Análisis de correlación moderno: ¿Qué alternativas existen para la correlación de Pearson?

Ana Belén Pazos Ruiz
Máster universitario en Bioinformática y bioestadística UOC-UB
Área de Estadística y bioinformática 16

Alexandre Sánchez Pla

05 de junio de 2018


Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Análisis de correlación moderno: ¿Qué alternativas existen para la correlación de Pearson?*
Nombre del autor:	*Ana Belén Pazos Ruiz*
Nombre del consultor/a:	*Alexandre Sánchez Pla*
Nombre del PRA:	*Alexandre Sánchez Pla*
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	*Plan de estudios del estudiante*
Área del Trabajo Final:	*Estadística y Bioinformática 16*
Idioma del trabajo:	*Castellano*
Palabras clave	*Análisis de correlación, independencia, asociación no lineal.*

Resumen del Trabajo (máximo 250 palabras):

El coeficiente de correlación de Pearson encuentra la dependencia lineal entre dos variables de forma sencilla y con un bajo coste computacional, pero teniendo que cumplir unos supuestos difíciles de asumir.

Actualmente, hay un elevado número de coeficientes que identifican la asociación entre variables, entre ellos se encuentran los coeficientes CorGC, RDC, dCor o MIC, este último bautizado como la correlación del s. XXI.

En este trabajo se han analizado estos coeficientes para encontrar el más completo, distinguiendo cuáles detectan más tipos de asociaciones con un bajo coste computacional y cumpliendo las siete propiedades fundamentales propuestas por Rényi. Fueron puestos a prueba con ocho tipos de asociaciones diferentes (pseudo-aleatoria, lineal, cuadrática, cúbica, exponencial, sinusoidal, escalón y círculo) además de con una base de datos génica en el que se precisaba saber que genes se expresaban significativamente. También se proporciona una aplicación sencilla para calcular cuatro de ellos.

A excepción de la Correlación de Pearson, el resto detectan asociaciones no lineales. Siendo RDC y dCor los únicos capaces de trabajar con datos multidimensionales. Se ha comprobado que CorGC y dCor son los que requieren de un mayor tiempo de ejecución, por el contrario, la r de Pearson y RDC los que tienen el coste computacional más bajo. El único que cumple con todas las propiedades de Rényi es RDC, además de ser el que en más ocasiones ha encontrado dependencia en los datos simulados y reales. Por lo que se erige como el coeficiente idóneo a sustituir la Correlación de Pearson.

Abstract (in English, 250 words or less):

The Pearson correlation coefficient finds the linear dependence between two variables with a straightforward and low computational cost, but having to meet some assumptions difficult to assume.

Currently, there is a high number of coefficients that identify the association between variables, among them are the coefficients CorGC, RDC, dCor or MIC, the latter named as the correlation of s. XXI.

In this work have been analysed these coefficients to find the most complete, distinguishing which detect more types of associations with a low computational cost and fulfilling the seven fundamental properties proposed by Rényi. They were tested with eight different types of associations (pseudo-random, linear, quadratic, cubic, exponential, sinusoidal, step and circle), as well as a gene database in which it was necessary to know which genes were expressed significantly. It also provides a straightforward application to calculate four of them.

With the exception of the Pearson Correlation, the other detects non-linear associations. Being RDC and dCor the only ones which are able to work with multidimensional data. It has been checked that CorGC and dCor are those that require a longer execution time, on the other side, the r Pearson's and RDC have the lowest computational cost. The only one that complies with all Rényi's properties is RDC, in addition to the one that has found dependency on simulated and real data. For this, it stands as the ideal coefficient to replace the Pearson correlation.

Índice

Lista de figuras

Lista de tablas

1. Introducción

1.1 Contexto y justificación del Trabajo

Poder identificar relaciones entre diferentes variables en los conjuntos de datos es uno de los cálculos más realizados en el ámbito científico, así como en el empresarial.

La r de Pearson desde su descubrimiento hasta la actualidad es uno de los estadísticos más utilizados por la sociedad para medir la asociación o relación entre dos variables, gracias a que es fácil de calcular y que es uno de los estadísticos más conocidos. Pero el coeficiente de correlación de Pearson sólo se puede utilizar para dos variables, cuando la relación buscada es lineal, además, los datos deben de ser continuos y distribuidos normalmente. Debido a estas restricciones en muchas ocasiones se está utilizando erróneamente el estadístico para buscar asociación entre variables.

Además del coeficiente r hay más estadísticos que miden la asociación entre variables, como la correlación de Sperman, la D de Hoeffding, la Correlación de distancia, el Coeficiente de información máxima, el Coeficiente de dependencia aleatorio o la Correlación a lo largo de una curva de generación entre muchas otras. Todas con sus ventajas e inconvenientes.

En este proyecto se procede a analizar algunos de estos coeficientes como alternativa a la Correlación de Pearson, generando una comparativa entre los diferentes estadísticos tanto teórica como prácticamente. Dicho estadístico debe de detectar una amplia gama de relaciones entre variables con un menor número de restricciones de uso.

Además, se ha creado una aplicación web en la que, dadas dos bases de datos, se calcula de una forma simple diferentes medidas de correlación, adaptándose a las necesidades del usuario y de los resultados obtenidos.

1.2 Objetivos del Trabajo

Los objetivos generales del trabajo son:

a. Encontrar una alternativa a la correlación de Pearson para estudios de alto rendimiento.
b. Desarrollar un software libre con el que la población pueda analizar si las variables están relacionadas.

Debido a que estos objetivos son muy generales se han desagregado en otros más específicos.

a. Alternativa a la correlación de Pearson:
    1. Identificar los diferentes estadísticos que existen actualmente para la identificación de asociación entre variables.
    2. Realizar una comparativa entre los estadísticos identificando en que casuísticas es más adecuado aplicar cada uno.
    3. Indicar cuales son las ventajas y desventajas de los estadísticos.
    4. Encontrar el estadístico óptimo para cualquier relación entre variables.
b. Desarrollo de software libre:
    1. Determinar si hay disponibles actualmente paquetes en R que calculen los diferentes estadísticos estudiados.
    2. Efectuar una comparación práctica con una base de datos real.
    3. Implementar con R una aplicación web, en que se apliquen los diferentes coeficientes estudiados y sea accesible a la población.
    4. Disponer el software a la población interesada.

1.3 Enfoque y método seguido

Se pueden realizar varias estrategias para llevar a cabo el trabajo, pero lo primero es investigar cuales son los estadísticos que hay actualmente para calcular las correlaciones entre variables cuantitativas. Esta fase no hace falta que sea muy detallada, sólo es una primera toma de contacto. Ya que en un primer análisis se detectaron 7 candidatos, además de la correlación de Pearson, a ser analizados.

A partir de aquí, se puede plantear el trabajo de varias formas:

- Primero desarrollar la parte teórica del proyecto (búsqueda de teoría, comparativa, etc.) y después la práctica (funciones disponibles en R, creación del paquete, obtención de una base de datos) incluida la creación del software y, por último, analizar los resultados obtenidos para identificar cual es el estadístico óptimo.

- Ir haciendo ambas partes del proyecto (teórica y práctica) a la vez.

- Una opción mixta entre las anteriores, se desglosaría las tareas por estadístico (teoría, ventajas/desventajas, funciones disponibles en R). Una vez hecha la investigación, centrarse en la parte práctica del proyecto. Analizando los resultados para llegar a la conclusión del estudio y por último realizar el software para hacerlo público.

La opción seleccionada fue la última, ya que se creaban subtareas, lo que ayudaba a centrar al investigador, ajustando los tiempos y proporcionando un soporte a la parte práctica, que podría ser la parte más complicada del proyecto, puesto que se tenía que generar un paquete en R y la aplicación web.

De esta forma, si por alguna razón fuese imposible analizar todos los estadísticos encontrados, se podría decidir cuántos entran en el estudio, sin alargar la parte teórica del proyecto y dando el tiempo suficiente para realizar la parte práctica. También da la posibilidad de priorizar los estadísticos a analizar, investigando primero los más completos y, tal vez complejos, y dejando para los últimos aquellos que ya se sabe, tras el primer análisis, que no van ser los óptimos debido a las restricciones o inconvenientes que tienen, o si es necesario descartar alguno.

1.4 Planificación del Trabajo

Debido al tema del trabajo, los recursos necesarios para realizar el trabajo son acceso a la documentación de los estadísticos y del software (libros, artículos, tesis, páginas web, …), el software libre R y RStudio, junto con el paquete Shiny, así como de servidores y repositorios donde alojar los diferentes productos obtenidos.

Respecto a las tareas, se pueden desagregar en las siguientes:

1. Identificar que estadísticos existen actualmente que analicen la relación entre variables (Objetivo Específico a.1. -OE-, con una duración de 8 horas -DT-).

2. Cada estadístico, se deshace en las siguientes subtareas (DT entre 5 y 9 horas dependiendo de la complejidad):

    i. Establecer en qué condiciones se puede utilizar (OE a.2).

    ii. Identificar las ventajas y desventajas (OE a.3).

    iii. Encontrar, si existe, una función o paquete en R que calcule el estadístico, y si no existe, programarlo (OE b.1).

    iv. Aplicarlo en una base de datos real (OE b.2, DT 1 horas).

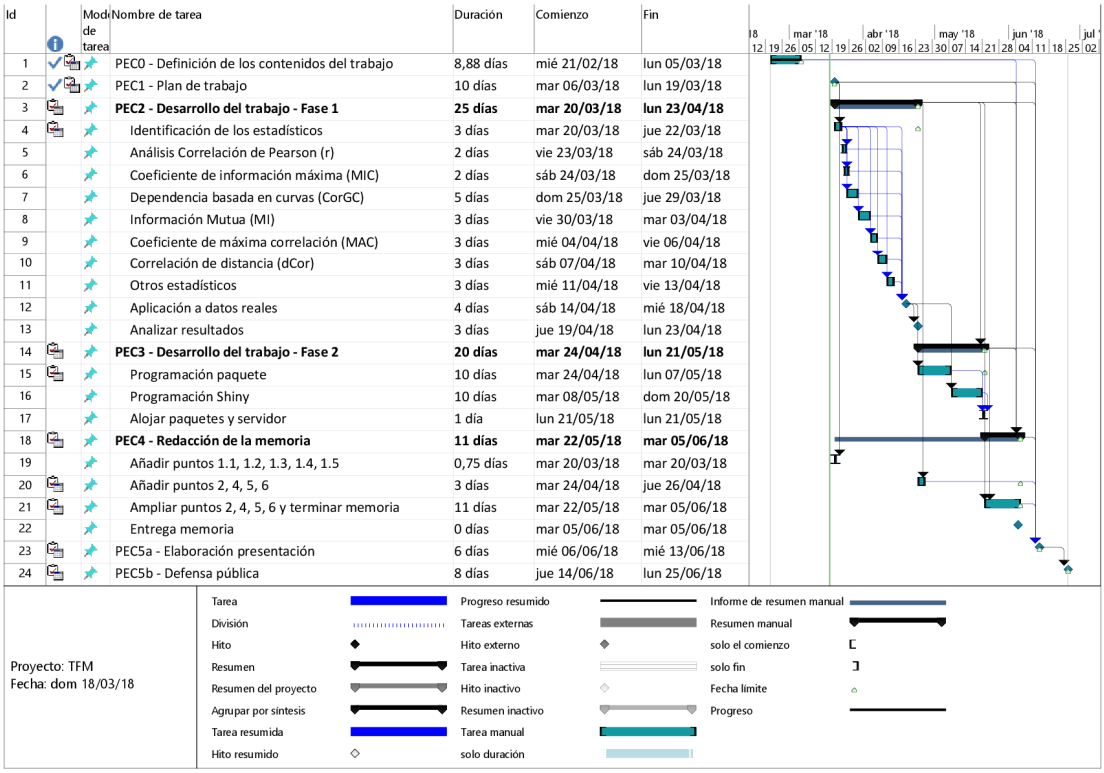
3. Analizar los resultados obtenidos, para identificar cuál de los estadísticos es el que se adapta a más casuísticas (OG a, OE a.4, DT 8 horas).

4. Programar un paquete de R y una aplicación Shiny donde se calculen los estadísticos estudiados (OG b, OE b.3, DT 65 horas).

5. Alojar aplicación en un hosting y el paquete en un repositorio (OE b.4, DT 2 horas).

A continuación, se presenta el Diagrama de Gantt junto con un diagrama de Pert en el que se indica las relaciones entre tareas y su temporalidad de la planificación inicial del proyecto, en el que se tuvieron en cuenta la disponibilidad del autor. Centrando la parte más fuerte del proyecto, recopilación de información y aplicación de los diferentes estadísticos, en los primeros dos meses y en mayo la generación del software.

Ilustración 1: Diagrama de Gantt inicial



Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
1		PEC0 - Definición de los contenidos del trabajo	8,88 días	mié 21/02/18	lun 05/03/18
2		PEC1 - Plan de trabajo	10 días	mar 06/03/18	lun 19/03/18
3		**PEC2 - Desarrollo del trabajo - Fase 1**	**25 días**	**mar 20/03/18**	**lun 23/04/18**
4		Identificación de los estadísticos	3 días	mar 20/03/18	jue 22/03/18
5		Análisis Correlación de Pearson (r)	2 días	vie 23/03/18	sáb 24/03/18
6		Coeficiente de información máxima (MIC)	2 días	sáb 24/03/18	dom 25/03/18
7		Dependencia basada en curvas (CorGC)	5 días	dom 25/03/18	jue 29/03/18
8		Información Mutua (MI)	3 días	vie 30/03/18	mar 03/04/18
9		Coeficiente de máxima correlación (MAC)	3 días	mié 04/04/18	vie 06/04/18
10		Correlación de distancia (dCor)	3 días	sáb 07/04/18	mar 10/04/18
11		Otros estadísticos	3 días	mié 11/04/18	vie 13/04/18
12		Aplicación a datos reales	4 días	sáb 14/04/18	mié 18/04/18
13		Analizar resultados	3 días	jue 19/04/18	lun 23/04/18
14		**PEC3 - Desarrollo del trabajo - Fase 2**	**20 días**	**mar 24/04/18**	**lun 21/05/18**
15		Programación paquete	10 días	mar 24/04/18	lun 07/05/18
16		Programación Shiny	10 días	mar 08/05/18	dom 20/05/18
17		Alojar paquetes y servidor	1 día	lun 21/05/18	lun 21/05/18
18		**PEC4 - Redacción de la memoria**	**11 días**	**mar 22/05/18**	**mar 05/06/18**
19		Añadir puntos 1.1, 1.2, 1.3, 1.4, 1.5	0,75 días	mar 20/03/18	mar 20/03/18
20		Añadir puntos 2, 4, 5, 6	3 días	mar 24/04/18	jue 26/04/18
21		Ampliar puntos 2, 4, 5, 6 y terminar memoria	11 días	mar 22/05/18	mar 05/06/18
22		Entrega memoria	0 días	mar 05/06/18	mar 05/06/18
23		PEC5a - Elaboración presentación	6 días	mié 06/06/18	mié 13/06/18
24		PEC5b - Defensa pública	8 días	jue 14/06/18	lun 25/06/18

Una vez realizada la planificación se identificaron cuáles son los hitos del proyecto, que son aquellos en los que si hay algún retraso en la tarea esto repercutiría negativamente en los plazos de otras tareas.

En la tabla 1 se puede encontrar detallados cada uno de los hitos, en qué PEC están incluidos y la fecha clave para cumplir el hito.

Tabla 1: Hitos del proyecto

Hito	PEC	Fecha crítica
Plan de trabajo	PEC1	19/03/2018
Aplicación a datos reales	PEC2	18/04/2018
Analizar resultados	PEC2	23/04/2018
Creación del paquete en R y aplicación Shiny	PEC3	20/05/2018
Entrega de memoria	PEC4	05/06/2018
Elaboración de la presentación	PEC5a	13/06/2018
Defensa pública	PEC5b	25/06/2018

En el apartado de conclusiones se puede ver el cumplimiento real de dicha planificación y los motivos de estas desviaciones

1.5 Breve sumario de productos obtenidos

Hasta el momento se han obtenido tres productos:

- Una comparativa teórica y práctica de los cinco estadísticos estudiados. Además de la prueba estadística HHG incluida en el anexo.
- Un paquete en R con las funciones necesarias para calcular los cuatro estadísticos seleccionados en el análisis anterior, junto con los p-valores y los p-valores ajustados.
- Una aplicación Web, que permite realizar de una forma simple el cálculo de los cuatro estadísticos.

Una vez se finalice el proyecto, además se habrá generado una presentación que resumirá el trabajo y resultados obtenidos en el TFM. Donde se expondrán oral y visualmente la información más importante del proyecto.

1.6 Breve descripción de los otros capítulos de la memoria

La memoria contiene dos capítulos más además de las conclusiones:

- Comparativa de coeficientes: en este capítulo se detalla para cada uno de los coeficientes el método de cálculo, así como sus ventajas e inconvenientes y las propiedades que cumplen. Además, de haber testado con ocho tipos de dependencias, de 500 simulaciones con 200 observaciones cada una, y con una base de datos real. Todo ello, para conseguir una comparativa teórica y práctica para averiguar cuál es el coeficiente idóneo a sustituir la correlación de Pearson.
- Software desarrollado: en él se indica las funciones necesarias para crear el paquete `AlterCorr` necesario para el funcionamiento de la aplicación web. Las funciones están diferenciadas según si ya estaban desarrolladas anteriormente o bien, las que han sido creadas expresamente o han sido preciso modificarlas. Junto con la explicación del funcionamiento de la aplicación Shiny creada.

2. Comparativa de coeficientes

2.1. Propiedades fundamentales de Rényi

El matemático Alfréd Rényi en 1959 instauró siete propiedades fundamentales en las que una medida de dependencia $\rho: X \times Y \rightarrow [0,1]$ entre las variables aleatorias $x \in X$ e $y \in Y$ que debe cumplir:

i. $\delta(X,Y)$ está definida para cualquier par de variables X e Y aleatorias no constantes.
ii. $\delta(X,Y) = \delta(Y,X)$
iii. $0 \leq \delta(X,Y) \leq 1$
iv. $\delta(X,Y) = 0$ sí y sólo sí X e Y son estadísticamente independientes.
v. $\delta(X,Y) = 1$ si para las funciones medibles de Borel f y g, $Y = f(X)$ o $X = g(Y)$.
vi. Para las funciones biyectivas de Borel $f, g: \mathbb{R} \rightarrow \mathbb{R}, \delta(X,Y) = \delta\big(f(X), g(Y)\big)$.
vii. Si $(X,Y) \sim N(\mu, \sum)$, entonces $\delta(X,Y) = |\rho(X,Y)|$, donde ρ es el coeficiente de correlación de X e Y.

2.2. Coeficientes

A continuación, se analizan cuatro coeficientes de los múltiples que hay en la actualidad para encontrar dependencia entre variables continuas, además de la propia Correlación de Pearson.

Correlación de Pearson (r)

Este coeficiente fue creado en 1888 por Francis Galton y modificada después por F.Y. Edgeworth y K. Pearson [1]. Es una medida de interdependencia lineal entre las variables X e Y. Las variables deben de ser aleatorias y distribuirse normalmente, además de seguir distribuciones normales.

Para calcular el coeficiente de correlación de Pearson se tiene que hacer de la siguiente forma:

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

Que es una covarianza estandarizada para las *n* observaciones emparejadas de las variables X e Y, por lo que carece de unidades de medida [2].

Toma los valores -1 y 1 cuando hay una correlación total (negativa y positiva respectivamente) entre las variables, y 0 cuando no existe relación lineal entre las variables (incorrelacionadas) [2, 3].

El requisito más exigente del estadístico es que la distribución conjunta de las variables debe de ser normal. Lo que implica que cada una de las variables se tiene que distribuir de forma normal. Deben de tener una varianza constante (homocedasticidad) y presentar pocos valores atípicos.

En la tabla 2 se indican las ventajas y desventajas que tiene este coeficiente.

Tabla 2: Ventajas e inconvenientes de la Correlación de Pearson

Ventajas [4, 5]	**Desventajas [4, 5]**
• Las variables no tienen por qué tener las mismas unidades de medida.	• Es sensible a los valores atípicos de las observaciones (no es un estadístico robusto).
• Es de fácil interpretación y de cálculo sencillo.	• Interpretar una relación entre variables con una posible causa-efecto.
• El coeficiente obtenido se encuentra entre valores acotados ($-\mathbf{1} \leq \boldsymbol{r} \leq \mathbf{1}$).	• Sólo detecta la relación lineal entre variables, reduciendo su efectividad para el resto de asociaciones no lineales.
• En la correlación parcial ($\boldsymbol{r_{XYZ}}$), en el caso del espacio lineal, se mantiene invariante al valor de Z.	• No soporta variables aleatorias multidimensionales.
• No es necesario el uso de parámetros.	• Varia si hay cambios en las distribuciones marginales.
• Tiene un bajo coste computacional respecto al tamaño de la muestra.	• No satisface las propiedades de Rényi.
• Debido a que las relaciones monótonas se pueden ajustar a funciones lineales, se pueden obtener resultados satisfactorios del coeficiente.	• No es consistente según el número de observaciones.

Una vez se tiene el coeficiente de correlación, interesa saber cuál es la significación del estadístico. La prueba de dependencia entre X e Y contrasta las siguientes hipótesis:

- H_0: $\rho = 0$, es decir, no hay relación lineal entre las variables.
- H_1: $\rho \neq 0$ las variables son dependientes.

Si las variables tienen una distribución conjunta normal, entonces el estadístico utilizado para probar la hipótesis nula es una t de Student con *n-2* grados de libertad:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}$$

Si no se distribuyen conjuntamente de forma normal, se utiliza la transformación de Fisher para obtener una distribución normal asintótica [5].

Coeficiente de Información Máxima (MIC)

Diseñado por Reshef et al. en 2011 para medir la dependencia de pares de variables, específicamente para la exploración rápida de conjuntos de datos multidimensionales, pertenece a una clasificación más grande de estadísticos de exploración no paramétricos basados en la información máxima (MINE). Clasifica las parejas según sus puntuaciones y

examinando las que obtienen mayor puntuación. Para ello, tiene que cumplir las propiedades heurísticas (generalidad y equidad) [1, 6].

La idea planteada con el coeficiente MIC, es que si existe una relación entre dos variables, se puede dibujar una cuadrícula en el diagrama de dispersión de las variables que dividen los datos para encapsular esa relación.

Para calcular el MIC se tienen que explorar todas las cuadriculas (con la resolución máxima que permita la muestra) y encontrar cual es la que tiene la mayor Información Mutua (MI). Los valores obtenidos de MI se normalizan para que sea una comparación apropiada y se encuentre entre los valores 0 y 1 [5, 7, 8].

La fórmula para calcular el estadístico MIC es:

$$MIC(x,y) = \max_{n_x n_y < B} \frac{I(x,y)}{\log_2 min\{x,y\}}$$

Donde $I(x,y) = \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$ es la máxima información mutua entre los datos x e y, $p(x,y)$ la proporción de datos que están dentro del contenedor (x,y) y los valores n_x y n_y el número de contenedores en los que x e y están partidos respectivamente [8].

Como las variables son reales, se podría crear un contenedor para cada uno de los pares de datos obteniendo un elevado valor de MI, por lo que los autores proponen tomar los contenedores ordenados n_x y n_y cuyo producto sea más pequeño que el tamaño muestral *n*. El valor predeterminado que sugieren es aquel que $n_x \times n_y < B$, donde $B = n^{0.6}$.

Para que la MI sea máxima los contenedores deben de tener un tamaño similar para que las entropías cumplan que $H(x) = H(y) = H(x,y)$.

Los autores proponen como valores por defecto para los parámetros $\alpha = 0.6$ y $c = 15$ que es el punto de inicio de la cuadrícula de búsqueda (partición).

A continuación, se detallan las ventajas y desventajas de dicho estadístico:

Tabla 3: Ventajas e inconvenientes de MIC

Ventajas [4, 5, 6, 7, 8, 9]	**Desventajas [4, 5, 6, 7, 8, 9]**
• Detecta una amplia gama de asociaciones entre variables, incluidas las no lineales.	• Para muestras grandes, se debe optimizar la velocidad computacional modificando los parámetros $B(n) = n^{\alpha}$ y c.
• Es de fácil interpretación, ya que el estadístico se encuentra entre los valores 0 y 1.	• Calcular todos los posibles "binning" incluso para una pequeña n es inviable.
• El coste computacional es bajo $O(n^{1.2})$, pero mayor que la correlación de Pearson.	• Uno de los inconvenientes más importantes es que en la práctica se puede o no encontrar el máximo real.
• Cumple las ventajas heurísticas de generalidad y equidad.	• La propiedad de equidad impone también un menor poder para detectar relaciones débiles que otros estadísticos.
• La equidad permite seleccionar las relaciones más fuertes del conjunto de datos.	• No soporta variables aleatorias multidimensionales.
• Para funciones lineales se aproxima a los valores del coeficiente de Pearson.	• Varia si hay cambios en las distribuciones marginales.
• Se puede utilizar para detectar asociaciones lineales entre variables, ya que MIC-r^2 es 0 cuando la función es lineal y 1 cuando hay una elevada relación no lineal.	• No satisface las propiedades de Rényi.
• MIC es una medida apropiada cuando hay un gran número de relaciones significativas en el conjunto de datos y se quiere buscar automáticamente las más fuertes	• Es preciso estimar parámetros para hacer el cálculo.

Para calcular la significatividad de MIC y aceptar o rechazar la hipótesis nula:

- H_0 las variables aleatorias son independientes.
- H_1 son dependientes.

Se puede utilizar los p-valores de tablas con diferentes muestras proporcionadas por los autores del coeficiente (disponibles en: http://www.exploredata.net/Downloads/P-value-Tables) o bien calculando permutaciones de los datos.

El procedimiento para calcular la prueba de permutación con dos variables aleatorias es [5]:

i. Construir un conjunto de datos permutados en virtud de la H_0, $(x_1, y_1^*), \cdots, (x_1, y_n^*)$ fijando x_i y permutando y_i.
ii. Calcular el coeficiente de información máxima con este conjunto de datos permutados (x, y^*).
iii. Repetir los pasos (i) y (ii) hasta el número deseado de permutaciones replicadas.

El p-valor obtenido del test de permutación es la fracción de las puntuaciones MIC permutadas (x, y^*) que son mayores o iguales que el MIC observado original (x, y).

Correlación a lo largo de una curva de generación (CorGC)

Esta correlación fue propuesta por P. Delicado y M. Smrekar en 2007, para dos variables aleatorias reales que no estaban relacionadas linealmente, utilizando curvas principales [10].

Usando la descomposición espectral de la matriz de varianza $\sum$ de las variables aleatorias X e Y,

$$\sum = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}$$

Se expresan la correlación y la covarianza, de las variables aleatorias X e Y, en términos del primer componente principal, siendo:

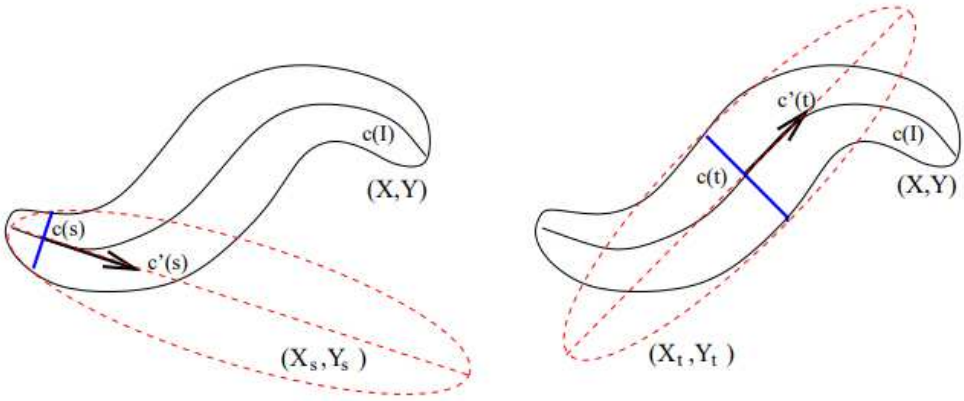
$$V(X) = \sigma_X^2 = \lambda_1 \cos^2\alpha + \lambda_2 \sin^2\alpha \qquad V(Y) = \sigma_Y^2 = \lambda_1 sin^2\alpha + \lambda_2 cos^2\alpha$$

$$Cov(X,Y) = \sigma_{XY} = (\lambda_1 - \lambda_2)\cos\alpha \sin\alpha$$

$$\rho = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}$$

Donde λ_i son los autovalores de la matriz de varianza $\sum$ y α el ángulo entre el autovalor λ_1 y el eje x.

Ilustración 2: Linealización de la distribución (X, Y) alrededor de c(s) y c(t)



Una vez definidas en términos de componente principales, se linealiza la distribución de (X,Y) alrededor de un punto $c(s)$ de la curva de generación $c(I)$, donde (S,T) es la variable aleatoria bivariante generadora [10].

Definiendo las varianzas, covarianza y correlación locales:

$$LV_X(s) = V(S)\ \cos^2\alpha(s) + V(T|S)\ \sin^2\alpha(s),$$
$$LV_Y(s) = V(S)\ \sin^2\alpha(s) + V(T|S)\ \cos^2\alpha(s),$$
$$LCov_{(X,Y)}(s) = \{V(S) - V(T|S=s)\}\cos\alpha(s)\sin\alpha(s),$$
$$LCor_{(X,Y)}(s) = \frac{LCov_{(X,Y)}(s)}{\{LV_X(s)LV_Y(s)\}^{1/2}}.$$

Obteniendo, la correlación global como la suma de la esperanza respecto la variable aleatoria S.

$$CorGC(X,Y) = \left(E_S\left[\left\{LCor_{(X,Y)}(S)\right\}^2\right]\right)^{1/2}$$

Para poder calcular esta correlación, se deben cumplir varias condiciones:

- Las variables (X,Y) deben de ser aleatorias distribuidas conjuntamente en $\mathbb{R}^2$.
- La función $\chi_c(s,t) = c(s) + tv(s)$, que para (S,T) su densidad es $h(S,T)$, que $E(T|S=s) = 0$, que $V(S) > V(T|S=s)$ y que χ_c es una aplicación uno a uno.
- $v(s)$ debe de ser un campo vectorial unitario ortogonal a la curva c.

En la tabla 4 se exponen las ventajas e inconvenientes de esta correlación.

Tabla 4: Ventajas e inconvenientes de CorGC

Ventajas [10, 11, 12]	**Desventajas [10, 11, 12]**
• Al estar comprendido entre los valores 0 y 1, es de fácil interpretación.	• El coste computacional es exponencial del orden de O(n^2), cuando se calcula la curva principal con el método de Hastie y Stuetzle (HS).
• Para funciones lineales se aproxima al valor absoluto de la correlación de Pearson.	• No satisface la última propiedad de Rényi ya que no se conservan las ortogonalidades al transformarse.
• Detecta algunas asociaciones no lineales entre variables, principalmente basadas en la curva.	• Tal y como se define la curva para generar el coeficiente CorGC, esta no es una curva principal según las definiciones de Hastie y Stuetzle (1989), Kégl et al. (2000) o Delicado (2001), aunque se aproxima a las tres definiciones.
• Cumple la segunda proposición de Rényi, debido al carácter simétrico de la función $\boldsymbol{\cos\alpha \sin\alpha}$ entre los valores $\boldsymbol{\alpha}$ y $\boldsymbol{\phi - \alpha}$. También verifica la tercera y la sexta propiedad.	• Sólo admite matrices de dimensión 2.
• Las propiedades primera y quinta de Rényi, se cumplen adaptándolas a la curva.	
• Casi verifica el cuarto axioma de Rényi, que es ligeramente más fuerte que las asumidas por los autores.	

Tal y como indican los autores, el coeficiente CorGC se puede utilizar para probar la independencia, la linealidad y la similitud entre variables [10, 13]. La prueba de independencia es la siguiente:

- H_0 Las variables aleatorias X e Y son independientes.
- H_1 Se distribuyen a lo largo de una curva.

Y proponen hacer una prueba aleatoria de permutación (como en el coeficiente MIC) que permite aproximar la distribución nula al test estadístico.

Coeficiente de dependencia aleatorio (RDC)

El RDC fue propuesto por Lopez-Paz et al. en 2013, como una medida de dependencia no lineal entre variables aleatorias de dimensión arbitraria basada en el coeficiente máximo de correlación de Hirschfeld-Gebelein-Rényi (HGR) [4, 14].

El estadístico calcula la dependencia entre las muestras aleatorias como la mayor correlación canónica entre las k proyecciones no lineales aleatorias de sus trasformaciones de cópula [4, 14].

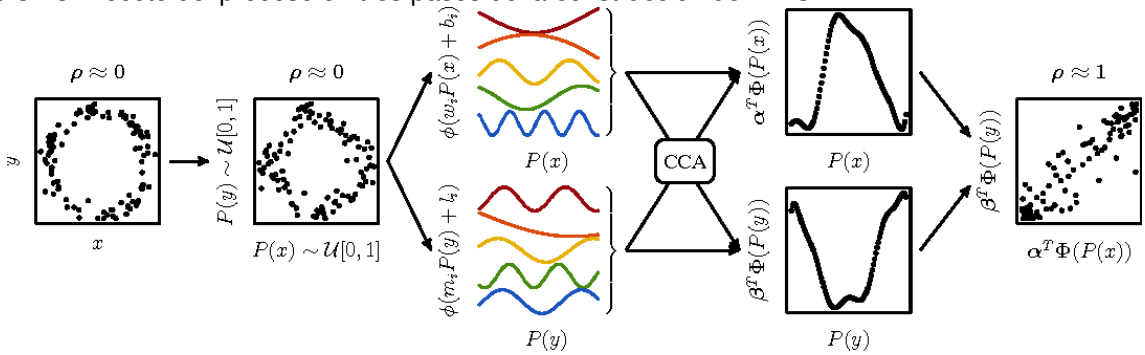
Para ello, el RDC se construye en tres pasos (en la ilustración 3, se puede ver un boceto del proceso) [4]:

1. **Estimación de la cópula-transformación**: Para que el estadístico sea invariante respecto a las distribuciones marginales, se realiza la transformación de la cópula empírica de los datos.

2. **Generación de proyecciones no lineales aleatorias**: Se aumentan las transformaciones de cópula empírica con k proyecciones no lineales elegidas al azar, de esta forma los métodos lineales pueden usarse para detectar asociaciones no lineales en los datos originales.

3. **Cálculo de correlaciones canónicas**: Por último, se calcula las combinaciones lineales de las transformaciones de cópula empírica ampliadas que tienen la correlación máxima. Donde la mayor correlación canónica entre las variables aleatorias es el supremo de los coeficientes de correlación sobre sus proyecciones lineales.

Ilustración 3: Boceto del proceso en tres pasos de la construcción del RDC



Por lo tanto, el Coeficiente de dependencia aleatorio entre las muestras X e Y se define como:

$$rdc(X, Y; k, s) = \sup_{\alpha,\beta} \rho\left(\alpha^T \Phi(P(X); k, s), \beta^T \Phi(P(Y); k, s)\right)$$

Donde $X \in \mathbb{R}^{p \times n}, Y \in \mathbb{R}^{q \times n}$, el parámetro $k \in \mathbb{N}_+$ es el número de proyecciones no lineales de la cópula y $s \in \mathbb{R}_+$ la varianza para dibujar i.i.d. coeficientes de proyección en $N{\sim}(0, sI)$. Siendo ρ la correlación

canónica, $P(X)$ la transformación de la cópula empírica y Φ las proyecciones no lineales.

La principal suposición de este coeficiente y que además es inevitable, es la elección de las no linealidades $\Phi: \mathbb{R} \to \mathbb{R}$. Los autores utilizan funciones aleatorias en lugar del método Nyström ya que necesita menos memoria y coste computacional. Para ello utilizan las proyecciones sinusoidales $(\Phi(w^T x + b) := \sin(w^T x + b))$, ya que los kernels invariantes por desplazamiento se aproximan con estas características cuando se utiliza la distribución de muestreo de parámetros aleatorios apropiada, además las funciones con transformadas de Fourier absolutamente integrables se aproximan a un error por debajo de $\mathrm{O}\left(1/\sqrt{k}\right)$ para estas k características [4].

En la tabla que aparece a continuación, se exponen las ventajas y desventajas del coeficiente de dependencia aleatorio.

Tabla 5: Ventajas e inconvenientes de RDC

Ventajas [4]	**Desventajas [4]**
• Opera con variables aleatorias de dimensión arbitraria.	• Es difícil de entender para personas no expertas.
• Encuentra dependencia entre variables no lineales.	• En algunas ocasiones no es posible calcular las correlaciones canónicas.
• Cumple todas las propiedades de Rényi.	
• Es invariante a las transformaciones marginales, gracias a la cópula-transformación.	
• Tiene un bajo coste computacional O(n log n).	
• Es de fácil de implementación.	

Las hipótesis para la prueba de independencia para este coeficiente son:

- H_0 es que los dos conjuntos de proyecciones no lineales no están correlacionados mutuamente.
- H_1 las proyecciones lineales están correlacionadas.

Para demostrar la prueba se puede utilizar la aproximación de Bartlett cuando se cumple las condiciones de normalidad, calculando $\left(\frac{2k+3}{2} - n\right) \log \prod_{i=1}^{k}(1-\rho_i^2) \sim \chi_{k^2}^2$, donde $\rho_1, \cdots, \rho_k$ son las correlaciones canónicas entre $\Phi(P(X); k, s)$ y $\Phi(P(Y); k, s)$. Alternativamente, para las distribuciones asintóticas no paramétricas pueden obtenerse del espectro de los productos internos de las matrices de proyección aleatorias no lineales para hacer el test [4].

Correlación de distancias (dCor)

La correlación de distancias (dCor) fue construida por Székely et. al (2007) y por Székely y Rizzo (2009) [15], junto con la covarianza de distancia (dCov), siendo análogas a la covarianza y correlación clásicas, pero para realizar los cálculos se basaron en el concepto de energía de distancias (euclídea), calculando las distancias entre las observaciones [5, 8, 15].

La covarianza de distancia de la población coincide con la covarianza con respecto al movimiento browniano, por lo tanto, ambos pueden llamarse covarianza y correlación de distancia browniana [16].

Para dos vectores aleatorios $X \in \mathbb{R}^p$ e $Y \in \mathbb{R}^q$, donde p y q son enteros positivos y $(X, Y) = \{(x_i, y_i): i = 1, \cdots, n\}$, se calculan las matrices de distancia transformadas A y B para las variables X e Y respectivamente.

Para calcular estas matrices, primero se calculan las distancias euclidianas ($\|\cdot\|$ o norma euclídea) entre los pares de observaciones para cada uno de los vectores, pueden estar elevadas a una potencia entre $(0, 2]$:

$$a_{i,j} = \|x_i - x_j\|_p, \quad b_{i,j} = \|y_i - y_j\|_q$$

para $i, j = 1, \cdots, n$, con lo que se obtiene dos matrices de distancias de $n \times n$, una para cada vector.

También se calculan la media de las filas $\bar{a}_{i\cdot} = \frac{1}{n}\sum_{j=1}^{n} a_{ij}$, la media de las columnas $\bar{a}_{\cdot j} = \frac{1}{n}\sum_{i=1}^{n} a_{ij}$ y la media total de la matriz $\bar{a}_{\cdot\cdot} = \frac{1}{n^2}\sum_{i,j=1}^{n} a_{ij}$ para la matriz de distancias del vector X. Análogamente se definen las medias para la matriz de distancias del vector Y ($\bar{b}_{i\cdot}$, $\bar{b}_{\cdot j}$ y $\bar{b}_{\cdot\cdot}$).

Una vez, hechos estos cálculos se procede a hacer un doble centrado de las dos matrices ($A_{ij} := a_{ij} - \bar{a}_{i\cdot} - \bar{a}_{\cdot j} + \bar{a}_{\cdot\cdot}$ y $B_{ij} := b_{ij} - \bar{b}_{i\cdot} - \bar{b}_{\cdot j} + \bar{b}_{\cdot\cdot}$) , de esta forma todas las filas y columnas suman 0.

Y se define covarianza de distancia entre X e Y como $dCov(X, Y) = \sqrt{\frac{1}{n^2}\sum_{i,j=1}^{n} A_{ij}B_{ij}}$ y las varianzas de X e Y como $dVar(X) = dCov(X, X)$ y $dVar(Y) = dCov(Y, Y)$ respectivamente. Por lo que, ya se puede definir la correlación de distancias como:

$$dCor(X, Y) = \frac{dCov(X, Y)}{\sqrt{dVar(X)dVar(Y)}}$$

Un punto crucial de la teoría es que la $dCov(X, Y) = 0$ sí y sólo sí X e Y son independientes. Esto no es válido para los espacios métricos generales. Para probar la independencia, es necesario y suficiente con que el espacio métrico sea de tipo fuertemente negativo [17].

Está definido sólo para un par de variables aleatorias X e Y con primeros momentos finitos, es decir, $E|X|_p < \infty$ y $E|Y|_q < \infty$ [15, 18].

A continuación, se procede a detallar las ventajas y desventajas de dicha correlación:

Tabla 6: Ventajas e inconvenientes de dCor

Ventajas [4, 15, 18, 19]	**Desventajas [4, 15, 19]**
• El coeficiente está comprendido entre 0 y 1.	• No es invariante respecto a cambios en las distribuciones marginales.
• $\boldsymbol{dCor(X,Y) = 0}$ sí y sólo sí $\boldsymbol{X}$ e $\boldsymbol{Y}$ son independientes.	• No satisface todas las propiedades fundamentales de Rényi para un coeficiente, cumple los puntos 2, 3, y 4, el resto parcialmente
• Vale 1 cuando hay una dependencia lineal perfecta.	• Tiene un coste computacional elevado del orden de $O(n^2)$ si la n es grande.
• No necesita que las variables tengan una distribución normal.	
• Detecta relaciones no lineales.	
• Maneja variables multidimensionales aleatorias.	
• Las variables no tienen por qué ser escalables y pueden ser de diferentes dimensiones.	
• Sólo requiere estimar un parámetro.	
• Es sencilla de calcular y de estimar.	
• Es invariante ante cambios de escala y de rotaciones.	
• Es un coeficiente consistente.	

Se puede realizar una prueba de independencia para variables multivariables utilizando la correlación de distancias, donde:

- H_0 las variables aleatorias son independientes.
- H_1 no son independientes.

Para ello, se tiene que contrastar con una prueba de permutación que $dCor(X,Y) = 0$ (que esto ocurre cuando $dCov(X,Y) = 0$, como ya se ha indicado anteriormente).

El procedimiento para calcularla con dos variables aleatorias es [5]:

iv. Construir un conjunto de datos permutados en virtud de la H_0, $(x_1, y_1^*), \cdots, (x_1, y_n^*)$ fijando x_i y permutando y_i.
v. Calcular la correlación de distancias ($dCor$) con este conjunto de datos permutados (x, y^*).
vi. Repetir los pasos (i) y (ii) hasta el número deseado de permutaciones replicadas.

El p-valor obtenido del test de permutación es la fracción de réplicas de $dCor$ en el conjunto de datos permutados (x, y^*) que son por lo menos tan grande como el estadístico observado del conjunto de datos original (x, y).

2.3. Propiedades de los coeficientes

En este apartado se procede a indicar las propiedades que tienen cada uno de los coeficientes estudiados.

La Correlación de Pearson es adimensional, ya que carece de unidades de medida. Además, es invariante a las posibles transformaciones lineales de las variables, tanto al cambio de origen como el de escala. Toma valores entre -1 y 1, coincidiendo con el signo de S_{xy} y los coeficientes de la recta de regresión e indicando que la correlación es positiva o negativa según los valores entre los que se encuentra. Si $r = \pm 1$ indica que hay una correlación total entre las variables, es decir, una de las variables es la combinación lineal de la otra $(y_i = a_r + b_r x_i\,, i = 1, \cdots, n)$, por el contrario, si vale 0 indica que no existe relación lineal entre las variables y se dice que las variables están incorrelacionadas. Cuando el estadístico está próximo a 1 existe una relación lineal fuerte entre las variables [2, 3].

En el caso de MIC, el estadístico está comprendido entre los valores 0 y 1, gracias a la normalización de los valores obtenidos de MI. Es simétrico debido a que la Información Mutua también lo es. Asigna el valor 0 cuando las variables son estadísticamente independientes. Y las puntuaciones que tienden a 1 para una clase más grande de relaciones sin ruido (aunque las funciones estén superpuestas) [1, 7, 8].

Para las variables aleatorias X e Y, Si Y es función de X no constante en un intervalo abierto, se obtendrá un MIC que tenderá a 1 con probabilidad uno cuanto más grande sea la muestra. Si el soporte de (X, Y) se describe como uniones finitas de curvas diferenciales de la forma $c(t) = [x(t), y(t)]$ para t en [0,1], entonces los datos extraídos de (X, Y), también obtendrá un MIC que tenderá a 1 con probabilidad uno a medida que aumenta la muestra siempre y cuando dx/dt y dy/dt sean cada uno 0 en un número finito de puntos. El MIC de los datos extraídos de (X, Y) converge a cero a medida que aumenta el tamaño de la muestra, sí y solo sí X e Y son estadísticamente independientes [7].

Al cumplir con la propiedad de equidad MIC para funciones relacionadas sin ruido no constantes, asigna puntuaciones que tienden a 1. Al añadir ruido a una función sin ruido, el MIC obtenido se aproxima a la r^2 de la función sin ruido. El valor desciende a medida que se añade ruido a la función relacional. Y asigna también valores similares a relaciones con ruidos iguales, aunque sean diferentes funciones [1, 7].

La Correlación de la Curva de Generación, como la correlación de Pearson, es un estadístico adimensional. Toma valores entre 0 y 1, cuando es 0 indica que no existe relación entre las variables y si es 1 indica que hay una correlación total entre las variables, es decir, se puede establecer una curva que asocie las variables. Cuando la asociación entre las variables es lineal, la CorGC es el valor absoluto de la correlación de Pearson [10].

Por otro lado, el RDC cumple con todas las propiedades fundamentales propuesta por Alfréd Rényi. Se mantiene invariante ante cambios en las distribuciones marginales. Encuentra dependencias no lineales. Además, puede utilizarse con variables aleatorias multidimensionales. Es un estimador del Coeficiente Máximo de Correlación (HGR) de Hirschfeld-Gebelein-Rényi, ya que HGR no es computable. Es de fácil implementación y por cómo está configurado, la complejidad computacional depende del cálculo de las transformaciones cópula, que es aproximadamente de O(n log n). Cuando el parámetro s tiende a 0 el coeficiente converge al Rango de Spearman y cuando el parámetro k tiende a ∞ converge al Kernel Canonical Correlation Analysis (KCCA) [4, 20].

Por último, se puede obtener un valor de la Correlación de distancias para X e Y de una dimensión arbitraria, por lo que se consiguen estimaciones multivariadas, y también basadas en el rango de esta métrica. Además, si $E(|X_p| + |Y_q|) < \infty$, entonces $0 \leq dCor(X,Y) \leq 1$, y $dCor(X,Y) = 0$ si y sólo sí X e Y son independientes. Si (X,Y) son normales bivariantes entonces $dCor(X,Y) \leq |r|$ y obtiene el valor 1 si $r = \pm 1$. Para X e Y multivariables, $0 \leq dCor_n(X,Y) \leq 1$ y si $dCor_n(X,Y) = 1$, entonces existe un vector a, un número real b distinto a cero y una matriz ortogonal C tal que $Y = a + bXC$ [15].

2.4. Resultados simulados

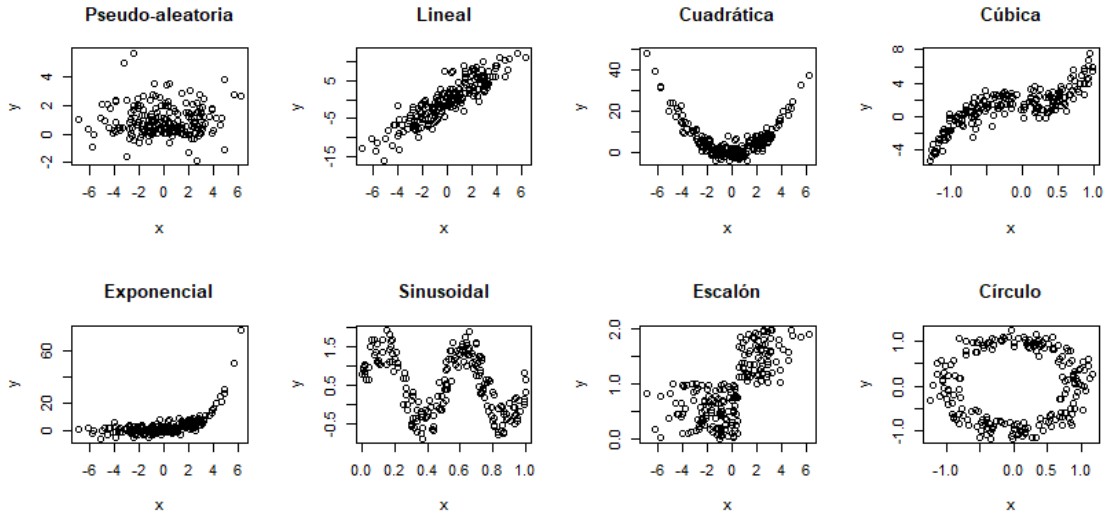
Seguidamente, se van a comparar los coeficientes estudiados anteriormente con 8 asociaciones diferentes que se han tenido en cuenta para hacer las simulaciones, para cada una se han seleccionado 200 observaciones y se han realizado 500 repeticiones. En todas ellas se les ha añadido un ruido para que fuese más difícil encontrar dicha asociación.

Tabla 7: Distribuciones de dependencia bivariadas de las simulaciones

#	Simulación	Variable dependiente	Variable independiente	Ruido
1	Pseudo-aleatoria	$y \sim N(2, 1.8) * \epsilon$	$x \sim N(0, 2.5)$	$\epsilon \sim U(0,1)$
2	Lineal	$y = 2x + \epsilon$	$x \sim N(0, 2.5)$	$\epsilon \sim N(0, 2.5)$
3	Cuadrática	$y = x^2 + \epsilon$	$x \sim N(0, 2.5)$	$\epsilon \sim N(0, 2.5)$
4	Cúbica	$y = 4x^3 + x^2 + \epsilon$	$x \sim U(-1.3, 1)$	$\epsilon \sim N(1.5, 0.85)$
5	Exponencial	$y = 4^{0.5x} + \epsilon$	$x \sim N(0, 2.5)$	$\epsilon \sim N(0, 2.5)$
6	Sinusoidal	$y = sin(4\pi x) + \epsilon$	$x \sim U(0,1)$	$\epsilon \sim U(0,1)$
7	Escalón	$y = \begin{cases} 0 + \epsilon & \text{si } x \leq 0.5 \\ 1 + \epsilon & \text{si } x > 0.5 \end{cases}$	$x \sim N(0, 2.5)$	$\epsilon \sim U(0,1)$
8	Círculo	$x = sin(z\pi) + \epsilon$ $y = cos(z\pi) + \epsilon$	$z \sim N(0, 2.5)$	$\epsilon \sim N(0, 1/8)$

En la ilustración 4 se pueden ver todas las distribuciones que se han estudiado y el efecto de haber añadido un ruido a la función para la primera simulación de las 500.

Ilustración 4: Distribuciones de dependencia bivariadas consideradas



Con el conjunto de datos pseudo-aleatorio se ha intentado generar una distribución aleatoria entre las variables, es decir, que no hubiese una dependencia, y así poder comprobar cómo se comportan los estadísticos con una supuesta independencia. Aunque todos los algoritmos de generación de números aleatorios siguen un patrón, de aquí que sea un conjunto pseudo-aleatorio.

La Correlación de Pearson (r) en valor absoluto es la única que está próxima a 0, lo que indica que no ha encontrado una relación lineal entre las variables x e y, aunque no quiere decir que sean independientes. Este coeficiente es el que tiene un menor tiempo de ejecución. El segundo coeficiente que ha obtenido de media el valor más bajo (0.12) ha sido la dCor, con una desviación estándar de 0.024 (la segunda más baja), por lo que no pude asumir que las variables sean independientes, pero indica que la relación es muy baja. Después se encuentran los coeficientes MIC y RDC ambos con valores de 0.22, pero RDC con una variación algo mayor a la de MIC. En el último lugar, se encuentra la Correlación a lo largo de una curva de generación obteniendo de media 0.27, la desviación estándar más elevada (0.105, por lo que es el menos fiable) y el tiempo computacional más elevado, para este coeficiente no se han podido calcular dos muestras debido a que vulneraba las condiciones de cálculo.

Al calcular las correlaciones, en la dependencia lineal, se ha perdido una observación en CorGC. De los resultados obtenidos se concluye que CorGC es el que tiene una media del coeficiente más alto, indicando que ha detectado en más ocasiones la linealidad entre las 500 simulaciones, aunque tiene un coste computacional más elevado que los coeficientes RDC y r de Pearson, cuando estos dos difieren en centésimas respecto al CorGC. Después de estos tres coeficientes se encuentra la correlación de

distancias (dCor) y por último, se encuentra, el coeficiente MIC que difiere en más de 0.16 unidades respecto a CorGC y es el que tiene una mayor variabilidad. Los coeficientes r, RDC y CorGC tienen un comportamiento similar ante la distribución lineal. La correlación dCor tiene una variabilidad parecida pero que tiende a obtener un coeficiente ligeramente inferior, ya que según las propiedades del coeficiente sólo es 1 cuando encuentra una dependencia lineal perfecta y en este caso, está afectada por el ruido. El coeficiente MIC como ya indican los autores tiene en cuenta el ruido y, por lo tanto, el valor obtenido es menor, aunque tendría que estar próxima al coeficiente de Pearson sin ruido.

Con la asociación del tipo cuadrática, los coeficientes han registrado resultados muy dispares. La media $|r|$ es la más baja 0.12, por lo que no logra detectar este tipo de dependencia (como ya era de esperar), el coeficiente máximo de las 500 repeticiones ha sido de 0.5116822 y es considerado un valor atípico. CorGC y dCor obtienen medias superiores, pero tampoco muy destacables. Por el contrario, en el caso de MIC la media del coeficiente es de 0.70 y RDC con una media de 0.95, encuentra dependencia en las variables en casi la totalidad de las repeticiones con valores por encima del 0.90 y un valor mínimo de 0.88, por lo que es un buen estimador para encontrar dependencias cuadráticas, además de tener el segundo mejor tiempo de ejecución. En el caso de CorGC se ha perdido un 9.6% de la muestra, mostrando que puede ser un gran inconveniente.

Como en el caso anterior, en la distribución cúbica entre variables, el coeficiente RDC es el que encuentra en más ocasiones dependencia entre las variables con una variabilidad muy baja. El siguiente coeficiente que ha obtenido la media más alta es CorGC, que en esta ocasión sólo tiene 3 muestras faltantes. El tercero es la r de Pearson con un valor promedio de 0.85, ya que logra ajustar una línea con la tendencia de la distribución cúbica. Seguido muy de cerca por dCor (con una media de 0.82). En última posición se encuentra MIC con una desviación estándar 4 veces mayor que el de RDC.

Los resultados de los estadísticos de las simulaciones con la distribución exponencial evidencian que el coeficiente RDC se erige como el que en más ocasiones ha detectado dicha asociación entre las variables (0.93) y con la menor variabilidad de los 5 coeficientes. Le sigue CorGC con una media de 0.91, pero sólo se ha podido calcular para el 73.6% de las repeticiones. Más alejados se sitúan dCor (0.66), el coeficiente de correlación de Pearson (0.57) y por último MIC con una media de 0.51.

Con las 500 simulaciones realizadas del tipo de dependencia sinusoidal, se ha comprobado que MIC es el coeficiente que obtiene la media más alta (0.90) y por lo tanto el que mejor ha detectado este tipo de dependencia, con una de las desviaciones estándar más bajas. Después le siguen los coeficientes RDC y CorGC con unas medias de 0.58 y 0.51, respectivamente, por lo que obtienen puntuaciones intermedias de asociación. Después se encuentra dCor con una media de 0.41 y la

variabilidad más baja de todos los coeficientes. Por último, la r de Pearson obtenida es negativa y con un valor de -0.36, por lo que este coeficiente no detecta la dependencia sinusoidal de una forma correcta.

Para la asociación en forma de escalón, el coeficiente MIC vuelve a ser el que presenta mejores resultados de los cinco coeficientes con una media de 0.94, con unos resultados comprendidos entre (0.84, 1). El segundo en obtener los mejores resultados es RDC (0.87) con la variabilidad más baja. Después se encuentra el coeficiente dCor con un buen resultado. Por último, con unos valores muy parecidos se encuentran Coeficiente de correlación de Pearson y CorGC, que en esta ocasión sólo ha perdido dos simulaciones. Destaca que con este tipo de dependencia todos los coeficientes han detectado en mayor o menor medida la asociación de las variables, incluida la r de Pearson.

Según las simulaciones realizadas, CorGC y RDC se erigen como los mejores coeficientes para encontrar una dependencia en forma de círculo, obteniendo unas medias superiores al 0.80, aunque el coeficiente CorGC sólo se ha podido calcular para 375 de las 500 replicaciones. El siguiente coeficiente es MIC con una media de 0.41, por lo que se aleja sustancialmente de los dos primeros. En el caso de dCor, este no logra encontrar la dependencia circular en la mayoría de las repeticiones (media de 0.19) y, por último, la media del valor absoluto de la r de Pearson es prácticamente 0, por lo que es ineficaz para este tipo de relación.

En la siguiente tabla se hace un resumen de los resultados obtenidos para cada función de dependencia y coeficiente de las 500 repeticiones de 200 observaciones cada una, además del tiempo medio de ejecución.

Tabla 8: Resumen de los coeficientes por tipo de asociación

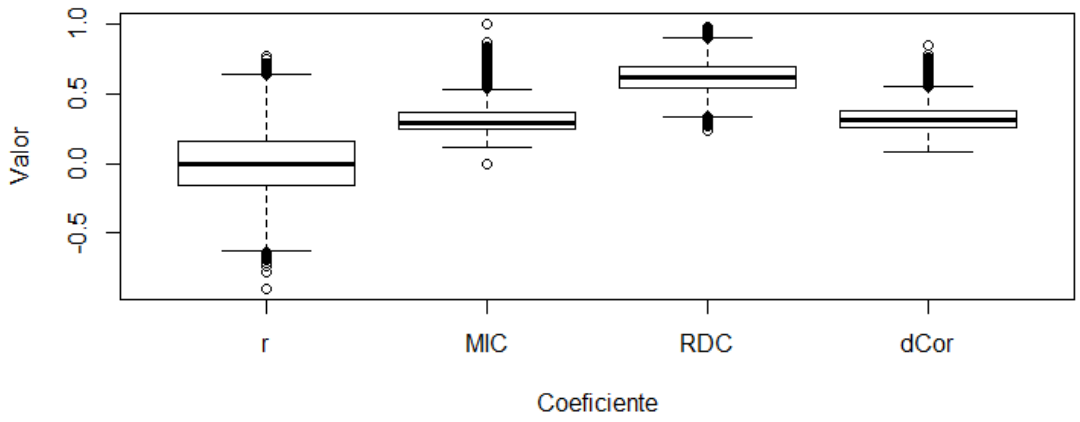
Distribución	Estadístico	# Repet.	Media	Desviación estándar	Tiempo medio
Pseudo-aleatorio	$\|r\|$	500	0.0542431	0.0414735	0.0001800
	MIC	500	0.2197186	0.0230497	0.0844000
	RDC	500	0.2176161	0.0435813	0.0024800
	dCor	500	0.1213537	0.0238863	0.0065200
	CorGC	498	0.2743532	0.1053090	0.1159438
Lineal	$\|r\|$	500	0.8937353	0.0138243	0.0002000
	MIC	500	0.7376346	0.0501943	0.0496400
	RDC	500	0.8954733	0.0147735	0.0044600
	dCor	500	0.8631295	0.0177517	0.0063000
	CorGC	499	0.8965103	0.0136034	0.0913427
Cuadrática	$\|r\|$	500	0.1171907	0.0876517	0.0003000
	MIC	500	0.7031888	0.0547503	0.0623200
	RDC	500	0.9536400	0.0183781	0.0018400
	dCor	500	0.5122542	0.0223313	0.0057200
	CorGC	452	0.3778143	0.0792358	0.1086504
Cúbica	$\|r\|$	500	0.8501664	0.0167685	0.0000600
	MIC	500	0.6498609	0.0433896	0.0509400
	RDC	500	0.9513424	0.0098559	0.0019600
	dCor	500	0.8226844	0.0224898	0.0064200
	CorGC	497	0.9032651	0.0163156	0.0870825
Exponencial	$\|r\|$	500	0.5693396	0.0702618	0.0002000
	MIC	500	0.5118739	0.0524885	0.0621200
	RDC	500	0.9333704	0.0306725	0.0026200
	dCor	500	0.6632335	0.0494574	0.0059800
	CorGC	368	0.9071072	0.0459373	0.1098913
Sinusoidal	$\|r\|$	500	0.3563127	0.0548508	0.0001000
	MIC	500	0.8969773	0.0402794	0.0625200
	RDC	500	0.5749231	0.0707863	0.0021600
	dCor	500	0.4081151	0.0364964	0.0064800
	CorGC	500	0.5101267	0.0845934	0.0966200
Escalón	$\|r\|$	500	0.6854694	0.0292742	0.0001000
	MIC	500	0.9440503	0.0323986	0.0521200
	RDC	500	0.8693319	0.0136974	0.0021600
	dCor	500	0.7568567	0.0214703	0.0062200
	CorGC	498	0.6827462	0.0258564	0.0905622
Círculo	$\|r\|$	500	0.0428617	0.0312623	0.0001600
	MIC	500	0.4068264	0.0266591	0.0738800
	RDC	500	0.8266646	0.0269493	0.0019200
	dCor	500	0.1934182	0.0161516	0.0056000
	CorGC	375	0.8715571	0.0377099	0.1132000

2.5. Resultados reales

Para probar los estadísticos con datos reales, se ha seleccionado los datos de expresión y de metilación de 25 individuos en 11.191 genes. Y se ha calculado la correlación entre la expresión y la metilación para cada uno de los genes con cada uno de los coeficientes estudiados, a excepción del CorGC, puesto que con las simulaciones se ha demostrado que no se puede calcular siempre, ya que se incumplen algunas de las condiciones.

A nivel informativo, el gráfico Box-Plot indica que la r de Pearson es el coeficiente que tiene más dispersión en los resultados, que MIC y dCor tienen un comportamiento similar y, por último, que RDC es el que encuentra más dependencia entre los datos.

Ilustración 5: Box-Plot los coeficientes de una base de datos con 25 individuos y 11.191 genes



Para poder identificar cuáles son los genes que se han expresado significativamente, hace falta calcular la prueba de significatividad y seleccionar aquellos que han rechazado la hipótesis nula de independencia entre la expresión y la metilación.

Tabla 9: Núm. de genes expresados significativamente por coeficiente y p-valor

p-valor	r	MIC	RDC	dCor
< 0.0001	12	4	40	0
< 0.0005	30	23	102	0
< 0.001	52	32	143	0
< 0.005	154	123	328	0
< 0.01	251	213	500	160
< 0.05	948	844	1340	976

Se observa que dCor no ha detectado ningún gen hasta el p-valor<0.01 y que supera en número a los otros dos coeficientes cuando el p-valor<0.05, a excepción de RDC. Paradójicamente, la r de Pearson ha rechazado en más ocasiones la independencia de las variables que el coeficiente MIC, encontrando más genes relacionados. Y el coeficiente RDC es el que

encuentra más dependencia entre los genes en todos los niveles de significación.

Los cuatro genes del coeficiente MIC que tienen una significatividad menor a 0.0001, son ANKRD26, NEURL2, SPRYD3 y ZNF670, estos son detectados también con la misma significatividad por RDC a excepción de NEURL2 que es de 0.005, pero no coinciden con ninguno de los doce localizados por el Coeficiente de correlación r con la misma significatividad.

De los doce detectados con r, los que obtienen los p-valores más bajos y, por lo tanto, que más fuertemente rechazan la hipótesis nula de independencia lineal son CENTB5, DUSP13 y GABBR1, pero no son significativos en el resto de coeficientes con un p-valor inferior a 0.0001.

Tal y como se puede ver en el tabla 10, de los 251 genes con p-valores menores a 0.01 de r, el 42.2% son detectados también por la correlación de distancias, pero esta proporción está por debajo del 25.0% para los genes coincidentes con RDC o MIC. El número de genes expresados significativamente es algo menor con el coeficiente MIC que con la correlación de Pearson y por lo tanto tiene menos coincidencias con RDC y dCor, sobre todo con este último. Respecto al coeficiente RDC, es el que más genes ha detectado a ese nivel de significación, pero el ratio de coincidencia con dCor es sólo del 8.6%.

Tabla 10: Número de genes coincidentes con p-valores <0.01

	r	**MIC**	**RDC**	**dCor**
r	251	37	62	106
MIC		213	56	43
RDC			500	43
dCor				160

De los 11.191 genes analizados, catorce han sido detectados por todos los coeficientes analizados, 42 más por tres coeficientes y aumenta hasta los 137 los genes significativos en dos coeficientes. En la tabla 11 se puede ver cuáles son estos 14 genes significativos, con los diferentes valores obtenidos en cada uno de los coeficientes junto con sus p-valores.

Tabla 11: Genes con una significación inferior a 0.01 en los cuatro coeficientes

Gen	r		MIC		RDC		dCor	
	Corr.	p-valor	Corr.	p-valor	Corr.	p-valor	Corr.	p-valor
ANKRD26	-0.5447	0.0049	0.9988	0.0000	0.8860	0.0002	0.6803	0.005
C6orf75	0.7078	0.0001	0.7785	0.0003	0.8129	0.0023	0.6976	0.005
CREM	0.7496	0.0000	0.6105	0.0040	0.8240	0.0005	0.7525	0.005
DEXI	-0.7362	0.0000	0.5638	0.0087	0.7834	0.0012	0.7131	0.005
DYNC1LI1	-0.5595	0.0036	0.5615	0.0095	0.6790	0.0020	0.5953	0.005
EEF1E1	0.7183	0.0001	0.7232	0.0005	0.8648	0.0081	0.7500	0.005
FAM57A	-0.5895	0.0019	0.6421	0.0028	0.8339	0.0007	0.6168	0.005
FER	-0.7319	0.0000	0.6815	0.0013	0.8663	0.0053	0.7445	0.005
LIG3	0.6897	0.0001	0.6757	0.0015	0.8255	0.0036	0.7334	0.005
NEURL2	-0.5786	0.0024	0.8388	0.0001	0.8431	0.0048	0.6783	0.005
STRN4	-0.6030	0.0014	0.7300	0.0005	0.8776	0.0051	0.7324	0.005
TOPBP1	0.6213	0.0009	0.7910	0.0003	0.8845	0.0000	0.7586	0.005
ZBTB12	0.5443	0.0049	0.5666	0.0073	0.7095	0.0039	0.5454	0.005
ZFP161	0.6191	0.0010	0.6768	0.0015	0.7514	0.0014	0.6474	0.005

2.6. Análisis de los resultados

Se ha tomado como base el coeficiente de correlación de Pearson y se ha comparado tanto teóricamente como de forma práctica con cuatro coeficientes más modernos, definidos entre los años 2007 y 2013.

De las simulaciones realizadas para los ocho tipos diferentes de asociaciones entre variables calculadas, el coeficiente RDC ha obtenido la media más alta en tres de ellas, concretamente en la Cuadrática, la Cúbica y la Exponencial (superior a 0.90). Y en cuatro ocasiones ha sido la segunda con mejor puntuación (Lineal, Escalón, Círculo y Sinusoidal), sólo en el conjunto de datos pseudo-aleatorios queda en tercer puesto.

El siguiente coeficiente que ha detectado más asociaciones ha sido CorGC pero no se puede considerar un buen estimador ya que sólo se ha podido estimar para una parte de la muestra.

MIC que se posicionaba como uno de los coeficientes más fuertes para encontrar asociaciones entre variables, sólo ha conseguido obtener en dos ocasiones la puntuación más alta de los cinco coeficientes, concretamente en la del Escalón y la Sinusoidal y un segundo puesto en la Cuadrática.

Por lo que, con el análisis realizado el Coeficiente de dependencia aleatoria (RCD) es el estadístico idóneo para encontrar dependencia entre variables, respecto al resto de coeficientes estudiados, ya que encuentra un elevado número de asociaciones, cumple con las propiedades propuestas por Rényi, es sencillo y requiere un coste computacional bajo.

3. Software desarrollado

3.1. Funciones disponibles en R

Actualmente la mayoría de los coeficientes estudiados están incluidos en paquetes en el repositorio CRAN o bien están disponibles el código libremente. A continuación, se detallan las funciones y los paquetes con los que calcular los estadísticos.

Correlación de Pearson (r)

Hay dos posibles maneras de calcular el coeficiente de correlación de Pearson.

La primera manera es con el paquete `stats` que proporciona dos funciones `cor(x, y, use=, method=)`[1] y `cor.test(x, y, alternative=, method=, exact=, conf.level=, continuity=, ...)`[2]. Con la primera se calcula el coeficiente de correlación y con la segunda calcula la prueba t-student para la correlación de muestras pareadas, devolviendo el p-valor de la significación.

La segunda manera es con el paquete `Hmisc` que con una única función devuelve una lista con los elementos: r, que es la matriz de correlaciones, n que es la matriz del número de observaciones utilizadas en el análisis para cada par de variables, y la matriz P con los p-valores asintóticos. `rcorr(x, y, type=)`[3] calcula los coeficientes de correlación de Pearson o Spearman de todos los pares de columnas de la matriz. Los valores ausentes se eliminan por pares en vez de eliminar todas las filas.

Para hacer la aplicación se ha utilizado la función `rcorr` del paquete `Hmisc` ya que con una única función se dispone de toda la información necesaria, además de eliminar menos datos ante valores missings.

Coeficiente de Información Máxima (MIC)

Los propios autores del MIC en la página web MINE indican que se puede bajar el Paquete `minerva` de R que calcular los estadísticos MINE (Maximal Information-based Nonparametric Exploration).

La función se llama `mine` y tiene la siguiente estructura:

```
mine(x, y=NULL, master=NULL, alpha=0.6, C=15, n.cores=1,
var.thr=1e-5, eps=NULL, est="mic_approx", na.rm=FALSE,
use="all.obs", ...)
```
[4]

[1] Documentación de la función: https://www.rdocumentation.org/packages/stats/versions/3.5.0/topics/cor

[2] Documentación de la función: https://www.rdocumentation.org/packages/stats/versions/3.5.0/topics/cor.test

[3] Documentación de la función: https://www.rdocumentation.org/packages/Hmisc/versions/4.1-1/topics/rcorr

[4] Documentación del paquete: https://cran.r-project.org/web/packages/minerva/minerva.pdf

Tanto `alpha` como `C` son los parámetros del Coeficiente de Información Máxima que por definición son 0.6 y 15 respectivamente. La función además de devolver el coeficiente MIC también devuelve en la misma salida los coeficientes de Puntuación máxima de asimetría (MAS), Valor de borde máximo (MEV), Número mínimo de celdas (MCN) y MIC-R2 que es la diferencia de MIC respecto a la correlación de Pearson.

Correlación a lo largo de una curva de generación (CorGC)

Las funciones necesarias para calcular la CorGC, están disponibles en la página PCOP: Principal Curves of Oriented Points [21] de Delicado y Smrekar (2008). Para ello es necesario bajarse el zip e instalar las funciones en R.

Para que funcione tiene que cargarse también los paquetes `princurve`, `KernSmooth`, y las funciones `pcop`, `localpolreg_no_plot`, tal y como se muestra a continuación:

```
source("CovrGC.R")
if (!require("princurve")) install.packages("princurve")
if (!require("KernSmooth")) install.packages("KernSmooth")
source("pcop.R")
source("locpolreg_no_plot.r")
```

Con la función `Covr` se calcula tanto la covarianza como la correlación generalizadas a lo largo de la curva principal. Se pueden calcular a través de dos métodos; usando la curva principal de Hastie-Stuetzle del paquete `princurve` o usando la función `pcop` según la curva principal de puntos orientados de Delicado. El formato de la función es `Covr(x, method="HS", h.method="dpill", plot.true=FALSE, …)`.

Devolviendo una lista con la curva principal ajustada a los puntos en x (PC), la Covarianza a lo largo de la curva principal de generación (CovGC), la Correlación a lo largo de la curva principal de generación (CorGC), la Covarianza local a lo largo de la curva principal de generación (LCovGC) y Correlación local a lo largo de la curva principal de generación (LCorGC).

Cuando se han realizado las pruebas con el estadístico, con el método *PCOP* ha dado problemas con Windows10 y no realiza los cálculos. Esto es debido a que el paquete no está actualizado con los nuevos sistemas operativos ya que la última modificación es del 6 de junio de 2007. Por este motivo al hacer los cálculos se ha utilizado la curva principal con el método de Hastie-Stuetzle.

Este método se descartó finalmente de incluirlo en la aplicación ya que tiene un alto porcentaje de casos en los que no se puede calcular, como se demostró en las simulaciones.

Coeficiente de dependencia aleatorio (RDC)

El cálculo del coeficiente lo proporcionan los autores del artículo en el apéndice, estando también disponible para MATLAB [4, 20].

Para el cálculo se llama a la función `cancor` del paquete `stats` para obtener las correlaciones canónicas, por lo que es necesario tener cargada dicho paquete anteriormente.

La función para calcular el RDC se indica a continuación, siendo una función simple y corta:

```
rdc <- function(x,y,k=20,s=1/6,f=sin) {
  x <- cbind(apply(as.matrix(x),2,function(u)rank(u)/length(u)),1)
  y <- cbind(apply(as.matrix(y),2,function(u)rank(u)/length(u)),1)
  x<- s/ncol(x)*x%*%matrix(rnorm(ncol(x)*k),ncol(x))
  y <- s/ncol(y)*y%*%matrix(rnorm(ncol(y)*k),ncol(y))
  cancor(cbind(f(x),1),cbind(f(y),1))$cor[4]
}
```

Los argumentos que se pasan en la función son los siguientes:

Argumento	Definición
x	Un vector, matriz o dataframe numérico
y	Un vector, matriz o dataframe numérico
k	Número de proyecciones no lineales de la cópula, por definición $k = 20$
s	Varianza para dibujar i.i.d. coeficientes de proyección en $N \sim (0, sI)$, por defecto es $1/6$
f	Función que se utiliza para la generación de las proyecciones no lineales aleatorias, si no se indica, utiliza las proyecciones sinusoidales (sin)

Devuelve una única correlación, por lo que si se necesita hacer comparaciones de una matriz por vectores se tiene que pasar uno a uno.

Correlación de distancias (dCor)

El paquete `energy`[5] de R, de los propios autores (M. Rizzo y G. Székely), contiene las funciones `dcov` y `dcor` para calcular la covarianza y la correlación de distancias, además de las funciones necesarias para calcular los test de independencia para una muestra, dos y muestras múltiples para comprobar las distribuciones multivariables, entre otras pruebas y estadísticos.

La función para calcular la correlación de distancias se llama `dcor(x, y, index=1.0)` y tiene la misma estructura que `dcov` y `DCOR`.

Como requerimiento, los tamaños de muestra (número de filas) de las dos muestras deben coincidir y no deben contener valores faltantes. Los argumentos `x` e `y` pueden opcionalmente ser objetos `dist`; de lo contrario, estos argumentos se tratan como datos.

Como se ha comentado anteriormente, se proporcionan dos métodos para calcular los estadísticos. `DCOR` es una función `R` independiente que devuelve una lista de estadísticos. `dcov` y `dcor` proporcionan interfaces `R`

[5] Documentación del paquete: https://cran.r-project.org/web/packages/energy/energy.pdf

a la implementación `C`, que generalmente es más rápida. `dcov` y `dcor` llaman a la función interna `dcov`, por lo que se tiene que tener en cuenta que es ineficiente calcular dCor manualmente con las funciones `dcov(x,y)`, `dcov(x,x)` y `dcov(y,y)` porque las diferentes llamadas a `dcov` implican una repetición innecesaria de cálculos. Si se quiere obtener los cuatro estadísticos es más eficiente llamar a la función `DCOR`.

Para calcular el estadístico junto con su p-valor existe la función `dcor.test (x, y, index=1.0, R)` que está en el mismo paquete, añadiendo un argumento más, R, que indica el número de replicaciones para realizar la prueba de permutación.

Devolviendo una lista con la descripción del test (`method`), el valor observado del estadístico (`statistic`), un vector con las estimaciones de dCov(x,y), dCor(x,y), dVar(x) y dVar(y) (`estimates`), replicaciones del test (`replicates`), el p-valor aproximado de la prueba (`p.value`), el tamaño de la muestra (`n`) y la descripción de los datos (`data.name`). Esta opción es la que se ha utilizado en el paquete.

3.2. Funciones creadas

Para poder disponer de todas las funciones necesarias para utilizar la aplicación Shiny se ha creado un paquete con el que con una única función se calculasen los cuatro estadísticos. El paquete está disponible en https://github.com/AnaBPazos/AlterCorr.git

El paquete `AlterCorr` calcula las correlaciones entre dos variables o matrices o dataframes, para cuatro coeficientes de independencia; Correlación de Pearson, Coeficiente de información máximo (MIC), Coeficiente de dependencia aleatorio (RDC) y Correlación de distancias (dCor). Junto con las correlaciones, calcula los p-valores y los p-valores ajustados para cada tipo de coeficiente.

Cuando la función se utiliza para matrices y dataframes, se diferencia el cálculo entre datos emparejados (debe tener las mismas dimensiones) o independientes (solo deben coincidir en el mismo número de columnas).

Consiste de cuatro funciones:

- `MIC` que calcula el coeficiente de información máximo y sus valores p con una prueba de permutaciones.
- `rdc` que calcula el coeficiente de dependencia aleatoria, junto con la prueba de significación utilizando la aproximación de Bartlett.
- `AlterCorr` que hace los cálculos para vectores de variables.
- `AlterCorrM` que realiza cálculos para matrices de valores utilizando la función `AlterCorr`.

Función MIC

En el apartado anterior se indicaba que los autores proporcionaban en el paquete `minerva` la función `mine` que calculaba los estadísticos MINE, incluido MIC. Pero no han incluido en el paquete el test de permutación para obtener los p-valores y así detectar qué variables rechazan la hipótesis nula de independencia. Por este motivo se ha tenido que generar una función que calculase la prueba de permutación.

La función tiene la estructura `MIC(x, y, R=100, ...)`, con los siguientes argumentos:

Argumento	**Definición**
x	Un vector, matriz o dataframe numérico
y	Un vector, matriz o dataframe numérico
R	Número de repeticiones que se usarán para calcular el p-valor con el test de permutaciones.
...	Parámetros adicionales definidos en la función de código `mine` del paquete `minerva`.

En el anexo 7.2. se incluye el código creado en R de la función, en él se genera un dataframe con el número de replicaciones indicada (si se proporciona un número no permitido se modifica a 100), se calcula los estadísticos mine para cada una de las repeticiones y por último se calcula el p-valor calculando la proporción de puntuaciones que son mayores o iguales que la puntuación original obtenida. Devolviendo un dataframe con los dos valores (coeficiente y p-valor).

Función rdc

En este caso, el autor proporcionaba el código para calcular el coeficiente e indicaba en el artículo que si se cumplían las condiciones de normalidad se podía calcular una chi-cuadrado con la aproximación de Bartlett con las correlaciones canónicas, si no con el espectro de los productos internos de las matrices de proyección aleatorias no lineales para hacer el test.

Se ha modificado el código proporcionado (ver anexo 7.3.) para realizar la aproximación de Bartlett. Guardando todas las correlaciones canónicas obtenidas y haciendo el cálculo $\left(\frac{2k+3}{2} - n\right) \log \prod_{i=1}^{k} (1 - \rho_i^2) \sim \chi^2_{k^2}$. Cómo no siempre coincidía el número de correlaciones obtenidas con el parámetro k proporcionado en la función, se contrastaba con la chi-cuadrado con el número de correlaciones obtenidas al cuadrado.

Para próximas versiones se tendría que implementar el test para distribuciones asintóticas no paramétricas.

Al no modificarse los argumentos de la función proporcionada por el autor e indicados en el apartado 3.1. no ha habido cambios en la llamada de la función.

Función AlterCorr

Esta función devuelve el valor de la correlación entre dos variables y su correspondiente p-valor para los cuatro coeficientes estudiados. Es sencilla (ver punto 7.4. del anexo), ya que lo único que hace es llamar a las funciones específicas de cada coeficiente y devolver un dataframe con la correlación y el p-valor obtenido.

A continuación, se indica la estructura de llamada de la función, así como los parámetros que se tienen que proporcionar.

```
AlterCorr(x, y, type=c("pearson", "MIC", "RDC", "dCor"), R=100)
```

Argumento	**Definición**
x	Un vector numérico.
y	Un vector numérico de la misma dimensión que `x`.
Type	Especifica el tipo de correlaciones para calcular. "Pearson" para la Correlación de Pearson, "MIC" para el Coeficiente de información máximo, "RDC" para el Coeficiente de dependencia aleatoria y finalmente "dCor" para la Correlación de distancias.
R	Número de permutaciones que se realizarán para MIC y dCor para calcular los p-valores. Por defecto es 100.

Para poder utilizar la función es necesario tener instalados los paquetes `Hmisc`, `minerva` y `energy`. Para no tener que indicar en la aplicación Shiny los diferentes argumentos de cada coeficiente, se ha decidido dejar los valores por defecto en todos los coeficientes.

Función AlterCorrM

Esta función devuelve el valor de la correlación entre dos matrices, su correspondiente p-valor y p-valor ajustado para los cuatro tipos de coeficientes analizados. Dependiendo de la opción de comparación seleccionada, la correlación entre cada columna de la matriz se calculará con respecto a todas las columnas la segunda matriz o se calculará en parejas. Por cómo está definida la función, es necesario poner en las columnas las variables y en las filas las observaciones.

La función es `AlterCorrM(X, Y, type=c("pearson", "MIC", "RDC", "dCor"), comparison=c("all", "pairs"), R=100, method=c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"))`, donde los argumentos a indicar son los siguientes:

Argumento	Definición
X	Un vector numérico o matriz o dataframe.
Y	Una matriz numérica o dataframe, con el mismo número de filas que X.
Type	Especifica el tipo de correlaciones para calcular. "Pearson" para la Correlación de Pearson, "MIC" para el Coeficiente de información máximo, "RDC" para el Coeficiente de dependencia aleatoria y finalmente "dCor" para la Correlación de distancias.
comparison	Método con el cual se calcularán las correlaciones. La opción "pairs" es para comparar dos matrices que comparten filas y columnas, pero los valores son de diferentes variables y se quiere hacer comparaciones por pares. El caso “alls” es para cuando se dispone de dos matrices con una dimensión común (observaciones, individuos, muestras), y se desea calcular las correlaciones de todas las variables con todas.
R	Número de permutaciones que se realizarán para MIC y dCor para calcular los p-valores. Por defecto es 100.
method	Método de corrección de los p-valores. Puede ser abreviado Los posibles valores son "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".

La función (para ver el código ir al anexo 7.5.) lo primero que hace es comprobar que el número de repeticiones proporcionado sea correcto, si no le asigna el valor 100. A continuación, se comprueba que las matrices cumplen las condiciones para realizar el tipo de comparación solicitado. Cuando la opción de comparación es “alls”, calcula la correlación y los p-valores entre todas las variables llamando a la función `AlterCorr`. Por el contrario, si la opción de comparación es “pairs”, calcula la función `AlterCorr` para la misma columna de las dos matrices. Una vez calculados los coeficientes y sus p-valores, se procede a calcular los p-valores ajustados según el método de corrección seleccionado en el argumento `method`.

La función devuelve una lista con la matriz de correlaciones (Correlation), los p-valores (p-value) y los p-valores ajustados (adjPval).

3.3. Aplicación Shiny

Para generar la aplicación web Shiny se utilizó el programa RStudio con un único archivo llamado app.R.

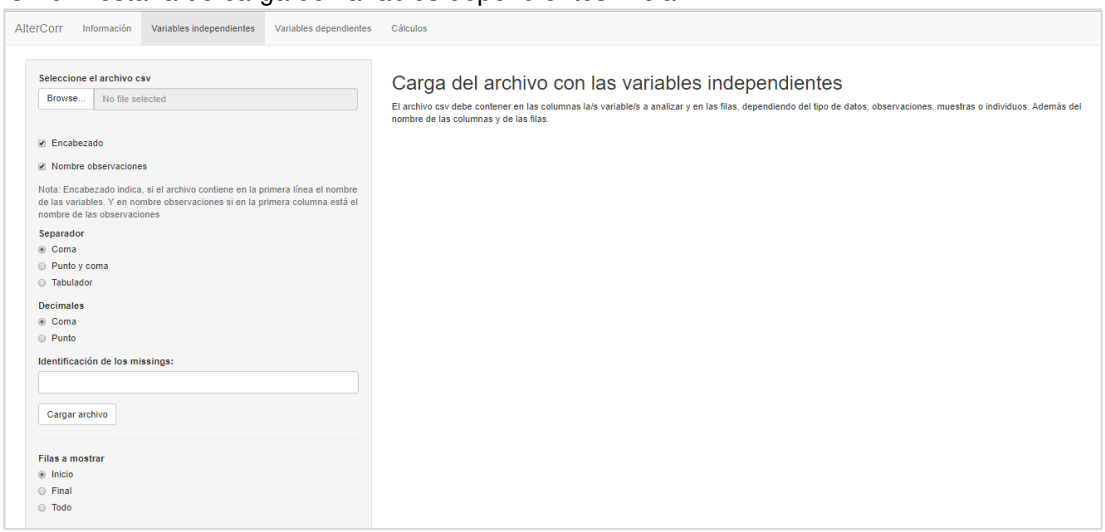
La aplicación se llama AlterCorrApp, está estructurada en cuatro pestañas y disponible en el servidor ShinyApps.io de RStudio, concretamente en la dirección https://abpazos.shinyapps.io/AlterCorrApp/.

Información es la primera pestaña, en ella se indica para qué sirve la aplicación y que se va a encontrar en cada pestaña. Todo esto para que el usuario, la primera vez que acceda, pueda utilizar la aplicación.

Las pestañas **Variables independientes** y **Variables dependientes**, están estructuradas de la misma forma (ver ilustración 6). A la izquierda aparece un sidebar con las opciones necesarias para importar los archivos csv y a la derecha un panel main, en el que inicialmente explica el formato que tiene que tener el archivo.

Debido a que los archivos pueden ser variados, se especifica que el archivo a cargar tiene que ser un csv, en el que las columnas son las variables a analizar y las filas, las observaciones. Con un checkbox se da la opción al usuario de indicar si el archivo a cargar contiene en la primera fila los nombres de las variables (encabezado) y/o si la primera columna incluye el nombre de las observaciones. La selección del tipo de separador de columnas y el símbolo utilizado para los decimales, se indica a través de dos radioButtons. Con un objeto del tipo textInput el usuario puede escribir como están identificados los valores faltantes en la base de datos.

Ilustración 6: Pestaña de carga de variables dependientes inicial

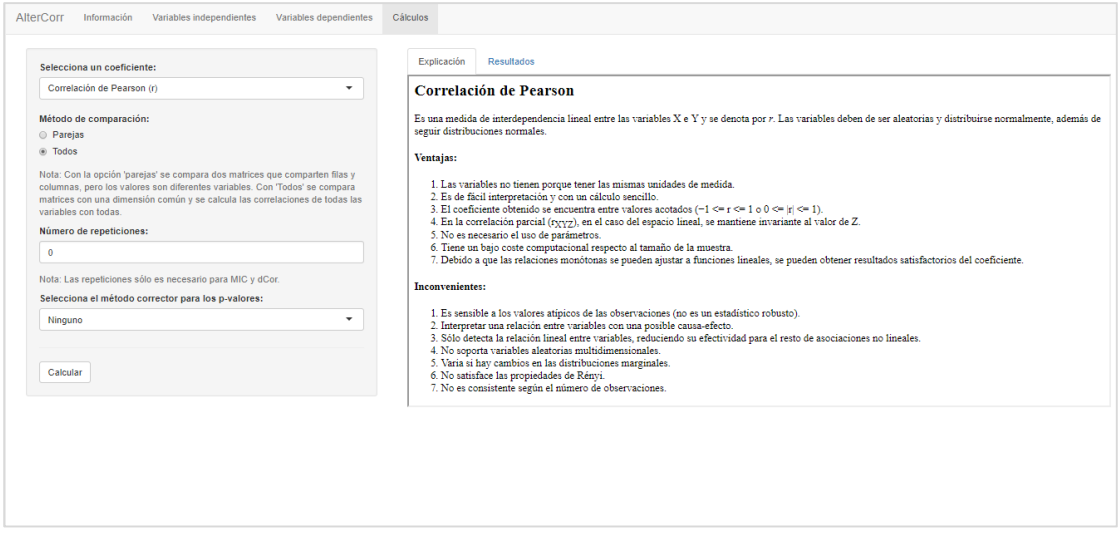


El archivo se carga sólo en el momento en el que se aprieta el botón de “cargar archivo”, modificando el contenido del panel main con una vista preliminar de los datos cargados. Según la opción seleccionada en filas a mostrar, se muestra sólo las primeras, las últimas filas o toda la base.

Ilustración 7: Pestaña variables dependientes con un archivo cargado

En la pestaña de **Cálculos** inicialmente muestra un sidebar a la izquierda con las diferentes opciones para calcular los coeficientes y con notas para un mayor entendimiento de la aplicación para el usuario. Y en la parte derecha, en el panel main, dos pestañas, la primera con una explicación del coeficiente y la segunda llamada Resultados en la que se mostraran los cálculos del coeficiente.

Ilustración 8: Estado inicial de la pestaña Cálculos

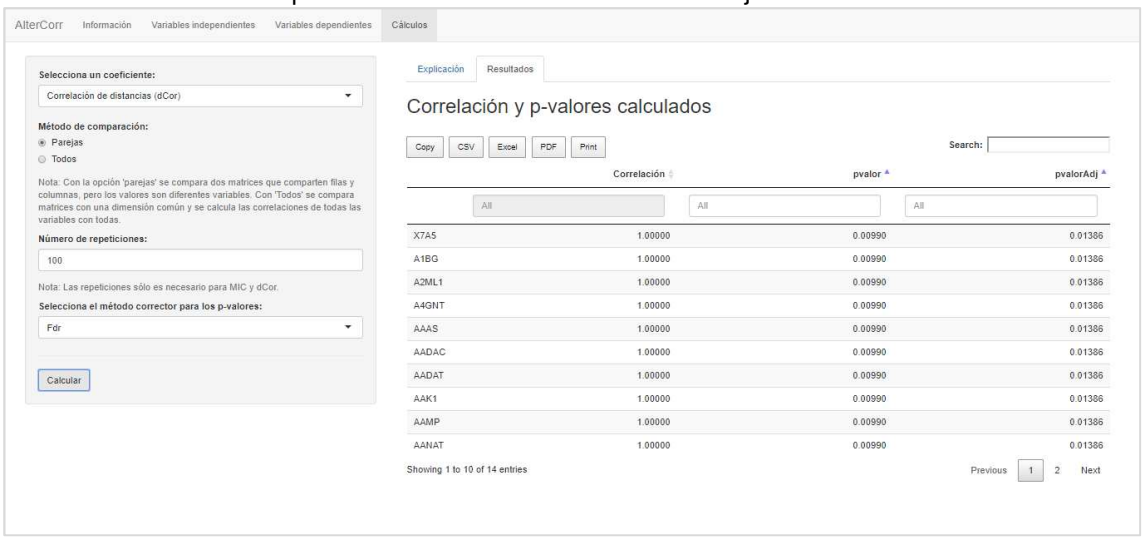


Cuando se selecciona un coeficiente de correlación de los cuatro estudiados con el selectInput, la información que se presenta en el tabPanel cambia, proporcionando al usuario la información necesaria para elegir el estadístico que más se adapta a sus datos.

En el sidebar está compuesto por todos los argumentos que se necesitan para llamar a la función `AlterCorrM`. Una vez seleccionados los parámetros y clicado el botón de Calcular, el panel derecho cambia según el método de comparación seleccionado.

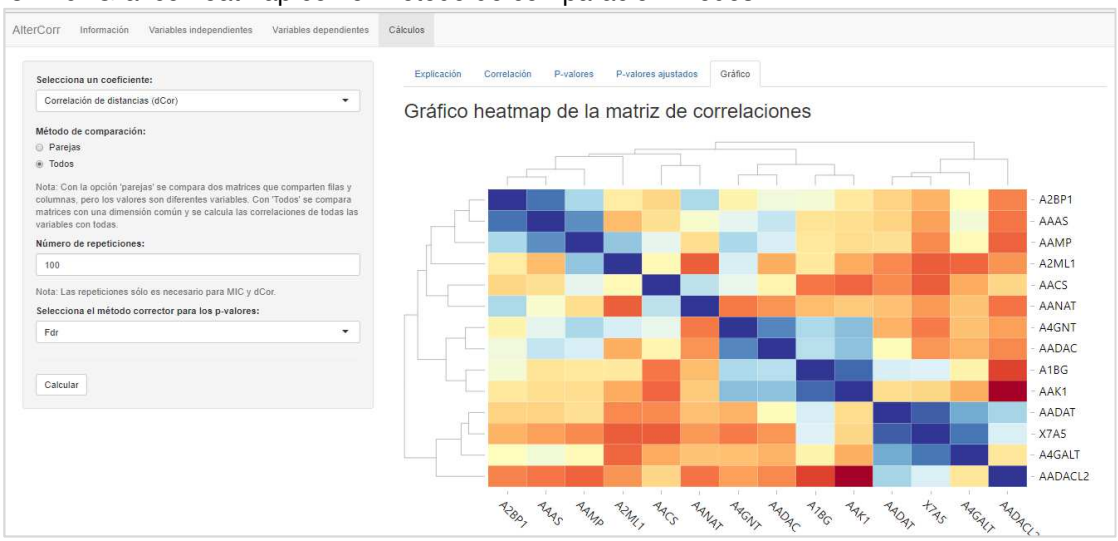
Si el usuario elige la comparación por parejas, en la pestaña de Resultados se muestra una tabla con los valores del coeficiente seleccionado, su p-valor y el p-valor ajustado (si se ha seleccionado un método corrector). Los datos se presentan ordenados ascendentemente por el valor del p-valor, de esta forma el usuario puede ver aquellas variables más significativas primero. La tabla da la opción de bajarse los datos en varios formatos, hacer búsquedas y ordenar por la columna que más le interese al usuario.

Ilustración 9: Estado final de la pestaña Cálculos con el método "Parejas"



Si, por el contrario, el usuario elige la opción de comparar todas las variables, los resultados se muestran en cuatro paneles diferentes. El primero (Correlación) muestra una tabla con las correlaciones de las variables, la segunda con los p-valores, la tercera con los p-valores ajustados y en la última (Gráfico) se muestra un gráfico heatmap con las correlaciones calculadas.

Ilustración 10: Gráfico heatmap con el método de comparación "Todos"



3.4. Valoración económica

Si esto no fuera un trabajo formativo y hubiese sido generado por una empresa como producto para ser ofrecido a sus clientes tendría unos gastos de desarrollo, así como de mantenimiento. En la siguiente tabla se pueden ver los gastos de desarrollo, investigación y contratación del servidor donde alojar dicha aplicación.

Concepto	Cantidad	Precio unidad	Importe
Análisis de resultados	187 h	20.00€	3740.00€
Analista desarrollador de Software	83 h	25.00€	2075.00€
Presentación de resultados	1 u	1500.00€	1500.00€
Alquiler de servidor	1 u	400.00€	400.00€
Total			7715.00€
21% de IVA			1620.15€
TOTAL			**9335.15€**

A partir del segundo año, se requeriría un mantenimiento por parte del analista de 20h al mismo coste económico, además del alquiler del servidor y del incremento del IPC anual, por lo que cada año tendría un gasto aproximado de unos 900.00€ sin IVA.

La aplicación web generada podría ser de utilidad a investigadores con un bajo perfil estadístico y que requieran de analizar datos génicos o de dependencia entre variables.

4. Conclusiones

La correlación de Pearson fue un gran descubrimiento para encontrar dependencia lineal entre dos variables de una forma sencilla y con un bajo coste computacional, pero que estaba condicionada a unos supuestos que en la práctica son difíciles de asumir y que en muchos casos se obviaba.

La correlación del s.XXI, como se llamó al coeficiente MIC, acotó los resultados de la Información Mutua a un coeficiente entre 0 y 1. Y como está creado para cumplir las propiedades heurísticas de generalidad y equidad, es capaz de encontrar un gran número de asociaciones entre dos variables, seleccionando las relaciones más fuertes del conjunto de datos. Cuando estas asociaciones son lineales se aproxima a la correlación de Pearson. Respecto al resto de coeficientes tiene un menor poder para detectar relaciones débiles. Otros inconvenientes son que no tiene por qué encontrar el máximo real del coeficiente y requiere un coste computacional elevado.

El coeficiente CorGC al estar basado en las curvas de generación consigue detectar asociaciones no lineales sobre todo en las que están relacionadas con curvas. Y si la relación entre las variables es lineal, el

valor obtenido coincide con el valor absoluto de la correlación de Pearson. Pero tiene importantes factores negativos en su contra, como su elevado coste computacional y que no siempre es posible calcularlo.

El estadístico más actual que se ha analizado ha sido el Coeficiente de dependencia aleatorio (RDC), siendo el único que cumple todas las propiedades propuestas por Rényi para un buen coeficiente de asociación, además de encontrar más tipos de dependencias entre las variables. De los coeficientes estudiados, es uno de los pocos capaz de soportar variables multidimensionales, todo esto con un bajo coste computacional y una fácil implementación.

El otro coeficiente que se puede calcular con variables multidimensionales es la Correlación de distancias o browniana, tiene la ventaja de que está definida de forma similar a la correlación de Pearson, por lo que es fácil de entender, la diferencia es que están basadas en distancias euclídeas. Mejorando este coeficiente, logrando establecer la dependencia entre asociaciones no lineales, cumplir con tres de las siete propiedades fundamentales propuesta por Rényi y el resto cumplirlas parcialmente. Pero requiere de un elevado coste computacional.

Gracias a las simulaciones realizadas y a la base de datos génica analizada, se ha constatado que el estadístico que más asociaciones ha encontrado ha sido el RDC.

Después de haber hecho el estudio teórico y práctico, el coeficiente MIC que se presentaba como el mejor estadístico de la actualidad, se ha comprobado que no es el más idóneo. De los cuatro coeficientes probados el más completo es RDC para sustituir la correlación de Pearson.

Con el desarrollo de la aplicación web, se ha profundizado en el lenguaje R y RStudio. Con la creación del paquete `AlterCorr`, se ha podido contrastar que la confección de paquetes ayuda a gestionar mejor las diferentes funciones necesarias para el trabajo diario. Al colgarlo en el repositorio GitHub, éste puede ser descargado por el propio autor y por otros usuarios para ser utilizado. Además de dar la oportunidad de mejorar el paquete a otros desarrolladores, fomentando el intercambio de conocimientos y una mejora continua. Por otro lado, con la aplicación Shiny creada se ha comprobado que, sin la necesidad de tener un gran conocimiento en la generación de páginas web, se puede construir una aplicación útil y actual sin grandes problemas, con los manuales y Cheat Sheet proporcionados por RStudio [22, 23].

Respecto a los objetivos plantados, como se ha podido comprobar, se han cumplido tanto los genéricos como los específicos indicados en el apartado 1.2.

Por otro lado, en la planificación no se ha seguido la planeada inicialmente en la Fase 1 del trabajo. Aunque, la metodología era la correcta, dando la posibilidad a adaptarla según los coeficientes seleccionados.

Tras la tarea de identificación de los estadísticos, se cambiaron dos de los estadísticos iniciales y se dejaron para un estudio futuro algunas variantes de la correlación de Pearson, además del coeficiente de correlación de rangos de Spearman, del coeficiente de correlación de rangos de Kendall, de la Distancia de Hoeffing y de la Información Mutua.

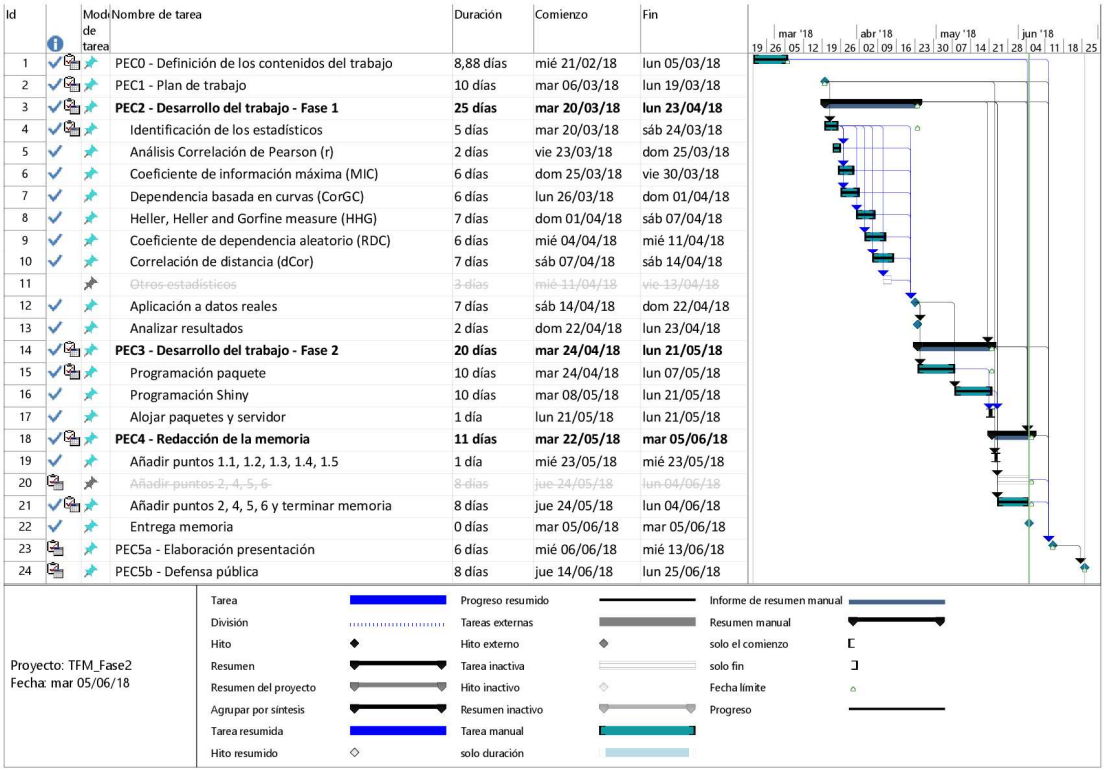
Debido a que se alargaron en el tiempo el análisis de todos los estadísticos estudiados, se decidió no realizar la tarea de análisis de otros estadísticos y dejar sólo los 5 estadísticos estudiados, además de la prueba HHG. Y alternar la lectura de documentación de un estadístico con la realización del resumen y pruebas de otro estadístico, de esta forma se maximizaba el uso del software.

Se dispuso generar las simulaciones de las ocho distribuciones de dependencia que no estaban planeadas inicialmente. Englobándose en la tarea de análisis de resultados.

También se había planificado ir redactando la memoria al finalizar cada una de las fases del trabajo, al final se redactó una vez finalizadas las dos primeras fases.

A continuación, se puede ver el diagrama de Gantt con el conograma real de las tareas realizadas.

Ilustración 11: Diagrama de Gantt final



Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
1		PEC0 - Definición de los contenidos del trabajo	8,88 días	mié 21/02/18	lun 05/03/18
2		PEC1 - Plan de trabajo	10 días	mar 06/03/18	lun 19/03/18
3		**PEC2 - Desarrollo del trabajo - Fase 1**	**25 días**	**mar 20/03/18**	**lun 23/04/18**
4		Identificación de los estadísticos	5 días	mar 20/03/18	sáb 24/03/18
5		Análisis Correlación de Pearson (r)	2 días	vie 23/03/18	dom 25/03/18
6		Coeficiente de información máxima (MIC)	6 días	dom 25/03/18	vie 30/03/18
7		Dependencia basada en curvas (CorGC)	6 días	lun 26/03/18	dom 01/04/18
8		Heller, Heller and Gorfine measure (HHG)	7 días	dom 01/04/18	sáb 07/04/18
9		Coeficiente de dependencia aleatorio (RDC)	6 días	mié 04/04/18	mié 11/04/18
10		Correlación de distancia (dCor)	7 días	sáb 07/04/18	sáb 14/04/18
11		~~Otros estadísticos~~	~~3 días~~	~~mié 11/04/18~~	~~vie 13/04/18~~
12		Aplicación a datos reales	7 días	sáb 14/04/18	dom 22/04/18
13		Analizar resultados	2 días	dom 22/04/18	lun 23/04/18
14		**PEC3 - Desarrollo del trabajo - Fase 2**	**20 días**	**mar 24/04/18**	**lun 21/05/18**
15		Programación paquete	10 días	mar 24/04/18	lun 07/05/18
16		Programación Shiny	10 días	mar 08/05/18	lun 21/05/18
17		Alojar paquetes y servidor	1 día	lun 21/05/18	lun 21/05/18
18		**PEC4 - Redacción de la memoria**	**11 días**	**mar 22/05/18**	**mar 05/06/18**
19		Añadir puntos 1.1, 1.2, 1.3, 1.4, 1.5	1 día	mié 23/05/18	mié 23/05/18
20		~~Añadir puntos 2, 4, 5, 6~~	~~8 días~~	~~jue 24/05/18~~	~~lun 04/06/18~~
21		Añadir puntos 2, 4, 5, 6 y terminar memoria	8 días	jue 24/05/18	lun 04/06/18
22		Entrega memoria	0 días	mar 05/06/18	mar 05/06/18
23		PEC5a - Elaboración presentación	6 días	mié 06/06/18	mié 13/06/18
24		PEC5b - Defensa pública	8 días	jue 14/06/18	lun 25/06/18

Para próximos estudios, sería conveniente comprobar los coeficientes estudiados con diferentes muestras, ya que pueden tener comportamientos diferentes con muestras más pequeñas donde no se pueda asumir normalidad con el teorema central del límite.

Además de incluir otros coeficientes que no se han podido tener en cuenta en este trabajo como el coeficiente de correlación de rangos de Spearman, el coeficiente de correlación de rangos de Kendall, la Distancia de Hoeffing, la Información Mutua, los nuevos estadísticos MINE (TIC, MIC_e o TIC_e), el HSIC, la correlación Global Gaussiana o el coeficiente RV entre otros que se han encontrado al investigar los cinco coeficientes para este trabajo.

Respecto al desarrollo del Software generado, en el paquete `AlterCorr` se podría mejorar la función `MIC` para calcular el resto de estadísticos MINE. En la función `rdc`, generar la prueba estadística cuando no se puede asumir normalidad en los datos. Además de incluir más estadísticos y, modificar los actuales, para que el usuario pueda elegir los argumentos de los diferentes coeficientes y no utilizar los que vienen por defecto.

En la aplicación web sería conveniente que los archivos a subir no tuviesen que ser necesariamente un csv, que el usuario pudiese indicar si las variables a analizar (p.ej. genes) están en alojados en las columnas o en las filas. Y, por último, dar la opción al usuario de guardar el heatmap generado, así como modificar su estética.

5. Glosario

Acrónimo/ Término	Definición
CorGC	Correlación a lo largo de una curva de generación
dCor	Correlación de distancias
HHG	Medida Heller, Heller y Gorfine
HS	Método de Hastie y Stuetzle para el cálculo de la curva principal
MI	Información Mutua, es una medida que indica cuánta información comparten dos variables, definida como $I(X,Y) = H(X) + H(Y) - H(X,Y)$. Siendo simétrica y tomando valores entre 1 y ∞.
MIC	Coeficiente máximo de información, del inglés Maximal Information Coefficient
MINE	Maximal Information-based Nonparametric Exploration
RDC	Coeficiente de dependencia aleatorio, en inglés Randomized Dependence Coefficient
TFM	Trabajo Final de Máster

6. Bibliografía

[1] T. Speed, «A Correlation for the 21st Century,» *Science,* vol. 334, nº 6062, pp. 1502-1503, 2011.

[2] F. Ríus y J. Wärnberg, Bioestadística, Segunda ed., Paraninfo, 2014, p. 56.

[3] C. M. Cuadras, Problemas de probabilidades y estadística, Segunda ed., vol. 1 Probabilidades, Barcelona, Barcelona: EUB, 1999, pp. 239-240.

[4] D. Lopez-Paz, P. Henning y B. Shölkopof, «Cornell University Library,» 29 Abril 2013. [En línea]. Available: https://arxiv.org/pdf/1304.7717v2.pdf. [Último acceso: 25 Marzo 2018].

[5] S. de Siqueira Santos, D. Takahashi, A. Nakata y A. Fujita, «A comparative study of statistical methods used to identify dependencies between gene expression signals,» *Briefings in Bioinformatics,* vol. 15, nº 6, pp. 906-918, 2014.

[6] «MINE: Maximal Information-based Nonparametric Exploration,» [En línea]. Available: http://www.exploredata.net. [Último acceso: 31 Enero 2018].

[7] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher y P. C. Sabeti, «Detecting Novel Associations in Large Data Sets,» *Science,* vol. 334, nº 6062, pp. 1518-1524, 2011.

[8] M. Clark, «A comparison of Correlation Measures,» Center for Social Research, University of Notre Dame, 2013.

[9] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher y P. C. Sabeti, «Equitability Analysis of the Maximal Information Coefficient, with Comparisons,» ArXiv e-prints, 2013.

[10] P. Delicado y M. Smrekar, «Measuring non-linear dependence for two random variables distributed along a curve,» *Statistics and Computing,* vol. 19, nº 3, pp. 255-269, 2009.

[11] S. Hoberman, «Carolina Digital Repository,» 01 Agosto 2014. [En línea]. Available: https://cdr.lib.unc.edu/record/uuid:e9656b36-1236-4137-a852-13db0c8de594. [Último acceso: 02 Abril 2018].

[12] K. Balázs, «Principal curves: learning, design, and applications,» 1999. [En línea]. Available: https://spectrum.library.concordia.ca/956/1/NQ47714.pdf. [Último acceso: 04 Abril 2018].

[13] D. Liu, X. Jiang, H. J. Zheng, B. Xie, H. Wang, T. He y X. Hu, «The Modularity of Microbial Interaction Network in Healthy Human Saliva: Stability and Specificity,» de *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Kansas City, MO, 2017.

[14] P. Fiedor, «Analysis of the Time Evolution of Non-Linear Financial Networks,» *Acta Universitatis Lodziensis. Folia Oeconomica,* vol. 3, nº 314, 2015.

[15] G. J. Székely, M. L. Rizzo y N. K. Bakirov, «Measuring and testing dependence by correlation of distances,» *Ann. Statist.,* vol. 35, nº 6, pp. 2769-2794, 2007.

[16] G. J. Székely y M. L. Rizzo, «Brownian distance covariance,» *Ann. Appl. Stat.,* vol. 3, nº 4, pp. 1236-1265, 2009.

[17] R. Lyons, «Distance covariance in metric spaces,» *Ann. Probab.,* vol. 41, nº 5, pp. 3284-3305, 2013.

[18] M. E. Paler, «On Modern Measures and Tests of Multivariate Independence,» Electronic Thesis or Dissertation, Bowling Green, 2015.

[19] J. R. Berrendero, «Correlación lineal y correlación de distancias,» FME, Barcelona, 2017.

[20] D. Lopez-Paz, P. Henning y B. Schölkopf, «The Randomized Dependence Coefficient,» Diciembre 2013. [En línea]. Available: https://pdfs.semanticscholar.org/2025/d548fb11c7bde92a47f20c77610d68477110.pdf. [Último acceso: 09 Abril 2018].

[21] P. Delicado y M. Huerta, «PCOP: Principal Curves of Oriented Points,» 06 Junio 2007. [En línea]. Available: http://www-eio.upc.es/~delicado/PCOP/. [Último acceso: 31 Marzo 2018].

[22] RStudio Inc., «Shiny from RStudio,» 2017. [En línea]. Available: https://shiny.rstudio.com/tutorial/. [Último acceso: 10 Mayo 2018].

[23] RStudio, Inc., «Cheatsheets RStudio,» Enero 2016. [En línea]. Available: https://www.rstudio.com/resources/cheatsheets/. [Último acceso: 08 Mayo 2018].

[24] R. Heller, Y. Heller y M. Gorfine, «A consistent multivariate test of association based on ranks of distances,» *Biometrika ,* vol. 100, nº 2, pp. 503-510, Enero 2012.

[25] J. Josse y S. Holmes, «Measuring multivariate association and beyond,» *Statistics surveys,* vol. 10, pp. 132-167, 2016.

[26] C. M. Cuadras, Problemas de probabilidades y estadística, Primera ed., vol. 2 Inferencia estadística, Barcelona, Barcelona: EUB, 2000, pp. 209-214.

[27] B. Brill, «The HHG Package - Multivariate and Univariate non-parametric Independence and K-Sample tests,» 26 Octubre 2016. [En línea]. Available: https://cran.r-project.org/web/packages/HHG/vignettes/HHG.html. [Último acceso: 07 Abril 2018].

7. Anexos

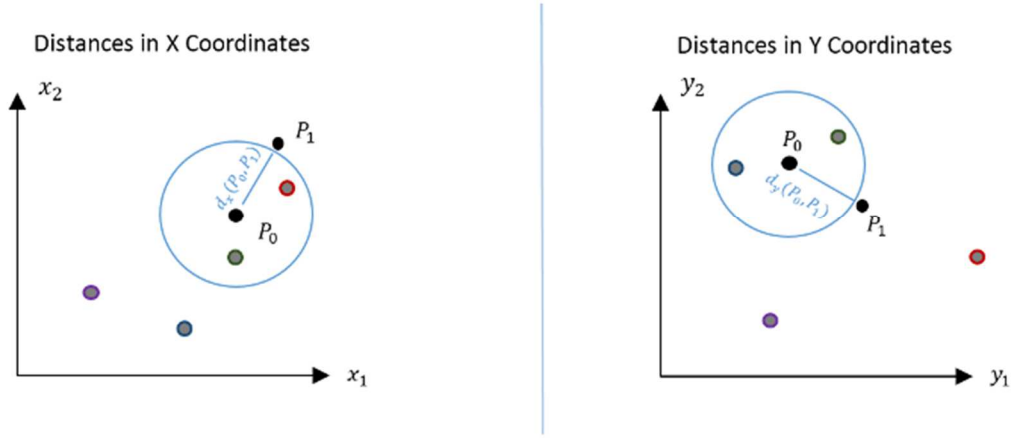
7.1. Heller, Heller and Gorfine measure (HHG)

A continuación, se expone la prueba de independencia HHG que, aunque no es un coeficiente es una prueba suficientemente fuerte de dependencia como para tenerla en cuenta.

Heller, Heller y Gorfine (HHG) la propusieron en 2012 afirmando que es una prueba sencilla aplicable en todas las dimensiones y consistente contra todas las posibles asociaciones [24]. Esta prueba está basada en la distancia por pares entre los valores X e Y ($d_x\ (x_i, x_j)$ y $d_y\ (y_i, y_j)$, donde $i, j \in (1, \cdots, n)$).

En el estadístico se tuvo en cuenta que, si X e Y son independientes y tienen una densidad conjunta continua, entonces hay un punto, digamos (x_0, y_0) en el espacio muestral de las variables X e Y, y unos radios R_x y R_y con centro en los puntos x_0 e y_0 respectivamente, de modo que la distribución conjunta de X e Y es diferente que las distribuciones marginales en el producto cartesiano de las esferas alrededor de (x_0, y_0) [5, 24].

Ilustración 12: Imagen de la partición HHG del espacio multivariante



Como no se puede establecer a priori este punto (x_0, y_0), la prueba lo que hace es contar el número de pares con ordenación concordante o discordante de las distancias, donde $d(\cdot,\cdot)$ es la distancia de la norma entre dos puntos [24, 25], de la siguiente forma:

$$A_{11} = \sum_{\substack{k=1,\\ k\neq i,\\ k\neq j}}^{n} I\left\{d(x_i, x_k) \leq d\left(x_i, x_j\right)\right\} I\left\{d(y_i, y_k) \leq d\left(y_i, y_j\right)\right\}$$

$$A_{12} = \sum_{\substack{k=1,\\ k\neq i,\\ k\neq j}}^{n} I\left\{d(x_i, x_k) \leq d\left(x_i, x_j\right)\right\} I\left\{d(y_i, y_k) > d\left(y_i, y_j\right)\right\}$$

$$A_{21} = \sum_{\substack{k=1,\\ k\neq i,\\ k\neq j}}^{n} I\{d(x_i, x_k) > d(x_i, x_j)\} I\{d(y_i, y_k) \leq d(y_i, y_j)\}$$

$$A_{22} = \sum_{\substack{k=1,\\ k\neq i,\\ k\neq j}}^{n} I\{d(x_i, x_k) > d(x_i, x_j)\} I\{d(y_i, y_k) > d(y_i, y_j)\}$$

Donde $I\{\cdot\}$ es la función indicadora.

Caso	$\boldsymbol{d(y_i, y_k) \leq d(y_i, y_j)}$	$\boldsymbol{d(y_i, y_k) > d(y_i, y_j)}$	**Total fila**
$\mathbf{d(x_i, x_k) \leq d(x_i, x_j)}$	$A_{11}(i,j)$	$A_{12}(i,j)$	$\mathbf{A_1.(i,j)}$
$\mathbf{d(x_i, x_k) > d(x_i, x_j)}$	$A_{21}(i,j)$	$A_{22}(i,j)$	$\mathbf{A_2.(i,j)}$
Total columna	$\boldsymbol{A_{.1}(i,j)}$	$\boldsymbol{A_{.2}(i,j)}$	**n-2**

Con la tabla de contingencia 2x2 obtenida (con $n-2$ observaciones para $k \in (1,\ldots,n)\ k \neq i\ y\ k \neq j$) se calcula el estadístico Chi-cuadrado, tal que así:

$$S(i,j) = \frac{(n-2)\{A_{12}(i,j)A_{21}(i,j) - A_{11}(i,j)A_{22}(i,j)\}^2}{A_{1\cdot}(i,j)A_{2\cdot}(i,j)A_{\cdot 1}(i,j)A_{\cdot 2}(i,j)}$$

donde $A_{1\cdot} = A_{11} + A_{12}$, $A_{2\cdot} = A_{21} + A_{22}$, $A_{\cdot 1} = A_{11} + A_{21}$ y $A_{\cdot 2} = A_{12} + A_{22}$. Para probar la independencia entre los vectores aleatorios X e Y, los autores proponen utilizar la prueba estadística:

$$T = \sum_{i=1}^{n} \sum_{\substack{j=1,\\ j\neq i}}^{n} S(i,j)$$

A comparar con el valor de la Chi-cuadrado $n(n-1)$, donde las hipótesis son [26]:

- H_0: Los vectores son independientes.
- H_1: Los vectores son dependientes, es decir, están relacionados.

Para ello, se realiza una prueba de permutación, en la que se permutan los vectores de Y de las observaciones y reasignando los pares: para cada permutación $(y_{\pi(1)}, \ldots, y_{\pi(n)})$, se computa el test estadístico para los pares reasignados $(x, y_{\pi(1)}), \ldots, (x, y_{\pi(n)})$. Si el test estadístico T de la muestra original es mayor o igual que el cuantil $(1-\alpha)$ de la población de estadísticos permutados, se puede rechazar la H_0 con un nivel de significación α. El p-valor es la fracción de reasignaciones para las cuales los test calculados son al menos tan grandes como la observada, donde la fracción se calcula a partir de las asignaciones $(B+1)$ que incluyen B muestras permutadas y la muestra observada [5, 24, 27].

A partir de aquí se podría continuar como en una prueba de independencia en tablas de contingencia. Si se rechaza la hipótesis de independencia, se calcula el coeficiente de Cramér

$$C = \sqrt{\frac{\chi^2/n}{q}},$$

donde $q = min(n, n-1) - 1 = n - 2$ [26].

La única restricción en las medidas de distancia $d_X(\cdot,\cdot)$ y $d_Y(\cdot,\cdot)$ es que están determinadas por normas.

Además, la prueba debe cumplir con las condiciones para calcular la Chi-cuadrado: una muestra superior a 30 y que las frecuencias esperadas sean mayores a 5. Si no las flexibilizadas que las frecuencias esperadas ninguna sea inferior a 1 y que el 80% sean mayores a 5.

Tabla 12: Ventajas e inconvenientes de HHG

Ventajas [5, 24, 27]	**Desventajas [25, 26]**
• Detecta una amplia gama de asociaciones entre variables (lineal, no lineal monótona / no monótona y no funcional), independientemente de la muestra	• El objetivo de HHG no es definir un coeficiente de asociación sino probar la asociación.
• Se puede utilizar en escenarios multivariantes.	• El coste computacional es alto O(n^2 log n), aunque con el algoritmo realizado menor al planteamiento original O(n^3).
• Es una prueba poderosa que tiene una forma sencilla, fácil de implementar y potente.	

Principales propiedades

- Se puede aplicar a todas las dimensiones.
- Las dimensiones de los vectores pueden ser mayores que el número de observaciones.
- La prueba es consistente, ya que está basada en el test Chi-cuadrado o en la razón de verosimilitud (likelihood).
- Es consistente para múltiples tipos de variables, ya que se puede contrastar con vectores aleatorios:
  - discretos con soporte contable,
  - continuos,
  - donde las coordenadas son discretas y otras continuas, si la densidad del vector aleatorio continuo es continua al rededor del punto de dependencia.

Función en R

En la página[6] de Ruth Heller, así como en el repositorio CRAN, se puede encontrar el paquete `HHG` en el que está implementado el test de independencia Heller-Heller-Gorfine (2013) y las pruebas de independencia e igualdad de distribución entre dos variables aleatorias univariadas introducidas en Heller et al. (2016).

[6] Documentación del paquete: http://www.math.tau.ac.il/~ruheller/Software.html

La función que calcula la prueba HHG estudiada es `hhg.test` cuya estructura es:

```
hhg.test(Dx, Dy, ties=T, w.sum=0, w.max=2, nr.perm=10000,
          is.sequential=F,          seq.total.nr.tests=1,
          seq.alpha.hyp=NULL,            seq.alpha0=NULL,
          seq.beta0=NULL,   seq.eps=NULL,   nr.threads=0,
          tables.wanted=F, perm.stats.wanted=F)
```

Donde los argumentos a indicar son:

Argumento	**Definición**
Dx	Es una matriz simétrica de doubles, donde el elemento $[i, j]$ es una distancia basada en normas entre las observaciones i y j de x.
Dy	Lo mismo que Dx pero con las distancias de y.
ties	Una variable booleana que indica si existen vínculos entre Dx y/o Dy y deben manipularse apropiadamente (requiere más cálculos)
w.sum	Frecuencia mínima esperada que se tiene en cuenta para calcular el estadístico `sum.chisq` (no negativa, la contribución de las tablas que tienen celdas con valores muy pequeños se truncará a cero)
w.max	Frecuencia mínima esperada que se tiene en cuenta para calcular el estadístico `max.chisq` (no negativa, la contribución de las tablas que tienen celdas con valores muy pequeños se truncará a cero)
nr.perm	Número de permutaciones a partir de las cuales se debe estimar el p-valor (debe de ser no negativo). Para más detalles ver la información de la función
is.sequential	Indicador booleano que indica si se quiere hacer la prueba secuencial de Wald, si no, se realiza el cálculo simple de Monte-Carlo de las permutaciones `nr.perm`. Cuando es TRUE, el usuario debe especificar `seq.total.nr.tests` o `seq.alpha.hyp`, `seq.alpha0`, `seq.beta0`, `seq.eps`)
seq.total.nr.tests	(opcional) Número total de hipótesis en la familia de hipótesis probadas simultáneamente. Cuando se proporciona, se usa para derivar valores predeterminados para los parámetros de la prueba secuencial de Wald. La derivación predeterminada se realiza asumiendo un nivel nominal de 0.05 FDR, y establece: `seq.alpha.hyp` = 0.05 / max (1, log (`seq.total.nr.tests`)), `seq.alpha0` = 0.05, `seq.beta0` = min (0.01, 0.05 / `seq.total.nr.tests`), `seq .eps` = 0.01. Alternativamente, uno puede especificar sus propios valores para estos parámetros usando los siguientes argumentos
seq.alpha.hyp	El tamaño de prueba nominal para este test único dentro del procedimiento de prueba múltiple
seq.alpha0	El tamaño de prueba nominal para probar la hipótesis nula lateral de p-valor> `seq.alpha.hyp`

seq.beta0	Uno menos el poder para probar la hipótesis nula lateral de p-valor> `seq.alpha.hyp`
seq.eps	Margen de aproximación alrededor de `seq.alpha.hyp` que define las regiones p-valor para el lado nulo p> `seq.alpha.hyp` * (1 + `seq.eps`) y la alternativa lateral p <`seq.alpha.hyp` * (1 - `seq. eps`)
nr.threads	Número de núcleos de procesamiento a usar para la permutación de p-valor. Si se deja como cero, intentará usar todos los núcleos disponibles
tables.wanted	Indicador booleano que determina si se generan tablas de contingencia locales detalladas 2x2
perm.stats.wanted	Indicador booleano que determina si se emiten valores estadísticos calculados para todas las permutaciones (que representan distribuciones nulas)
w.max	Frecuencia mínima esperada que se tiene en cuenta para calcular el estadístico `max.chisq` (no negativa, la contribución de las tablas que tienen celdas con valores muy pequeños se truncará a cero)

La función devuelve cuatro estadísticos especificados en el documento original de HHG [27]:

- `sum.chisq`: Suma de los estadísticos Chi-cuadrado de Pearson de las tablas de contingencia 2x2 consideradas.
- `sum.lr`: Suma de los valores de la relación de verosimilitud ("Estadístico G") de las tablas 2x2.
- `max.chisq`: Máximo estadístico Chi-cuadrado de Pearson de cualquiera de las tablas 2x2.
- `max.lr`: Máximo estadístico G de cualquiera de las tablas 2x2.

Resultados simulados

A continuación, se indican los valores obtenidos para cada una de las simulaciones, en la que todas se rechaza la hipótesis nula de independencia a excepción de los resultados del conjunto de datos pseudo-aleatorios. La prueba al ser multivariable, es posible pasarle las 500 simulaciones a la vez y dar como resultado un único estadístico.

Distribución	**Estadístico**	**Valor**	**p-valor**
Pseudo-aleatorio	sum.chisq	232046.091	0.982
	sum.lr	118573.962	0.988
	max.chisq	15.88682	1
	max.lr	9.665826	0.752
Lineal	sum.chisq	30367614.540	0.000999
	sum.lr	14310953.616	0.000999
	max.chisq	273.2264	0.000999
	max.lr	135.6648	0.000999
Cuadrática	sum.chisq	4691200.987	0.000999
	sum.lr	2318351.174	0.000999
	max.chisq	90.74689	0.000999
	max.lr	46.77841	0.000999

Cúbica	sum.chisq	21241928.803	0.000999
	sum.lr	10162492.607	0.000999
	max.chisq	227.5142	0.000999
	max.lr	114.5269	0.000999
Exponencial	sum.chisq	3578108.677	0.000999
	sum.lr	1779874.660	0.000999
	max.chisq	80.87216	0.000999
	max.lr	33.26879	0.000999
Sinusoidal	sum.chisq	478202.877	0.000999
	sum.lr	241647.626	0.000999
	max.chisq	26.21158	0.16
	max.lr	12.86142	0.05
Escalón	sum.chisq	3747893.769	0.000999
	sum.lr	1854502.578	0.000999
	max.chisq	79.25219	0.000999
	max.lr	36.39601	0.000999
Círculo	sum.chisq	652394.146	0.000999
	sum.lr	328319.189	0.000999
	max.chisq	35.79214	0.00699
	max.lr	15.49309	0.003

7.2. Código de la función MIC

```
MIC <- function(x,y,R=100,...) {
  #MIC
  mic<-mine(x=x,y=y,...)

  #For calculate p-value de MIC
  if (! is.null(R)) {
    R <- floor(R)
    if (R < 1) R <- 100
  } else {
    R <- 100
  }

  Rep<-as.data.frame(rep(as.data.frame(y),R))
  Rep2<-as.data.frame(apply(Rep,2,sample))
  permic<-matrix(NA,nrow=R,ncol=7)
  colnames(permic)<-c("MIC", "MAS", "MEV", "MCN", "MIC-R2",
      "GMIC", "TIC")
  for (i in 1:R){
    p<-mine(x=x,y=Rep2[,i],...)
    permic[i,1:7]<-c(p$MIC, p$MAS, p$MEV, p$MCN, p$`MIC-R2`,
      p$GMIC, p$TIC)
  }

  permic<-as.data.frame(permic)
  pvalor<-nrow(permic[which(permic$MIC>=mic$MIC),])/nrow(permic)

  mic2<-as.data.frame(cbind(mic$MIC,pvalor))
  colnames(mic2)<-c("MIC","p-value")
```

```
  return(mic2)
}
```

7.3. Código de la función rdc

```
rdc <- function(x,y,k=20,s=1/6,f=sin) {
  x <- cbind(apply(as.matrix(x), 2,
            function(u) rank(u)/length(u)), 1)
  y <- cbind(apply(as.matrix(y), 2,
            function(u)rank(u)/length(u)),1)
  x<- s/ncol(x)*x%*%matrix(rnorm(ncol(x)*k),ncol(x))
  y <- s/ncol(y)*y%*%matrix(rnorm(ncol(y)*k),ncol(y))
  can<-cancor(cbind(f(x),1),cbind(f(y),1))$cor

  k<-length(can)
  chi<-(((2*k+3)/2)-nrow(x))*log(prod(1-can^2))
  rdc<-as.data.frame(cbind(can[1], pchisq(chi, k^2,
      lower.tail = FALSE)))
  colnames(rdc)<-c("rdc","p-value")
  return(rdc)
}
```

7.4. Código de la función AlterCorr

```
AlterCorr<-function(x, y, type=c("pearson","MIC","RDC","dCor"),
R=100){
  type <- match.arg(type)
  if (! is.null(R)) {
    R <- floor(R)
    if (R < 1) R <- 100
  } else {
    R <- 100
  }
  if (type=="pearson") {
    corrP <- rcorr(x,y, type="pearson")
    result<-as.data.frame(cbind(corrP$r[1,2], corrP$P[1,2]))
  }
  if (type=="MIC") {
    result <- MIC(x, y, R=R)
  }
  if (type=="RDC") {
    result <- rdc(x, y, k=20, s=1/6, f=sin)
  }
  if (type=="dCor") {
    dcor <- dcor.test(x, y, index=1.0, R=R)
    result<-as.data.frame(cbind(dcor$statistic, dcor$p.value))
  }
  colnames(result)<-c("Correlation","pvalue")
  return(result)
}
```

7.5. Código de la función AlterCorrM

```
AlterCorrM<-function(X,Y,type=c("pearson","MIC","RDC","dCor"),
                     comparison=c("all","pairs"),R=100 ,
                     method =c("holm", "hochberg", "hommel",
                               "bonferroni", "BH", "BY", "fdr",
```

```
                                  "none")){

  type <- match.arg(type)
  comp <- match.arg(comparison)
  method <- match.arg(method)

  if (! is.null(R)) {
    R <- floor(R)
    if (R < 1) R <- 100
  } else {
    R <- 100
  }

  if ((nrow(X)!=nrow(Y)) && type=="all") stop('The number of obs
ervations does not match')
  if ((nrow(X)!=nrow(Y) ||(ncol(X)!=ncol(Y))) && type=="pairs")
{
    stop('Matrices dimensions do not match, select comparation="
alls"')
  }

  if (comp=="all"){
    Corrs<-matrix(NA,nrow=ncol(X),ncol=ncol(Y))
    Pvalue<-matrix(NA,nrow=ncol(X),ncol=ncol(Y))

    t<-1
    k<-matrix(NA,nrow=ncol(X)*ncol(Y),ncol=3)
    for (i in 1:ncol(X)){
      for (j in 1:ncol(Y)){
        z<-AlterCorr(X[,i], Y[,j], type=type,R=R)
        Corrs[i,j] <- z$Correlation
        Pvalue[i,j] <- z$pvalue
        k[t,]<-cbind(i,j,z$pvalue)
        t<-t+1
      }
    }
    k<-as.data.frame(k)
    colnames(k)<-c("X","Y","pvalue")
    adj<- p.adjust(k$pvalue,method=method)
    adjPval <- matrix(adj,nrow=ncol(X),ncol=ncol(Y),byrow = TRUE
)
    colnames(Corrs)<-colnames(Y)
    colnames(Pvalue)<-colnames(Y)
    colnames(adjPval)<-colnames(Y)
    rownames(Corrs)<-colnames(X)
    rownames(Pvalue)<-colnames(X)
    rownames(adjPval)<-colnames(X)
  }

  if (comp=="pairs"){
    Corrs<-matrix(NA,nrow=ncol(X),ncol=1)
    Pvalue<-matrix(NA,nrow=ncol(X),ncol=1)
    for (i in 1:ncol(X)){
      z<-AlterCorr(X[,i], Y[,i], type=type,R=R)
```

```
      Corrs[i,1] <- z$Correlation
      Pvalue[i,1] <- z$pvalue

    }
    adjPval <- matrix(p.adjust(Pvalue, method=method),
                      nrow=ncol(X), ncol=1)
    rownames(Corrs)<-colnames(X)
    rownames(Pvalue)<-colnames(X)
    rownames(adjPval)<-colnames(X)
  }

  res<-(list(Correlation=Corrs, pvalue=Pvalue, adjPval=adjPval))
  return(res)
}
```

7.6. Código de la aplicación Shiny AlterCorrApp

```
library(shiny)
library(markdown)
library(AlterCorr)
library(ggplot2)
library(d3heatmap)
# Definición del UI para la aplicación
ui <- shinyUI(navbarPage("AlterCorr", #id = "tabs",
                tabPanel("Información",
                         h1("Información de la aplicación"),
                         br(),
                         p("Con esta aplicación se puede subir dos matrices de datos
                             y calcular cuatro coeficientes de correlación diferentes:"),
                         br(),
                         p("- La Correlación de Pearson (r)"),
                         p("- El Coeficiente Máximo de Información (MIC)"),
                         p("- El Coeficiente de dependencia aleatorio (RDC)"),
                         p("- La Correlación de distancias (dCor)"),
                         br(),
                         p("En las pestañas 'Variables independientes' y
                           'Variables dependientes' se suben los archivos csv con los
                           datos necesarios para calcular las correlaciones."),
                         br(),
                         p("Y en la pestaña de 'Cálculos' una vez seleccionado el coeficiente
                           con el que se quiere calcular la correlación y el
                           tipo de comparación (pareada o todas las
                           variables), se presentan la correlación, el p-valor
                           (sin ajustar y ajustado) y si procede un gráfico
                           heatmap.")
                         ),
                tabPanel("Variables independientes",
                         fluidPage(
                           #Barra Lateral
                           sidebarLayout(
                             #Panel de la barra lateral con los inputs
                             sidebarPanel(
                               #Input: Selección del primer archivo
                               fileInput("archivo1",
                                         "Seleccione el archivo csv",
                                         multiple=FALSE,
                                         accept=c("text/csv",
                                                  "text/comma-separated-values,
```

```
                                        text/plain",
                                        ".csv")
                            ),

                            #Input: Checkbox si el archivo contiene encabezado
                            checkboxInput("header1", "Encabezado", TRUE),

                            #Input: Checkbox si el archivo contiene en la
                            #primera columna el nombre de las filas
                            checkboxInput("obs1", "Nombre observaciones", TRUE),

                            # Texto explicatorio
                            helpText("Nota: Encabezado indica, si el archivo contiene
                                     en la primera línea el nombre de las variables.
                                     Y en nombre observaciones si en la primera columna está
el nombre de las observaciones"),

                            #Input: Selección del separador
                            radioButtons("sep1", "Separador",
                                         choices=c(Coma=",",
                                                   "Punto y coma"=";",
                                                   Tabulador="\t"),
                                         selected=","),

                            #Input: Selección del carácter utilizado para los decimales
                            radioButtons("dec1", "Decimales",
                                         choices=c(Coma=",",Punto="."),
                                         selected=","),

                            #Input: Tratamiento de los missings
                            textInput("na1",
                                      "Identificación de los missings:",
                                      ""),

                            #Input: Hasta que no se clica no se carga el archivo
                            actionButton("Carga1","Cargar archivo"),

                            hr(),

                            #Input: Número de filas a mostrar
                            radioButtons("dis1", "Filas a mostrar",
                                         choices=c(Inicio="head", Final="tail", Todo="all"),
                                         selected="head")
                            ),

                          mainPanel(
                            # Output: Explicación y texto, según si se ha cargado
                            # o no los datos
                            uiOutput("Textx"),
                            # Output:
                            tableOutput("Vistax")
                          ))),
                tabPanel("Variables dependientes",
                         fluidPage(
                           #Barra lateral
                           sidebarLayout(
                             #Panel de la barra lateral con los inputs
                             sidebarPanel(
```

```
                              #Input: Selección del primer archivo
                              fileInput("archivo2",
                                        "Seleccione el archivo csv",
                                        multiple=FALSE,
                                        accept=c("text/csv",
                                                 "text/comma-separated-values,
                                                 text/plain",
                                                 ".csv")
                              ),

                              #Input: Checkbox si el archivo contiene encabezado
                              checkboxInput("header2", "Encabezado", TRUE),

                              #Input: Checkbox si el archivo contiene en la
                              #primera columna el nombre de las filas
                              checkboxInput("obs2", "Nombre observaciones", TRUE),

                              # Texto explicatorio
                              helpText("Nota: Encabezado indica, si el archivo contiene
                                       en la primera línea el nombre de las variables.
                                       Y en nombre observaciones si en la primera columna está
el nombre de las observaciones"),

                              #Input: Selección del separador
                              radioButtons("sep2", "Separador",
                                           choices=c(Coma=",",
                                                     "Punto y coma"=";",
                                                     Tabulador="\t"),
                                           selected=","),

                              #Input: Selección del carácter utilizado para los decimales
                              radioButtons("dec2", "Decimales",
                                           choices=c(Coma=",",Punto="."),
                                           selected=","),

                              #Input: Tratamiento de los missings
                              textInput("na2",
                                        "Identificación de los missings:",
                                        ""),

                              #Input: Hasta que no se clica no se carga el archivo
                              actionButton("Carga2","Cargar archivo"),

                              hr(),

                              #Input: Número de filas a mostrar
                              radioButtons("dis2", "Filas a mostrar",
                                           choices=c(Inicio="head", Final="tail", Todo="all"),
                                           selected="head")
                              ),

                            mainPanel(
                              # Output: Explicación y texto, según si se ha cargado
                              # o no los datos
                              uiOutput("Texty"),
                              # Output:
                              tableOutput("Vistay")
                            ))),
```

```
                tabPanel("Cálculos",
                         fluidPage(
                           #Barra lateral
                           sidebarLayout(
                             #Panel de la barra lateral con los inputs
                             sidebarPanel(
                               #Input: Coeficiente a calcular
                               selectInput("type","Selecciona un coeficiente:",
                                           c("Correlación de Pearson (r)"="pearson",
                                             "Coeficiente Máximo de Información (MIC)"="MIC",
                                             "Coeficiente de dependencia aleatorio
(RDC)"="RDC", "Correlación de distancias (dCor)"="dCor"),
                                           selected = "pearson"
                               ),
                               #Input: Comparación
                               radioButtons("comp", "Método de comparación:",
                                            choices=c(Parejas="pairs", Todos="all"),
                                            selected="all"),

                               #Texto explicatorio
                               helpText("Nota: Con la opción 'parejas' se compara dos matrices
que comparten filas y columnas, pero los valores son diferentes variables. Con 'Todos' se
compara matrices con una dimensión común y se calcula las correlaciones
de todas las variables con todas."),

                               #Input: Número de repeticiones
                               numericInput("R",
                                            "Número de repeticiones:",
                                            value=0, min=0),

                               #Texto explicatorio
                               helpText("Nota: Las repeticiones sólo es necesario para MIC y
dCor."),

                               #Input: Método corrector
                               selectInput("method","Selecciona el método corrector para los p-
valores:",
                                           c("Ninguno"="none",
                                             "Fdr"="fdr",
                                             "Bonferroni"="bonferroni",
                                             "Holm"="holm",
                                             "Hochberg"="hochberg",
                                             "Hommel"="hommel",
                                             "BH"="BH",
                                             "BY"="BY"),
                                           selected = "none"
                               ),

                               hr(),

                               #Input: Hasta que no se clica no se realiza el cálculo
                               actionButton("Calcular","Calcular")
                             ),
                             mainPanel(
                               fluidPage(
                                 tabsetPanel(
                                   tabPanel("Explicación", uiOutput("Textcoef")),
                                   tabPanel("Resultados",
```

```
                                        titlePanel("Correlación y p-valores calculados"),
                                        br(),
                                        #Output: Tabla que muestra toda la información
                                        #de las correlaciones para la comparación por pares
                                        #o cuando sólo tiene una columna,
                                        #cuando es la opción "all" .
                                        DT::dataTableOutput("TablaDT1")),
                                tabPanel("Correlación",
                                        titlePanel("Matriz de correlaciones"),
                                        br(),
                                        #Output: Tabla que las correlaciones
                                        #cuando es la opción "all" y tiene más de
                                        #una variable independiente.
                                        DT::dataTableOutput("TablaDT2")),
                                tabPanel("P-valores",
                                        titlePanel("Matriz de p-valores"),
                                        br(),
                                        #Output: Tabla con los p-valores para la opción
"all"
                                        #con más de una variable independiente.
                                        DT::dataTableOutput("TablaDT3")),
                                tabPanel("P-valores ajustados",
                                        titlePanel("Matriz de p-valores ajustados"),
                                        br(),
                                        #Output: Tabla con los p-valores para la opción
"all"
                                        #con más de una variable independiente.
                                        DT::dataTableOutput("TablaDT4")),
                                tabPanel("Gráfico",
                                        titlePanel("Gráfico heatmap de la matriz de
correlaciones"),
                                        br(),
                                        d3heatmapOutput("heatmap", width="100%",
height="600px")
                                        )
                                , id="tabs")
                            )
                        )
                    )
                )

        )
)

# Define la lógica del server requerido para la aplicación
server <- function(input, output, session) {
  #Carga del archivo con las variables independientes
  X<-eventReactive(input$Carga1,
                   {
                     #Inicialmente está en nulo
                     req(input$archivo1)

                     #Lee el archivo csv, si hay un error en la
                     #carga lo gestiona
                     tryCatch(
                       {
                         if(input$obs1==TRUE){
```

```
                        df<-read.csv(input$archivo1$datapath,
                                     header=input$header1,
                                     sep=input$sep1,
                                     dec=input$dec1,
                                     row.names = 1,
                                     na.strings = input$na1)
                      } else {
                        df<-read.csv(input$archivo1$datapath,
                                     header=input$header1,
                                     sep=input$sep1,
                                     dec=input$dec1,
                                     na.strings = input$na1)
                      }

                    },
                    error =function(e){
                      #Devuelve un safeError si se produce un error
                      stop(safeError(e))
                    }
                  )
                },
                ignoreNULL = FALSE)

#Muestra el archivo cargado
output$Vistax<-renderTable({
  dataset <- X()
  if(input$dis1 =="head"){
    return(head(dataset))
  } else if (input$dis1=="tail"){
    return(tail(dataset))
  } else {
    return(dataset)
  }
})

#Texto del main de las variables indepedientes
output$Textx = renderUI({
  if (input$Carga1==0) {
    #Sin cargar los datos
    text<-list(
      tags$h2("Carga del archivo con las variables independientes"),
      tags$div("El archivo csv debe contener en las columnas la/s variable/s
               a analizar y en las filas, dependiendo del tipo de datos;
               observaciones, muestras o individuos. Además del nombre de las
               columnas y de las filas.")
      )
  } else{
    #Con los datos cargados
    text<-list(tags$h2("Vista de los datos cargados:"))

  }
  return(tagList(text))
})



#Carga del archivo con las variables dependientes
Y<-eventReactive(input$Carga2,
```

```
                {
                  #Inicialmente está en nulo
                  req(input$archivo2)

                  #Lee el archivo csv, si hay un error en la
                  #carga lo gestiona
                  tryCatch(
                    {
                      if(input$obs2==TRUE){
                        df<-read.csv(input$archivo2$datapath,
                                     header=input$header2,
                                     sep=input$sep2,
                                     dec=input$dec2,
                                     row.names = 1,
                                     na.strings = input$na2)
                      } else {
                        df<-read.csv(input$archivo2$datapath,
                                     header=input$header2,
                                     sep=input$sep2,
                                     dec=input$dec2,
                                     na.strings = input$na2)
                      }
                    },
                    error =function(e){
                      #Devuelve un safeError si se produce un error
                      stop(safeError(e))
                    }
                  )
                },
                ignoreNULL = FALSE)

#Muestra el archivo cargado
output$Vistay<-renderTable({
  dataset <- Y()
  if(input$dis2 =="head"){
    return(head(dataset))
  } else if (input$dis2=="tail"){
    return(tail(dataset))
  } else {
    return(dataset)
  }
})

#Texto del main con la información del coeficiente
output$Texty = renderUI({
  if (input$Carga2==0) {
    #Sin cargar los datos
    text<-list(
      tags$h2("Carga del archivo con las variables dependientes"),
      tags$div("El archivo csv debe contener en las columnas la/s variable/s
               a analizar y en las filas, dependiendo del
               tipo de datos; observaciones, muestras o individuos.
               Además del nombre de las columnas y de las filas. El número de
               columnas tiene que coincidir con el archivo subido en variables
               dependientes.")
      )
  } else{
    #Con los datos cargados
```

```
    text<-list(tags$h2("Vista de los datos cargados:"))

  }
  return(tagList(text))
})

#Texto del main de las variables dependientes
output$Textcoef = renderUI({
  archivo=paste(input$type,".html",sep="")
  tags$iframe(src=archivo, seamless=NA, scrolling="auto",
              width="100%", height="500px")
})

#Calcula las correlaciones
datos<-eventReactive(input$Calcular,
                     {
                       #Inicialmente está en nulo
                       req(X(), Y())

                       #Lee el archivo csv, si hay un error en la
                       #carga lo gestiona
                       tryCatch(
                         {
                           Z<-AlterCorrM(X(), Y(),
                                         type=input$type,
                                         comparison=input$comp,
                                         R=input$R,
                                         method=input$method)
                         },
                         error =function(e){
                           #Devuelve un safeError si se produce un error
                           stop(safeError(e))
                         }
                       )
                     },
                     ignoreNULL = FALSE)
#Añade la tabla con las correlaciones dependiendo del número de
#variables y tipo de comparación
output$TablaDT1<-DT::renderDataTable(
  if(input$Calcular!=0){
    if(input$method=="none"){
      if (input$comp=="pairs"){

        Tabla<-as.data.frame(cbind(datos()$Correlation,datos()$pvalue))
        colnames(Tabla)<-c("Correlación", "pvalor")
        DT::datatable(Tabla,
                      filter = 'top',
                      extensions = 'Buttons',
                      options=list(
                        order = list(list(2, 'asc')),
                        dom = 'Bfrtip',
                        buttons = c('copy', 'csv', 'excel', 'pdf', 'print')
                      ))%>%
          formatRound(1:2,5)
      }else if (input$comp=="all" && ncol(X())==1){

        Tabla<-as.data.frame(cbind(t(datos()$Correlation),
                                   t(datos()$pvalue)))
```

```
        colnames(Tabla)<-c("Correlación", "pvalor")
        DT::datatable(Tabla,
                      filter = 'top',
                      extensions = 'Buttons',
                      options=list(
                        order = list(list(2, 'asc')),
                        dom = 'Bfrtip',
                        buttons = c('copy', 'csv', 'excel', 'pdf', 'print')
                        ))%>%
          formatRound(1:2,5)
      } else {
        return()
      }

    }else{
      if (input$comp=="pairs"){

        Tabla<-as.data.frame(cbind(datos()$Correlation,datos()$pvalue,datos()$adjPval))
        colnames(Tabla)<-c("Correlación", "pvalor", "pvalorAdj")
        DT::datatable(Tabla,
                      filter = 'top',
                      extensions = 'Buttons',
                      options=list(
                        order = list(list(2, 'asc'), list(3, 'asc')),
                        dom = 'Bfrtip',
                        buttons = c('copy', 'csv', 'excel', 'pdf', 'print')
                      ))%>%
          formatRound(1:3,5)
      }else if (input$comp=="all" && ncol(X())==1){

        Tabla<-as.data.frame(cbind(t(datos()$Correlation),
                                   t(datos()$pvalue),
                                   t(datos()$adjPval)))
        colnames(Tabla)<-c("Correlación", "pvalor", "pvalorAdj")
        DT::datatable(Tabla,
                      filter = 'top',
                      extensions = 'Buttons',
                      options=list(
                        order = list(list(2, 'asc'), list(3, 'asc')),
                        dom = 'Bfrtip',
                        buttons = c('copy', 'csv', 'excel', 'pdf', 'print')
                      ))%>%
          formatRound(1:3,5)
      } else {
        return()
      }
    }
  } else {
    return()
  }
)
#Añade la tabla con las correlaciones para comparaciones "all"
output$TablaDT2<-DT::renderDataTable(
  if(input$Calcular!=0){
    if (input$comp=="all" && ncol(X())>1){
      Tabla<-as.data.frame(datos()$Correlation)
      DT::datatable(Tabla,
                    filter = 'top',
```

```
                    extensions = 'Buttons',
                    options=list(
                      dom = 'Bfrtip',
                      buttons = c('copy', 'csv', 'excel', 'pdf', 'print'))
                    )%>%
        formatRound(1:ncol(Tabla),5)
    } else {
      return()
    }
  } else {
    return()
  }
)

#Añade la tabla con las p-valores para el tipo de comparación "all"
#con más de una variable independiente
output$TablaDT3<-DT::renderDataTable(
  if(input$Calcular!=0){
    if (input$comp=="all" && ncol(X())>1){
      Tabla<-as.data.frame(datos()$pvalue)
      DT::datatable(Tabla,
                    filter = 'top',
                    extensions = 'Buttons',
                    options=list(
                      dom = 'Bfrtip',
                      buttons = c('copy', 'csv', 'excel', 'pdf', 'print'))
                    )%>%
        formatRound(1:ncol(Tabla),5)

    } else {
      return()
    }
  } else {
    return()
  }
)

#Añade la tabla con las p-valores ajustados para el tipo de comparación "all"
#con más de una variable independiente
output$TablaDT4<-DT::renderDataTable(
  if(input$Calcular!=0){
    if (input$comp=="all" && ncol(X())>1){
      Tabla<-as.data.frame(datos()$adjPval)
      DT::datatable(Tabla,
                    filter = 'top',
                    extensions = 'Buttons',
                    options=list(
                      dom = 'Bfrtip',
                      buttons = c('copy', 'csv', 'excel', 'pdf', 'print'))
                    )%>%
        formatRound(1:ncol(Tabla),5)

    } else {
      return()
    }
  } else {
    return()
  }
```

```
  )

  #Activa y desactiva tabset dependiendo de los resultados a obtener
  showTab(inputId="tabs", target="Resultados")
  hideTab(inputId="tabs", target="Correlación")
  hideTab(inputId="tabs", target="P-valores")
  hideTab(inputId="tabs", target="P-valores ajustados")
  hideTab(inputId="tabs", target="Gráfico")

  observeEvent(input$Calcular,{
      #Requiere este elemento
      req(X())
      if (input$comp=="all" && ncol(X())>1){
        showTab(inputId="tabs", target="Correlación", select=TRUE)
        showTab(inputId="tabs", target="P-valores")
        if(input$method=="none"){
          hideTab(inputId="tabs", target="P-valores ajustados")
        }else{
          showTab(inputId="tabs", target="P-valores ajustados")
        }
        showTab(inputId="tabs", target="Gráfico")
        hideTab(inputId="tabs", target="Resultados")
      } else {
        showTab(inputId="tabs", target="Resultados", select=TRUE)
        hideTab(inputId="tabs", target="Correlación")
        hideTab(inputId="tabs", target="P-valores")
        hideTab(inputId="tabs", target="P-valores ajustados")
        hideTab(inputId="tabs", target="Gráfico")
      }
  }
  )

  #Gráfico heatmap

  output$heatmap <- renderD3heatmap({
    if (input$comp=="all" && ncol(X())>1){
      d3heatmap(datos()$Correlation)
    }else{
      return()
    }
  })
}


# Run the application
shinyApp(ui = ui, server = server)
```