
while (i <= length(secuencia)) {
  subsecuencia <- c() #Calculamos las subsecuencias completas
  while (length(unique(subsecuencia)) < t &&
  i <= length(secuencia)) {
    subsecuencia <- c(subsecuencia, secuencia[i])
    i <- i + 1}
  if (length(subsecuencia) >= t) {
  long_subsecuencias_completas <-c(long_subsecuencias_completas,
  length(subsecuencia))}}
n_compl <- length(long_subsecuencias_completas)
#Calculamos probabilidades
media <- 0
for (i in 1:t){media <- media + n/(n-i)}
m <- floor(3*media)
sapply(t:m, FUN=function(x){stirling_2(x-1,t-1)/(n^x)}) ->
st_denom
choose(n, t) * factorial(t) * st_denom -> p_r
#Calcular automáticamente el valor de tmin y de tmax.
#Juntar por abajo (t elementos diferentes)
tmin <- t
sum1 <- 0
for (i in p_r){
```

```
28    if (i*n_compl < 5){sum1 <- sum1 + i
29    tmin <- tmin + 1}
30    else {break()}}
31  p_r[tmin] <- sum1 + p_r[tmin]
32  #Juntar por arriba (t elementos diferentes)
33  tmax <- m
34  sum2 <- 0
35  for (i in rev(p_r)){
36    if (i*n_compl < 5){sum2 <- sum2 + i
37    tmax <- tmax - 1}
38    if (i*n_compl >= 5){break()}}
39  p_r[tmax] <- p_r[tmax] + sum2
40  p_r <- p_r[tmin:tmax]
41  #Colecciones de longitud mayor/menor que tmax y tmin
42  for (j in 1:length(long_subsecuencias_completas)) {
43    long_subsecuencias_completas[j] <-
44    if(long_subsecuencias_completas[j] < tmax)
45    long_subsecuencias_completas[j] else tmax
46    long_subsecuencias_completas[j] <-
47    if(long_subsecuencias_completas[j] > tmin)
48    long_subsecuencias_completas[j] else tmin}
49  longfac <-factor(long_subsecuencias_completas, levels=tmin:tmax)
50  test <- chisq.test(table(longfac), p_r)
51  return(list(estadistico = test$statistic,
52  p.valor = test$p.value, cat.min.long = tmin, cat.max.long =
53  tmax))}
```

1.2.5. Test de las Rachas (Run Test)

Dada una secuencia de números enteros vamos a dividirla en rachas estrictamente crecientes. No obstante, para su formación tendremos en cuenta que cada vez que se finalice una de las rachas, se eliminará el elemento inmediatamente posterior. Esto lo hacemos para conseguir que exista independencia entre las rachas.

Una vez se han obtenido todas las rachas presentes en la secuencia, se contará la longitud, se hallan las frecuencias y se aplica chi-cuadrado.

No obstante, como para poder aplicar chi-cuadrado es necesario que la frecuencia esperada de cada categoría sea mayor o igual que 5, tendremos que unir las categorías a partir de una longitud t.

Finalmente, asumiendo que la secuencia puede tomar valores entre 0 y $d-1$, se tiene que la probabilidad de que se dé una racha de longitud r viene dada por

$$p_r = \sum_{j=r-1}^{d-1} p_j^r \left(\frac{j}{d}\right),$$

donde los p_k^t son las probabilidades de llevar t elementos de forma ascendente (la racha no tiene por qué haber finalizado) en donde el último elemento hasta el momento es k (con $k = 0, 1, \ldots d-1$; $t = 1, 2, \ldots d$). Estos p_k^t pueden calcularse de forma recursiva como

$$p_k^t = \frac{1}{d} \sum_{j=0}^{k-1} p_j^{t-1}$$

donde $p_j^1 = \frac{1}{d}$ para $j = 0, \ldots, d-1$.

Ejemplo. Se presenta la secuencia

3141592653589793238462643383279502884197169399375105820974944

formada por los 150 primeros decimales de pi.

En este caso, tenemos que nuestras rachas van a ser las siguientes: 3·4·59·6·3589·9·238·6·6·338·279·0288·19·169·99·7·1·58·09·49·459·3·7·16·06·8·2·899·6·8·348·5·4·117·679·2·48·8·5·3·8·3·66·7·9·8·46·9·5·58·23·7·5·59·08·2.

Observamos las siguientes frecuencias (observamos que para longitudes mayores a 3, la frecuencia esperada es menor que 5 por lo que juntaremos categorías).

Longitud	1	2	3
Frec. esp.	30.8000	18.5360	5.8564
Frec. obs.	30	16	10

Luego, calcularemos el estadístico de chi-cuadrado para comparar las frecuencias dadas anteriormente.

$$V = \frac{(30-30.8)^2}{30.8} + \frac{(16-18.536)^2}{18.536} + \frac{(10-5.8564)^2}{5.8564} = 3.2887.$$

Finalmente, como nuestro estadístico es mayor que 0.352, valor crítico para dos grados de libertad y $\alpha = 0.05$, el test concluye que la secuencia no es aleatoria.

Finalmente, se ha realizado una función en R para determinar la aleatoriedad de la secuencia a partir del test de las rachas.

```
test_rachas <- function(secuencia){
n <- length(secuencia)
d <- length(unique(secuencia))
rachas <- 1
i <- 2
```

```
while (i <= n) {
 #En la posición i hay un cambio de racha
 #(el elemento i es el primer elemento de la nueva racha)
  if (secuencia[i] <= secuencia[i-1]) {
    rachas <- c(rachas, i+1)
    i <- i+2}
  else i <- i+1}
  # Añadimos índice de finalización última racha.
 rachas <- c(rachas, n+2) #Para poder calcular su longitud.
  # Calculamos la longitud de cada racha.
 long_rachas <- diff(rachas) - 1
 long_rachas[length(long_rachas)] <- rachas[length(rachas)] -
 rachas[length(rachas)-1]
#Calculamos la probabilidad de cada longitud.
p_r <- c()
p_kt <- matrix(0, nrow = n, ncol = d)
p_kt[1, ] <- 1/d
p_kt[, 1] <- (1/d)^(1:n)
for (t in 2:n){
  for (k in 2:d){
    p_kt[t, k] <- 1/d * sum(p_kt[t-1, 1:(k-1)])}}
for (r in 1:n){
  h <- c()
  for (j in 1:d){
    if (j < (r-1)){h <- c(h, 0)}
    else{h <- c(h, p_kt[r, j]*(j/d))}}
p_r <- c(p_r, sum(h))}
#Calcular automáticamente el valor de t.
t <- 0
sum <- 0
for (i in 1:length(secuencia)){
  p_i <- p_r[i]
  if (length(long_rachas)*p_i < 5){break}
  else {t <- t + 1}}
p_r <- c(p_r[1:(t-1)], sum(p_r[t:length(p_r)]))
#Rachas de longitud > t, se cuentan junto las de longitud t
for (i in 1:length(long_rachas)) {
   long_rachas[i] <- if(long_rachas[i] < t) long_rachas[i]
   else t}
#Aplicamos Chi-Cuadrado
longfac <- factor(long_rachas, levels=1:t)
```

```
47  test <- chisq.test(table(longfac), p_r)
48  return(list(estadistico = test$statistic,
49  p.valor = test$p.value, cat.max.long = t))}
```

1.2.6. Test de las Colisiones (Collision Test)

Este test se va a basar en el siguiente experimento. Tenemos m cajas (m categorías) y n pelotas, con $m > n$, es decir, el número de cajas es mayor que el número de pelotas, por lo que si se procede a tirar todas las pelotas encima de las cajas, podemos afirmar con total seguridad, que va a ser inevitable que algunas cajas queden completamente vacías, no sabemos cuántas con exactitud, pero como mínimo serán $m-n$ cajas. Diremos que se ha producido una colisión cuando dos pelotas caigan en la misma caja.

Aplicamos este test a secuencias de números enteros donde m (número de categorías) es mucho mayor que n (longitud de la secuencia).

Este test va a calcular el número de colisiones que se producen dentro de una misma secuencia. Posteriormente, se comprobará que no genera ni muchas ni muy pocas colisiones. Además, tenemos que dada una categoría, la probabilidad de que k observaciones pertenezcan a ella es:

$$p_k = \binom{n}{k} \frac{1}{m^k} \left(1 - \frac{1}{m}\right)^{n-k}.$$

Nótese que para cada observación de las n que tenemos, la probabilidad de éxito, es decir, de que caiga en una categorá determinada, es $\frac{1}{m}$. Por lo tanto, el número total de las que caen en esa categoría se distribuye como una binomial, $Bi(n, p = \frac{1}{m})$.

Por otro lado, cabe destacar que usando la expresión

$$\sum_{k\geq 1}(k-1)p_k = \sum_{k\geq 0} kp_k - \sum_{k\geq 1} p_k = \frac{n}{m} - 1 + p_0,$$

con $p_0 = (1 - \frac{1}{m})^n$, obtenemos el número medio de colisiones que han tenido lugar por categoría.

Demostración. En primer lugar, aplicamos la propiedad distributiva y propiedades de de los sumatorios.

$$\sum_{k\geq 1}(k-1)p_k = \sum_{k\geq 1}(kp_k - p_k) = \sum_{k\geq 1} kp_k - \sum_{k\geq 1} p_k = \sum_{k\geq 0} kp_k - \sum_{k\geq 1} p_k.$$

Recordamos que nuestra probabilidad p_k viene de una binomial, por lo que podemos ver a $\sum_{k\geq 0} kp_k$ como la esperanza de la distribución $Bi(n, p = \frac{1}{m})$, es decir,

$$\sum_{k\geq 0} kp_k = n \cdot p = n \cdot \frac{1}{m} = \frac{n}{m}.$$

Además, sabemos que la suma de todas las probabilidades tiene que dar 1, es decir, $\sum_{k\geq 0} p_k = 1$. Luego, obtenemos

$$\sum_{k\geq 1} p_k = \sum_{k\geq 0} p_k - p_0 = 1 - p_0.$$

Por último, si sustituimos, llegamos a

$$\sum_{k\geq 0} kp_k - \sum_{k\geq 1} p_k = \frac{n}{m} - (1 - p_0) = \frac{n}{m} - 1 + \left(1 - \frac{1}{m}\right)^n.$$

■

Por otro lado, tenemos que la probabilidad de que se den exactamente c colisiones, viene dada de la siguiente forma:

$$P(c \text{ colisiones}) = \frac{m(m-1)\cdots(m-n+c+1)}{m^n} \left\{ {n \atop n-c} \right\}.$$

Demostración. Tenemos n observaciones que vamos a clasificar en m categorías. En primer lugar, se tienen m^n formas de asignar las diferentes categorías a las n observaciones.

En segundo lugar, vamos a calcular los casos favorables. Como se tienen que dar c colisiones, sabemos que se van a utilizar $n - c$ categorías (las que se quedan llenas). Como el orden en el que se asignan las categorás influye y no se pueden repetir, tomamos las $m(m-1)\cdots(m-n+c+1)$ variaciones sin repetición de m categorías tomadas de $n - c$ en $n - c$ observaciones.

Además, tenemos que considerar

$$\left\{ {n \atop n-c} \right\},$$

ya que nos devuelve el número de formas de dividir un conjunto de n elementos en $n - c$ partes. Es decir, el número de formas de ordenar las n categorías en $n - c$ partes diferentes. ■

Para aplicar el test sobre secuencias, vamos a considerar que tenemos $m = d^t$ categorías, donde d es el número de elementos diferentes que puede obtener la secuencia y donde t va a ser la longitud que vamos a darle a cada observación. A continuación, se deberá calcular el número total de colisiones que se han dado en la secuencia.

Finalmente, una vez calculado el número de colisiones, obtendremos el p-valor correspondiente. Para ello, en primer lugar, obtendremos las probabilidades de que se hayan producido c colisiones para $c = 1, 2, 3, \ldots$. Posteriormente, las ordenaremos de menor a mayor. Nuestro p-valor será el resultado de la suma desde la probabilidad más pequeña hasta la probabilidad del número de colisiones de nuestra secuencias principal.

Ejemplo. Dada la secuencia

$$31415926535897932384626433832795028841971693993751$$

que ha sido generada con los 50 primeros números de pi. Vamos a calcular el número total de colisiones que han tenido lugar en ella. En primer lugar, sabemos que para aplicar este test, el número de categorías tiene que ser mucho mayor que la longitud de la secuencia. Por ello, vamos a dividir la secuencia en 25 grupos de longitud 2, donde cada uno será considerado una observación.

$$31 \cdot 41 \cdot 59 \cdot 26 \cdot 53 \cdot 58 \cdot 97 \cdot 93 \cdot 23 \cdot 84 \cdot 62 \cdot 64 \cdot 33 \cdot 83 \cdot 27 \cdot 95 \cdot 02 \cdot 88 \cdot 41 \cdot 97 \cdot 16 \cdot 93 \cdot 99 \cdot 37 \cdot 51$$

Tenemos un total de $m = 10^2$ categorías ($100 > 25$). A continuación, contaremos el número de colisiones que se han dado en la secuencia.

Vemos que todas las parejas aparecen una vez salvo 41, 97 y 93 que aparecen dos veces. Diremos que cada uno de los pares anteriores ha producido una colisión. Han tenido lugar, por tanto, un total de tres colisiones.

Finalmente, calcularemos la probabilidad teórica de que se hayan producido tres colisiones.

$$P(3 \text{ colisiones}) = \frac{100(100-1)\cdots(79)}{100^{25}} \left\{ {25 \atop 22} \right\} = 0.2638.$$

El siguiente código muestra la implementación del test en R.

```
numero_colisiones <- function(secuencia, t){
#t es la longitud de cada observación.
n <- length(secuencia) %/% t #número de observaciones.
observaciones <- split(secuencia, rep(1:n, each = t))
numeros_enteros <- c()
for (i in 1:length(observaciones)) {
   grupo <- observaciones[[i]]
   num <- c()
   cont <- t
   for (i in grupo){
     num <- c(num, i*10^(cont-1))
    cont <- cont-1}
    numeros_enteros <- c(numeros_enteros, sum(num))}
```

```
#Tenemos el número de observaciones que caen en cada categoría.
repeticiones <- table(numeros_enteros)
#Tenemos el número de colisiones que caen en cada categoría.
colisiones <- repeticiones - 1
colisiones_totales <- sum(colisiones)
return (colisiones_totales)}
```

```
test_colisiones <- function(secuencia, t, m){
n <- length(secuencia)%/%t
if (m<n){print("El número de categorías debe ser mayor que la
longitud de la secuencia")}
else{c <- numero_colisiones(secuencia, t)
p_c <- ((factorial(m)/factorial(m-n+c))/(m^n))*
stirling_2(n, n-c)}
#Obtener todas las probabilidades para el pvalor
sapply((n-1):0, FUN=function(x){stirling_2(n,n-x)}) -> st
cumprod(m:(m-n+1)) * st / (m^n) -> p_r
p_ordenado <- sort(p_r)
# Calcular el p_valor
p_valor <- sum(p_ordenado[p_ordenado <= p_c])
return(list(colisiones = c, probabilidad = p_c,
p.valor = p_valor))}
```

1.3. Aplicación de los contrastes a secuencias de enteros

1.3.1. Descripción secuencias

- **Thue-Morse:** La secuencia de Thue-Morse es una secuencia binaria infinita que se construye de manera iterativa aplicando reemplazamiento. Comienza con cero y en cada iteración se duplica el tamaño de la secuencia, ya que se concatena la secuencia original con la secuencia invertida. La formación de esta última consiste en convertir 0 por 1 y viceversa. Concretamente, trabajaremos con los mil primeros elementos.

$$0110100110010110100101100110 10\ldots$$

- π: La secuencia del número π es una secuencia cuyos elementos corresponden con los decimales de π. En nuestro caso, tomaremos también las cifras de la parte entera como los primeros elementos de la secuencia.

$$3141592653589793238462643383279\ldots$$

- $\sqrt{2}$: La secuencia está formada por todas las cifras, tanto enteras como decimales, de dicho entero. En particular, estudiaremos los mil primeros términos.

$$14142135623730950488016887242\ldots$$

- **ADN:** Una secuencia de ADN (ácido desoxirribonucleico) es una cadena de moléculas que representan la información genética en los seres vivos. Está compuesta por una secuencia de nucleótidos, que son unidades básicas que contienen una base nitrogenada (adenina, timina, citosina o guanina). En nuestro caso, diremos que cada ácido nucléico va a estar asociado a un número entero entre uno y cuatro. Concretamente, tenemos la siguiente asignación adenina (1), citosina (3), guanina (7) y timina (20). Nótese que cada entero corresponde con la posición de la inicial en el abecedario. En contreto, nuestra secuencia corresponde con un segmento de ADN orginal de un ser humano.

$$GATCACAGGTCTATCACCCTATTAACC$$

- **Lanzamiento de monedas**: La secuencia representa los resultados de los diez lanzamientos de una moneda realizados por cada alumno de una clase. Cada dígito corresponde a un lanzamiento, donde 0 representa cruz y 1 representa cara. Concretamente, se han unido los resultados de 68 alumnos, por lo que tenemos una secuencia de longitud 680.

$$01100001011000011100000010000\ldots$$

- **Simulación lanzamiento monedas:** Esta secuencia presenta los resultados de 100 lanzamientos simulados de una moneda realizados mentalmente por un alumno.

$$00010110111010001101011101 1000\ldots$$

- **Generada por** R: La secuencia ha sido generada por R utilizando la función de semilla "seed(1)" con el fin de poder garantizar la reproducibilidad de los datos. La secuencia se obtuvo mediante la repetición aleatoria de los números entre 0 y 9 hasta lograr una secuencia de longitud mil. En cada repetición, R selecciona un número al azar de manera aleatoria entre 0 y 9.

$$83601612044959684488441980 3259\ldots$$

1.3.2. Tabla de contrastes en la batería de secuencias

Secuencia	Chi^2	Series	Póker	Colecc.	Rachas	Colis
Thue-Morse	1	$t = 2$: 4.77e-108, 4.43e-12	$k = 4$: 0.2231	-	0.1991	-
π	0.8613	$t = 2$: 0.1217, 0.2188	$k = 4$: 0.1991	$t = 6$: 0.2242	0.2133	0.2906
$\sqrt{2}$	0.5161	$t = 3$: 0.0974, 0.2191, 0.8249	$k = 4$: 0.1991	$t = 6$: 0.2423	0.2133	0.8325
ADN	6.868e-16	$t = 2$: 4.4032e-103, 1.0320e-113	$k = 4$: 0.1991	$t = 4$: 0.2600	0.2133	0.4275
Lanz. monedas	0.4900	$t = 2$: 0.2171, 0.6352	$k = 4$: 1	$t = 3$: 0.2381	0.1991	0.6830
Simul. monedas #1	0.4190	$t = 2$: 0.8308, 0.7610	$k = 3$: 1	$t = 2$: 1	0.1991	0.5216
Sample de R	0.6496	$t = 4$: 0.7345, 0.1389, 0.5179, 0.8825	$k = 6$: 0.2133	$t = 7$: 0.2426	0.2133	0.8325

Tabla 1.1. Resultados de la aplicación de los tests a una batería de secuencias enteras. Se indica el p-valor.

En la tabla podemos observar los p-valores obtenidos al ejecutar cada secuencia con los diferentes contrastes.

Concretamente, observamos que la mayoría de los p-valores son mayores a los niveles de significación típicos ($\alpha = 0.05, 0.1, \dots$). De esta manera, identificamos que existen secuencias no aleatorias que superan satisfactoriamente algunas de las pruebas.

Destacamos el caso de la secuencia generada por el algoritmo de Thue-Morse, ya que, según el test de chi-cuadrado, puede considerarse como aleatoria (se obtiene el valor p-valor más alto posible). Como este test compara las frecuencias de cada categoría y, en este caso, la secuencia de Thue-Morse tiene la misma cantidad de ceros que de unos, es lógico esperar que supere esta prueba. Sin embargo, al aplicar el test de las series, determinamos que no es aleatoria, ya que se obtuvieron valores p-valores bastante pequeños (inferiores a los niveles de significancia). Por lo tanto, aquí se evidencia la necesidad de aplicar múlti-

ples tests a una misma secuencia, ya que pueden existir patrones que algunos contrastes no son capaces de identificar mientras que otros sí lo hacen.

Finalmente, nos fijamos en los resultados obtenidos por la secuencia de lanzamientos de monedas (aleatoria) y por la simulación de lanzamientos (propuesta por el alumno). Resulta notable que la secuencia generada mediante simulaciones haya logrado superar un mayor número de pruebas en comparación con la secuencia de lanzamientos aleatorios. Por ende, se puede concluir que, en este caso, el estudiante ha sido capaz de determinar resultados que podrían ser considerados como aleatorios, en el sentido de que no se han encontrado patrones o estructuras predecibles en los datos.

Después de notar que la secuencia aleatoria obtuvo peores resultados que la simulada mentalmente por un alumno, aplicamos las diferentes pruebas a las simulaciones de varios de los alumnos. El objetivo es verificar si los p-valores obtenidos en las pruebas anteriores se mantienen consistentes en todos los casos o si hay variaciones significativas entre las simulaciones. La elección de las simulaciones ha sido aleatoria. No obstante, las correspondientes a los alumnos 9, 17 y 42 fueron seleccionadas intencionalmente. Se eligieron, puesto que presentaban características como presentar bloques de ceros y unos consecutivos (9), la primera mitad de la secuencia coincide exactamente con la segunda mitad (17) y, finalmente, la primera mitad es todo unos salvo un cero y la segunda es todo ceros.

Después de aplicar los tests a diferentes secuencias de simulaciones, hemos observado que para la mayoría los p-valores son aproximadamente iguales. No obstante, destacamos que todas las simulaciones que fueron elegidas intencionadamente por la presencia de diferentes patrones en ellas, han sido definidas como no aleatorias por, al menos, un test. Por tanto, concluimos que parte de los tests presentados (en particular, el test de las series y el de las colisiones) son capaces de diferenciar entre secuencias aleatorias y las que no lo son, pero para hacerlo es necesario que los patrones estén bien definidos.

Secuencia	Chi^2	Series $t = 2$	Series $t = 3$	Rachas	Colis
Simul. monedas #1	0.4190	0.8308, 0.7610	0.1886, 0.3326, 0.5397	0.1991	0.5216
Simul. monedas #8	0.2254	0.6026, 0.5385	0.2187, 0.7798, 0.8850	0.1991	0.5216
Simul. monedas #9	0.06902	1.1075e-09, 1.6218e-06	1.5096e-10, 2.3880e-11, 5.8200e-13	0.1991	7.8569e-10
Simul. monedas #17	0.0263	0.0013, 0.0017	0.2187, 0.7798, 0.8850	0.1991	7.8568e-10
Simul. monedas #37	0.6862	0.7520, 0.6823	0.9271, 0.8850, 0.9271	0.1991	0.5216
Simul. monedas #42	0.8399	5.4652e-09, 1.3910e-09	8.6118e-15, 2.5139e-16, 5.3819e-15	0.1991	4.2016e-22
Simul. monedas #59	0.2254	0.4175, 0.4753	0.5397, 0.5397, 0.5397	0.1991	0.5216
Simul. monedas #66	0.6862	0.9075, 0.2122	0.8850, 0.8352, 0.5397	0.1991	0.5216
Simul. monedas Aleat 1	0.8399	0.5036 , 0.9189	0.9271, 0.8850, 0.7798	0.1991	0.5216
Simul. monedas Aleat 2	0.8399	0.6026 , 0.3430	0.5992, 0.7207, 0.4289	0.1991	0.0223
Simul. monedas Aleat 3	0.2254	0.5682 , 0.5724	0.7798, 0.5397, 0.3787	0.1991	0.5216

Tabla 1.2. Resultados de la aplicación de los tests a una batería de simulaciones de alumnos de 2º curso del Grado de Matemáticas y a otras aleatorias.

2

Pruebas de aleatoriedad para secuencias de números reales

2.1. Pruebas de bondad de ajuste: Kolmogorov-Smirnov

En el capítulo 1 vimos el test chi-cuadrado para probar si unos datos provienen de una distribución discreta. Kolmogorov-Smirnov es un test de bondad de ajuste a una distribución continua que viene dada por su función de distribución.

Dadas n observaciones independientes, el test compara la función de distribución acumulada observada, $F(x)$, de nuestros datos con la distribución teórica que queremos contrastar, $G(x)$. Es decir, este contraste tiene como hipótesis nula que las observaciones proceden de una distribución determinada.

$$\begin{cases} H_0 \equiv & F(x) = G(x) \\ H_1 \equiv & F(x) \neq G(x). \end{cases}$$

Para comparar las dos distribuciones, consideramos la diferencia en valor absoluto entre ambas, $|F(x) - G(x)|$, para todo punto x.

Para ello, el test de Kolmogorov-Smirnov nos da los dos siguientes estadísticos:

1. Por un lado, vamos a tener un estadístico que determina la desviación máxima (diferencia entre ambas) cuando el valor de la distribución empírica es mayor que el de la teórica

$$K^+ = \sqrt{n} \cdot \sup_{-\infty<x<\infty} (F(x) - G(x))$$

2. Por otro lado, vamos a tener un estadístico que determina la desviación máxima (diferencia entre ambas) cuando el valor de la distribución teórica es mayor que el de la empírica

$$K^- = \sqrt{n} \cdot \sup_{-\infty<x<\infty} (G(x) - F(x))$$

Se puede observar que en ambos estadísticos se multiplica por $\sqrt{n}$. Esto se debe a que tenemos que la desviación de $F(x)$ se puede obtener multiplicando

por $\frac{1}{\sqrt{n}}$ a una combinación de la distribución teórica. Por ello, si multiplicamos por $\sqrt{n}$ obtendríamos que los valores de ambos estadísticos serán independientes del número de observaciones, n. A continuación se presenta la demostración que muestra la proporcionalidad entre ambas distribuciones.

Demostración. Asumimos que se tienen n observaciones $X_1, X_2, \ldots, X_n$ independientes. La función de distribución empírica va a venir dada por

$$F(x) = P(X \leq x) = \frac{\text{Número de observaciones } X_i \text{ que son } \leq x}{n}.$$

Se tiene que el valor medio de $F(x)$ es:

$$E[F(x)] = E\left[\frac{\text{cantidad de } X_i \text{ que son } \leq x}{n}\right] = \frac{1}{n}E[\sum_{i=1}^{n} I(X_i \leq x)] =$$

$$= \frac{1}{n}\sum_{i=1}^{n} E[I(X_i \leq x)] = \frac{1}{n}\sum_{i=1}^{n} P(X_i \leq x) = G(x),$$

donde tendremos que $I(X_i \leq x)$ representa la siguiente función indicadora:

$$I(X_i) = \begin{cases} 1 & \text{si } X_i \leq x. \\ 0 & \text{en otro caso.} \end{cases}$$

Por otro lado, se tiene que la varianza de $F(x)$ es:

$$\text{Var}(F(x)) = \text{Var}\left[\frac{\text{cantidad de } X_i \text{ que son } \leq x}{n}\right] = \frac{1}{n^2}\,\text{Var}[\sum_{i=1}^{n} I(X_i \leq x)] =$$

$$= \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}(I(X_i \leq x)),$$

puesto que $I(X_i \leq x)$ es una Bernuilli de parámetro $P(X_i \leq x) = G(x)$, su varianza será $G(x)(1 - G(x))$ y por tanto

$$\text{Var}(F(x)) = \frac{1}{n^2}\sum_{i=1}^{n}[G(x) \cdot (1 - G(x))] = \frac{G(x) \cdot (1 - G(x))}{n}.$$

Nótese que la desviación típica es la raíz de la varianza, quedándonos que la desviación entre ambas distribuciones es

$$\text{DS}(F(x)) = \frac{1}{\sqrt{n}} \cdot \sqrt{G(x) \cdot (1 - G(x))}.$$

■

Finalmente, teniendo en cuenta que

$$K = \text{máx}\{K_n^+, K_n^-\} = \sqrt{n} \cdot \sup_{-\infty<x<\infty} |F(x) - G(x)|,$$

ya que $|F(x)-G(x)| = |G(x)-F(x)|$, se aplicará a las observaciones el estadístico K que, como demostró D. Knuth (1938), sigue la distribución

$$P_r(K \leq s) = 1 - e^{-2s^2}\left(1 - \frac{2s}{3\sqrt{n}} + \mathcal{O}\left(\frac{1}{n}\right)\right).$$

De esta manera, determinaremos si están ajustadas o no, esto lo haremos comparándolo con el valor crítico que depende del nivel de significación α y del tamaño muestral. Para obtenerlo, se acudirá a la tabla de valores críticos de Kolmogorov-Smirnov y diremos que se rechaza la hipótesis nula solo si el valor de K es mayor que el valor crítico.

Ejemplo. Se tiene una secuencia cuyos elementos son

$$0.6828, 0.3137, 0.4805, 0.0843, 0.2907, 0.4767, 0.3705, 0.1638, 0.6519, 0.2593$$

Vamos a comprobar si la distribución empírica de nuestra secuencia corresponde con una distribución exponencial con $\lambda = 1$.

En este caso, tenemos que nuestra función de distribución teórica es

$$G(x) = 1 - e^{-x}, \quad \forall x \geq 0.$$

Observaciones	$F(x)$	$G(x)$	$\lvert F(x) - G(x)\rvert$
0.0843	0.1	0.0808	0.0192
0.1638	0.2	0.1511	0.0489
0.2593	0.3	0.2284	0.0716
0.2907	0.4	0.2523	0.1477
0.3137	0.5	0.2693	0.2307
0.3705	0.6	0.3096	0.2903
0.4767	0.7	0.3792	0.3208
0.4805	0.8	0.3815	0.4184
0.6519	0.9	0.4789	0.4211
0.6828	1	0.4517	0.5052

Observamos que

$$K = \sqrt{10} \cdot \sup_{-\infty<x<\infty} |F(x) - G(x)| = 1.5976.$$

Vemos que el valor del estadístico es mayor que el valor crítico, 0.442 (para $n = 10$ y $\alpha = 0.05$), por lo que tendremos que rechazar la hipótesis nula, afirmando que las observaciones no siguen una distribución de probabilidad exponencial.

2.2. Tests de aleatoriedad

En este apartado estudiaremos siete tests diferentes. Las secuencias sobre las que vamos a aplicar las diferentes pruebas están formadas por números reales,

$$\langle U_n \rangle = U_0, U_1, U_2, \ldots.$$

y pretendemos comprobar si se distribuyen uniforme e independientemente entre 0 y 1.

Por otro lado, cabe destacar que todos los tests descritos en el capítulo 1 pueden ser aplicados a secuencias reales considerando un valor de d conveniente contrastando la aleatoriedad de la secuencia

$$\langle Y_n \rangle = \lfloor dU_0 \rfloor, \lfloor dU_1 \rfloor, \lfloor dU_2 \rfloor, \ldots.$$

2.2.1. Test de Equidistribución (Equidistribution Test)

Esta prueba se puede abordar mediante dos procedimientos diferentes:

- Kolmogorov-Smirnov: Aplicamos el test de K-S tomando como función teórica $G(x) = x$, para $0 < x < 1$, esto es, comprobamos si nuestras observaciones pueden provenir de una distribución uniforme en el intervalo $[0, 1]$.

  Ejemplo. Vamos a considerar una secuencia cuyos elementos van tener 0 como parte entera y a los términos del número pi de tres en tres como parte decimal. Concretamente, tendremos unas secuencia de longitud 10 generada por los 30 primeros números de pi, es decir, obtendremos la secuencia

  $$0.314, 0.159, 0.265, 0.358, 0.979, 0.323, 0.846, 0.264, 0.338, 0.327.$$

  Vamos a comprobar si la distribución empírica de nuestra secuencia corresponde con la distribución teórica $G(x) = x, \quad 0 \leq x \leq 1$.

Observaciones	$F(x)$	$G(x)$	$\lvert F(x) - G(x) \rvert$
0.159	0.1	0.159	0.059
0.264	0.2	0.264	0.064
0.265	0.3	0.265	0.035
0.314	0.4	0.314	0.086
0.323	0.5	0.323	0.177
0.327	0.6	0.327	0.273
0.338	0.7	0.338	0.362
0.358	0.8	0.358	0.442
0.846	0.9	0.846	0.054
0.979	1	0.979	0.021

Observamos que

$$K = \sqrt{10} \cdot \sup_{-\infty<x<\infty} |F(x) - G(x)| = 1.3977.$$

Tenemos que es mayor que nuestro valor crítico, 0.442 (para $n = 10$ y $\alpha = 0.05$), por lo que rechazaremos la hipótesis nula, concluyendo que la muestra no sigue una distribución uniforme en [0,1].

- Chi-cuadrado: siendo d un número conveniente, que nos dará cuántas categorías consideraremos tenemos que, en este caso, utilizaremos la secuencia

$$\langle Y_n \rangle = \lfloor dU_0 \rfloor, \lfloor dU_1 \rfloor, \lfloor dU_2 \rfloor, \ldots,$$

Una vez hemos contado las frecuencias, aplicamos el test chi-cuadrado con $k = d$, siendo $p_s = \frac{1}{d}$ la probabilidad de que se dé cada categoría.

Ejemplo. Siguiendo el ejemplo anterior, para poder aplicar chi-cuadrado, multiplicaremos todos los valores de la secuencia por $d = 10$ (tamaño muestral). De esta manera, obtenemos los valores

$$3.14, 1.59, 2.65, 3.58, 9.79, 3.23, 8.46, 2.64, 3.38, 3.27.$$

que presentan las diferentes categorías y sus respectivas frecuencias esperadas. Sin embargo, todas ellas son menores que cinco, por lo que el resultado al aplicar el test chi-cuadrado a esta secuencia no será fiable.

Este código permite llevar a cabo el contraste anterior usando R.

```
test_equidist <- function(secuencia, metodo){
d <- length(unique(secuencia))
#OPCIÓN 1
if (metodo == 'chi'){
secuencia_enteros <- floor(d*secuencia)
test <- chisq.test(table(secuencia_enteros))
return(list(estadistico = test$statistic,
p.valor = test$p.value))}
#OPCIÓN 2
else if (metodo == 'ks'){
  Fdistr <- function(x) {
  ifelse(x <= 0, 0, ifelse(x >= 1, 1, 1/x))}
test <- ks.test(secuencia, Fdistr)
return(list(estadistico = test$statistic,
p.valor = test$p.value))}
}
```

2.2.2. Test de Separación (Gap Test)

Este test va a basarse en los saltos que hay dentro de una secuencia. Definimos como salto a la cantidad de números entre dos valores que cumplen cierta condición. Nosotros queremos analizar la longitud de los saltos entre los U_j que pertenecen a un rango determinado. Dicho rango vendrá determinado por los valores reales $\alpha, \beta \in [0,1], \alpha < \beta$. Por tanto, la formación de un salto vendrá dada de la siguiente forma:

1. Comenzamos a recorrer la secuencia. Cuando nos encontremos con un valor U_i perteneciente al intervalo $[\alpha, \beta]$, iniciaremos un hueco (U_i no se incluye en el hueco).
2. Continuaremos recorriendo la secuencia. En el momento que aparezca otro elemento U_j perteneciente a dicho intervalo, el hueco anterior habría finalizado (U_j no se incluye en el hueco).
3. Una vez se han determinado todos los huecos presentes en la secuencia, se cuentan cuántos hay de cada longitud.

A continuación, se aplicará la prueba chi-cuadrado. Como ya sabemos, para aplicar este contraste, es necesario conocer la probabilidad de cada categoría. En este caso, tenemos que las longitudes de los huecos se distribuyen como una geométrica que truncaremos en cierta longitud t para que no nos queden categorías con frecuencias esperadas demasiado pequeñas:

$$p_r = p(1-p)^r, \quad 0 \leq r < t \qquad \text{y} \qquad p_t = (1-p)^t \quad \text{donde} \quad p = \beta - \alpha.$$

De esta manera, determinamos una categoría que va a incluir todas las rachas de longitudes mayores o iguales a t.

Ejemplo. Consideremos la secuencia de longitud 70 cuyos elementos son números decimales cuya parte entera es cero y la decimal son los números de pi de tres en tres. Vamos determinar los saltos que se producen en ella dado el intervalo $[0.2, 0.7]$.

Vamos a recorrer la secuencia para identificar a todos los elementos pertenecientes al intervalo, que señalaremos en morado:
0.314, 0.159, 0.265, 0.358, 0.979, 0.323, 0.846, 0.264, 0.338, 0.327, 0.950, 0.288, 0.419, 0.716, 0.939, 0.937, 0.510, 0.582, 0.097, 0.494, 0.459, 0.230, 0.781, 0.640, 0.628, 0.620, 0.899, 0.862, 0.803, 0.482, 0.534, 0.211, 0.706, 0.798, 0.214, 0.808, 0.651, 0.328, 0.230, 0.664, 0.709, 0.384, 0.460, 0.955, 0.058, 0.223, 0.172, 0.535, 0.940, 0.812 0.848, 0.111, 0.745, 0.028, 0.410, 0.270, 0.193, 0.852, 0.110, 0.555, 0.964, 0.462, 0.294, 0.895, 0.493, 0.038, 0.196, 0.442, 0.881, 0.097.

Seguidamente, contaremos el número de elementos que hay entre cada dos pertenecientes al intervalo. No obstante, como conocemos las frecuencias esperadas y hemos obtenido un total de 35 rachas (y una última no acabada por

lo que no la incluiremos en el conteo) se tiene que para longitudes mayores o iguales que 2 se espera una frecuencia, $35 * p_2 = 35 \cdot (0.5) \cdot (1 - 0.5)^2 = 4.375$, menor que 5, por lo que todos los datos de longitud mayores que 2 se incluirán también en esta categoría.

Longitud	0	1	2
Frec. esperada	17.5	8.8	8.5
Frec. observada	17	11	7

Finalmente, usando las frecuencias anteriores calculamos el estadístico de chi-cuadrado

$$V = \frac{(17 - 17.5)^2}{17.5} + \frac{(11 - 8.75)^2}{8.75} + \frac{(7 - 8.5)^2}{8.5} = 0.8746.$$

Como tenemos que el estadístico es mayor que el p-valor para dos grados de libertad y $\alpha = 0.05$ (0.103), se rechazará la hipótesis nula. Concluyendo que hay evidencias suficientes para afirmar que existen diferencias significativas entre las distribuciones, es decir, consideramos que la secuencia no es aleatoria.

A continuación, podemos observar una función definida en R para la realización del test.

```
test_huecos <- function (secuencia, alpha, beta){
# Cada True (elem. del intervalo) indica comienzo racha.
x <- secuencia >= alpha & secuencia <= beta
# Encontrar las longitudes de las rachas de False
rachas_false <- rle(x)$lengths[rle(x)$values == FALSE]
#El comando rle cuenta la longitud de las rachas en vector.
#Calcular automáticamente el valor de t.
t <- 0
for (i in 1:length(rachas_false)){
  p_i <- (beta-alpha)*((1-beta+alpha)**i)
  if (length(rachas_false)*p_i < 5){break}
  else {t <- t + 1} }
t <- t+1
#Grupos de longitud > t, se cuentan junto las de longitud t
 for (j in 1:length(rachas_false)) {
  rachas_false[j] <- if(rachas_false[j] < t) rachas_false[j]
  else t}
p_r <- c()
for (r in 1:(t-1)){
p_r <- c(p_r, (beta-alpha)*((1-beta+alpha)**r))}
p_r <- c(p_r, 1-sum(p_r))
# Test chi-cuadrado
```

```
23  rachasfac <- factor(rachas_false, levels=1:t)
24  test <- chisq.test(table(rachasfac), p_r)
25  return(list(estadistico = test$statistic,
26  p.valor = test$p.value, cat.max.long = t))}
```

2.2.3. Test de las Permutaciones (Permutation Test)

Este test se va a aplicar sobre una secuencia de números reales que vamos a dividir en n grupos de t elementos cada uno (es decir, la secuencia es de longitud $n \cdot t$). Los elementos de cada grupo tienen $t!$ posibles órdenes relativos. Además, consideramos que no se pueden dar repeticiones, puesto que al trabajar con números reales vamos a asumir que todos los números son diferentes, por lo que siempre se va a poder determinar cuál es mayor y cuál es menor.

Para realizar la prueba, contamos la frecuencia de cada ordenación relativa en las n subsecuencias de tamaño t. Posteriormente, aplicamos el test chi-cuadrado con $k = t!$ categorías y probabilidad $1/t!$, ya que todas ordenaciones relativas tienen la misma probabilidad.

Para poder determinar dichas frecuencias, asignaremos un número entero a cada ordenación relativa. Sea la ordenación $(U_1, U_2, \ldots, U_t)$ seguiremos los siguientes pasos:

1. Crearemos un vector asociado a la secuencia donde se le asignará el número 1 al elemento de menor valor, el número 2 al segundo más pequeño, etc., estos serán colocados en el mismo orden.
2. Dado el vector de posiciones que obtengamos en el paso 1, formaremos el número entero uniendo todos los elementos del vector (vamos a considerar $t \leq 10$).

Finalmente, una vez se ha asignado un número entero a cada ordenación relativa, nótese que existirán $t!$ números enteros diferentes, aplicaremos el test chi-cuadrado donde cada categoría tiene probabilidad $1/t!$.

Ejemplo. A partir de la secuencia cuyos elementos tienen como parte entera 0 y parte decimal los términos de pi, vamos a separarla en treinta grupos de longitud 3.

0.314, 0.159, 0.265 | 0.358, 0.979, 0.323 | 0.846, 0.264, 0.338
0.327, 0.950, 0.288 | 0.419, 0.716, 0.939 | 0.937, 0.510, 0.582
0.097, 0.494, 0.459 | 0.230, 0.781, 0.640 | 0.628, 0.620, 0.899
0.862, 0.803, 0.482 | 0.534, 0.211, 0.706 | 0.798, 0.214, 0.808
0.651, 0.328, 0.230 | 0.664, 0.709, 0.384 | 0.460, 0.955, 0.058
0.223, 0.172, 0.535 | 0.940, 0.812, 0.848 | 0.111, 0.745, 0.028
0.410, 0.270, 0.193 | 0.852, 0.110, 0.555 | 0.964, 0.462, 0.294
0.895, 0.493, 0.038 | 0.196, 0.442, 0.881 | 0.097, 0.566, 0.593

0.344, 0.612, 0.847 | 0.564, 0.823, 0.378 | 0.678, 0.316, 0.527
0.120, 0.190, 0.914 | 0.564, 0.856, 0.692 | 0.346, 0.348, 0.610.

Tendremos que los números enteros asociados a cada orden relativo son: 312, 231, 312, 231, 123, 312, 132, 132, 213, 321, 213, 213, 321, 231, 231, 213, 312, 231, 321, 312, 321, 321, 123, 123, 123, 231, 312, 123, 132, 123.

Luego, tenemos las siguientes frecuencias de los números enteros (tenemos en cuenta que la probabilidad de cada categoría es $\frac{1}{3!}$.

Núm. ent.	123	132	213	231	312	321
Frec. esp.	5	5	5	5	5	5
Frec. obs.	6	3	4	6	6	5

Por último, hallamos el estadístico del test que se distribuye como una chi-cuadrado de 5 grados de libertad

$$V = \frac{(6-5)^2}{5} + \frac{(3-5)^2}{5} + \frac{(4-5)^2}{5} + \frac{(6-5)^2}{5} + \frac{(6-5)^2}{5} + \frac{(5-5)^2}{5} = 1.6$$

a cuyo valor de V le asociamos un p-valor de 0.9. Por tanto, como el p-valor es menor que el valor del estadístico, se rechaza la hipótesis nula obteniendo que las distribuciones no son iguales, es decir, este test determina que la secuencia no es aleatoria.

```
test_permutaciones <- function(secuencia, t){
  install.packages("combinat")
  library(combinat)
# Dividimos la secuencia en grupos de longitud t
n <- length(secuencia) %/% t #Dividimos en n grupos.
#Matriz donde cada fila es un grupo.
grupos <- matrix(secuencia, nrow = n, byrow = T, ncol = t)
vector_grupos <- split(grupos, row(grupos))
#Asignamos un entero a cada grupo
index_ordenadas <- lapply(vector_grupos, function(x) order(x))
enteros_f <- as.integer(lapply(index_ordenadas, function(v)
paste(v, collapse = "")))
# Test Chi-Cuadrado
posibles_enteros <- permn(1:t)
niveles <- as.integer(lapply(posibles_enteros, function(v)
paste(v, collapse = "")))
entfac <- factor(enteros_f, levels=niveles)
test <- chisq.test(table(entfac))
```

```
19  return(list(estadistico = test$statistic,
20  p.valor = test$p.value))}
```

2.2.4. Máximo del t-test (Maximum of t-test).

Primero dividimos nuestra secuencia en subsecuencias de longitud t y para cada una hallamos su máximo:

$$V_j = \text{máx}\{U_{t\cdot j}, U_{t\cdot j+1}, \ldots, U_{t\cdot j+t-1}\}, \quad 0 \leq j < n.$$

A continuación, tenemos dos formas de proceder:

1. Aplicamos Kolmogorov-Smirnov a la secuencia $(V_0, \ldots, V_{n-1})$, tomando como función de distribución $F(x) = x^t$ para $0 \leq x \leq 1$.
2. Aplicamos el test de Equidistribución a la secuencia $(V_0^t, \ldots, V_{n-1}^t)$, es decir, a la misma secuencia que el anterior con la diferencia de que elevamos a t cada uno de los elementos de la misma, esto se debe a que la probabilidad de que $\text{máx}\{U_1, \ldots, U_t\} \leq x$ es equivalente a la probabilidad de que $U_1 \leq x, \ldots, U_t \leq x$. Por tanto, por la independencia de los U_i tendremos que será el producto de las probabilidades $P(U_i < x) = x$ y por lo tanto es x^t.

   Ejemplo. Tomamos la secuencia formada por elementos cuyas partes enteras son 0 y sus partes decimales son los términos de pi de tres en tres. Dividimos la secuencia en 10 grupos de longitud 9 y obtendremos el valor máximo de cada uno.

   $$\begin{aligned}
   &\text{máx}\{0.314, 0.159, 0.265, 0.358, 0.979, 0.323, 0.846, 0.264, 0.338\} = 0.979\\
   &\text{máx}\{0.327, 0.950, 0.288, 0.419, 0.716, 0.939, 0.937, 0.510, 0.582\} = 0.950\\
   &\text{máx}\{0.097, 0.494, 0.459, 0.230, 0.781, 0.640, 0.628, 0.620, 0.899\} = 0.899\\
   &\text{máx}\{0.862, 0.803, 0.482, 0.534, 0.211, 0.706, 0.798, 0.214, 0.808\} = 0.862\\
   &\text{máx}\{0.651, 0.328, 0.230, 0.664, 0.709, 0.384, 0.460, 0.955, 0.058\} = 0.955\\
   &\text{máx}\{0.223, 0.172, 0.535, 0.940, 0.812, 0.848, 0.111, 0.745, 0.028\} = 0.940\\
   &\text{máx}\{0.410, 0.270, 0.193, 0.852, 0.110, 0.555, 0.964, 0.462, 0.294\} = 0.964\\
   &\text{máx}\{0.895, 0.493, 0.038, 0.196, 0.442, 0.881, 0.097, 0.566, 0.593\} = 0.895\\
   &\text{máx}\{0.344, 0.612, 0.847, 0.564, 0.823, 0.378, 0.678, 0.316, 0.527\} = 0.847\\
   &\text{máx}\{0.120, 0.190, 0.914, 0.564, 0.856, 0.692, 0.346, 0.348, 0.610\} = 0.914
   \end{aligned}$$

   Posteriormente, se aplicará a la secuencia

   $$0.979, 0.950, 0.899, 0.862, 0.964, 0.895, 0.847, 0.914$$

   el test de Kolgomorov-Smirnov o el test de Equidistribución.

En el siguiente cuadro, se presenta la programación en R de la prueba.

```
test_maximos<- function(secuencia, t, metodo){
#t longitud de las subsecuencias.

num_subsec <- length(secuencia) %/% t #número de grupos
#número de elementos distintos en cada grupo.
maximos <- numeric(num_subsec)
for (i in 1:num_subsec){
  maximos[i] <- max(secuencia[(t*i-(t-1)):(t*i)])}
#OPCIÓN 1
if (metodo == 'ks'){
Fdistr<- function(x) {
  ifelse(x <= 0, 0, ifelse(x >= 1, 1, x**t))}
test <- ks.test(maximos, Fdistr)
return(list(estadistico = test$statistic,
p.valor = test$p.value))}
#OPCIÓN 2.
else if (metodo == 'equid_ks'){
test_equidist(maximos, 'ks')}
#OPCIÓN 3.
else if (metodo == 'equid_chi'){
test_equidist(maximos, 'chi')}
}
```

	$t = 5$			$t = 10$		
Secuencia	KS	Equid_Chi	Equid_KS	KS	Equid_Chi	Equid_KS
`runif(1000)`	0.5087	0	0.2922	0.4747	0	0.0001
`runif(1000)`	0.8155	0	0.03430	0.6402	0	0.0029
decimales de π	0.1025	0	0.0237	0.9787	0	0.0025
decimales de $\sqrt{2}$	0.7503	0	0.0289	0.5343	0	0.0015

Tabla 2.1. Aplicación del test a varias secuencias para distintos valores de t mediante los distintos métodos.

2.2.5. Test de Correlación Serial (Serial Correlation Test).

Mediante esta prueba vamos a comprobar si existe alguna relación entre un elemento de la secuencia y su inmediato sucesor. Por ello, se aplicará en

$$(U_0, U_1, U_2, \ldots, U_{n-1}) \text{ y } (U_1, U_2, \ldots, U_{n-1}, U_0).$$

El estadístico

$$C = \frac{n(U_0U_1 + U_1U_2 + \cdots + U_{n-2}U_{n-1} + U_{n-1}U_0) - (U_0 + U_1 + \cdots + U_{n-1})^2}{n(U_0^2 + U_1^2 + \cdots + U_{n-1}^2) - (U_0 + U_1 + \cdots + U_{n-1})^2}$$

presenta el Coeficiente de Correlación Serial. Este va a tomar valores pertenecientes al intervalo $[-1, 1]$ y sirve para medir el grado de dependencia de U_j sobre U_{j+1}.

Seguidamente, para probar la validez del estadístico anterior, se presenta la siguiente demostración.

Demostración. En primer lugar, consideramos

$$C_p = \frac{\mathrm{Cov}(U, V)}{\sqrt{\mathrm{Var}(U)\,\mathrm{Var}(V)}}$$

conocido como el Coeficiente de Correlación de Pearson, ya que nos permite medir la relación entre dos variables.

En nuestro caso, para poder identificar la relación existente entre un elemento y su inmediatamente posterior, donde

$U = (U_0, \ldots, U_{n-1})$ $V = (U_1, \ldots, U_{n-1}, U_0$.

A continuación, calculamos la media y la varianza para ambas variables.

- Variable U:

$$\bar{U} = \frac{1}{n}\sum U_j; \qquad \mathrm{Var}(U) = \frac{1}{n-1}\sum (U_j - \bar{U})^2$$

- Variable V:

$$\bar{V} = \frac{1}{n}\sum V_j = \bar{U}; \qquad \mathrm{Var}(V) = \frac{1}{n-1}\sum (V_j - \bar{V})^2 = \mathrm{Var}(U).$$

Seguidamente, obtemos que la covarianza, entre U e V, viene dada como

$$\mathrm{Cov}(U, V) = \frac{1}{n-1}\sum (U_j - \bar{U})(V_j - \bar{V}) =$$

$$= \frac{1}{n-1}\left[\sum U_j V_j - \sum U_j \bar{V} - \sum V_j \bar{U} + \sum \bar{U}\bar{V}\right].$$

Sustituyendo y simplificando en la expresión de Pearson, obtenemos lo siguiente:

$$C_p = \frac{\frac{1}{n-1}\left[\sum U_j V_j - \sum U_j \bar{V} - \sum V_j \bar{U} + \sum \bar{U}\bar{V}\right]}{\sqrt{\frac{1}{n-1}\sum (U_j - \bar{U})^2 \cdot \frac{1}{n-1}\sum (V_j - \bar{V})^2}} =$$

$$= \frac{\sum U_j V_j - \bar{V}\sum U_j - \bar{U}\sum V_j + n\bar{U}\bar{V}}{\sqrt{\sum (U_j - \bar{U})^2 \cdot \sum (V_j - \bar{V})^2}} =$$

$$= \frac{\sum U_j V_j - n\bar{V}\bar{U} - n\bar{U}\bar{V} + n\bar{U}\bar{V}}{\sqrt{\left[\sum U_j^2 - \sum 2U_j\bar{U} + \sum \bar{U}^2\right] \cdot \left[\sum V_j^2 - \sum 2V_j\bar{V} + \sum \bar{V}^2\right]}} =$$

$$= \frac{\sum U_j V_j - n\bar{U}\bar{V}}{\sqrt{\left[\sum U_j^2 - 2n\bar{U}^2 + n\bar{U}^2\right] \cdot \left[\sum V_j^2 - 2n\bar{V}^2 + n\bar{V}^2\right]}} =$$

A continuación, vamos a poner la expresión anterior en términos de U.

$$\frac{\sum U_j U_{j+1} - n \cdot \frac{1}{n}\sum U_j \cdot \frac{1}{n}\sum U_{j+1}}{\sqrt{\left[\sum U_j^2 - n\left(\frac{1}{n}\sum U_j\right)^2\right] \cdot \left[\sum U_{j+1}^2 - n\left(\frac{1}{n}\sum U_{j+1}\right)^2\right]}} =$$

$$= \frac{n\sum U_j U_{j+1} - \sum U_j \sum U_{j+1}}{\sqrt{n^2\left[\sum U_j^2 - \frac{1}{n}\left(\sum U_j\right)^2\right] \cdot \left[\sum U_{j+1}^2 - \frac{1}{n}\left(\sum U_{j+1}\right)^2\right]}} =$$

Finalmente, llegamos a nuestro Coeficiente de Correlación Serial.

$$\frac{n(U_0U_1 + \cdots U_{n-1}U_0) - (U_0 + \cdots U_{n-1})(U_1 + \cdots U_{n-1} + U_0)}{\sqrt{\left[n(U_0^2 + \cdots U_{n-1}^2) - (U_0 + \cdots + U_{n-1})^2\right]\left[n(U_1^2 + \cdots + U_0^2) - (U_1 + \cdots + U_0)^2\right]}} =$$

$$= \frac{n(U_0U_1 + \cdots U_{n-1}U_0) - (U_0 + \cdots U_{n-1})^2}{n(U_0^2 + \cdots U_{n-1}^2) - (U_0 + \cdots + U_{n-1})^2}.$$

■

Cuando calculamos nuestro coeficiente, tenemos tres posibilidades:

- Valor de C es mayor que 0, la correlación es positiva, es decir, a medida que aumenta una variable aumenta la otra también.
- Valor de C igual a 0, la correlación lineal entre las variables es nula.
- Valor de C es menor que 0, la correlación es negativa, por lo que a medida que aumenta una de las variables la otra va disminuyendo.

Si la secuencia fuera aleatoria, realmente no deberíamos esperar relación entre un término y el siguiente. En nuestro caso, diremos que la secuencia se considera aleatoria cuando C se encuentre en el intervalo: $[\nu_n - 2\sigma_n^2, \nu_n - 2\sigma_n^2]$ con

$$\nu_n = \frac{-1}{n-1} \quad \text{y} \quad \sigma_n^2 = \frac{n^2}{(n-1)^2(n-2)}.$$

Este intervalo fue propuesto por D. Knuth en el año 1900. Por otro lado, se resalta que también se podría estudiar, con este test, la relación que existe entre un elemento y el q-ésimo posterior $(0 < q < n)$.

Ejemplo. Tomamos la secuencia de longitud 10 formada por números con parte entera 0 y parte decimal los términos de pi de 3 en 3. En este caso, tenemos que diferenciar las dos siguientes secuencias:

1. 0.314, 0.159, 0.265, 0.358, 0.979, 0.323, 0.846, 0.264, 0.338, 0.327.
2. 0.159, 0.265, 0.358, 0.979, 0.323, 0.846, 0.264, 0.338, 0.327, 0.314.

En este caso, obtenemos que nuestro intervalo es $[-0.1587, -0.063]$.

Seguidamente, procederemos a calcular el estadístico ara comprobar si su valor pertenece al intervalo dado.

$$C = \frac{10(0.314 \cdot 0.159 + \cdots + 0.327 \cdot 0.314) - (0.314 + \cdots + 0.327)^2}{10(0.314^2 + \cdots + 0.327^2) - (0.314 + \cdots + 0.327)^2} =$$

$$= \frac{10 \cdot 1.649528 - 17.413929}{10 \cdot 2.391621 - 17.413939} = -0.1412810366.$$

Observamos que el es coeficiente de correlación sí pertenece, por lo que se concluye que, según el test de Correlación Serial, esta secuencia es aleatoria.

Se muestra, a continuación, la implementación del contraste con R.

```
test_correlacion <- function(secuencia){
secuencia_1 <- secuencia
#Le quito U_0 de delante y lo pongo al final.
secuencia_2 <- c(secuencia_1[-1], secuencia[1])
n <- length(secuencia)
#Estadístico C.
C <- cor(secuencia_1, secuencia_2)
#Cálculo intervalo.
v_n <- -1/(n-1)
sigma_n <- n^2 / ((n-1)^2 * (n-2))
intervalo_inf <- v_n - 2*sigma_n^2
intervalo_sup <- v_n + 2*sigma_n^2
#Comprobación contenido en intervalo.
if (C >= intervalo_inf && C <= intervalo_sup){
  test <- c("la secuencia es aleatoria")
  return(list(Coeficiente = C, conclusión = test))}
else{
test <- c("la secuencia no es aleatoria")
return(list(Coeficiente = C, conclusión = test))}}
```

2.2.6. Test en Subsecuencias (Test on Subsequences)

El test consiste en dividir la secuencia en grupos de tamaño t. A continuación, se crearán las subsecuencias formadas por los elementos cuyas posiciones en los grupos coinciden, es decir, todos los elementos que están en la posición t

de cada grupo formarán una secuencia (el primer elemento pertenece al grupo 1, el segundo al grupo 2,...).

Luego, habrá que comprobar si existe alguna relación dentro de cada una de las t subsecuencias obtenidas anteriormente. Esto lo haremos aplicando a cada una de ellas algunos de los tests expuestos previamente. Para poder afirmar que la secuencia original es aleatoria por una prueba determinada, será imprescindible que todas las subsecuencias pasen de manera satisfactoria la respectiva prueba.

Aquí, se muestra un ejemplo de implementación computacional de la prueba con *R*. (La implementación, en este caso, consiste en la obtención de las distintas subsecuencias. Para determinar aleatoriedad aplicar los tests descritos anteriormente.)

```
test_subsecuencias <- function(secuencia, t){
#La máquina nos pide introducir los elementos de t en t.
matriz_vectores <- matrix(secuencia, nrow = t,
ncol = length(secuencia)/t)
# Subsecuencia formada por el t elemento de cada grupo.
subsecuencia_posicion_t <- matriz_vectores[t,]
subsecuencia_posicion_t}
```

2.3. Aplicación de los contrastes a secuencias de reales

2.3.1. Descripción secuencias

- π **decimal**: Esta secuencia de longitud 1000 consiste en una serie de números reales en la que las partes enteras son cero y las partes decimales corresponden a las cifras del número pi agrupadas de tres en tres.

$$0.314, 0.159, 0.265, 0.358, 0.979, \ldots$$

- $\sqrt{2}$ **decimal**: La secuencia contiene mil elementos cuyas partes enteras son cero, mientras que las partes decimales corresponden a las cifras decimales de la raíz cuadrada de 2. Estas cifras decimales se agrupan de tres en tres para formar los números de la secuencia.

$$0.141, 0.421, 0.356, 0.237, 0.309, \ldots$$

- **money:** Los 257 elementos de esta secuencia corresponden con los datos de un indicador económico que mide la velocidad de circulación del dinero en la categoría M2 de la oferta monetaria. Se tiene que el M2 es una medida ampliada de la oferta monetaria que incluye el efectivo en manos del público, depósitos a la vista, depósitos de ahorro y ciertos activos financieros de corto

plazo. Los datos fueron recuperados de la página web del Federal Reserve Bank of St. Louis.

$$1.773, 1.789, 1.773, 1.779, 1.817, 1.797, \ldots$$

- **AMBNS:** Los 214 componentes de esta secuencia pertenecen a los datos de la base monetaria ajustada de St. Louis. Los datos se localizan en la página web del Federal Reserve Bank of St. Louis.

$$0.4874, 0.4834, 0.4914, 0.4993, 0.4954, \ldots$$

- **Runif de R:** Esta secuencia ha sido generada utilizando la función "runif"de R contiene números reales entre cero y uno. Se ha establecido la semilla "seed(1)"para poder reproducir los valores.

2.3.2. Tabla de contrastes en la batería de secuencias

Secuencia	Equid. (KS)	Huecos	Permut. $t = 4$	Máximo (KS) $t = 6$	Correl.
π decimal	0.7183	0.2424	0.0809	0.3723	-0.0369
$\sqrt{2}$ decimal	0.3696	0.2289	0.7961	0.0924	-0.0105
money	0	0.3575	2.1126e-24	0	0.9831
AMBNS	0	0.2381	5.2316e-26	0	0.9231
`runif(1000)`	0.5927	0.2426	0.1876	0.3015	0.0342
`runif(100)`	0.6058	0.2231	0.3066	0.6748	0.0116

Tabla 2.2. Resultados de la aplicación de los tests a una batería de secuencias reales. Se indica el p-valor.

En la tabla anterior podemos observar los p-valores de la aplicación de diferentes contrastes a las secuencias introducidas anteriormente.

En cuanto a los resultados de la prueba de correlación, se tiene que el valor devuelto no es un p-valor, si no el valor del coeficiente de correlación. Para todas las secuencia planteadas se ha obtenido un coeficiente que no pertenece al intervalo dado por el contraste, dando como resultado que ninguna es aleatoria. Por tanto, concluimos que el test de las correlaciones es muy restrictivo, ya que ni la secuencia generada aleatoriamente por R ha sido capaz de superarala.

Finalmente, destacamos los resultados obtenidos por las secuencias de los decimales de pi y de la raíz de dos, ya que han superado todos los contrastes (salvo el de las correlaciones). Luego, las cifras de ambos números exhiben un alto grado de aleatoriedad.

3

Análisis del criterio de selección de números en Euromillones

Este capítulo tiene como objetivo analizar el criterio de selección de números de los alumnos al rellenar un boleto de Euromillones. Se realizó una actividad en la que se les proporcionó a los estudiantes la oportunidad de elegir los números que conformarían su apuesta, y ahora nos disponemos a evaluar si sus selecciones fueron aleatorias o si se identifican patrones en las mismas.

Recordamos, que una apuesta consiste en la elección de cinco números entre los 50 posibles. Para poder estudiar la presencia de patrones, consideraremos que cada dígito seleccionado va a tomar el valor 1, mientras que aquellos que no lo han sido valdrán cero.

Sin embargo, si queremos estudiar estos datos tenemos que considerar que los elementos de la muestra no están bajo la hipótesis de equiprobabilidad, ya que al solo poder escoger cinco números de 50, tendremos que la probabilidad de que un número concreto sea elegido es $\frac{5}{50}$, pero que no son independientes. De esta forma nos fallan las hipótesis que asumíamos de uniformidad y de independencia.

Por este motivo, en este capítulo tendremos que plantear nuevos contrastes que se adapten a esta condición.

3.1. Nuevos contrastes para datos bidimensionales

3.1.1. Patrón según pequeñas estructuras

Se plantea analizar la matriz según la diferentes submatrices de dimensión $n \times m$ que se encuentran en ella. De esta forma, se estudiarán las secuencias formadas por la unión de las filas de las submatrices.

```
submatrices <- function(matriz, n, m) {
  filas <- nrow(matriz)
  columnas <- ncol(matriz)
  submatrices <- list()
  numeros <- numeric()
```

```
  for (i in 1:(filas-n+1)) {
    for (j in 1:(columnas-m+1)) {
      submatriz <- matriz[i:(i+n-1), j:(j+m-1)]
      # agregar solo si es del tamaño deseado
      if (nrow(submatriz) == n && ncol(submatriz) == m) {
      numero <- as.numeric(paste(submatriz, collapse = ""))
      submatrices[[length(submatrices) + 1]] <- submatriz
      numeros <- c(numeros, numero)}}}
  return(list(submatrices, numeros))}
```

Para este tipo de datos, hemos planteado el test de las submatrices. En este se obtendrá la frecuencia de cada submatrices para, posteriormente, aplicar la prueba chi-cuadrado. Para obtener las probabilidades, llevaremos a cabo la siguiente simulación:

```
as.vector(replicate(30000,{
  vector <- numeric(54)
  sorteo <- sample(1:50, size=5)
  vector[sorteo] <- 1
  boleto <- matrix(vector, nrow=9, ncol=6, byrow=F)
  boleto[6:9,6] <- 2
  submatrices(boleto, 2, 2)[[2]]
  }
)) -> simul
simulfac <- factor(simul, c(0000, 0001, 0010, 0011, 0100, 0101,
0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111))
round(table(simulfac)/length(simulfac), 7)
```

```
simulfac
      0       1      10      11     100     101     110     111
0.6470417 0.0771556 0.0770833 0.0069806 0.0770417 0.0072074 0.0070537 0.0005231
   1000    1001    1010    1011    1100    1101    1110    1111
0.0768019 0.0071389 0.0073287 0.0005083 0.0071657 0.0004944 0.0004593 0.0000157
```

Figura 3.1. Probabilidades por simulación del Tests de las submatrices para el problema de Euromillones.

A continuación, presentamos la implemetación del test de las submatrices en R para el caso de Euromillones.

```
test_submatrices <- function(submatrices){
submatricesfac <- factor(submatrices, levels = c(0000, 0001,
0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011,
1100, 1101, 1110, 1111))
#probabilidades calculadas por simulación
p_r <- c(0.6470417, 0.0771556, 0.0770833, 0.0069806, 0.0770417,
```

```
0.0072074, 0.0070537, 0.0005231, 0.0768019, 0.0071389,0.0073287,
0.0005083, 0.0071657, 0.0004944, 0.0004593, 0.0000157)
test <- chisq.test(table(submatricesfac), p_r)
return(list(estadistico = test$statistic,
p.valor = test$p.value))}
```

3.1.2. Test de las filas

Este contraste consiste en contar el número de filas en una matriz que tienen, al menos, un valor igual a 1 y luego comprobar si ese valor coincide con el esperado. Con el propósito de calcular la probabilidad de obtener el número de filas obtenido, se llevó a cabo la siguiente simulación en *R*.

```
replicate(30000,{
  vector <- numeric(54)
  sorteo <- sample(1:50, size=5)
  vector[sorteo] <- 1
  boleto <- matrix(vector, nrow=9, ncol=6, byrow=F)
  sum(apply(boleto, FUN=sum, MARGIN=1) != 0) #margin=1 por filas
  }
) -> simul
round(table(simul)/length(simul), 7)
```

De esta manera, obtuvimos las probabilidades:

Filas	1	2	3	4	5
Probabilidades	0.0000333	0.0082333	0.1646000	0.5147333	0.3124000

```
test_filas <- function(matriz){
numfilas_con_1 <- sum(rowSums(matriz == 1) > 0)
p_r <- c(0.0000333, 0.0082333, 0.1646000, 0.5147333, 0.3124000)
#Calculamos p-valor
p_ordenado <- sort(p_r)
p_valor <- sum(p_ordenado[p_ordenado <= p_r[numfilas_con_1]])
return(p_valor)}
```

3.1.3. Test de las columnas

Este test se comporta de manera similar al anterior, pero en lugar de contar el número de filas contaremos el número de columnas. Al igual que en el caso anterior, realizaremos una simulación para obtener las probabilidades correspondientes.

```
replicate(30000,{
  vector <- numeric(54)
  sorteo <- sample(1:50, size=5)
  vector[sorteo] <- 1
  boleto <- matrix(vector, nrow=9, ncol=6, byrow=F)
  #margin=2 suma por columnas
  sum(apply(boleto, FUN=sum, MARGIN=2) != 0)
  }
) -> simul
round(table(simul)/length(simul), 7)
```

Obtuvimos las siguientes probabilidades:

columnas	1	2	3	4	5
Probabilidades	0.0003000	0.0443667	0.3612333	0.4900000	0.1041000

```
test_columnas <- function(matriz){
numcol_con_1 <- sum(colSums(matriz == 1) > 0)
p_r <- c(0.0003000, 0.0443667, 0.3612333, 0.4900000, 0.1041000)
#Calculamos p-valor
p_ordenado <- sort(p_r)
p_valor <- sum(p_ordenado[p_ordenado <= p_r[numcol_con_1]])
return(p_valor)}
```

Finalmente, vamos a aplicar los tests presentados anteriormente a una batería de resultados del sorteo de Euromillones.

Matriz	Submatrices	Filas	Columnas
Apuesta aleatoria 1	0.3250	0.4853	1
Apuesta aleatoria 2	0.3522	0.4853	0.5100
Apuesta aleatoria 3	0.3250	0.9999	0.5100
Apuesta aleatoria 4	0.3522	0.1729	1
Apuesta aleatoria 5	0.3522	0.1729	0.5100
Apuesta aleatoria 6	0.3250	0.9999	0.5100

Tabla 3.1. Resultados de la aplicación de los tests a una batería de Euromillones rellenadas aleatoriamente por R.

Matriz	Submatrices	Filas	Columnas
Apuesta alumno 1	0.3522	0.0083	0.5100
Apuesta alumno 2	0.3380	0.9999	1
Apuesta alumno 3	0.3522	0.4853	0.1488
Apuesta alumno 4	0.3522	3.33e-05	0.1488
Apuesta alumno 5	0.3259	0.4853	0.1488
Apuesta alumno 6	0.3521	0.4853	0.0447

Tabla 3.2. Resultados de la aplicación de los tests a una batería de Euromillones de los alumnos de 2º curso del Grado de Matemáticas.

Tras aplicar los tests definidos a diferentes ejemplos de Euromillones, hemos observado que todos los que han sido generados aleatoriamente han conseguido superar todas las pruebas satisfactoriamente. Sin embargo, entre los realizados por los alumnos, existen casos en los que las pruebas han identificado diferentes patrones. Concretamente, tenemos las apuestas por el alumno 1 (todos los unos en la primera submatriz de 2×2 y el quinto en la posición (2,5)), por el alumno 4 (todos los unos en la primera fila) y por el alumno 6 (Todos los unos concretados en las dos primeras columnas).

Por tanto, podemos concluir que los tests que hemos definido han resultado bastante útiles, ya que las apuestas con patrones definidos han sido detectadas por, al menos, un test.

Conclusiones

En este trabajo se ha estudiado la presencia de aleatoriedad, es decir, la ausencia de patrones en datos unidimensionales (secuencias) y bidimensionales (matrices).

Determinar la presencia de patrones mediante este método no es tarea sencilla, ya que gran parte de los datos no aleatorios, como hemos visto, son capaces de superar con éxito contrastes. Esto se debe a que según el test que estemos utilizando tendremos un nivel de restrictividad diferente. Por ejemplo, la prueba de chi-cuadrado determina como aleatoria a la secuencia de Thue-Morse, puesto que la frecuencia de ceros y unos es la misma. Sin embargo, dicha serie presenta un patrón por definición. Por esto motivo, se tiene que tener mucho cuidado y no podemos considerarla aleatoria habiendo aplicado un test solamente, es decir, será necesario que varios contrastes lo determinen.

Concretamente, tomamos una batería de datos, entre los que se pueden encontrar secuencias aleatorias y no aleatorias, y les aplicamos todos los contrastes para identificar la veracidad de los resultados. Con los valores obtenidos, nos fijamos que hay algunos contrastes que son demasiado restrictivos, como por ejemplo el test de correlación, que después de haberlo aplicado a numerosas secuencias (aleatorias y no aleatorias), ninguna consiguió superar el test.

El análisis en muestras bidimensionales es mucho más restrictivo, ya que cada matriz deberá superar cada uno de los tests, por lo menos, cinco veces (cada vez con una subsecuencia distinta). Conseguir en todos los casos p-valores favorables es muy complicado, por lo que es muy difícil que una matriz pueda considerarse como aleatoria mediante la técnica de aplicación de contrastes.

En nuestro caso, como trabajamos con Euromillones, además de ser muestras bidimensionales, se nos presentaba el problema de la no equiprobabilidad entre los elementos de la muestra. Cuando adaptamos algunos de los tests presentados en el primer capítulo, los resultados obtenidos no podían considerarse fiables, ya que cuando la matriz era convertida en secuencia, se tenía una longitud demasiado pequeña para estos contrastes. Como alternativa, decidimos definir nuevos contrastes para poder evaluar nuestros boletos de Euromillones.

Concretamente, al aplicarlos sobre las matrices, obtuvimos los resultados esperados, ya que en la mayoría de los casos se pudo concluir que la matriz no había sido generada aleatoriamente.

En definitiva, este trabajo nos ha permitido profundizar en el análisis de patrones en las secuencias y en la implementación de técnicas estadśticas avanzadas, lo que puede resultar de gran utilidad en múltiples campos de aplicación.

Bibliografía

[1] *The Art of Computer Programming Vol. 1*, third Edition, Donald E. Knuth (1997). Addison-Wesley, Boston.

[2] *The Art of Computer Programming Vol. 2*, third Edition, Donald E. Knuth (1997). Addison-Wesley, Boston.

[3] Pearson K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, Series 5, 50, 157–175.

[4] Kolmogorov, A. (1933). *Sulla determinazione empirica di una legge di distribuzione. Giornale dell'Istituto Italiano degli Attuari, 4*(1), 83-91.

[5] Marsaglia, G. (1968). Random numbers fall mainly in the planes. *Proceedings of the National Academy of Sciences, 61*(1), 25-28.

[6] Kendall, M. G. (1949). *The Advanced Theory of Statistics*, Volume 2: Inference and Relationship. Griffin.

[7] Kendall, M. G. (1948). The treatment of ties in the calculation of the runs up and down in a sequence. *Annals of Mathematical Statistics, 19*(1), 1-10.

[8] Carlitz, L. & von Mises, R. (1956). The birthday problem. *Journal of the American Statistical Association, 51*(273), 365-367.

[9] Kendall, M. G. & Babington-Smith, B. (1938). The problem of m rankings. `Journal of the Royal Statistical Society, 101`, 147–166.

[10] Kendall, M. G. & Babington-Smith, B. (1939). Supplement to "The problem of m rankings". `Journal of the Royal Statistical Society, 6`, 51–61.

[11] Hartley, H. O. (1952). A Serial Test for Randomness. *Journal of the American Statistical Association, 47*(260), 575-584.

[12] Doob, J. L. (1940). Randomness and the Ranks of Matrices. *The Annals of Mathematical Statistics, 11*(2), 211-214.

[13] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Journal of Economic Entomology, 38*(6), 617-627.

[14] Kendall, M. G. (1948). *The Advanced Theory of Statistics.* London: Griffin.

[15] MacWilliams, F. J. (1957). Probability problems in the theory of error-correcting codes. *Bell System Technical Journal, 36* (2), 429-433.

[16] *Federal Reserve Bank of St. Louis. (s.f.). M2 Velocity.* Recuperado de https://fred.stlouisfed.org/series/M2V [10-05-2023]

[17] *Federal Reserve Bank of St. Louis. (s.f.). Assets: Total Assets: Total Assets (Less Eliminations From Consolidation): Wednesday Level.* Recuperado de https://fred.stlouisfed.org/series/AMBNS

[18] R Core Team. *R: A language and environment for statistical computing.* 2023. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

Todos los códigos de este trabajo se encuentran disponibles en el siguiente notebook de Jupyter:

[19] https://colab.research.google.com/drive/1FDsXO3oPA063cqqlpNa93VwqqyrYUxpq?usp=sharing

RANDOMNESS TESTS: DETECTION OF NON-RANDOM PATTERNS IN DATA SAMPLES

Shaina Daryanani Hassani
Facultad de Ciencias · Sección de Matemáticas
Universidad de La Laguna
alu0101353104@ull.edu.es

ABSTRACT

In this paper, different contrasts will be presented to determine if a one-dimensional sample has been generated by a random process or not.
It is established that it must pass most of the tests to be considered random. In addition, methods are proposed to apply these contrasts to two-dimensional samples, such as matrices. All these contrasts have been computationally implemented using functions in R to determine the presence of patterns or not in an automated way.

OBJECTIVES

- Determine if sequences with integer values are random.
- Identify randomness in sequences of real numbers.
- Randomness in situations of two-dimensionality without equiprobability.
- Computational implementation in R of all the tests.

UNIDIMENSIONAL TESTS

Different tests are presented to analyze the existence of patterns in entire sequences. Above all, they consist of the identification of repetitions, the division into groups comparing their characteristics (number of different elements, frequency of each one,...), among other things. In most of them, we use the chi-square test to compare the expected values with the observed ones.

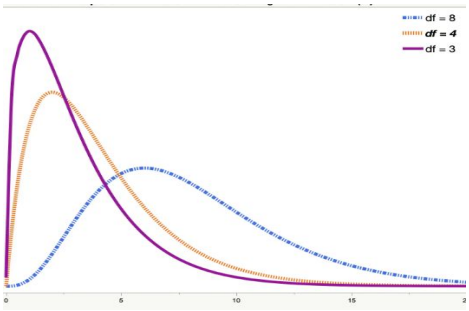

Figure 1: Chi-Square Distributions with Different Degrees of Freedom (df).

We study tests for sequences whose elements are between zero and one. In this case, patterns are sought regarding the relationship of a number with its subsequent number, the division by groups identifying their relative order, etc. Specifically, the Kolmogorov-Smirnov test is introduced to verify whether the distribution followed by the elements of the sequence coincides with a known distribution.

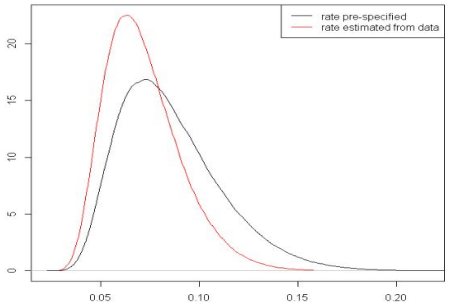

Figure 2: Empirical and theoretical distributions to be compared by the Kolmogorov-Smirnov test.

TWO-DIMENSIONALITY

We studied two-dimensionality and non-equiprobability in the following problem:

EUROMILLONES

Figure 3: Random bet for euromillions.

We convert the bet into a matrix of zeros and ones to analyze the patterns by rows, columns, diagonals, and 2x2 submatrixes.

Specifically, we analyze the patterns of euromillions bets where the probability of chosen numbers is much lower than that of those not chosen.

REFERENCES

[1] *The Art of Computer Programming Vol. 2*, third Edition, Donald E. Knuth (1997). Addison-Wesley Boston.

TRABAJO FIN DE GRADO, Convocatoria de junio, 2023

Sección de Matemáticas
Universidad de La Laguna