

Mariano Córdoba

Pablo Paccioretti

Franca Giannini Kurina

Cecilia Bruno

Mónica Balzarini

Guía para el análisis de datos espaciales. Aplicaciones en agricultura

DISEMINACIÓN CIENTÍFICA Y TRASNFERENCIA DE
RESULTADOS DE INVESTIGACIÓN, PROMOVIDAS POR
EL MINISTERIO DE CIENCIA Y TECNOLOGÍA DE LA
PROVINCIA DE CÓRDOBA

La cita bibliográfica para el presente documento

Córdoba M, Paccioretti P, Giannini Kurina F, Bruno C, Balzarini M. 2019. Guía para el análisis de datos espaciales aplicaciones en agricultura. Serie Estadística Aplicada. Com. Balzarini M. Brujas. Córdoba, Argentina.

Guía para el análisis de datos espaciales: aplicaciones en agricultura. / Mariano Córdoba ... [et.al.]; dirigido por Mónica Balzarini.- 1a ed. – Córdoba: Brujas, 2019. 250p.; 23 x 15 cm.

ISBN 978-987-760-272-2

1. Estadísticas. 2. Agronomía. 3. Agricultura. I. Córdoba, Mariano II. Balzarini, Mónica, dir.

CDD 630

©Córdoba Mariano; Paccioretti Pablo; Giannini Kurina Franca;
Bruno Cecilia; Balzarini Mónica.

1º Edición

Primera Impresión

Impreso en Argentina

ISBN: 978-987-760-272-2

Queda hecho el depósito que prevé la ley 11.723

Queda prohibida la reproducción total o parcial de este libro en forma idéntica o modificada por cualquier medio mecánico o electrónico incluyendo fotocopia, grabación o cualquier sistema de almacenamiento y recuperación de información no autorizada por los autores.

A los generadores de conocimiento



Índice general

Prólogo	IX
I Aproximaciones metodológicas	1
1 Manejo de datos espaciales	3
1.1 Transformación y conversión de coordenadas	5
1.2 Manipulación de múltiples capas de datos	6
1.3 Depuración de datos	8
2 Caracterización de variabilidad espacial	13
2.1 Semivariogramas	16
2.1.1 Ajuste de semivariogramas	22
2.2 Correlación espacial bivariada	25
2.2.1 Coeficiente de correlación	25
2.2.2 Coeficiente de co-dispersión	27
2.3 Interpolación Kriging	29
2.3.1 Kriging ordinario	30
2.3.2 Kriging en bloques	32
2.3.3 Kriging local	33
2.3.4 Kriging universal	34
2.3.5 Validación cruzada	35
3 Caracterización de variabilidad espacial con múltiples capas de datos	39
3.1 Análisis de componentes principales	39
3.2 Análisis de conglomerados	43
4 Predicción con múltiples capas de datos	47

4.1	Regresión con errores correlacionados espacialmente vía REML	51
4.2	Regresión con efectos aleatorios de sitio vía INLA	53
4.3	Regresión vía modelos basados en árbol . .	56
4.3.1	Bosques aleatorios	58
4.3.2	Árboles de regresión generalizados . .	59
II	Análisis de datos a escala fina	61
1	Implementación con R	63
1.1	Conversión de coordenadas espaciales	64
1.2	Eliminación de <i>outliers</i> e <i>inliers</i>	69
1.3	Detección de tendencias espaciales	83
1.4	Cálculo del índice de Moran	87
1.5	Análisis basado en semivariogramas	88
1.5.1	Mapeo de la variabilidad espacial	97
1.5.2	Validación cruzada	107
1.6	Caracterización de variabilidad espacial con múltiples capas de datos	109
1.6.1	Análisis de componentes principales	109
1.6.2	Análisis de conglomerados	114
1.7	Predicción con múltiples capas de datos . .	121
1.7.1	Kriging con deriva externa	123
1.7.2	Kriging desde modelo de regresión .	125
1.7.3	Árboles aleatorios	128
2	Implementación con InfoStat	133
2.1	Conversión de coordenadas espaciales	133
2.2	Eliminación de <i>outliers</i> e <i>inliers</i>	136
2.3	Detección de tendencias espaciales	141
2.4	Cálculo del índice de Moran	143
2.5	Análisis basado en semivariogramas	145
2.5.1	Mapeo de variabilidad espacial	149
2.5.2	Validación cruzada	153

2.6	Caracterización de variabilidad espacial con múltiples capas de datos	156
2.6.1	Análisis de componentes principales	156
2.6.2	Análisis de conglomerados	158
2.7	Predicción con múltiples capas de datos	161
2.7.1	Kriging con deriva externa	162
2.7.2	Kriging desde modelo de regresión	168
2.7.3	Árboles aleatorios	172
III	Análisis de datos a escala regional	179
1	Bases de datos regionales	181
1.1	Manejo de datos espaciales	182
1.2	Confección de grillas de predicción	187
1.3	Agregado de capas de información	191
2	Predicción con múltiples capas de datos	199
2.1	Regresión con errores correlacionados espacialmente vía REML	200
2.2	Regresión con efectos aleatorios de sitio vía INLA	206
2.3	Regresión vía modelos basados en árbol	217
Anexo	225	
A	Herramientas de software	225
A.1	Introducción al manejo de datos espaciales con R	226
A.2	Intérprete de R en InfoStat	226
A.3	RStudio	227



Prólogo

En las últimas décadas se ha impulsado el desarrollo y la utilización de nuevas tecnologías que permiten capturar datos espaciales, *i.e.* datos de una variable regionalizada o asociados a una localización en el espacio.

La infraestructura de datos espaciales es cada vez mayor en tamaño y calidad, especialmente la asociada a la generación de datos que provienen de sensores ya sea remotos o proximales. Los volúmenes de datos espaciales no sólo son vastos y variados, sino que también, en la mayoría de los escenarios, son accesibles. Estos datos generan nuevas oportunidades para la investigación en agricultura.

La variabilidad en los procesos aleatorios que generan datos espaciales se modela con diversas herramientas de la Estadística Espacial y se representa gráficamente en mapas de variabilidad espacial donde puede observarse cómo cambian los valores de una o más variables aleatorias según su posición en el espacio.

Aún cuando se estudian dominios espaciales continuos con alta densidad de datos, usualmente no existen observaciones de la variable de interés para todos las localizaciones o sitios del espacio analizado; así se hace necesario obtener predicciones espaciales, *i.e.* predecir el valor de la variable en sitios sin datos. Con grillas de predicción densa, es posible obtener mapas de contorno casi continuos espacialmente.

Con varias variables para cada sitio, una de ellas interpreta-

da como resultante de un proceso y otras como explicativas o potenciales predictores, es posible obtener predicciones espaciales a partir de modelos que consideran la correlación espacial de los datos. Los modelos pueden estimarse tanto en un marco teórico frecuentista (Cressie y Wikle, 2015; Schabenberger y Gotway, 2005) como desde el marco teórico bayesiano (Correa Morales et al., 2018). También, desde la Ciencia de Datos con base computacional, se encuentran disponibles algoritmos de aprendizaje automático que incorporan la espacialidad en el análisis de datos (Li et al., 2011).

En esta guía se ilustra el manejo y procesamiento de datos espaciales con distintos métodos estadísticos y su aplicación en agricultura. El texto está organizado en tres partes; la primera contiene bases conceptuales para el análisis de datos georreferenciados provenientes de procesos espaciales continuos. La segunda, la implementación de protocolos de análisis completos sobre datos distribuidos a escala fina en el espacio, con códigos de programa listos para ejecutar en el software estadístico R (R. C. Team, 2019) y en el software *InfoStat* (Di Rienzo et al., 2019). La tercera parte del texto ilustra la implementación del manejo y análisis de datos distribuidos a escala regional con códigos en R. La versión digital de este libro puede obtenerse desde www.agro.unc.edu.ar/~estadisticaaplicada donde también se encuentran los códigos de programación y los datos usados en este texto.

Parte I

Aproximaciones metodológicas



1

Manejo de datos espaciales

Cuando los datos georreferenciados se representan en fotografías, imágenes o mapas es posible visualizar patrones espaciales, *i.e.* estructuraciones de los datos en el espacio. Los datos espaciales involucran, no solo la realización (valor) de una variable, sino también las coordenadas geográficas que posicionan el dato en el dominio espacial. Las coordenadas pueden ser uni, bi o tridimensionales y expresarse según distintos sistemas. Los tipos de datos espaciales usados son: datos geoestadísticos, datos regionales (látices) y patrones de puntos. Los primeros son datos de dominio continuo, es decir, supone que entre dos sitios pueden existir infinitos datos. Se refiere a continuidad en la estructura espacial del proceso aleatorio espacial subyacente a partir del cual se han generado las observaciones que se tienen. A causa de la continuidad del dominio espacial, los datos geoestadísticos también se llaman “datos espaciales con variación continua”. La continuidad se asocia entonces con el proceso aleatorio subyacente y no con el atributo medido (que la variable sea de naturaleza continua o discreta no determina si los datos son geoestadísticos o no). Los datos regionales o de áreas son aquellos donde el dominio es fijo y conformado por un conjunto discreto de áreas, superficies o polígonos. Los datos del tipo patrones de puntos son aquellos que provienen de un proceso puntual aleatorio conformado por los puntos o sitios donde ocurren los eventos.

En el trabajo con datos espaciales puede ser necesario rea-

lizar preprocesamientos previos al análisis estadístico. Es menester realizar la lectura de la o las capas de información (variables), proveer una descripción estadística de los datos y mapear o visualizar los datos en el espacio, eliminar valores atípicos. También puede requerirse la expresión del conjunto de variables en un mismo sistema de información geográfica o la necesidad de interpolar las variables de interés a una misma identidad espacial. Para el tratamiento de datos espaciales hay que considerar el formato de la información espacial (ráster o vectorial) y las particularidades del sistema de referencia.

La tecnología de sistematización para información georreferenciada más conocida es la de los Sistemas de Información Geográfica (SIG). Actualmente, éstos abarcan un complejo de sistemas de bases de datos, programas de escritorio, lenguajes de programación, dispositivos gráficos, aplicaciones web y servidores. En un SIG, los datos de cada variable pueden manejarse como capas de información, y diversas técnicas de análisis de datos pueden aplicarse simultáneamente en todas las capas de manera independiente o integrándolas en un único análisis multivariado. Las capas pueden estar correlacionadas entre sí, los datos dentro de cada capa pueden presentar estructura de correlación espacial. Los Sistemas de Información Geográfica (SIG) ofrecen funciones para crear, integrar, transformar, visualizar y analizar de manera exploratoria estas capas de datos espaciales. Los SIG más avanzados también disponen de funciones que generan interfase con softwares estadísticos ampliando así la capacidad para la modelación conjunta de varias capas de información.

1.1 Transformación y conversión de coordenadas

Para localizar el sitio (coordenadas) con el cual se asocia un dato espacial, se necesita un sistema de referencia. Existen dos tipos de coordenadas, cartesianas y geográficas. Las coordenadas cartesianas se miden desde el centro de la tierra, mientras que las geográficas desde una superficie de referencia o *datum*. Para Sudamérica el *datum* comúnmente utilizado es WGS84 (*World Geodetic System 84*). Éste es el *datum* estándar por defecto para coordenadas en los dispositivos GPS comerciales. Para combinar capas de información o para realizar otros procesamientos de datos espaciales es necesario conocer el *datum* y frecuentemente transformar o convertir las coordenadas. Transformar implica pasar de un sistema de referencia a otro (cambiar el *datum*), mientras que cuando se convierten coordenadas no se cambia de *datum*.

Por una cuestión de practicidad, es usual proyectar el sistema de coordenadas geográficas (expresados en grados, minutos y segundos) a un sistema de coordenadas cartesianas, como por ejemplo el sistema de proyección UTM (*Universal Transverse Mercator*). Esta operación permite que las distancias entre los sitios desde donde se leen los datos se expresen como distancias absolutas (metros) en vez de distancias relativas (grados). Por ello, un paso inicial en el análisis de datos espaciales es convertir las coordenadas geográficas en coordenadas cartesianas (UTM). La mayoría del software SIG tiene la capacidad para realizar la transformación o conversión de coordenadas.

1.2 Manipulación de múltiples capas de datos

Cuando se recolectan datos de más de una variable georreferencia (múltiples capas de datos especializados) es poco probable que se registre la misma ubicación para cada variable o tiempo de medición. Por ejemplo, rara vez las mediciones de propiedades del suelo y los índices derivados de imágenes satelitales de cultivos no son obtenidas exactamente para la misma localización y frecuentemente existen capas de datos en distintas escalas. Esta variabilidad en las coordenadas espaciales dificulta la fusión de datos para realizar análisis estadísticos multivariados, *i.e.* análisis que contemplen simultáneamente las distintas capas de datos.

Se necesita organizar los datos en una grilla común a todas las capas, de manera que cada celda de la grilla cuente con la información de su ubicación espacial y cada una de las variables medidas. Existen diversas alternativas metodológicas para crear este tipo de grillas. Una de ellas consiste en generar una grilla regular de una determinada dimensión la cual se interseca con cada una de las variables medidas. Luego los valores de cada capa son asignados al nodo de la celda más cercana al punto medido. Cuando se tiene más de un dato de una variable para el mismo nodo, se suele calcular una medida de posición como la media o mediana de los datos e inclusive en algunos casos puede ser de interés tomar una medida de variabilidad como el desvió estándar o coeficiente de variación de los datos que comparten la celda. Otra alternativa metodológica es generar la grilla regular y utilizar la información recolectada para realizar una interpolación espacial en sitios no medidos y así obtener una predicción espacial de la variable de interés en cada celda de la grilla. Este proceso se realiza para cada

una de las variables medidas empleando la misma grilla. Diversos métodos de interpolación pueden ser usados, uno frecuente es la interpolación kriging.

El espaciado de la grilla debe reflejar el nivel de detalle requerido y la capacidad de procesamiento computacional. Por ejemplo, en aplicaciones de agricultura de precisión (escala de lote) puede utilizarse una grilla de celdas cuadradas de $5\text{ m} \times 5\text{ m}$ que se aproxima a la mitad del ancho operativo básico de muchas maquinarias. Esto genera unos 400 puntos de grilla por hectárea. Con lotes grandes puede ser conveniente utilizar una cuadricula de $10\text{ m} \times 10\text{ m}$ para superar problemas computacionales y al mismo tiempo mantener una resolución de mapa adecuada para la visualización y análisis de los patrones espaciales.

La normalización de los datos es otra práctica comúnmente usada en el manejo de múltiples capas de datos. Con esta técnica se busca ajustar los valores de variables no commensurables, incluso medidos en diferentes escalas a una escala común. La normalización puede realizarse en base a la media de cada capa o variable y expresar la unidad como un porcentaje (%) de la media. La normalización también suele realizarse utilizando el máximo de la capa como referente o calculando la diferencia de la variable respecto al valor mínimo y dividiendo por el rango. Finalmente, cabe citar a la estandarización (sustracción de la media y división por el desvío estándar) como una transformación usual para expresar variables un conjunto de variables no commensurables en un conjunto de variables normal estándar.

Un paso importante en el análisis exploratorio de datos geoestadísticos es explorar la distribución de la variable. Para ello, puede realizarse una estadística descriptiva que incluye la elaboración de gráficos de distribución de fre-

cuencias y medidas resumen (media, mediana y coeficiente de asimetría) de la variable en análisis. Cuando el método de análisis supone distribución normal de los datos, estas medidas exploratorias pueden ayudar a verificar el cumplimiento de los supuestos. Se considera que una distribución de frecuencias es simétrica y está próxima a la de una variable normal cuando la media y la mediana son prácticamente iguales y el coeficiente de asimetría es inferior a 1. La distribución de la variable también provee información para la depuración de datos raros.

1.3 Depuración de datos

Los *outliers*, datos raros o atípicos, son observaciones con valores que se encuentran fuera del patrón general o distribución del conjunto de datos. La eliminación de los *outliers* es fundamental en el análisis de datos espaciales ya que las varianzas espaciales son muy sensibles a la presencia de datos raros. Los *outliers* deben eliminarse cuando el conjunto de datos no se limita dentro del rango de variación esperable con valores máximos y mínimos derivados de conocimiento previo sobre la distribución de la variable. También pueden eliminarse desde un criterio estadístico, cuando luego de calcular la media y la desviación estándar (SD), se identifican los valores que se encuentran fuera de la media ± 3 SD. Según conocimiento teórico, el 89 % de los datos de una variable debieran encontrarse entre la media ± 3 SD cualquiera sea la distribución de la variable. Es recomendable, antes de la eliminación de los *outliers*, graficarlos utilizando coordenadas espaciales para visualizar su localización. De esta manera será posible identificar si los datos seleccionados para ser eliminados se relacionan

con algún patrón sistemático o se corresponden a errores aleatorios.

Al eliminar los *outliers* globales se eliminan los extremos del conjunto de datos, pero no los extremos locales (*outliers* espaciales). Los *outliers* espaciales, conocidos también como *inliers*, son datos que difieren significativamente de su vecindario, pero se sitúan dentro del rango general de variación del conjunto de datos. Existen estadísticos para identificar *inliers*, tal es el caso del índice autocorrelación espacial local de Moran (LM) (Anselin, 1995). Dado un grupo de datos que pertenecen a diferentes vecindarios, el LM es aplicado a cada dato individualmente y da idea del grado de similitud o diferencia entre el valor de una observación respecto al valor de sus vecinos. La fórmula del índice de autocorrelación espacial local de Moran es la siguiente:

$$LM_i = \frac{n}{(n-1)s^2} \sum_{j=1}^n [w_{ij} (Z(s_j) - \bar{z})]$$

donde $Z(s_i)$ es el valor de la variable z en la posición i ; \bar{z} y s^2 son la media y varianza muestral de z , respectivamente; $Z(s_j)$ es el valor de la variable z en todos los otros sitios (donde $j \neq i$); w_{ij} es el peso espacial entre las ubicaciones i y j .

Para el cálculo del Índice de Moran se debe identificar el vecindario de cada dato, es decir el dominio donde existen datos que pueden ser interpretados como vecinos espaciales y que serán usados como referencia para decidir si el dato correspondiente es o no diferente a sus vecinos. Los vecindarios se definen a través de redes de conexión las que si bien pueden ser de distintos tipos pueden expresarse en el formato de una matriz de ponderación espacial W . Cuando W es binaria, *i.e.* compuesta por ceros y unos, se indica con

1 si la posición j se considera vecina a la posición i . Otra posibilidad para construir la matriz de ponderaciones espaciales es usando una función de la distancia d (usualmente distancia Euclídea) entre los sitios i, j como elemento de W . Una función de amplio uso es la inversa de la distancia, es decir: $w_{ij} = 1/d_{ij}$. Así, valores muy cercanos en el espacio tendrán mayor ponderación. Existen diferentes opciones para definir el tamaño y la forma de los vecindarios de un dato espacial.

El índice de Moran local está estandarizado por lo que su nivel de significación puede ser evaluado en base a una distribución normal estándar. Los valores positivos del LM se corresponden con agrupamiento espacial de valores similares ya sean altos o bajos (autocorrelación positiva), mientras que un valor de LM negativo indica un agrupamiento de valores diferentes, por ejemplo, un sitio con valor bajo de la variable se encuentra rodeado de vecinos con valores altos (autocorrelación negativa).

Para determinar la significancia estadística de LM, se calcula el *valor-p* asociado a la prueba de hipótesis que establece que la correlación de la información de un sitio con la de sus vecinos es nula. El *valor-p* para un índice determinado debe ser lo suficientemente pequeño para considerar el valor en cuestión como un *outlier* espacial o *inlier* (rechazar la hipótesis nula). Dado que se realiza una prueba de hipótesis para cada uno de los puntos espaciales, se recomienda el ajuste de los *valores-p* por el criterio de Bonferroni.

Para visualizar el índice LM se puede representar en un diagrama de dispersión la similitud de cada valor observado respecto a las observaciones vecinas. Usualmente en el eje horizontal se expresan los valores de las observaciones mientras que en el vertical se representa el retardo espacial de la variable. Adicionalmente, se puede ajustar y añadir a

este diagrama modelos de regresión lineal y estadísticos de influencia para identificar sitios con datos raros.



2

Caracterización de variabilidad espacial

Denotamos el proceso espacial en d dimensiones como: $Z(s) : s \in D \subset R^d$ donde Z denota el atributo que observamos, s es la ubicación en la cual Z es observada y es un vector de coordenadas de dimensiones $n \times 2$ y D es el dominio. Los procesos espaciales que se abordarán en este libro son procesos bidimensionales, $d = 2$ y $s = x, y$ son tratadas como coordenadas cartesianas. Cuando la d es mayor a 1, el proceso estocástico subyacente es definido como un campo aleatorio.

La colección de n observaciones georreferenciadas que conforman un conjunto de datos espaciales deben entenderse como una muestra de tamaño uno de una distribución n -dimensional. En este caso la $E [Z(s)] = \mu(s)$ representa el promedio del atributo en la ubicación s sobre la distribución de una posible realización. Si quisieramos determinar el valor esperado para un sitio no observado, para s_0 , sería necesario repetir las observaciones en ese punto, pero usualmente solo se tiene una observación por sitio. Sólo se puede hacer inferencia basada en una muestra de tamaño uno, bajo condiciones de estacionariedad, es decir cuando la esperanza es la misma en todos los puntos. Por ello, un supuesto importante para el análisis de los datos espaciales será el de estacionariedad. Bajo estas condiciones la variabilidad espacial podrá ser caracterizada a través de funciones basadas solo en varianzas y covarianzas o autocorrelaciones en los datos espaciales.

La autocorrelación espacial mide la correlación lineal entre los valores de una variable aleatoria y los de otra construida a partir del rezago de la primera. Puede interpretarse como medida de la coincidencia de valores similares de una variable en espacios geográficos cercanos, es decir, la variable tiende a asumir valores similares en unidades geográficamente cercanas. Mediado por la distancia, queremos saber qué tan semejante o diferente es el valor de la variable “consigo misma”. Luego, para una variable espacialmente autocorrelacionada, los valores observados en el espacio no serán aleatorios, sino que estarán espacialmente relacionados.

La autocorrelación puede ser global o local. El primer tipo considera los valores de todas las observaciones, mientras el segundo solo los valores de las observaciones de un sitio respecto a los de observaciones vecinas. En ambos casos, la autocorrelación espacial puede ser medida en términos de su intensidad; una autocorrelación espacial fuerte significa que los valores del atributo de las unidades de observación geográfica vecinas muy parecidos o predecibles desde el valor del sitio, el caso contrario se produce cuando la distribución en sitios vecinos refleja un patrón aleatorio. El análisis de autocorrelación espacial requiere contar con una medida de correlación lineal apropiada para medir grados de semejanza entre las observaciones en un sitio y en su entorno.

Los índices de autocorrelación espacial expresan de manera formal el grado de correlación lineal entre las variables aleatorias representadas funcionalmente por el vector de valores observados y el vector de medias ponderadas espacialmente en las unidades vecinas, llamado el vector con *lag espacial*. El cálculo de estos índices en un espacio continuo requiere la definición de una matriz de ponderación espacial. Ésta puede tener elementos binarios para indicar cuáles son las

observaciones que pertenecen al vecindario de cada dato, *i.e.* las observaciones “conectadas” con cada dato.

También puede tener como elementos, los valores de un coeficiente de continuidad que mide el grado de conexión entre un par de datos. El elemento w_{ij} de la matriz de ponderaciones W , es el peso aplicado a la comparación de las observaciones en la posición i y la posición j . Usualmente se utilizan redes de conexión que derivan en un matriz de pesos espaciales. La red de vecindarios también puede ser definida considerando puntos vecinos a aquellos contiguos ubicados entre un límite inferior y superior, previamente preestablecido. Cuando las entidades se encuentran distribuidas en forma homogénea en el espacio, suele recomendarse la red de conexión obtenida por el método de triangulación de Delaunay. Las redes de conexión también pueden ser adaptadas manualmente pudiéndose excluir contactos entre sitios cercanos o incluir relaciones entre sitios lejanos.

Por ejemplo, el índice de autocorrelación espacial de Geary (GI), expresa la magnitud de las desviaciones entre observaciones en diferentes localizaciones. Siendo $w..$ la suma de todos los pesos, la expresión del índice es:

$$GI = \frac{(n - 1) \sum_i \sum_j w_{ij} (Z(s_i) - Z(s_j))^2}{2w.. \sum_i (Z(s_i) - \bar{z})^2}$$

El valor de GI se encuentra en el intervalo [0,2]. Si no hay autocorrelación espacial, el valor esperado de GI es 1. Valores del índice entre 1 y 2 indican autocorrelación espacial negativa, y entre 0 y 1 autocorrelación espacial positiva. Este índice se relaciona inversamente con el índice MI, es decir valores más cercanos a 0 sugieren autocorrelaciones positivas más fuerte. GI es más sensible a pequeñas diferencias entre posiciones vecinas que el IM. Los índices

de autocorrelación espacial local son calculados para cada sitio y usan solo ponderadores para las distancias entre las observaciones de ese sitio y las restantes. El índice LM fue descripto anteriormente para ejemplificar su uso en la detección de *outliers*. Otro índice de autocorrelación espacial local es el índice de Getis Ord (GO) el que se calcula como la suma de los valores observados para la j – éSIMA variable en el vecindario centrado del i – éSIMO píxel, en relación con la suma de todas las observaciones. Su expresión estandarizada es:

$$GO_i = \frac{\sum_{i=1}^m w_{i,i'} Z(s_{i'}) - \bar{Z} \sum_{i=1}^m w_{i,i'}}{S \sqrt{\frac{m \sum_{i=1}^m w_{i \neq i'}^2 - (\sum_{i=1}^m w_{i \neq i'})^2}{n-1}}}$$

donde w_i representa pesos espaciales en un vecindario del i – éSIMO píxel de tamaño m . Valores positivos de GO indican grupos locales de valores altos para la variable alrededor de la i – éSIMA ubicación, mientras que valores negativos indican grupos locales de valores bajos alrededor de la i – éSIMA ubicación. Para evaluar la significancia estadística de estos índices es posible utilizar procedimientos del tipo Monte Carlo (Babai, 1979). Las ubicaciones son permutadas en el espacio para obtener la distribución del índice bajo la hipótesis nula de distribución aleatoria.

2.1 Semivariogramas

La dependencia espacial o autocorrelación espacial, puede modelarse mediante un semivariograma. Esta función permite analizar la estructura y la naturaleza de la dependencia espacial en un conjunto de observaciones geo-referenciadas.

El proceso espacial puede ser representado por el siguiente modelo estadístico:

$$Z(s) = \mu + \varepsilon(s)$$

donde μ es la media del proceso y $\varepsilon(s)$ es un término de error aleatorio con media cero y covarianza $C(h)$, donde h es el *lag* o separación en el espacio entre dos sitios particulares. Un campo aleatorio $Z(s) : s \in D \subset R^d$ es estrictamente estacionario si la distribución espacial es invariante bajo traslación de las coordenadas a través de todo el dominio (estacionariedad en sentido fuerte). La estacionariedad de segundo orden, o estacionariedad en sentido débil, se produce cuando $E[Z(s)] = \mu(s)$ y $Cov[Z(s), Z(s+h)] = C(h)$. Es decir, en un campo aleatorio estacionario de segundo orden, la media es constante y la covarianza entre observaciones sobre diferentes posiciones, es función de la separación espacial entre los sitios en las que son tomadas, $C(h)$ es la función de covarianza del proceso espacial. La estacionariedad de primero orden implica la estacionariedad de segundo orden, pero la inversa no es cierta.

Dado que $C(h)$ no depende del valor de las coordenadas y $Cov[Z(s), Z(s+0)] = Var[Z(s)] = C$, en procesos estacionarios de segundo orden, la variabilidad es la misma en todas partes, *i.e.* $Var[Z(s)] = \sigma^2$ no es una función de la ubicación espacial. En síntesis, un proceso espacial estacionario de segundo orden tiene media y varianza constantes y la función de covarianza no depende en absoluto de las coordenadas. A $C(h)$ también se la conoce como función de autocovarianza y depende de la escala en la cual Z fue medida. Resulta más conveniente y fácil de interpretar si se la hace adimensional convirtiéndola en autocorrelación $\rho(h) = \frac{C(h)}{C}$. La función $\rho(h)$ se denomina correlograma del proceso espacial.

Aún si $Z(h)$ no es estacionaria de segundo orden, el incremento $Z(s) - Z(s+h)$ puede serlo. Un proceso que tiene esta característica se dice que tiene estacionaridad intrínseca. Esto se produce si $E[(Z(s))] = \mu$ y $\frac{1}{2}Var[Z(s) - Z(s+h)] = \gamma(h)$.

La función $\gamma(h)$ es llamada semivariograma del proceso espacial. La clase de procesos intrínsecamente estacionario es más grande que la clase de procesos estacionarios de segundo orden. Notar que un proceso espacial que presenta estacionaridad intrínseca no es necesariamente estacionario de segundo orden. En condiciones de estacionaridad de segundo orden la función de covarianza es el semivariograma.

Un proceso que parece estacionario en una escala podría no serlo a otra escala (*i.e.* presentar una tendencia o un componente sistemático). En el modelo, μ será remplazado por $\mu(s)$, *i.e.* término de tendencia determinístico para el sitio s . El semivariograma, en estos casos se calcula sobre los residuos del modelo. El semivariograma, puede interpretarse como función de la varianza de la diferencia entre las observaciones. Si el semivariograma es sólo una función de la distancia entre observaciones, entonces es conocido como semivariograma isotrópico, *i.e.* no depende de la dirección. El semivariograma y covariograma son parámetros del proceso espacial y juegan un rol crítico en los métodos geoestadísticos de análisis de datos espaciales.

Un primer paso para caracterizar la variación espacial en un dominio continuo es construir un semivariograma experimental o empírico. Una fórmula usual para computar semivariogramas, es conocida como estimador de los momentos de Matheron

$$\hat{\gamma}(h) = \frac{1}{2m(h)} \sum_{i=1}^{m(h)} \left\{ Z(s_i) - Z(s_i + h) \right\}^2$$

donde $m(h)$ es el número de pares de puntos separados por la particular distancia h . El otro estimador ampliamente usado es el estimador de Cressie- Hawkins o estimador robusto cuya fórmula se expresa como

$$2\tilde{\gamma}(h) = \frac{\left[\frac{1}{m(h)} \sum_{i=1}^{m(h)} \left| Z(s_i) - Z(s_i + h) \right|^{\frac{1}{2}} \right]^4}{0,457 + \frac{0,494}{m(h)} + \frac{0,045}{m^2(h)}}$$

Este estimador puede ser menos sesgado que $\hat{\gamma}(h)$ cuando la varianza residual es relativamente pequeña siendo también menos sensible a la presencia de valores externos. El estimador muestra típicamente menor variación en distancias pequeñas y también resulta en valores generalmente más pequeños que el estimador de los momentos de Matheron. Computando cualquiera de los dos estimadores, para las distancias h , obtenemos un conjunto ordenado de semivarianzas. Tales semivarianzas graficadas en función h constituye el semivariograma empírico o experimental.

Los parámetros de un semivariograma son: la varianza nugget o simplemente nugget (C_0), la varianza estructural o sill parcial (C) y el rango (R). La asíntota es llamada la meseta del semivariograma o C y el lag o distancia h^* en el cual la meseta es alcanzada se denomina R o rango. Observaciones $Z(s_i)$ y $Z(s_j)$ para las cuales $\|Z(s_i) - Z(s_j)\| \geq h^*$ son espacialmente independientes. Si el semivariograma alcanza la meseta asintóticamente, se define el rango práctico (R_P) como la distancia en el cual la semivarianza alcanza el 95 % de la varianza umbral o total.

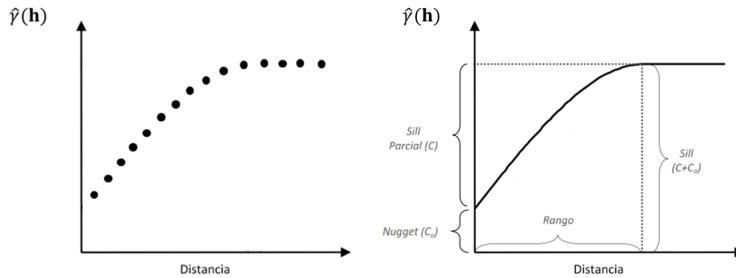


Figura 2.1: a) Semivariograma empírico. b) Semivariograma teórico, modelo esférico. Se representan los tres parámetros que lo definen: rango, sill y efecto pepita o nugget.

En la práctica el semivariograma empírico $\hat{y}(h)$ puede no pasar a través del origen. La ordenada al origen del semivariograma representa a C_0 , por lo tanto $C_0 = \lim_{h \rightarrow 0} g(h) \neq 0$. Este parámetro representa la suma de errores aleatorios o no estructurados espacialmente, así como errores asociados con la variabilidad espacial a escalas más finas que la usada para realizar las mediciones. Un alto valor de C_0 indica que la mayoría de la variación espacial no es explicada por el semivariograma. La varianza umbral o sill se obtiene sumando las varianzas antes mencionadas ($C_0 + C$) y es la varianza de observaciones independientes, es decir observaciones que fueron tomadas a mayor distancia que R .

Un semivariograma se define como anisotrópico si cambia en alguna forma respecto a la dirección que se considere. Si el semivariograma no solo depende de la longitud del vector h sino también de la dirección del vector entonces el semivariograma es anisotrópico. En los casos isotrópicos, los contornos de isocorrelación son esféricos, mientras que en el caso que haya anisotropía los contornos de isocorrelación son elípticos. Se reconocen dos tipos de anisotropía: anisotropía geométrica y anisotropía zonal. Anisotropía geométrica ocurre cuando el rango del semivariograma

cambia en las distintas direcciones, pero no la varianza sill, por lo tanto, la correlación es más fuerte en una dirección que en otra. Anisotropía zonal existe cuando la varianza estructural del semivariograma cambia con la dirección. Anisotropía geométrica significa que la correlación es más fuerte en una dirección que en otra.

Una forma en que la anisotropía geométrica puede ser identificada es graficando un semivariograma experimental direccional. Diferencias en el semivariograma muestral usando diferentes ángulos al computarlo, es indicador de anisotropía. La anisotropía geométrica puede ser modelada cambiando el modelo de semivariograma por un proceso isotrópico transformando las coordenadas. Los modelos teóricos de semivariograma más usados en predicción espacial están basados son isotrópicos, por lo que es necesario una corrección en casos de anisotropía para poder utilizar la metodología clásica de predicción en geoestadística. El radio de anisotropía, es decir, el cociente entre los rangos de la dirección de máximo y mínima variación es usada para medir anisotropía. Algunos autores consideran que existe anisotropía significativa si el radio de anisotropía es mayor a 2,5.

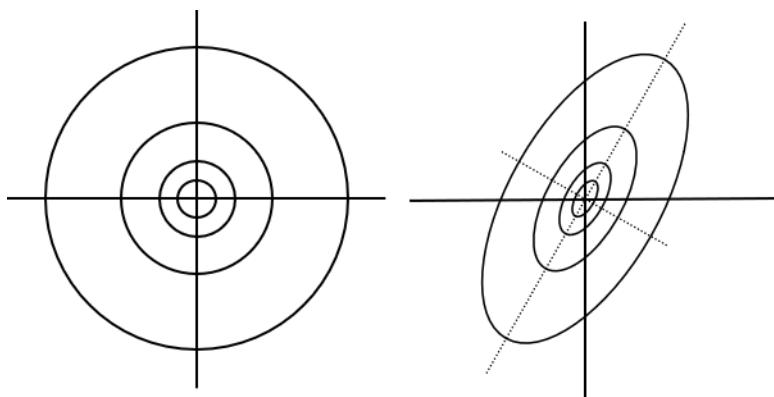


Figura 2.2: a) Modelo isotrópico. b) Modelo anisotrópico, con ángulo de anisotropía de 45° y un radio de anisotropía de 0,5.

En los procesos espaciales continuos, caracterizados por semivariogramas suelen obtenerse medidas del grado de estructuración espacial. Una de éstas es la varianza estructural relativa (RSV):

$$RSV = \left(\frac{C}{C + C_0} \right) \times 100\%$$

Un valor alto de RSV indica que las predicciones geostadísticas serán más eficientes que aquellas obtenidas con métodos de predicción que ignoran la información espacial. Un valor alto de RSV también indica una continuidad mayor del proceso espacial. Zimback (2001) establece que el grado de dependencia espacial puede ser clasificado como: $RSV \leq 25\%$ bajo, RSV entre 25% y 75% medio y $RSV \geq 75\%$ alto. También se puede calcular el cociente $\frac{C_0}{C_0+C}$ y en función de éste hablar de estructura espacial fuerte cuando el cociente es: $\leq 25\%$, intermedia si el mismo se encuentra entre 25% y 75% y débil si el mismo es mayor al 75% .

2.1.1 Ajuste de semivariogramas

El semivariograma empírico $\hat{\gamma}(h)$, es un estimador insesgado de $\gamma(h)$, pero provee solo estimaciones para un conjunto finito de distancias. Para obtener estimaciones de $\gamma(h)$, para cualquier lag, al semivariograma empírico se le ajusta un modelo teórico. El análisis geoestadístico sigue entonces estos dos pasos: 1) obtención del semivariograma empírico y 2) ajuste de un modelo teórico de semivariograma al semivariograma empírico.

Las funciones que sirven como modelos teóricos de semivariograma deben ser condicionalmente definidas positivas. Existen varios modelos teóricos para funciones semivario-

gramas, entre los que se encuentran el modelo nugget, el lineal, el esférico, el gaussiano y el exponencial (Figura 2.3). El semivariograma de un proceso de ruido blanco (modelo nugget), donde los valores $Z(s)$ se comportan como muestras aleatorias, todas con igual media y varianza sin correlación entre ellas. Este modelo suele ajustar el semivariograma empírico cuando la menor distancia de muestreo en los datos es mayor que el rango del proceso espacial.

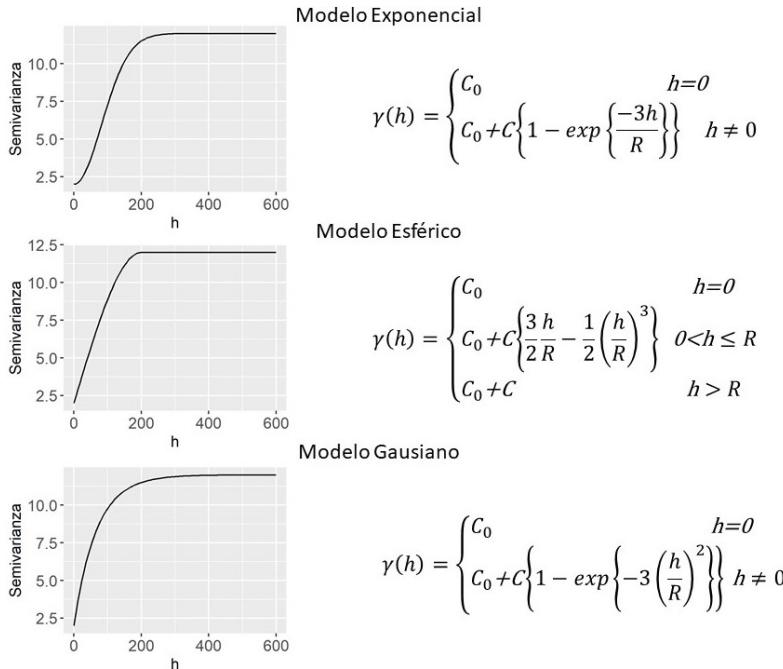


Figura 2.3: Funciones de semivariograma para el modelo exponencial, esférico y gaussiano. $C_0=2$, $C=10$ y $R=200$

El modelo esférico es uno de los más populares entre los modelos de semivariograma. Tiene dos características principales: un comportamiento lineal cerca del origen y el hecho de que a la distancia R el semivariograma encuentra la meseta y después de esta se mantiene llano. El modelo exponencial se aproxima a la meseta del semivariograma

asintóticamente cuando $\| h \| \rightarrow \infty$. En la parametrización mostrada en la Figura 2.3, el parámetro R es el rango práctico del semivariograma. Frecuentemente el modelo puede encontrarse en una parametrización donde el exponente es $- \| h \| /R$. Entonces el R_p corresponde a $3R$. Para el mismo rango y meseta de un modelo esférico, el modelo exponencial alcanza el rango más rápidamente, es decir, a menor distancia que el modelo esférico. El modelo gaussiano exhibe un comportamiento cuadrático cerca del origen y produce una correlación de corto rango que son las más altas que para cualquier modelo estacionario de segundo grado con el mismo rango práctico. Además, es el más continuo cerca del origen de los considerados aquí. En la parametrización el rango práctico es $\sqrt{3R}$.

Es importante notar que, si se realiza un análisis basado en semivariogramas y se pretende comparar los parámetros de los semivariogramas obtenidos bajo distintas condiciones, la utilización de modelos teóricos diferentes resulta poco útil. Hay que tener en cuenta que, por ejemplo, los rangos del modelo esférico y el exponencial no son directamente comparables. El modelo esférico es el único que tiene un umbral verdadero, ya que tanto el modelo exponencial como el gaussiano alcanzan el umbral de forma asintótica, o lo que es lo mismo, no lo alcanzan nunca y el modelo lineal no tiene umbral. En consecuencia, los rangos no son directamente equivalentes entre modelos. En este caso, es más conveniente elegir un único modelo para realizar comparaciones de procesos espaciales.

Los modelos de semivariograma son no lineales a excepción del modelo nugget. Por ello, para la estimación de parámetros estas funciones se usan métodos basados en aproximaciones numéricas. El método de ajuste por mínimos cuadrados ponderados (WLS) es común en la práctica. Para ello, se elige una función y valores iniciales de los

parámetros basados en la observación del semivariogramas empírico. El tamaño del conjunto de datos a partir del cual el modelo de semivariograma es ajustado depende del número de *lags* que se elija. Los valores de las clases de *lag* en las cuál el número de pares no es mayor a 30 debieran ser removidos si se ajusta el semivariograma por mínimos cuadrados. Journel y Huijbregts 1978 recomiendan solo usar *lags* menores a la mitad del máximo *lag* en el conjunto de datos. La distribución de los puntos en el espacio determinará para qué *lags* esto es posible.

2.2 Correlación espacial bivariada

2.2.1 Coeficiente de correlación

El coeficiente de correlación lineal de Pearson (r) es una medida de la magnitud de la correlación lineal entre dos variables. Para calcularlo se supone que se tiene una muestra aleatoria de unidades de análisis donde se han registrado simultáneamente dos variables. El intervalo de confianza para r y el valor p usados para decidir si la correlación poblacional entre ambas variables es cero o distinta de cero, dependen del tamaño de la muestra n . El tamaño de la muestra es el número de unidades de análisis independientes.

Cuando las variables en estudio exhiben autocorrelación espacial, las observaciones de cada una de éstas estarán correlacionadas dentro de un determinado vecindario, es decir, no serán independientes entre sí. Luego, en el caso de datos espaciales, se viola la suposición de observaciones independientes para la prueba de significancia r . Una propuesta para contemplar las correlaciones generadas por

patrones espaciales es calcular el coeficiente de correlación haciendo un ajuste para determinar el número de observaciones independientes (tamaño de muestra efectivo) para acompañar la inferencia necesaria.

El coeficiente de correlación modificado (Clifford et al., 1989; Dutilleul et al., 1993), permite evaluar correlación entre dos variables espacializadas en el mismo dominio espacial. La prueba se basa en la modificación de las varianzas y los grados de libertad de la prueba t estándar usada para evaluar significancia del coeficiente de correlación de Pearson y requiere de la estimación del tamaño efectivo de la muestra.

Considerando $A \subset D$ un grupo de n sitios $A = s_1, s_2, \dots, s_n$, se supone que $Z = Z(s_1), Z(s_2), \dots, Z(s_n)$ y $Y = Y(s_1), Y(s_2), \dots, Y(s_n)$ con media constante y matriz de varianzas y covarianzas Σ_Z y Σ_Y . Se divide D en los estratos D_0, D_1, D_2, \dots . Entonces $Cov(Z(s_i), Z(s_j)) = C_Z(k)$ y $Cov(Y(s_i), Y(s_j)) = C_Y(k)$, con $s_i, s_j \in D_k$, para $k = 0, 1, \dots$ (Clifford et al., 1989) sugieren como estimador de $\hat{C}_Y(h)$

$$\hat{C}_Y(h) = \frac{\sum_{s_i, s_j \in A_k} (Y(s_i) - \bar{Y})(Y(s_j) - \bar{Y})}{n_k}$$

donde n_k es la cardinalidad de D_k y similaridad para $C_Z(k)$. Luego, Clifford et al. (1989) sugirió utilizar $n^{-2} \sum_h n_h \hat{C}_Z(h) \hat{C}_Y(h)$ como un estimador de la varianza condicional de $s_{ZY} = n^{-1} \sum_D (Z(s) - \bar{Z})(Y(s) - \bar{Y})$. Como resultado se obtiene la prueba t modificada basada en el estadístico W

$$W = n s_{ZY} \left(\sum_h n_h \hat{C}_Z(h) \hat{C}_Y(h) \right)^{-2}$$

El cual a partir de una serie de aproximaciones a la varianza del coeficiente de correlación de Pearson (σ_r^2) entre los procesos $Z(s)$ e $Y(s)$ se puede escribir de la siguiente manera $W = (\hat{M} - 1)^{1/2}r$, $\hat{M} = 1 + \hat{\sigma}_r^{-2}$ y $\hat{\sigma}_r^2 = \frac{\sum_h n_h \hat{C}_Z(h) \hat{C}_Y(h)}{n^2 s_Z^2 s_Y^2}$.

Se define W como una prueba t modificada con $\hat{M} - 2$ grados de libertad, donde \hat{M} es el tamaño de muestra efectivo asumiendo bajo hipótesis nula la no correlación entre $Z(s)$ e $Y(s)$. Cuando se presenta una estructura de correlación espacial positiva, generalmente $\hat{M} < n$, si existe estructura de autocorrelación negativa se espera que $\hat{M} > n$.

2.2.2 Coeficiente de co-dispersión

Otra forma usada en estadística espacial para explorar patrones de correlaciones o covariaciones entre dos variables espacializadas, es el coeficiente de co-dispersión, que cuantifica la correlación entre dos procesos espaciales para un lag espacial particular sobre un espacio bidimensional. Para dos procesos espaciales intrínsecamente estacionarios $Z(s) : s \in D \subset R^2$ y $Y(s) : s \in D \subset R^2$ definidos en una parte de la región $D \subset R^2$, el coeficiente de co-dispersión es definido como:

$$\rho_{ZY}(h) = \frac{E\left[\left(Z(s+h) - Z(s)\right)\left(Y(s+h) - Y(s)\right)\right]}{\sqrt{E\left[Z(s+h) - Z(s)\right]^2 E\left[Y(s+h) - Y(s)\right]^2}}$$

La estructura de ρ_{ZY} es computacionalmente similar al coeficiente de correlación de Pearson. Al igual que ese coeficiente, $\rho_{ZY}(h)$, donde los límites superior e inferior definen una asociación espacial negativa o positiva perfecta, respectivamente. Sin embargo, a diferencia del coeficiente de

correlación de Pearson, ρ_{ZY} depende del lag h , que enfatiza la idea de que la correlación espacial es un valor asociado con una distancia en el plano. El cálculo de la correlación se realiza para diferentes distancias y direcciones en el espacio. Cuando el coeficiente de co-dispersión se calcula para muchas direcciones, es útil mostrar esos valores en un solo gráfico. Vallejos et al. (2015) proponen el mapa de co-dispersión para resumir en un plano los valores de los coeficientes de co-dispersión obtenidos para distintos *lag* espaciales (direcciones y distancias). El gráfico resume la información sobre la correlación entre dos procesos espaciales en forma radial sobre un plano que circunscribe los coeficientes en una semiesfera de radio no mayor al rango del proceso espacial 2.4. En general las correlaciones espaciales que se observan desde un gráfico de co-dispersión permanecen ocultas en el análisis exploratorio usual y pueden ser distintas a las correlaciones lineales de Pearson no restringidas espacialmente. El gráfico de co-dispersión no debe ser confundido con un mapeo de la co-dispersión de las variables en el espacio de interés. No captura similitudes relacionadas con los patrones o formas que están presentes en los procesos espaciales, sino que captura la dependencia espacial entre los procesos para una distancia h . Los ejes del gráfico de co-dispersión hacen referencia a los *lag* y direcciones y no a las coordenadas de los sitios muestrales originales.

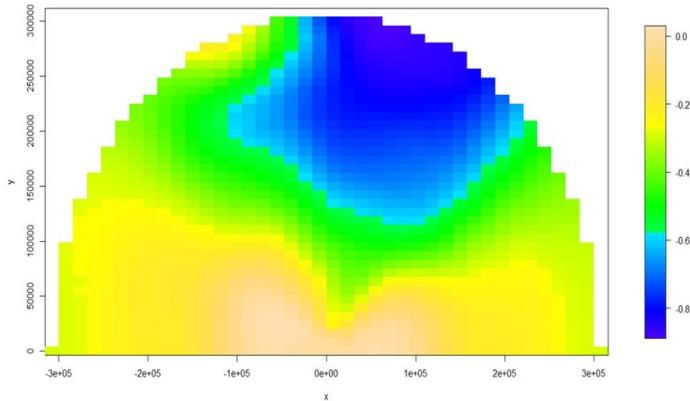


Figura 2.4: Gráfico de co-dispersión mostrando la correlación espacial entre dos variables para varios lag espaciales.

2.3 Interpolación Kriging

La predicción espacial, es decir la predicción de valores de la variable en sitios del campo espacial donde no existen observaciones, usualmente se hace por el método kriging basándose en el semivariograma ajustado. Kriging proporciona el mejor estimador lineal insesgado del valor esperado para el sitio y un error de estimación conocido como varianza kriging. Esta varianza depende del modelo de semivariograma ajustado y de la ubicación en el espacio de los datos observados ya que son los datos observados en distintos sitios los que proveen información para aproximar el valor en el sitio sin dato. Las interpolaciones basadas en semivariograma, se denominan geoestadísticas y tienen

ciertas ventajas respecto a interpolaciones determinísticas, como las obtenidas por el método IDW que se basa en las distancias geométricas entre los sitios con datos y el sitio a interpolar. Las observaciones que participan en la predicción se ponderan de forma distinta según la distancia estadística a la que se encuentran.

Los parámetros del semivariograma son los que gobiernan la asignación de los pesos o ponderaciones de las observaciones vecinas al sitio al cual se le asignará la predicción. El parámetro *nugget* es determinante en la asignación de pesos. si la varianza del error es muy alta, todas las observaciones tenderán a tener el mismo peso en la interpolación. Por el contrario, si la varianza del error es baja, los coeficientes de ponderación serán distintos. Si el rango aumenta, cada punto tendrá mayor peso en la interpolación de otras observaciones. Entre los métodos de interpolación geoestadísticos que utilizan todos los datos simultáneamente se destacan los métodos de kriging ordinario, simple y universal. En el kriging ordinario la media de la variable es estimada localmente. En caso de conocer la media poblacional de la variable, hecho que raramente ocurre, se utiliza el kriging simple. En el kriging universal se estima también la influencia de una tendencia espacial de los datos. La predicción asignada a los puntos incógnita puede realizarse de manera puntual (kriging puntual) o definiendo bloques (kriging en bloques) (Schabenberger y Gotway, 2005; R. Webster y Oliver, 2007).

2.3.1 Kriging ordinario

El kriging ordinario supone que la variación es aleatoria, que existe dependencia espacial y que el proceso espacial subyacente es intrínsecamente estacionario con media constante y varianza que depende solo de la separación en

distancia entre los sitios y no de su posición. La predicción kriging resulta de una combinación lineal de los datos observados. Supongamos que los valores de Z , han sido muestreados en los puntos s_1, s_2, \dots, s_n , para generar N datos $z(s_i)$, $i = 1, 2, \dots, N$. Para el caso del kriging ordinario puntual se predice Z en cualquier nuevo punto s_0 mediante:

$$\hat{Z}(s_0) = \sum_{i=1}^N w_i z(s_i)$$

donde w son los pesos asignados a cada observación. Para asegurar que la estimación del valor esperado para el sitio sea insesgada y de mínima varianza, los pesos son dado de manera que:

$$\sum_{i=1}^N w_i = 1$$

$$E[\hat{Z}(s_0) - z(s_0)] = 0$$

$$\begin{aligned} Var[\hat{Z}(s_0)] &= E[\hat{Z}(s_0) - z(s_0)^2] \\ &= 2 \sum_{i=1}^N w_i \gamma(s_i - s_0) - \sum_{i=1}^N \sum_{j=1}^N w_i w_j \gamma(s_i - s_j) \end{aligned}$$

donde la cantidad $\gamma(s_i - s_0)$ es la semivarianza de Z entre el punto de muestreo i y el punto objetivo x_0 y $\gamma(s_i - s_j)$ es la semivariancia entre los puntos de muestreo i y j . Las semivarianzas se derivan del modelo teórico de semivariograma, debido a que no hay existen valores de semivarianzas entre los sitios con datos y los sitios objetivos donde no existen valores observados. Si un sitio objetivo también es un punto de muestreo, el kriging puntual devuelve el

valor observado en ese sitio y la varianza de estimación es cero. El kriging puntual es un interpolador exacto en este sentido. El siguiente paso en kriging es encontrar los pesos que minimizan la varianza de la predicción sujeto a la restricción de que la suman de los pesos se igual a 1.

$$\sum_{i=1}^N w_i \gamma(s_i - s_0) + \psi(s_0) = \gamma(s_j - s_0) \forall j$$

La cantidad $\psi(s_0)$ es el multiplicador de Lagrange introducido para lograr la minimización. La solución de las ecuaciones de kriging proporciona los pesos para las ponderaciones y la varianza de predicción se obtiene de la siguiente forma:

$$\sigma^2(s_0) = \sum_{i=1}^N w_i \gamma(s_i - s_0) + \psi(s_0)$$

2.3.2 Kriging en bloques

El kriging en bloques consiste en estimar directamente el valor promedio de la variable sobre un soporte mayor que el soporte de los datos (bloque). Intuitivamente, la idea es calcular mediante kriging puntual los valores en todos los puntos de una superficie o bloque y usar la media de las predicciones como estimador del valor esperado para el sitio. La estimación para cualquier bloque sigue siendo un promedio ponderado de los datos, $z(s_1)$, $z(s_2)$, ..., $z(s_N)$:

$$\hat{Z}(B) = \sum_{i=1}^N w_i z(s_i)$$

Los factores de ponderación se obtienen nuevamente para minimizar la varianza del error y para obtener un estimador

insesgado de la media. La grilla de predicción sobre la que se construye el mapa de variabilidad espacial presenta una dimensión menor que la de los bloques, asegurándose la obtención de un mapa más suavizado respecto al obtenido con kriging puntual. El kriging en bloques ha mostrado ser efectivo a la hora de reducir errores que pueden trasladarse en los mapas como consecuencia de inexactitudes de datos puntuales.

2.3.3 Kriging local

Hemos dicho que los pesos de las observaciones en la predicción geoestadística son funciones de las semivarianzas entre las observaciones en sitios en el vecindario, $\gamma(s_i - s_j)$, y entre cada punto de muestreo y el punto a predecir, $\gamma(s_i - s_0)$. En general solo los puntos cercanos al punto a predecir tienen un peso significativo. Cuando la relación nugget:sill es pequeña el interpolador kriging es visto como un predictor local, donde para la predicción de $Z(s_0)$ participarán sólo datos de puntos dentro de la proximidad de s_0 (*kriging neighborhood* o *kriging local*). El kriging local esencialmente asigna pesos $w(s_0) = 0$ para todos los puntos s_i fuera de la zona en la que se quiere predecir. Por otra parte, esto permite que podemos aceptar el supuesto de estacionariedad local (o cuasi estacionariedad), es decir se puede restringir el supuesto de estacionariedad de la media a los vecindarios del kriging. Lo que sucede en distancias mayores a las del vecindario del sitio no será de importancia para la predicción en el sitio. La predicción y varianza kriging dependen principalmente de la parte del semivariograma cercana al origen, por ello es de importancia modelar bien el semivariograma en estos lugares, *i.e.* dar más peso a las semivarianzas experimentales cercanas al origen. No hay reglas para definir el vecindario para implementar el

kriging local, aunque cuando el nugget es relativamente bajo se puede definir un radio de vecindad cercano al rango o rango práctico del modelo de semivariograma ajustado. Cuando el efecto nugget es importante el radio de vecindad debería ser mayor al rango ya que es probable que puntos más distantes tengan aún peso significativo. Otra alternativa para definir el vecindario se basa en términos de un número mínimo y máximo de datos cercanos al punto a predecir. Algunos autores recomiendan utilizar un mínimo de 7 vecinos y un máximo de 20.

2.3.4 Kriging universal

La suposición de estacionariedad intrínseca no se cumple cuando existen tendencias geográficas pronunciada de naturaleza sistemática y no aleatoria. La tendencia puede ser regional, es decir, una variación sistemática en toda la región de interés o local de un punto a otro dentro de la región estudiada. La existencia de tendencias puede ser explorada graficando los datos de la variable analizada en función a la variable que se supone genera la tendencia espacial. La tendencia también se manifiesta en los semivariogramas experimentales con un incremento de la semivarianza con la distancia que no tiene límites. Si hay tendencia, entonces μ ya no es constante, sino que depende de s . Además, el semivariograma experimental de los datos ya no estima el semivariograma de los errores aleatorios, $\varepsilon(s)$. Se necesita estimar el semivariograma de $\varepsilon(s) = Z(s) - \mu(s)$. Cuando este variograma es el input del kriging, el proceso de interpolación se conoce como “kriging universal” y la predicción se obtiene como:

$$\hat{Z}(s_0) = \sum_{i=1}^N w_i f_k z(s_i)$$

donde f_k es función de las coordenadas espaciales.

2.3.5 Validación cruzada

La predicción implica asignar nuevos valores de las variables respuesta a contextos o escenarios que no corresponden al conjunto de escenarios medidos, es decir no se trata de aquellos sitios que utilizaron para realizar la predicción espacial. Entre las alternativas para estimar la exactitud de la predicción existen las técnicas de validación cruzada o técnicas de partición del conjunto de datos en datos de calibración y datos de validación (Efron y Hastie, 2016). Es necesario identificar un grupo de observaciones sobre las que se ajustará el modelo o el método que permite predecir, usualmente llamado datos de calibración, y otro grupo que se usará para validar, llamado datos de validación. El modelo (semivariograma teórico) se ajusta sobre el conjunto de datos de entrenamiento y posteriormente se usa para la predicción de interés, con observaciones del subconjunto de validación. Seguidamente, los valores observados del conjunto de validación se comparan con los valores predichos por el modelo. Usualmente el proceso se repite cruzando el rol de los subconjuntos de datos, es decir el que era de validación pasa a ser de calibración y viceversa.

Sin embargo, otras estrategias pueden ser usadas para la selección de los datos de entrenamiento y validación. Una es particionar en forma aleatoria los datos en ambos conjuntos. Otro tipo de validación cruzada es dejando uno fuera (*Leave-One-Out*) donde se utiliza una sola observación para conformar el subconjunto de validación y se deja al resto como subconjunto de entrenamiento. El modelo se ajusta utilizando las $n-1$ observaciones de entrenamiento y se obtiene una predicción para la observación excluida. Este proceso se repite n veces. Otro tipo de validación cruzada es

k – fold, donde las observaciones se dividen aleatoriamente en *k* grupos de aproximadamente igual tamaño. Uno de los *k* grupos se emplea como subconjunto de validación, mientras que el resto de los grupos se emplean para entrenar el modelo. El proceso de validación cruzada es repetido durante *k* iteraciones, con cada uno de los subconjuntos de datos de prueba. Un valor común de *k* que puede dar buenos resultados en cuanto al equilibrio sesgo-varianza para estimar el error de predicción es *k* = 10. Si el modelo tuvo un buen desempeño, los residuos de la validación cruzada serán pequeños, su media será cercana a cero y no presentarán estructura.

En la evaluación de modelos geoestadísticos, los valores predichos de kriging $\hat{Z}(s_i)$ se comparan con los observados $z(s_i)$, y se calcula una medida resumen que caracteriza el resultado de la comparación. Algunas de estas medidas resumen son:

Error medio

$$ME = \frac{1}{N} \sum_{i=1}^N \{z(s_i) - \hat{Z}(s_i)\}$$

donde *N* es el número de observaciones, $z(s_i)$ es el valor verdadero en s_i y $\hat{Z}(s_i)$ es el valor predicho en ese punto.

Error cuadrático medio (Mean Square Error, MSE):

$$MSE = \frac{1}{N} \sum_{i=1}^N \{z(s_i) - \hat{Z}(s_i)\}^2$$

Raíz del error cuadrático medio:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \{z(s_i) - \hat{Z}(s_i)\}^2}$$

Media del cociente del error cuadrático (Mean Squared Deviation Ratio, MSDR):

$$MSDR = \frac{1}{N} \sum_{i=1}^N \frac{\{z(s_i) - \hat{Z}(s_i)\}^2}{\hat{\sigma}^2(s_i)}$$

donde $\hat{\sigma}^2(s_i)$ es la varianza de la predicción kriging en el punto s_i .

Para el caso de datos espaciales, no solo es necesario disponer de una medida de error de predicción global, sino que también hay que evaluar del error de la predicción en cada sitio específico, *i.e.* dimensionar el error puntual de la predicción espacial.



3

Caracterización de variabilidad espacial con múltiples capas de datos

3.1 Análisis de componentes principales

Diferentes objetivos pueden surgir cuando analizamos un conjunto de datos que además de ser espaciales o georreferenciados es multivariado (*i.e.* múltiples capas de información sobre el mismo dominio espacial o varias variables por sitio). Por un lado, se puede querer resumir la variabilidad de los sitios usando unas pocas variables sintéticas que representen bien la variabilidad en las variables originales. Por otro, se puede querer resumir patrones espaciales usando unas pocas variables sintéticas que combinan las múltiples capas de información considerando la correlación espacial subyacente. Una solución al primer problema es usar el Análisis de Componentes Principales (PCA (Pearson, 1901)). Mientras que el segundo objetivo puede ser abordado mediante el Análisis de Componentes Principales Espaciales propuesto por (Dray et al., 2008), también conocido como MULTISPATI-PCA. Éste se basa en el PCA, pero incorpora la restricción dada por los datos espaciales mediante el cálculo del índice de Moran antes de obtener las variables sintéticas o componentes principales (PC). Los datos multivariados son generalmente registrados en una matriz X con n filas (observaciones) y p columnas (variables). El PCA permite identificar las variables que

explican la mayor parte de la variabilidad total contenida en los datos, explorar las correlaciones entre variables y reducir la dimensión del análisis al combinar las variables originales en nuevas variables sintéticas. El PCA opera sobre la matriz de covarianza de las variables originales o de las variables estandarizadas con el fin de encontrar una base ortogonal de tal manera que el primer eje del nuevo espacio considera la dirección de mayor variación de los datos originales. La descomposición espectral de la matriz de covarianzas proporciona un conjunto de autovectores y sus correspondientes autovalores. Los autovectores contienen los coeficientes de ponderación para construir variables sintéticas como combinaciones lineales de las variables originales. Los coeficientes de cada variable en estas combinaciones lineales indican la importancia relativa de las variables para explicar la variabilidad entre las observaciones. Las combinaciones lineales obtenidas con PCA se llaman componentes principales (PC), son ortogonales y en conjunto explican la variabilidad de los datos originales. Existen tantas PC posibles de formar como columnas en la matriz X . La primera componente (PC1) explica la mayor parte de la variación en el conjunto de datos y la segunda (PC2), la mayor parte de la variabilidad remanente o no explicada por la PC1, y así sucesivamente.

Los resultados del PCA se pueden visualizar en un gráfico denominado Biplot (Karl Ruben Gabriel, 1971) el cual permite representar en un plano óptimo para el estudio de variabilidad, las diferencias entre sitios, la correlación entre variables y las variables que mejor explican las principales componentes de variabilidad. La incorporación de la información geográfica o la característica espacial de los datos suele realizarse a posteriori del PCA mediante la asignación de los valores de las componentes a cada uno de los sitios georreferenciados o ajustando semivariogramas a las PC.

El objetivo de MULTISPATI-PCA, otra forma de trabajar con datos espaciales, es encontrar variables sintéticas independientes que optimicen el producto de la varianza total y la autocorrelación espacial. Para delimitar los vecindarios, MULTISPATI-PCA utiliza una matriz de pesos espaciales determinando cuáles y cuántas observaciones cercanas a cada sitio deben ser consideradas para el cálculo del índice de autocorrelación espacial. Este análisis permite estudiar las relaciones entre las variables considerando su estructura espacial. Para la implementación del análisis es necesario primero definir cómo la información espacial será incorporada. En MULTISAPTI - PCA, la detección de la estructura espacial se realiza a través del índice de Moran. Es necesaria la construcción de una red de conexión (también llamada gráficos de vecinos) la cual usa un criterio objetivo para definir que entidades son vecinas y cuáles no. Existen diferentes opciones o alternativas metodológicas para definir los vecindarios que dependen de los diferentes tipos de arreglos espaciales presente en los datos. Para muestras irregulares los vecindarios suelen definirse a partir de la red de conexión propuesta por Gabriel (K Ruben Gabriel y Sokal, 1969), mediante la triangulación de Delaunay (Lee y Schachter, 1980). Otro método es el de los vecinos más cercanos (Cover y Hart, 1967) o el basado en la especificación de una distancia como radio del vecindario de cada sitio.

Una vez que la red de conexión es definida, la información espacial es almacenada en una matriz de conexión binaria C (en la cual $c_{ij} = 1$ si las unidades espaciales i y j son vecinas o $c_{ij} = 0$ en caso contrario), la cual es simétrica y tiene tantas filas y columnas como sitios. Esta matriz de conectividad C en general es escalada para obtener la matriz de pesos espaciales (representación matemática de la disposición geográfica de los sitios en el dominio espacial).

Los pesos espaciales reflejan a priori la ausencia ($w_{ij} = 0$), presencia ($w_{ij} = 1$) o intensidad ($w_{ij} > 0$) de la relación espacial entre los sitios. Una vez que los pesos espaciales han sido definidos, el índice de autocorrelación de Moran es computado.

El método MULTISPATI-PCA opera sobre la matriz $\widetilde{X} = WX$ que está compuesta por los promedios ponderados de los valores de los vecinos de cada sitio según indique la matriz de conexión espacial, esta matriz es llamada matriz lagged. Las dos matrices X y \widetilde{X} tienen las mismas cantidades de columnas (variables) y de filas (sitios). El análisis MULTISPATI-PCA consiste en analizar la correlación entre este par de matrices (\widetilde{X} y X) mediante un análisis de co-inercia (Dray et al., 2003). MULTISPATI-PCA maximiza el producto escalar entre una combinación lineal de las variables originales y una combinación lineal de variables *lagged*. La ventaja de MULTISPATI-PCA respecto al PCA es que las componentes principales espaciales del MULTISPATI-PCA (sPC) contemplan la autocorrelación espacial entre los sitios, maximizándola en las primeras componentes. Por lo tanto, las primeras sPC del MULTISPATI-PCA muestran fuertes estructuras espaciales o altos índices de autocorrelación y no sólo mayores varianzas como en el PCA clásico. El método MULTISPATI-PCA constituye una herramienta multivariada útil no sólo para mapear la variabilidad conjunta de múltiples capas de datos dentro del dominio espacial estudiado sino también para la delimitación de zonas o áreas homogéneas en sentido multivariado cuando las componentes espaciales se usan como inputs de algoritmos de clasificación.

3.2 Análisis de conglomerados

Los métodos multivariados, utilizados para la clasificación de sitios de un dominio espacial, suelen basarse en algoritmos de agrupamiento no supervisados como los algoritmos de conglomerados jerárquicos o en algoritmos de conglomerados no jerárquico como k-means o fuzzy k means. Contrariamente al algoritmo k-means u otros métodos determinísticos de agrupamiento en los que cada observación sólo puede pertenecer a un único clúster, los métodos de clasificación basados en la teoría difusa (como fuzzy k-means), permiten que cada observación pueda asignarse a más de un clúster, con diferentes grados de pertenencia para cada clúster. Aplicado a datos espaciales puede generar alta fragmentación ya que el algoritmo de agrupación no tiene en cuenta la información espacial asociada a cada observación. Frogbrook y Oliver (2007) y Milne et al. (2012) propusieron introducir la restricción espacial mediante la incorporación de nuevas variables asociadas a parámetros del variograma co-regionalizado o del variograma de la componente principal de las variables originales. Córdoba et al. (2012) propusieron implementar fuzzy k means usando las componentes principales espaciales como variables de entrada para la clasificación con datos espaciales, logrando así disminuir la fragmentación e incrementar la contigüidad de los conglomerados espaciales. Otra alternativa, para delimitar conglomerados espaciales es aplicar filtros espaciales a la clasificación resultante de un método de clasificación estándar (Galarza et al., 2013; Ping y Dobermann, 2003).

En el método fuzzy k-means además de la matriz de datos X se genera la matriz de pertenencia difusa U , que contiene los valores o asignaciones parciales de cada una de las n

observaciones en cada uno de los k clusters o conglomerados, con la restricción que se debe cumplir para cualquier $i = 1, \dots, n$ y para cualquier $j = 1, \dots, k$:

$$u_{ij} \in [0, 1] \quad \forall_{i,j}$$

$$\sum_{j=1}^k u_{ij} = 1, \quad \forall_j$$

La partición difusa óptima de los datos es la que minimiza la función objetivo j_m igual a la suma ponderada de las distancias cuadráticas entre las observaciones y los centroides de cada clúster que conforman la matriz V :

$$j_m(U, V) = \sum_{i=1}^n \sum_{j=1}^k (u_{ij})^m (d_{ij})^2$$

donde m es el coeficiente de ponderación difuso ($1 \leq m < \infty$) cuya función es controlar el grado de solapamiento que se establece entre los clusters y $(d_{ij})^2$ es el cuadrado de la distancia en el espacio de los atributos entre el punto i y la clase centroide j . Distintas métricas de distancia pueden ser usadas. La distancia Euclídea se utiliza cuando las variables son independientes y de igual varianza. En caso contrario la distancia de Mahalanobis es usada. El algoritmo difuso *fuzzy k-means* utiliza un proceso iterativo que hace óptima la partición difusa de los datos X . La estructura del algoritmo (Bezdek et al., 1981) es la misma para cualquier conjunto de variables de entrada. Cuando el algoritmo ha asignado pesos o probabilidades de pertenencia a cada grupo para cada observación, se computaban una serie de índices para validar los distintos arreglos de conglomerados.

Para evaluar la clasificación conseguida con un determinado número de grupos, existen diferentes índices como el

coeficiente de partición (o fuzziness performance index-FPI, (Bezdek et al., 1981)), el índice de entropía de la clasificación o *normalized classification entropy* (NCE, (Bezdek et al., 1981)), el índice de Xie-Beni (Xie y Beni, 1991) y el de Fukuyama-Sugeno (Fukuyama y Sugeno, 1989), entre otros.

El coeficiente de partición (CP) mide el grado de solapeamiento (grado de fuzziness) entre los grupos formados. Se considera que mientras menos difusa es la partición, mejor es la clasificación. Por tanto, se prefiere la estructura con un número de conglomerados para la cual el coeficiente de partición es mayor. El máximo equivale a una clasificación en la que cada observación pertenece a un único clúster. El mínimo se da cuando cada observación pertenece, con la misma probabilidad, a cada clúster (mayor incertidumbre).

Otro índice que se puede usar para decidir con cuantos conglomerados quedarse es el conocido como entropía de la partición (EP) que cuantifica el grado de desorganización de la clasificación. Para este índice los valores próximos a 0 son indicativos de una mejor clasificación, es decir, con mayor grado de organización o menos difusos. El índice de Xie-Beni (XB) evalúa el cociente entre las distancias intracluster e intercluster. Se prefieren particiones donde la distancia intra-cluster es mínima y la distancia inter-cluster máxima. El índice XB es considerado como una medida de compacidad. Un valor bajo de XB, representa una clasificación con grupos compactos y separables. Por consiguiente, la mejor partición se obtiene mediante la minimización de XB. El índice Fukuyama-Sugeno (FS) es función de la separación entre los centroides de los grupos y la media de todos los centroides. El mínimo de FS corresponde a una partición con clases compactas y separables. Es importante considerar que, para un conjunto de datos, los índices no son necesariamente consistentes

463 *Caracterización de variabilidad espacial con múltiples capas de datos*

entre sí sugiriendo diferentes números de clúster como partición óptima. Una propuesta es promediar el valor de estos índices normalizados por el máximo usando para CP su reciproco, $CP^* = 1/CP$, para que el valor mínimo en todos los índices represente la estructura optima.

4

Predicción con múltiples capas de datos

La predicción del valor de una variable en un sitio no observado, realizada por interpolación kriging local, se alimenta de datos de la misma variable que se quiere predecir, pero obtenidos en sitios del vecindario de aquel para el que se requiere la predicción. Aún cuando la predicción pueda ser óptima, resulta informativo conocer el impacto de otras variables (variables secundarias, auxiliares o explicativas) sobre la predicción.

El método conocido como regresión kriging (Hengl et al., 2007) permite predecir o interpolar los valores de una variable en sitios no muestreados que se encuentran entre los sitios con observaciones a partir de capas de datos secundarios que actúan como variables predictoras. La incorporación de covariables de sitio puede mejorar sustancialmente la predicción de una variable respuesta espacializada. Para tal fin es necesario el ajuste de un modelo de regresión múltiple que describe la relación entre la variable observada y las capas de variables predictoras. Luego de ajustar la regresión, se obtienen los residuos (diferencia entre valor observado y valor predicho por el modelo a partir de las variables secundarias) y se realiza una interpolación kriging sobre los residuos para contemplar la espacialidad que no se encuentra relacionada con las capas de información usadas en el ajuste previo. Finalmente, los resultados del ajuste de regresión y de la interpolación kriging son combinados para producir la predicción.

Matemáticamente, esta interpolación es equivalente a la que hemos llamado kriging universal, ya que allí se realiza un ajuste de la variable de interés y las coordenadas espaciales (un tipo de variable auxiliar) cuyo impacto es descontado antes de realizar la interpolación espacial. Kriging desde modelo de regresión es también equivalente matemáticamente al método de interpolación conocido como kriging con deriva externa donde múltiples variables auxiliares son usadas como predictoras (covariables de sitio que son externas o distintas a las coordenadas espaciales) e intervienen directamente para resolver los pesos de la interpolación espacial.

La predicción sitio-específica de una variable respuesta en función de covariables de sitio puede realizarse también desde el marco teórico de los modelos lineales mixtos de covarianza residual (Balzarini et al., 2004). Es decir, ajustando un único modelo, donde la componente determinística es un modelo de regresión lineal múltiple que relaciona la respuesta con las covariables de sitio y las componentes sistemáticas (patrón espacial) y estocástica (variación sin estructura) de la variación espacial son expresadas conjuntamente en la matriz de varianza y covarianza de los términos de error (matriz de covarianza residual). Bajo este marco teórico, se utiliza máximo verosimilitud restringida o REML (Patterson y Thompson, 1971) para estimar el modelo predictivo. El modelo relaciona la variable de interés para el estudio de variación espacial con las variables explicativas también distribuidas espacialmente y contempla la dependencia espacial en las observaciones de la variable a predecir incorporando en la matriz de covarianza residual covariaciones derivadas de una función basada en la distancia que separa dichas observaciones en el espacio.

La construcción de un modelo predictivo para datos es-

paciales también puede realizarse desde un marco teórico Bayesiano, donde sistemática relacionada con la media de la respuesta sigue siendo el modelo de regresión lineal múltiple, pero se incorporan parámetros aleatorios de sitio para modelar la componente sistemática de los datos espaciales, los cuales se predicen independientemente de la componente de error. Trabajar el efecto de sitio como aleatorio, junto a la suposición de procesos espaciales locales, ha permitido eficientizar la estimación en un único paso de modelos para datos espaciales. Bajo el marco teórico bayesiano y el supuesto de campo aleatorio gaussiano markoviano para el proceso espacial, la estimación por aproximación de Laplace anidada conocida como INLA (del término en inglés *Integrated Nested Laplace Approximation*) es uno de los métodos elegidos para ajustar el modelo espacial (Rue et al., 2009).

Alternativamente, la construcción del modelo predictivo espacial puede realizarse mediante métodos de aprendizaje automático, también bajo el concepto de ajustar un modelo para la componente determinística, obtener los residuos y modelar el proceso espacial en los residuos para finalmente combinar ambos resultados (Li et al., 2011). Los modelos basados en árboles de regresión (Breiman et al., 2017) son particularmente útiles para abordar la primera etapa, aquella donde la variable respuesta es relacionado con las múltiples covariables que definen capas de información espacial.

Cualquiera sea la aproximación usada en la construcción del modelo para realizar la predicción espacial y posterior mapeo de la variable respuesta, para lograr buenos resultados se requiere hacer uso del criterio experto en la disciplina, realizar un adecuado preprocesamiento de los datos, identificar los predictores influyentes y validar el modelo estimado (Kuhn y Johnson, 2013).

La validación cruzada es una técnica de amplio uso para evaluar el desempeño de modelos predictivos con múltiples capas de información. La forma más adecuada de muestrear para dividir el conjunto de datos en estudio y los criterios de validación es discutida (Brenning, 2012; Efron y Tibshirani, 1997), pero como regla general se puede establecer que en cuanto menor es la proporción de datos que conforma el subconjunto de validación, menor capacidad de extrapolación tendrá el modelo ajustado y que éste no puede crecer indefinidamente ya que también se requiere buen tamaño del subconjunto de calibración para ajustar un buen modelo. El compromiso se relaja a medida que la base de datos observados es de mayor tamaño.

Una medida comúnmente usada en la evaluación de modelos de regresión lineal estimados por mínimos cuadrados es el R^2 o coeficiente de determinación. Éste expresa la bondad del ajuste del modelo respecto a los datos observados más que la capacidad predictiva que interesa evaluar cuando el modelo es usado como predictor. Otras medidas de bondad de ajuste, como los criterios de información de Akaike y BIC usados para evaluar modelos lineales de covarianza residual estimados por métodos basados en la verosimilitud o DIC usado en contextos de ajustes de modelos bayesianos, permite calificar el ajuste del modelo a los datos observados, pero no su desempeño para predecir observaciones no realizadas. La validación cruzada es, por el contrario, una estrategia transversal a distintas aproximaciones metodológicas o marcos teóricos que provee información de la capacidad predictiva del modelo, y por tanto una medida de cómo podría funcionar en la interpolación espacial. No obstante, es de resaltar que un buen modelo predictivo debería representar también un buen ajuste de los datos de entrenamiento, y por ello métricas comunes de bondad

de ajuste suelen ser usadas para una primera selección de modelos predictivos.

4.1 Regresión con errores correlacionados espacialmente vía REML

Se asume una relación lineal determinística entre la variable respuesta y las covariables espacializadas que se modela a través de coeficientes de regresión, y el proceso espacial subyacente en los datos se modela sobre los términos de error del modelo. Éstos, en lugar de considerarse independientes como en aproximaciones estadísticas clásicas, se suponen espacialmente correlacionados, *i.e.* con correlaciones entre pares de términos de error expresadas como función de la distancia que separa los sitios desde los que se obtuvieron las observaciones. La varianza residual del modelo mide la variabilidad no estructurada espacialmente y caracteriza la componente estocástica del modelo. Se ajusta un modelo directamente a los datos, y no a las semivarianzas como en la geoestadística clásica. La estructura de covariación espacial se define en función de la distancia entre las observaciones al igual que en la geoestadística, pero se estima simultáneamente con los parámetros del modelo de regresión, *i.e.* los coeficientes de regresión que interpretan como pendientes o cambios en la respuesta por unidad de cambio de la covariable de sitio.

El modelo de regresión lineal múltiple con errores correlacionados espacialmente para una variable Y asume la siguiente distribución para la i -ésima observación:

$$Y_i \sim N(E(Y), V(Y))$$

$$Y_i = \beta_0 + \sum_{j=1}^p x_{ij}\beta_j + \varepsilon_i$$

$$\varepsilon(s_i) \sim N(0, \sigma^2)$$

$$Cov(\varepsilon_i, \varepsilon_j) = \begin{cases} \sigma_s^2 + \sigma_e^2 & si \quad i = j \\ \sigma_e^2 f(d_{ij}) & si \quad i \neq j \end{cases}$$

donde β_0 es el intercepto; β_j es el vector de efectos fijos para las variables regresoras x_j ; x_{ij} la valuación de la covariante x_j en el sitio i y ε_i es el término de error que se asume distribuido Normal con media 0 y varianza σ^2 . $Cov(\varepsilon_i, \varepsilon_j)$ es la covarianza de los errores de los sitios i y j determinada a partir de una función de covarianza espacial dependiente de la distancia d_{ij} entre observaciones.

La estimación REML es la elegida ya que ha demostrado que reduce el sesgo en las estimaciones de los parámetros de covarianza (Morrell, 1998). Si el proceso de ajuste se realiza en etapas, se recomienda postular un modelo saturado o con máximo número de covariables en primera instancia y sobre los residuos de este modelo, vía REML, estimar las varianzas y covarianzas en los datos. Seleccionado el modelo para la matriz de covarianza residual, se intentará reducir la componente determinística y en este momento, cuando se evalúan los coeficientes de regresión con un modelo de varianza-covarianza adecuado, se puede utilizar estimaciones maximum likelihood clásica (ML). El método de REML, ajusta los grados de libertad de los efectos fijos (estructura de medias) antes de estimar los componentes de varianza y por ello es preferido a ML al momento de identificar la matriz de covarianza residual más apropiada para modelar los términos de error. La correlación espacial en los errores podría explicarse con una función de correlación espacial

exponencial, gaussiana, esférica o lineal, entre otras. Ellas expresan como la correlación entre dos observaciones decrece con la interdistancia (usualmente Euclídea) entre los sitios desde los cuales se obtienen y consecuentemente la selección de un modelo de correlación espacial es equivalente a la selección de un modelo teórico para el ajuste de un semivariograma experimental de la geoestadística clásica. Los modelos de correlación espacial pueden contener o no el efecto nugget, *i.e.* una estimación de la varianza a una escala más fina que la de la grilla entre observaciones. Finalmente, el modelo para la matriz de covarianza residual puede ser homocedástico (*i.e.* con varianza residual única) o heterocedástico (*i.e.* con varianza residual diferente para distintos subgrupos de datos). Comúnmente se evalúan varios modelos alternativos para la matriz de covarianza residual y se selecciona aquel para el cual el ajuste del modelo tenga un menor valor para los criterios de información penalizada como AIC y BIC.

4.2 Regresión con efectos aleatorios de sitio vía INLA

Los modelos lineales de covarianza residual pueden no ser eficientes para modelar procesos espaciales continuos que involucran matriz de covarianzas grandes y densas, por ejemplo, el modelado de correlaciones espaciales en un conjunto de datos con más de 10000 observaciones es computacionalmente intensivo y podría no ser logrado en computadores de escritorio actuales. Nuevas implementaciones de la regresión lineal múltiple para datos espaciales, desarrolladas en el marco teórico bayesiano, facilitan esta estimación

En estadística bayesiana se considera que los parámetros del modelo son variables aleatorias y se calculan distribuciones de probabilidad para los parámetros de las cuales se deriva medidas de incertidumbre (Correa Morales et al., 2018). La información previa sobre los parámetros debe resumirse en distribuciones de probabilidad denominadas distribuciones *a priori*, a partir de las cuales se estima la distribución de probabilidad *a posteriori* dadas las observaciones. Estimaciones puntuales de los parámetros de interés se pueden obtener calculando medidas resumen de la distribución *a posteriori*, como la media o el modo, y se informan juntos a intervalos de credibilidad calculados desde percentiles de la distribución *a posteriori*. La credibilidad se interpreta como la probabilidad de que el valor estimado para el parámetro pertenezca al intervalo reportado, dado los datos observados.

Los métodos de simulación por cadenas de Markov Monte Carlo (MCMC), han permitido resolver modelos complejos sin la necesidad de imponer estructuras que lo simplifiquen. Éstos han sido usados para la estimación de modelos con datos espaciales. Sin embargo, el método MCMC también conlleva desafíos computacionales. Rue et al. (2009) propusieron una alternativa para obtener la distribución *a posteriori* en contextos de datos espaciales a partir de aproximaciones basadas en el algoritmo INLA que bajo el supuesto de que la variación espacial subyacente se describe como un campo aleatorio gaussiano Markoviano simplifica las estimaciones de covarianzas espaciales en procesos espaciales continuos con una gran cantidad de observaciones. Sobre la base de las aproximaciones por INLA y la implementación de la alternativa en el lenguaje de programación R (R-INLA) se han popularizado las aplicaciones de la regresión bayesiana a datos espacial y espaciotemporales (Cameletti et al., 2013). En este contexto es posible obtener

la matriz de precisión de los efectos aleatorio de sitio utilizando aproximaciones por ecuaciones diferenciales parciales estocásticas (spde) (Lindgren y Rue, 2015). Bajo este enfoque se construye una malla de predicción con unidades de celda triangulares que cubre todo el dominio espacial para el cual se requiere la predicción, cada vértice de los triángulos representa un nodo sobre el que se predice la variable respuesta por interpolación (Blangiardo y Cameletti, 2015). Es posible trabajar con dominios espaciales de límites y bordes complejos (Bakka et al., 2018) y asignar medidas de incertidumbre de cada predicción puntual ya que lo que se obtiene del modelo bayesiano es la distribución a posteriori de los valores predichos para cada sitio más que un único valor de predicción.

Bajo este enfoque el modelo de regresión bayesiana para una variable Y asume la siguiente distribución para la i -ésima observación:

$$Y_i \sim N(\eta_i, \sigma_e^2)$$

$$\eta_i = \beta_0 + \sum_{j=1}^p x_{ij}\beta_j + \xi(s_i)$$

donde β_0 es el intercepto; β_j es el vector de efectos fijos de las variables explicativas x_j ; x_{ij} la valuación de la covariante x_j en el sitio i y $\xi(s_i)$ el efecto aleatorio de sitio que se asume una realización de un proceso gausiano markoviano latente $\xi(s_i) \sim MVN(0, \Sigma)$, siendo Σ la matriz de varianza y covarianza de los efectos de sitio definidos por la función de covariación espacial de Matérn (Matérn, 1986). En R-INLA la estimación de la inversa de Σ (matriz de precisión) se resuelve eficientemente usando **spde**.

Para estimar el modelo, es necesario obtener una represen-

tación de la estructura de dependencia desde una estructura de vecindario para datos continuos (malla). La malla se obtiene mediante triangulación de Delaunay restringida comenzando sobre la estructura base correspondiente a los vértices iniciales de las observaciones, luego se agregan o eliminan vértices adicionales para satisfacer las restricciones de la triangulación que se encuentran definidas por los siguientes parámetros: 1) `offset` define hasta qué punto se debe extender la malla hacia lo interno (es decir, dentro del área donde se pretende predecir) y hacia el exterior (es decir, fuera del área donde se pretende predecir); 2) `cutoff` define la distancia mínima entre vértices permitida; 3) `maximumedge` que refiere a la longitud máxima del borde de cada triángulo; 4) `minimumangle` o ángulo mínimo interior de cada triángulo. Construida la malla es necesario seleccionar un modelo de correlación espacial. En R-INLA con SPDE esta función se define parametrizando la función de correlación espacial de Matérn definiendo su parámetro `alpha` (variando entre 0 y 2). El valor por defecto de `alpha` es 2 y corresponde a una función de correlación espacial del tipo exponencial. La matriz de covarianza de los efectos aleatorios es una matriz rala dado el proceso espacial que se supone, y sus valores son aproximados por suavizado vía `spde`.

4.3 Regresión vía modelos basados en árbol

El término aprendizaje de máquina o aprendizaje automático corresponde a una rama de la inteligencia artificial que hace referencia a algoritmos o procedimientos de cálculo basados en intenso proceso computacional que “aprenden” de los datos intentando minimizar la intervención huma-

na. Algunos se basan en particiones binarias recurrentes de los datos y evaluaciones de éstas hasta identificar la mejor para explicar variabilidad en la variable respuesta. Los árboles de clasificación y regresión (algoritmos CART) (Breiman, 2001) se utilizan con fines predictivos y son particularmente útiles para interpretar relaciones (no necesariamente lineales) en contextos de regresión múltiple con variables explicativas correlacionadas. Estos algoritmos pueden ser empoderados mediante métodos de remuestreo que se usan para obtener muestras aleatorias a partir de los datos observados o muestra original, derivar modelos para cada muestra y ensamblar los resultados para optimizar la predicción. Otros algoritmos de aprendizaje automático basados en árboles son la regresiones por bosques aleatorios (RF por el término en inglés *Random Forest*) y los árboles de regresión generalizados (GBR por el término en inglés *Generalized Boosting Regression*) (Efron y Hastie, 2016). Los algoritmos basados en árboles engloban así a un conjunto de técnicas supervisadas no paramétricas (*i.e.* sin supuestos distribucionales) para segmentar el espacio de los predictores en regiones simples con máxima diferencia en la variable respuesta. Es necesario tener cuidados particulares al momento de estimar el árbol que será usados como modelo predictivo ya que puede existir sobreajuste, *i.e.* construirse un modelo sólo útil para los datos disponibles cuyas predicciones pueden cambiar con pequeños cambios en el conjunto de datos observados.

Si bien estos algoritmos se han utilizado para datos espaciales (Kanevski et al., 2009), es raro que se modele la estructura espacial. Una propuesta para incorporar la correlación espacial en los datos es utilizar las coordenadas geográficas o las matrices de distancias entre observaciones covariables en la construcción del modelo (Pejović et al., 2018). Otra propuesta, es modelar el residuo remanente

del ajuste del algoritmo de aprendizaje automático con una función de autocorrelación espacial (Li et al., 2011) y finalmente combinar los resultados de la predicción determinística dada por el árbol y la predicción espacial obtenida mediante kriging de los residuos.

4.3.1 Bosques aleatorios

El método de bosques aleatorios o *Random Forest* (RF) es una modificación del proceso de ensamblaje de varios árboles (*Bagging*) donde se ajustan múltiples árboles desde cada muestra obtenida por remuestreo formando un “bosque”. En cada nueva predicción, todos los árboles que forman el “bosque” participan aportando su predicción. Como valor final, se toma la media de todas las predicciones en el caso de variables respuesta continuas. El método RF a diferencia de *Bagging* realiza una selección aleatoria de m predictores antes de evaluar cada división. Si $m = p$ los resultados de RF y *Bagging* son equivalentes. Este método trabaja bien con grandes bases de datos presentando mayor facilidad en la implementación y baja tendencia al sobreajuste. Para implementar RF es necesario optimizar el parámetro m , no obstante, existe la recomendación de usar $m = \frac{p}{3}$ por defecto para regresión.

Para contemplar la estructura espacial es posible combinar los resultados de RF con una interpolación geoestadística basada en kriging (Li et al., 2011). La predicción de los residuos se complementa con la predicción de RF de manera aditiva. Es recomendable que el ajuste del modelo espacial para realizar kriging se logre con un subconjunto de datos de entrenamiento, diferente al grupo de validación, para evitar sobreajustes.

4.3.2 Árboles de regresión generalizados

El método *Boosting* es otro método de ensamblaje que consiste en ajustar secuencialmente modelos sencillos, de manera que cada modelo aprende de los errores del anterior. Los algoritmos de *Boosting* trabajan minimizando una función de pérdida (deviance) para maximizar la proporción de varianza que explica el modelo. Los árboles de regresión generalizados, conocido en inglés como *Generalized Boosted Regression Trees* (GBR) particularmente utiliza *Boosting* para ensamblar los árboles obtenidos de múltiples muestras obtenida mediante remuestreo de la muestra original. El algoritmo ajusta árboles de regresión a los datos de entrenamiento de manera iterativa. El primer árbol que se ajusta es aquel que, según la complejidad del árbol seleccionada, minimiza la deviance. El siguiente árbol se ajusta a los residuos del primer árbol. Luego, se vuelven a realizar predicciones para las observaciones que tienen en cuenta los dos árboles. En cada uno de los pasos siguientes se ajusta un nuevo árbol sobre los residuos de la combinación de los árboles anteriores. El procedimiento es parametrizado por varias constantes que es necesario identificar probando numerosas o todas las combinaciones posibles de parámetros ya que estos son dependientes entre sí.

Una vez encontrada la combinación optima de parámetros se ajusta GBR en el grupo de entrenamiento. Luego, a partir de los residuos del modelo se ajusta un kriging y se guardan los parámetros de la función de semivarianza ajustados. Finalmente se utiliza el modelo GBR construido para predecir los datos en el grupo de validación adicionando a los resultados obtenidos desde el árbol la predicción de los residuos de cada sitio.



Parte II

Análisis de datos a escala fina



1

Implementación con R

A continuación, se muestran los procedimientos para realizar el análisis exploratorio utilizando datos de rendimiento de trigo (**datosRinde.txt**) con n=9810 observaciones que fueron recolectado con un monitor de rendimiento en un lote agrícola de 84 ha. La base de datos cuenta con tres columnas, las primeras dos identifican las coordenadas espaciales bidimensionales (x e y) y la tercera corresponde al rendimiento expresado en $t\ ha^{-1}$. Para cargar una base de datos puede utilizarse la función `read.table()`. Esta función permite abrir distintos tipos de archivos entre ellos aquellos de extensión .txt. El siguiente ejemplo crea un objeto llamado “datos” de clase `data.frame` cuya información es leída desde un archivo de texto (.txt). El argumento `header=TRUE` indica que la primera fila de los datos contiene los nombres de las columnas.

Para seguir la ilustración, cargar los paquetes específicos de R que albergan las funciones que se utilizarán para el análisis.

```
library(sp)
library(sf)
library(tmap)
library(e1071)
library(spdep)
library(gstat)
library(caret)
```

```
library(geoR)
library(nlme)
library(ade4)
```

```
datos <- read.table("datos/datosRinde.txt",
                     header = TRUE)
```

Para visualizar las filas de un objeto, basta con escribir su nombre o, en el caso de objetos de clase data.frame es posible utilizar la función `head()`. En el panel de resultados, se despliega el contenido del objeto.

```
head(datos)
#> x y Rinde
#> 1 -59,1 -37,9 0,348
#> 2 -59,1 -37,9 0,360
#> 3 -59,1 -37,9 0,367
#> 4 -59,1 -37,9 0,001
#> 5 -59,1 -37,9 0,382
#> [ reached 'max'
#> / getOption("max.print") --
omitted 1 rows ]
```

1.1 Conversión de coordenadas espaciales

Dado que la función utilizada para la generación del objeto `datos` no es específica para datos espaciales, es necesario transformar este objeto. Esta transformación permite correr funciones estadísticas que solo trabajan sobre objetos de datos espaciales (clase `SpatialPointsDataFrame` o `sf`).

Para ello puede utilizarse funciones de los paquetes `sp` o `sf`.

La función `coordinates()` del paquete `sp` transforma el `data.frame` en un objeto de datos espaciales e indica al software que las columnas “x” e “y” son coordenadas espaciales. A continuación, se genera un nuevo objeto espacial denominado `datos_sp` que contiene esta información.

```
datos_sp <- datos
coordinates(datos_sp) <- ~x+y
```

Luego de indicar las columnas que pertenecen a la información de las coordenadas, es necesario especificar el sistema de referencia. La función `CRS()`, Coordinate Reference System, tiene una variedad de argumentos que permiten hacer referencia a diferentes sistemas de proyecciones y asociar esta información al objeto con el que se está trabajando. La proyección *longlat* es utilizada en esta aplicación. Esta proyección solo admite valores de longitud mayores a -180 y menores a 180 y valores de latitudes que se encuentren entre -90 y 90. El *datum* especificado será WGS84.

```
proj4string(datos_sp) <-
CRS("+proj=longlat +datum=WGS84")
```

Una alternativa para hacer referencia a un sistema de coordenadas particular es utilizar su código EPSG. La EPSG es una organización científica vinculada a la industria del petróleo europea (<http://www.epsg-registry.org/>) la cual desarrolló un repositorio que contiene información sobre sistemas de referencia, proyecciones cartográficas y elipsoides de todo el mundo. La función `make_EPSG()` del paquete `rgdal` permite visualizar todos ellos. Para realizar este procedimiento sólo se modifican los argumentos de

la función CRS. En este caso el código EPSG para la proyección *longlat* es 4326.

```
proj4string(datos_sp) <- CRS("+init=epsg:4326")
```

La función `spTransform()` permite convertir las coordenadas. Cuando se realiza la transformación del sistema de proyección geográfico a cartesiano, es necesario indicar a cuál zona o faja pertenecen los datos bajo análisis, para este caso la zona es 21. Al igual que en la sentencia anterior, se debe indicar el *datum* y *elipsoide* que en ambos casos corresponde a WGS84.

```
datos_sp <-
  spTransform(datos_sp,
              CRS("+proj=utm +zone=21 +south
+ellps=WGS84 +datum=WGS84"))
```

Esta transformación también puede realizarse en base al código EPSG en este caso para el sistema de coordenadas UTM faja 21 sur el código es 32721.

```
datos_sp <-
  spTransform(datos_sp,
              CRS("+init=epsg:32721"))
```

Otra opción para generar este tipo de datos espaciales es utilizando el paquete `sf`. Para ello se usan las funciones `st_as_sf()` la cual convierte el `data.frame` en un objeto `sf` y se le asigna el sistema de coordenadas de referencia, utilizando el argumento `crs = 4326`. La función `st_transform()` transforma el sistema de coordenadas.

```
datos_sf <- st_as_sf(datos,
                      coords = c("x", "y"),
                      crs = 4326)
st_crs(datos_sf) <- 4326
datos_sf <- st_transform(datos_sf, crs = 32721)
```

Si se visualizan las primeras filas del objeto, se puede observar, entre otras cosas, la clase de objeto (**sf**), su geometría (puntos) y la información referida a su sistema de coordenadas.

```
head(datos)
#> #> x y Rinde
#> 1 -59,1 -37,9 0,348
#> 2 -59,1 -37,9 0,360
#> 3 -59,1 -37,9 0,367
#> 4 -59,1 -37,9 0,001
#> 5 -59,1 -37,9 0,382
#> [ reached 'max'
#> / getOption("max.print") --
omitted 1 rows ]
```

Esta transformación también puede realizarse en base al código EPSG en este caso para el sistema de coordenadas UTM faja 21 sur el código es 32721.

```
datos_sp <- spTransform(datos_sp,
                        CRS("+init=epsg:32721"))
```

Otra opción para generar este tipo de datos espaciales es utilizando el paquete **sf**. Para ello se usan las funciones **st_as_sf()** (convierte el **data.frame** en un objeto de clase **sf**), **st_crs()** (asigna el sistema de coordenadas de

referencia) y `st_transform()` (transforma el sistema de coordenadas).

```
datos_sf <- st_as_sf(datos, coords=c("x","y"),
                      crs=4326)
datos_sf <- st_transform(datos_sf, crs=32721)
```

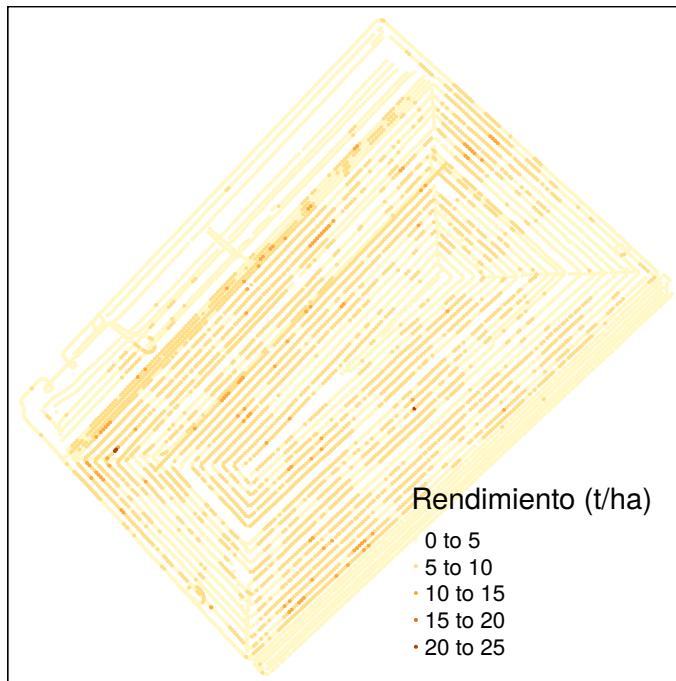
Si se visualizan las primeras filas del objeto, se puede observar entre otras cosas la clase de objeto (`sf`), su geometría (puntos) y la información referida a su sistema de coordenadas.

```
head(datos_sf)
#> Simple feature collection with 6 features
and 1 field
#> geometry type: POINT
#> dimension: XY
#> bbox: xmin: 312000 ymin: 5800000 xmax:
313000 ymax: 5800000
#> epsg (SRID): 32721
#> proj4string:
+ proj=utm
+ zone=21
+ south
+datum=WGS84 +units=m +no_defs
#> First 3 features:
#> Rinde geometry
#> 1 0,348 POINT (313088 5800921)
#> 2 0,360 POINT (311983 5800811)
#> 3 0,367 POINT (312933 5800910)
```

La visualización de la información georreferenciada permite un rápido diagnóstico de la distribución de los datos y de los valores observados. Trabajando con datos de clase `sf`

es posible realizar una rápida visualización de los datos espaciales. El paquete `tmap` permite realizar mapas temáticos. Para comenzar a realizar un mapa debe especificarse los datos con los que se desea trabajar mediante la función `tm_shape()` y luego se suman elementos creados con funciones de este paquete utilizando el símbolo `+`.

```
tm_shape(datos_sf) +  
  tm_dots("Rinde", title="Rendimiento (t/ha)")
```



1.2 Eliminación de *outliers* e *inliers*

En un `data.frame`, una forma sencilla para obtener medidas resumen de una variable es con la función `summary()`.

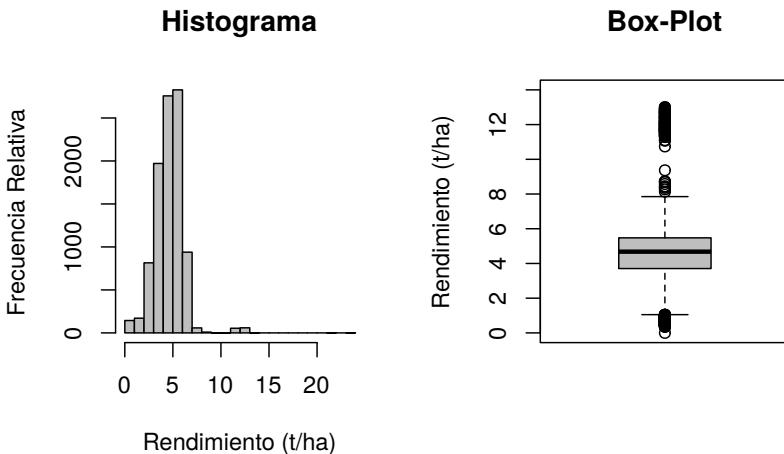
Se utiliza \$ para hacer referencia a una columna particular dentro de un objeto.

```
summary(datos_sf$Rinde)
#> Min. 1st Qu. Median Mean 3rd Qu. Max.
#> 0,00 3,71 4,68 4,62 5,48 24,00
```

Las funciones `hist()` y `boxplot()` realizan gráficos de histogramas y box-plots, respectivamente. Sus múltiples argumentos permiten la edición de cada gráfico. La función `par()` permite dividir la ventana gráfica de R, en el siguiente ejemplo se divide la ventana gráfica de R en dos columnas y una fila.

```
par(mfrow = c(1, 2))
hist(
  datos_sf$Rinde,
  col = 'grey',
  nclass = 20,
  main = "Histograma",
  ylab = 'Frecuencia Relativa',
  xlab = 'Rendimiento (t/ha)'
)

boxplot(
  datos$Rinde,
  col = 'grey',
  ylab = 'Rendimiento (t/ha)',
  main = "Box-Plot",
  ylim = c(0, 14)
)
```



a función `skewness()` del paquete `e1071` permite calcular el coeficiente de asimetría. Existen 3 fórmulas para su cálculo (por defecto usa el tipo 3). Para más información, se puede utilizar `help(skewness)`.

```
skewness(datos_sf$Rinde)
#> [1] 1,45
```

En el histograma se observa asimetría derecha en la distribución de los datos. La asimetría también puede advertirse con los estadísticos con el coeficiente de asimetría el cual es de 1,45. En el gráfico box-plot se observan valores extremos de la variable que se encuentran principalmente por encima de la media $+ 3 \text{ SD}$.

Las siguientes instrucciones calculan y crean objetos para la media, el DE y los límites superior ($\text{media} + 3\text{DE}$) e inferior ($\text{media} - 3\text{DE}$) con los que pueden detectarse los *outliers* y eliminar estos valores.

```
Media <- mean(datos_sf$Rinde)
DE <- sd(datos_sf$Rinde)
```

```
LI <- Media-3*DE
LS <- Media+3*DE
```

Los símbolos `|` y `&` son operadores lógicos que significan *or* y *and*, respectivamente. Las siguientes instrucciones generan dos objetos. El objeto `datos_1` es la base depurada, es decir sin *outliers*, mientras que el objeto `outliers` presenta los datos que han sido eliminados en esta etapa.

```
datos_1 <-
  subset(datos_sf,
         datos_sf$Rinde < LS &
         datos_sf$Rinde > LI)
outliers <-
  subset(datos_sf,
         datos_sf$Rinde > LS | 
         datos_sf$Rinde < LI)
```

Para ver el impacto de la eliminación de *outliers* pueden obtenerse nuevamente las medidas resumen, coeficiente de asimetría, histograma y box-plot.

```
summary(datos_1$Rinde)
#> Min. 1st Qu. Median Mean 3rd Qu. Max.
#> 0,35 3,69 4,66 4,52 5,45 8,73
skewness(datos_1$Rinde)
#> [1] -0,536
```

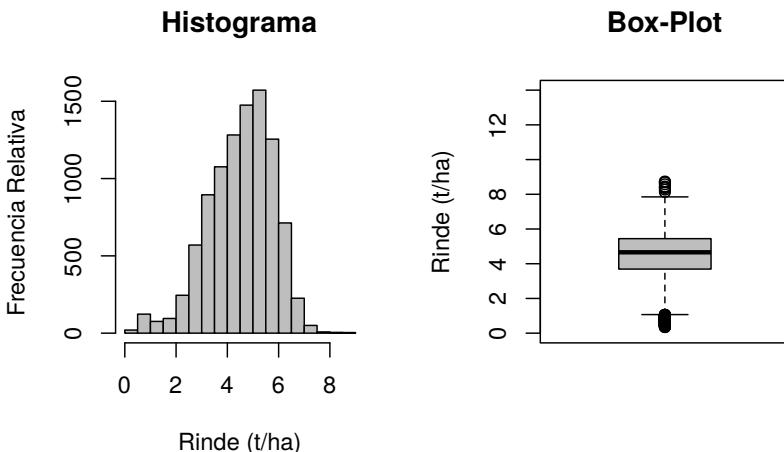
```
par(mfrow = c(1, 2))
hist(
  datos_1$Rinde,
  col = 'grey',
```

```

nclass = 20,
main = "Histograma",
ylab = 'Frecuencia Relativa',
xlab = 'Rinde (t/ha)'

)
boxplot(
  datos_1$Rinde,
  col = 'grey',
  ylab = 'Rinde (t/ha)',
  main = "Box-Plot",
  ylim = c(0, 14)
)

```



Las medidas resumen muestran un cambio principalmente a nivel de los valores máximos, pasando de 24 a 8,734 $t\ ha^{-1}$. El coeficiente de asimetría presenta un valor de -0,54 que se aproxima a valores recomendados para el análisis geoestadístico (-1 a 1). En la figura anterior se presenta el histograma y box-plot luego de la eliminación de los *outliers*. Para la variable en análisis, se eliminaron durante la depuración 120 casos que representan un 1,2% del total de sitios ($n=9810$) con mediciones. Puede observarse una

mejora en la simetría de la distribución de la variable y una marcada disminución de valores extremos.

La identificación y eliminación de *inliers* requiere de la creación de una matriz de ponderación espacial. La función `dnearest()` se utiliza para identificar el vecindario de cada sitio. Para ello es necesario calcular la distancia espacial entre los puntos para lo cual se usa la sintaxis `$geom` que permite acceder a las coordenadas del objeto `datos`. En este ejemplo, se consideran datos vecinos a aquellos que se encuentran a una distancia Euclídea de 0 a 15 m. La función `nb2listw()` transforma el objeto `vecindarios` que contiene las distancias a una matriz de pesos estandarizados por filas (`style = "W"`). Este objeto es denominado `pesos_sp`. Para generar la matriz de pesos espaciales es necesario que todos los puntos tengan al menos un vecino, caso contrario la función `nb2listw()` genera un error advirtiendo este hecho. Para superar el inconveniente es posible probar con distancias mayores hasta lograr que todos los puntos tengan al menos un vecino. Hay que tener la precaución de no generar un excesivo solapamiento entre los vecindarios. Otra opción es incorporar el argumento `zero.policy=T` dentro de la función `nb2listw()` que permite generar la matriz de pesos espaciales con observaciones que no presentan dato/s vecino/s. El mismo argumento debe agregarse luego cuando se calcula el índice de Moran local o gráfico de Moran (funciones `localmoran()` y `moran.plot()`, respectivamente). Las frecuencias del número de puntos vecinos para cada observación puede obtenerse mediante la función `summary()` del objeto `vecindarios`. En el ejemplo se observa que 3113 puntos tienen vecindarios conformados con 4 datos. Mientras que sólo un dato presenta 18 observaciones consideradas como vecinas.

```

vecindarios <- dnearneigh(datos_1$geom,
                           d1 = 0, d2 = 15)
summary(vecindarios)
#> Neighbour list object:
#> Number of regions: 9690
#> Number of nonzero links: 62578
#> Percentage nonzero weights: 0,0666
#> Average number of links: 6,46
#> Link number distribution:
#>
#> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
#> 5 83 302 3113 1292 999 893 870 393 457 547
#> 473 171 43 31 13
#> [ reached getOption("max.print") -- omitted
#> 2 entries ]
#> 5 least connected regions:
#> 3033 7885 9153 9598 9681 with 1 link
#> 1 most connected region:
#> 869 with 18 links

```

```

pesos_sp <- nb2listw(vecindarios,
                      style = "W")

```

La función `localmoran()` calcula el índice local de Moran que permite identificar potenciales *outliers* espaciales. También permite el ajuste de los *valores-p* por el criterio de Bonferroni. La información referida al valor del índice local de Moran de cada punto se encuentra en la columna `Ii` mientras que su significancia estadística en la columna `Pr(z < 0)`.

```

moranl <-
  localmoran(datos_1$Rinde,

```

```

    pesos_sp,
    p.adjust.method = "bonferroni",
    alternative = "less")
head(moranl)
#> Ii E.Ii Var.Ii Z.Ii Pr(z < 0)
#> 1 5,32 -0,000103 0,0832 18,44 1
#> 2 6,83 -0,000103 0,1110 20,49 1
#> 3 5,83 -0,000103 0,0832 20,21 1
#> [ reached getOption("max.print") -- omitted
3 rows ]

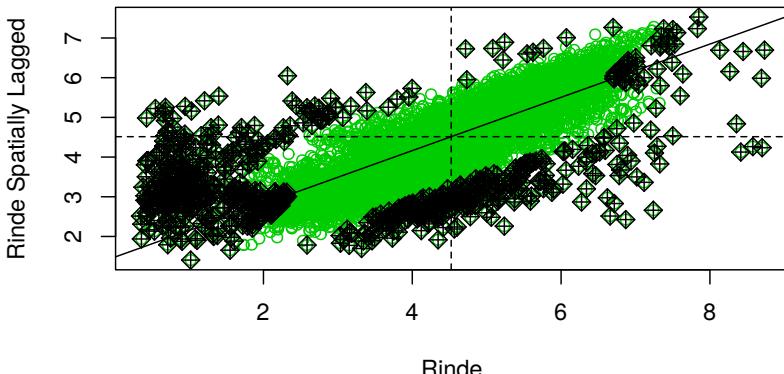
```

El gráfico de Moran permite la identificación de puntos influyentes. La función `moran.plot()` construye el gráfico y devuelve los estadísticos de diagnóstico para cada punto. En el eje horizontal se expresan los valores de la variable rendimiento mientras que en el vertical se representa el retardo espacial de la variable. Adicionalmente, se ajusta y añade a este diagrama modelos de regresión lineal y estadísticos de influencia para identificar sitios con datos raros. Un punto se determina como influyente si al menos uno de los estadísticos así lo considera. En la figura siguiente los puntos negros con forma romboidal fueron identificados como influyentes y se los considera como *inliers*.

```

moranp <-
  moran.plot(
    datos_1$Rinde,
    col = 3,
    pesos_sp,
    labels = F,
    quiet = T,
    xlab = "Rinde",
    ylab = "Rinde Spatially Lagged"
  )

```



Para visualizar en una tabla los puntos potencialmente influyentes y sus estadísticos de diagnóstico puede imprimir el objeto `moranp`. Datos con * en la columna `inf` se los considera como influyente y por lo tanto posible *outlier* espacial.

```
summary(moranp)
```

Desde el objeto `moranp` puede extraerse una matriz de valores lógicos (verdadero/falso) para los estadísticos diagnóstico que identifican un punto como influyente o no.

```
influ <- moranp$is.inf
head(influ)
#> dfb.1_ dfb.x dffit cov.r cook.d hat
#> 1 FALSE FALSE TRUE TRUE FALSE TRUE
#> 2 FALSE FALSE FALSE TRUE FALSE TRUE
#> [ reached getOption("max.print") -- omitted
4 rows ]
```

En la siguiente sentencia se adiciona al objeto `datos_1` los valores de los objetos `moran1` e `influ`, que tienen información para detectar los *outliers* espaciales detectados con el índice de Moran local y gráfico de Moran, respectivamente.

```
datos_1 <- cbind(datos_1, moranl, influ)
```

Posteriormente procedemos a eliminar los datos con Índice de Moran Local negativo y estadísticamente significativos ($p<0,05$). La función `subset()` selecciona datos que cumplen con cierta condición lógica. El operador lógico *or* que indica que extraiga los datos que cumplen con alguna de las dos condiciones. El nuevo objeto es denominado `datos_2`. Además, se crea un nuevo objeto que tendrá los datos que han sido eliminados en este proceso (`inliers_ml`).

```
datos_2 <-
  subset(datos_1,
         datos_1[["Ii"]] >= 0 | 
         datos_1[["Pr.z...0."]] > 0.05)
inliers_ml <-
  subset(datos_1,
         datos_1[["Ii"]] < 0 &
         datos_1[["Pr.z...0."]] < 0.05)
```

Existen varias formas de eliminar las filas de la tabla que fueron identificadas como *inliers* con la función `moran.plot()`. Una alternativa es usando sentencias lógicas con los operadores `==` y `&` que significan igualdad lógica y *and* respectivamente. Como en el caso anterior se genera una nueva base (`datos_3`) la cual no tendrá los datos considerados como *outliers* y *outliers* espaciales. También se genera una nueva base que tendrá solo los datos considerados aquí como *outliers* espaciales (`inliers_mp`).

```
datos_3 <-
  datos_2[datos_2$dfb.1_ == FALSE &
          datos_2$dfb.x == FALSE &
```

```
    datos_2$dffit == FALSE &
    datos_2$cov.r == FALSE &
    datos_2$cook.d == FALSE &
    datos_2$hat == FALSE, ]
```

La sentencia anterior instruye al software para que cree un objeto llamado `datos_3` a partir de las filas del objeto `datos_2` cuyas columnas `dfb.1`, `dfb.x`, `dffit`, `cov.r`, `cook.d` y `hat` son simultáneamente iguales a FALSE.

```
inliers_mp <-
  datos_2[datos_2$dfb.1 != FALSE |
           datos_2$dfb.x != FALSE |
           datos_2$dffit != FALSE |
           datos_2$cov.r != FALSE |
           datos_2$cook.d != FALSE |
           datos_2$hat != FALSE, ]
```

Luego de identificar y eliminar los *inliers* detectados con el índice de Moran y posteriormente con el gráfico de Moran, la nueva base de datos presenta 9009 casos, es decir, se eliminaron 681 casos (7 % de los datos) respecto a la base sin *outliers*.

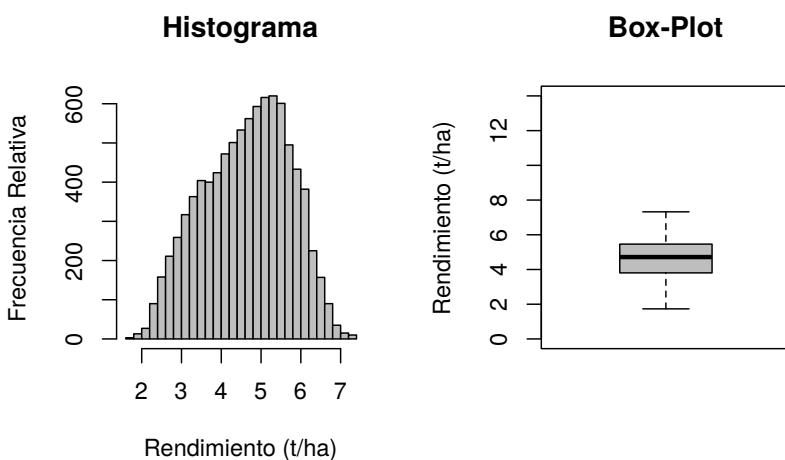
Los estadísticos descriptivos de los datos depurados muestran una mejora en el coeficiente de asimetría (-0,19) lo cual se refleja en el histograma y box-plot. Este último también muestra la ausencia de valores extremos.

```
summary(datos_3$Rinde)
#> Min. 1st Qu. Median Mean 3rd Qu. Max.
#> 1,73 3,81 4,72 4,63 5,46 7,32
skewness(datos_3$Rinde)
#> [1] -0,197
```

```

par(mfrow = c(1, 2))
hist(
  datos_3$Rinde,
  col = 'grey',
  nclass = 20,
  main = "Histograma",
  ylab = 'Frecuencia Relativa',
  xlab = 'Rendimiento (t/ha)'
)
boxplot(
  datos_3$Rinde,
  col = 'grey',
  ylab = 'Rendimiento (t/ha)',
  main = "Box-Plot",
  ylim = c(0, 14)
)

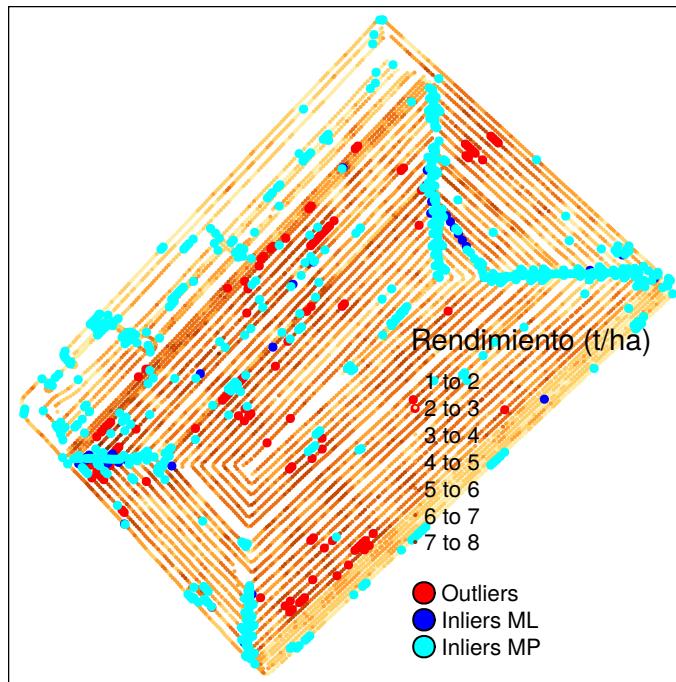
```



Las siguientes líneas muestran la visualización conjunta de los datos originales y aquellos detectados como *outliers* y

outliers espaciales. En este último se diferencian los detectados por el índice de Moran local (*Inliers ML*) respecto a los identificados por el gráfico de Moran (*Inliers MP*).

```
tmap_mode("plot")
#> tmap mode set to plotting
tm_shape(datos_3) +
  tm_dots("Rinde", title="Rendimiento (t/ha)") +
  tm_shape(outliers) +
  tm_dots(col = "red", size = 0.1) +
  tm_shape(inliers_ml) +
  tm_dots(col = "blue", size = 0.1) +
  tm_shape(inliers_mp) +
  tm_dots("cyan", size = 0.1) +
  tm_add_legend("symbol",
                col = c("red", "blue", "cyan"),
                labels = c("Outliers",
                          "Inliers ML",
                          "Inliers MP"))
```



Finalmente las líneas siguientes permiten la exportación de los datos depurados en diferentes formatos (.csv, .shp, .gpkg). para ello se utiliza la función `st_write()`. Previo a ello se selecciona solo la variable Rinde del objeto `datos_3`.

```
datos_3 <- datos_3[, c("Rinde")]
st_write(datos_3,
          "base_depurada.csv",
          layer_options = "GEOMETRY=AS_XY",
          delete_layer = T)
st_write(datos_3,
          "base_depurada.shp",
          delete_layer = T)
st_write(datos_3,
          "base_depurada.gpkg",
          delete_layer = T)
```

1.3 Detección de tendencias espaciales

Para evaluar las tendencia de la media del rendimiento con las coordenadas geográficas primero se incorpora al `data.frame` del objeto `sf` las coordenadas. De esta forma el objeto `datos_3` presenta la variable Rinde, su geometría y las coordenadas x e y.

```
datos_3$x <- st_coordinates(datos_3) [,1]
datos_3$y <- st_coordinates(datos_3) [,2]
head(datos_3)
#> Simple feature collection with 6 features
and 3 fields
#> geometry type: POINT
#> dimension: XY
#> bbox: xmin: 312000 ymin: 5800000 xmax:
313000 ymax: 5800000
#> epsg (SRID): 32721
#> proj4string:
+ proj=utm
+ zone=21
+ south
+datum=WGS84 +units=m +no_defs
#> First 3 features:
#> Rinde geometry x y
#> 256 1,73 POINT (311987 5800811) 311987
5800811
#> 257 1,73 POINT (313167 5800905) 313167
5800905
#> 263 1,78 POINT (312511 5800721) 312511
5800721
```

Para visualizar tendencias espaciales graficamos la variable

en estudio en función de las coordenadas. Si se desea desplegar los gráficos para la coordenada x e y en una misma ventana gráfica, se puede particionar la ventana en una fila y dos columnas utilizando la siguiente función:

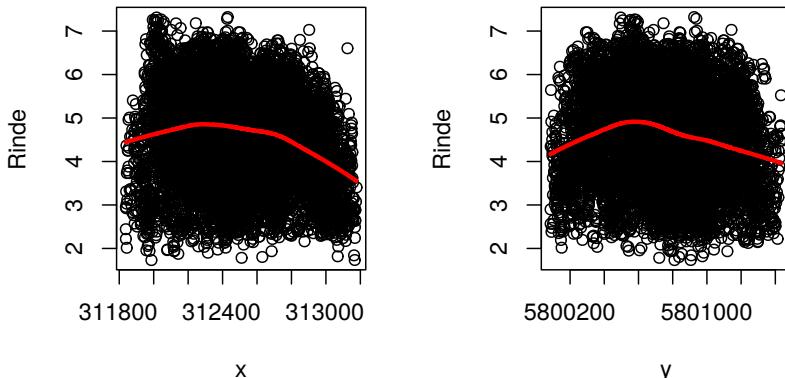
```
par(mfrow=c(1,2))
```

La función `plot()` permite realizar gráficos de dispersión. Además, puede adicionarse una línea de suavizado *lowess* con la función `lines()`. Esta última, realiza el ajuste sobre una ventana gráfica preexistente.

```
par(mfrow = c(1, 2))

with(datos_3, {
  plot(Rinde ~ x)
  lines(lowess(Rinde ~ x),
        col = "red", lwd = 3)

})
with(datos_3, {
  plot(Rinde ~ y)
  lines(lowess(Rinde ~ y),
        col = "red", lwd = 3)
})
```



Mediante un modelo lineal de regresión, puede ajustarse la tendencia entre la variable en estudio y las coordenadas. Si la tendencia lineal resulta significativa, debería descontarse trabajando con los residuos del modelo, que se obtienen con la función `residuals()`.

```
regresion <- lm(formula = Rinde ~ 1 + x + y ,
                  data = datos_3,
                  na.action = na.omit)
```

La siguiente línea despliega una tabla resumen del modelo:

```
summary(regresion)
#>
#> Call:
#> lm(formula = Rinde ~ 1 + x + y, data =
datos_3, na.action = na.omit)
#>
#> Residuals:
#> Min 1Q Median 3Q Max
#> -3,165 -0,811 0,102 0,819 2,736
#>
#> Coefficients:
```

```
#> Estimate Std. Error t value Pr(>|t|)  
#> (Intercept) 1,84e+03 2,34e+02 7,85 4,6e-15  
***  
#> x -5,56e-04 4,10e-05 -13,54 < 2e-16 ***  
#> y -2,86e-04 4,10e-05 -6,98 3,2e-12 ***  
#> ---  
#> Signif. codes: 0 '***'  
     0,001 '**'  
     0,01 '*'  
0,05 '.'  
     0,1 '  
     '  
     1  
#>  
#> Residual standard error: 1,07 on 9006  
degrees of freedom  
#> Multiple R-squared: 0,0348, Adjusted  
R-squared: 0,0346  
#> F-statistic: 162 on 2 and 9006 DF, p-value:  
<2e-16
```

En este caso, los gráficos exploratorios no marcan una tendencia marcada con las coordenadas. Aún, cuando los valores-p del modelo de regresión son significativos ($p<0,05$), se decidió trabajar con la variable original debido a que el coeficiente de determinación del modelo acusa un ajuste pobre (0,035).

1.4 Cálculo del índice de Moran

Para la conformación de la matriz de ponderadores espaciales se definieron los vecindarios de cada sitio mediante una red de conexión construida en base a la distancia euclídea. Se consideraron sitios vecinos a aquellos contiguos ubicados hasta 15 m de distancia. El procedimiento es similar al empleado para el cálculo de índice de Moran local. En este caso se agrega el argumento `zero.policy=T` dentro de la función `nb2listw()` y `moran.mc()`. Esto permite que se genera la matriz de pesos espaciales sin la restricción de que todos los puntos tengan al menos un dato vecino.

```
vecindarios <- dnearneigh(datos_3$geom,  
                           0, 15)  
pesos_sp <- nb2listw(vecindarios,  
                      style = "W",  
                      zero.policy = TRUE)
```

Para realizar el cálculo del Índice de Moran y determinar su significancia estadística mediante simulación Monte Carlo, se utiliza `moran.mc()`. Se debe especificar la variable en estudio, la lista con los pesos de las ponderaciones espaciales y el número de simulaciones.

```
i.moran <-  
  moran.mc(  
    datos_3$Rinde,  
    listw = pesos_sp,  
    nsim = 999,  
    zero.policy = T  
)
```

```
i.moran
#>
#> Monte-Carlo simulation of Moran I
#>
#> data: datos_3$Rinde
#> weights: pesos_sp
#> number of simulations + 1: 1000
#>
#> statistic = 0,8, observed rank = 1000,
p-value = 0,001
#> alternative hypothesis: greater
```

Estos resultados permiten concluir que existe autocorrelación espacial positiva (0,78196) y que esta es estadísticamente significativa ($p=0,001$).

1.5 Análisis basado en semivariogramas

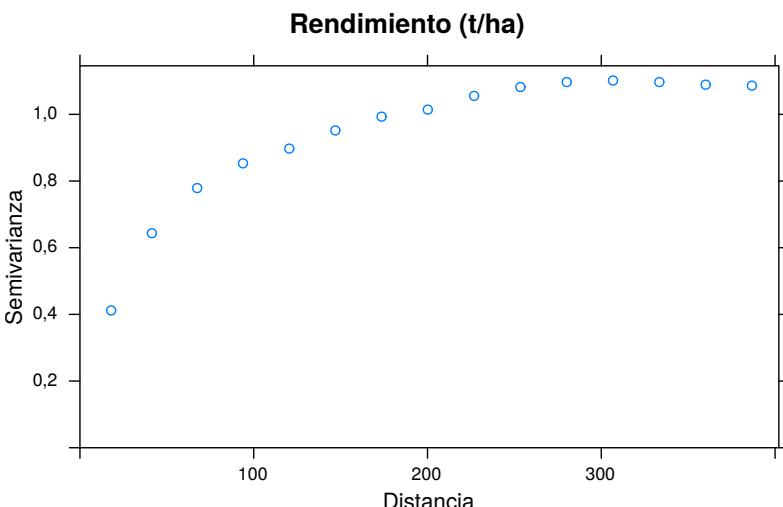
Las semivariogramas empíricos pueden obtenerse usando la función `variogram()` del paquete `gstat`. Esta tiene múltiples argumentos, entre ellos una fórmula, un objeto de datos espaciales y la distancia hasta la cual los pares de puntos son incluidos en la estimación de semivarianza (`cutoff`). Dado que el objeto a tratar (`datos_3`) es de clase `sf`, no es necesario realizar su transformación a un objeto del tipo espacial. Utilizando la función `plot()` se visualiza el semivariograma empírico ajustado.

```
semiv_emp <- variogram(Rinde ~ 1,
                         datos_3,
```

```
cutoff = 400)

head(semiv_emp)
#> np dist gamma dir.hor dir.ver id
#> 1 97403 18,0 0,412 0 0 var1
#> 2 281953 41,4 0,643 0 0 var1
#> [ reached 'max'
/getOption("max.print") --
omitted 4 rows ]
```

```
plot(semiv_emp,
      main = "Rendimiento (t/ha)",
      xlab = "Distancia",
      ylab = "Semivarianza")
```



En el caso anterior la fórmula utilizada (`Rinde~1`) asume que el proceso es estacionario. Si se requiere adicionar una tendencia sea reemplaza el carácter 1 por el nombre de la covariable. Por ejemplo, si existe una tendencia dada por las coordenadas la fórmula se escribe como `Rinde~x+y`.

Es posible solo colocar una de las coordenadas o incluir también términos polinómicos para las mismas. También se pueden usar otras covariables distintas a las coordenadas.

Por defecto para el cálculo del semivariograma empírico la función `variogram()` utiliza el estimador de los momentos de Matheron. Para emplear el estimador robusto de Cressie-Hawkins se adiciona en la función el argumento `cresice=TRUE`. Por defecto la función emplea para el cálculo del semivariograma un tercio de la diagonal del “cuadro” que contiene las observaciones. Con el argumento `cutoff` se puede cambiar esta distancia. El argumento `width` permite cambiar el ancho de los intervalos de distancia en los que se agrupan los pares de puntos de datos para las estimaciones de semivarianza. Por defecto se calcula como `cutoff/15`. El argumento `alpha` permite el cálculo de los semivariogramas en distintas direcciones en el plano (`x`, `y`), tomando valores en grados positivos en sentido horario desde y (Norte). Para `alpha = 0` la dirección es Norte y para `alpha = 90` la dirección es Este. Esto es útil para evaluar la presencia de anisotropía. Usualmente se suelen calcular los semivariogramas direccionales para 45° , 90° , 135° y 180° . Otras opciones pueden encontrarse mediante la función `help()`.

A continuación se ajusta un modelo de semivariograma teórico sobre el semivariograma empírico usando las funciones `fit.variogram()` y `yvgm()`. Esta última ajusta el modelo teórico, sus argumentos indican el tipo de modelo a ajustar y los parámetros de ajuste (*partial sill*, rango y efecto *nugget*). Estos parámetros iniciales son de referencia y pueden obtenerse a partir del semivariograma empírico. Cambiar los parámetros modifica la suma de cuadrados del error (SCE).

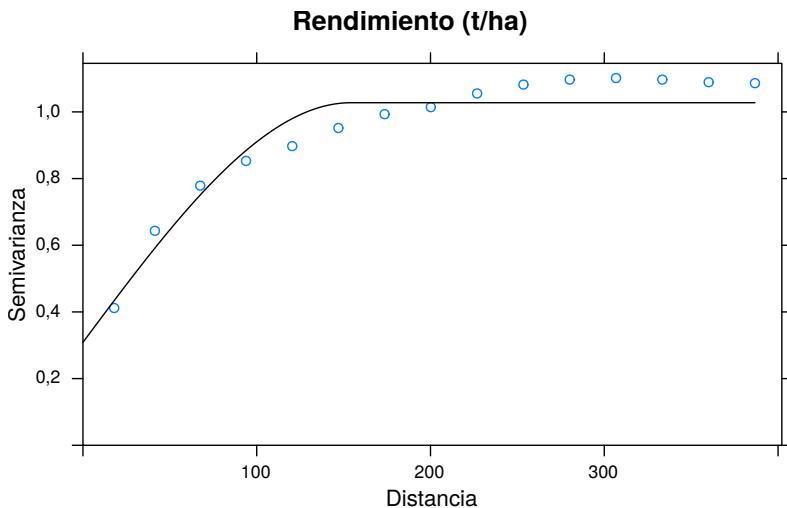
Se ajusta un modelo esférico, con valores 0,6, 200 y 0,2

como parámetros iniciales para estimar el *sill* parcial, rango y *nugget*, respectivamente. La salida del software R muestra los parámetros del semivariograma teórico ajustado: *nugget* ($C_0 = 0,31$), sill parcial ($C = 0,72$) y rango (154 m). Nota: bajo la columna *psill*, para la fila Nug, se debe leer el valor C_0 .

```
mod_esf <- fit.variogram(
  semiv_emp,
  vgm(0.6, "Sph", 200, 0.2))
mod_esf
#>   model psill range
#> 1   Nug 0,309     0
#> 2   Sph 0,719    154
```

El semivariograma empírico (puntos) y teórico ajustado (línea), para un modelo esférico, puede graficarse de la siguiente manera:

```
plot(semiv_emp,
      mod_esf,
      main = "Rendimiento (t/ha)",
      xlab = "Distancia",
      ylab = "Semivarianza")
```



El modelo que mejor ajusta será el de menor SCE. La función `attr()` devuelve atributos de un objeto y puede usarse para consultar la SCE del modelo ajustado.

```
attr(mod_esf, 'SSErr')
#> [1] 1,7
```

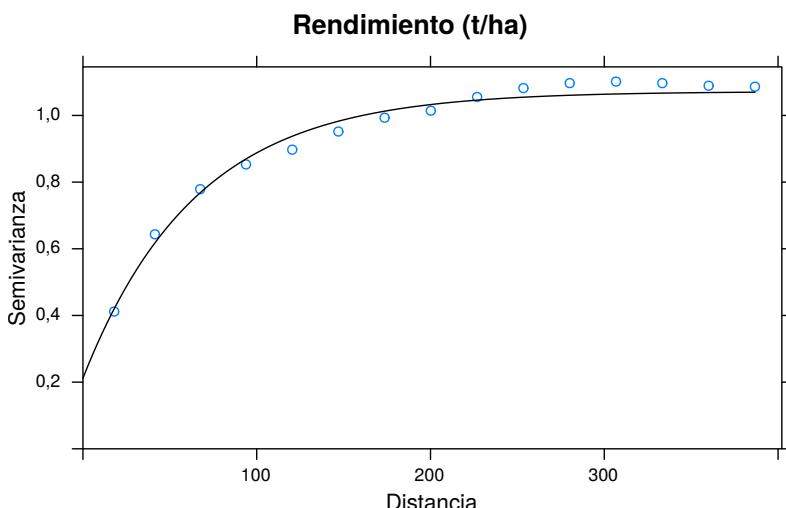
Repetiendo el procedimiento para un modelo exponencial (especificado en la función `vgm()`) se obtiene un menor SCE indicando mejor ajuste. Abajo se presenta la salida del software R que contiene los parámetros del semivariograma teórico ajustado: nugget ($C_0 = 0,21$), sill parcial ($C = 0,86$) y rango (64,88 m) o Rango Practico ($R_p = 64,88m \times 3$). Nota: bajo la columna “psill”, para la fila Nugget, se debe leer el valor C_0 .

```
mod_exp <- fit.variogram(
  semiv_emp,
  vgm(0.6, "Exp", 200, 0.2))
mod_exp
```

```
#>   model psill range
#> 1   Nug 0,212    0,0
#> 2   Exp 0,860   64,9
```

```
attr(mod_exp, 'SSErr')
#> [1] 0,377
```

```
plot(semiv_emp,
      mod_exp,
      main = "Rendimiento (t/ha)",
      xlab = "Distancia",
      ylab = "Semivarianza")
```

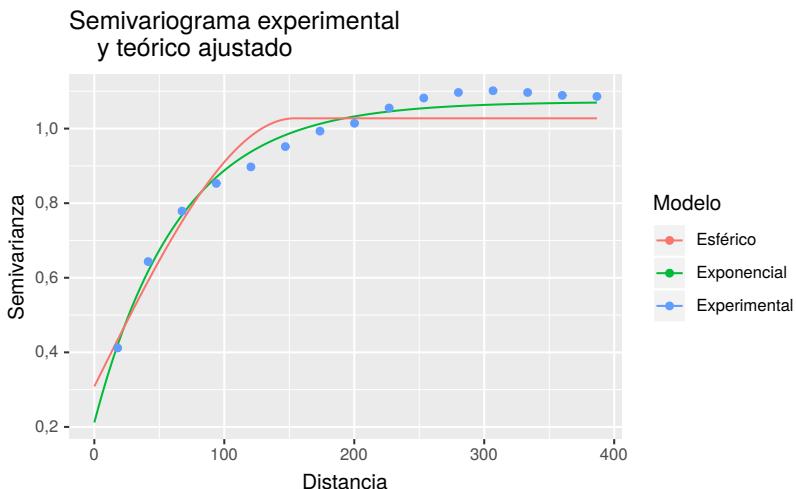


Las siguientes líneas permiten la visualización conjunta de los dos ajustes realizados.

```
vgLine <- rbind(
  cbind(
```

```
variogramLine(
  mod_exp, maxdist = max(semiv_emp$dist)),
  id = "Exponencial"),
cbind(
  variogramLine(
    mod_esf, maxdist = max(semiv_emp$dist)),
    id = "Esférico")
)

ggplot(semiv_emp, aes(x = dist, y = gamma,
                      color = id)) +
  geom_line(data = vgLine) +
  geom_point() +
  labs(
    title = "Semivariograma experimental
y teórico ajustado") +
  xlab("Distancia") +
  ylab("Semivarianza") +
  scale_color_discrete(
    name = "Modelo",
    labels = c("Esférico",
              "Exponencial",
              "Experimental"))
```



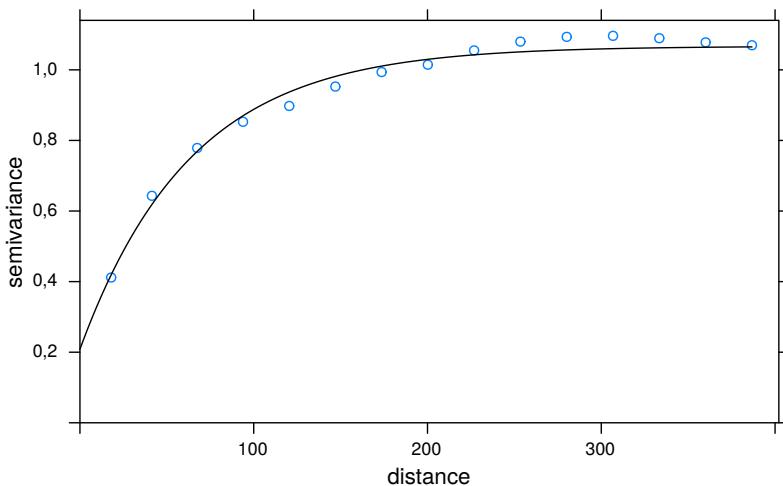
Algunas alternativas para aplicar las funciones de ajuste de los semivariogramas teóricos incluyen opciones de ajuste automático donde el usuario sólo especifica los modelos a ajustar sin tener que dar valores iniciales de los parámetros del semivariograma. La función estima valores iniciales razonables y selecciona aquel modelo de mejor bondad de ajuste en función a la SCE. A continuación, se presenta este ejemplo ajustando los modelos exponencial y esférico. Como era de esperar, el modelo de mejor ajuste fue el esférico.

```
modelos <- fit.variogram(semiv_emp,
                           vgm(c("Exp", "Sph")))
modelos
#>   model psill range
#> 1   Nug 0,212    0,0
#> 2   Exp 0,860   64,9
```

```
attr(modelos, 'SSERr')
#> [1] 0,377
```

Si bien, como se mostró en el análisis exploratorio de los datos no se evidencia una tendencia en la variable rendimiento con las coordenadas, en las siguientes líneas se ilustra cómo se realiza el ajuste del semivariograma empírico con tendencia dada por las coordenadas (x e y) y el posterior ajuste del modelo teórico. Los resultados muestran que no se observan diferencias importantes en los parámetros estimados del semivariograma teórico. En casos donde la tendencia es importante su efecto se puede reflejar en el ajuste del semivariograma empírico el cual suele mostrar un incremento de la semivarianza a medida que aumenta la distancia que no alcanza a estabilizarse dentro del dominio bajo estudio.

```
semiv_emp_t <- variogram(Rinde ~ x + y,
                           datos_3, cutoff = 400)
modelos_t <- fit.variogram(semiv_emp_t,
                            vgm(c("Exp", "Sph")))
modelos_t
#> model psill range
#> 1 Nug 0,208 0,0
#> 2 Exp 0,859 63,8
attr(modelos_t, 'SSErr')
#> [1] 0,345
plot(semiv_emp_t , modelos_t)
```



1.5.1 Mapeo de la variabilidad espacial

Para el mapeo de la variabilidad espacial se confeccionará una grilla de predicción donde se realiza el kriging. Las dimensiones de la grilla se establecerá mediante un polígono de los límites del lote. El archivo **limites.txt** contiene los puntos georreferenciados de cada arista del polígono.

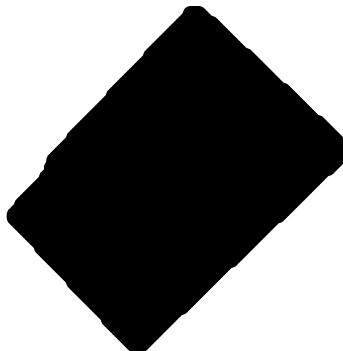
```
limites <- read.table("datos/limites.txt",
                      header = TRUE)
head(limites)
#>      x      y
#> 1 311842 5800614
#> 2 311950 5800728
#> 3 311998 5800788
#> 4 312006 5800835
#> 5 312566 5801431
#> 6 312590 5801435
```

La función `pred_grid()` del paquete `geoR` genera una grilla

regular de puntos de 10 metros de distancia entre estos. La función `polygrid()` recorta el polígono en la grilla siguiendo los límites del lote. Posteriormente se definen los nombres de las coordenadas y se transforma la clase del objeto a `sf` asignando el sistema de coordenadas.

```
grid <- pred_grid(límites, by = 10)
grid <- polygrid(grid, bor = límites)

names(grid) <- c("x", "y")
grid <- st_as_sf(grid, coords = c("x", "y"),
                  crs = 32721)
plot(grid)
```



La función `krige()` del paquete `gstat` realiza interpolación kriging y simulaciones condicionales mediante diferentes métodos de predicción. En este caso, se presenta la interpolación por kriging ordinario con el modelo de semivariograma exponencial estimado con geoestadística clásica. Los argumentos de esta función incluyen, la fórmula en la cual se especifica que el proceso es estacionario (`Rinde~1`), la base de datos (`datos_3`), el objeto sobre el cual se hará la predicción (`grid`) y la información del modelo del modelo de semivariograma teórico ajustado (`model`). La información de este último se encuentra dentro del objeto `modelos`. Los argumentos `nmin` y `nmax` permiten realizar

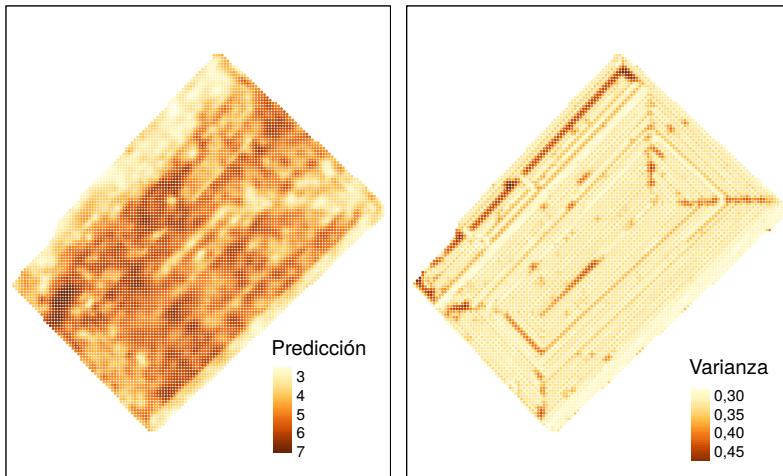
el proceso de interpolación en un contexto local, con un número mínimo y máximo de vecinos de cada punto a predecir de 7 y 25, respectivamente. En caso de omitir estos últimos argumentos la interpolación se realiza en un contexto global.

```
kriging_o <-
  krig(
    Rinde ~ 1,
    datos_3,
    st_as_sf(grid),
    nmin = 7,
    nmax = 25,
    model = modelos
  )
#> [using ordinary kriging]
```

A continuación, se realiza la visualización de la predicción y su varianza.

```
predK_otm <- tm_shape(kriging_o) +
  tm_dots("var1.pred", style = "cont",
          title = "Predicción")
varK_otm <- tm_shape(kriging_o) +
  tm_dots("var1.var", style = "cont",
          title = "Varianza")

tmap_arrange(predK_otm, varK_otm)
```



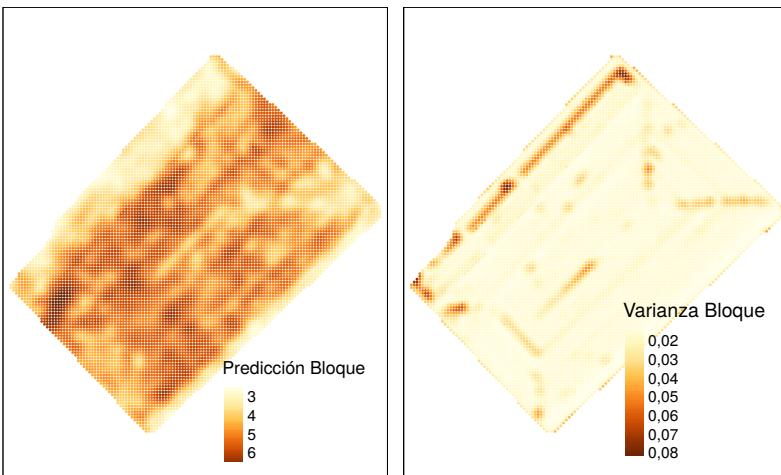
La siguiente línea de comando permite realizar la predicción kriging en bloques en un contexto local para lo cual solo se adiciona en la función `krige()` el argumento `block`. En este ejemplo se definió la dimensión del bloque de 40×40 m. Posteriormente se realiza la visualización de forma similar al caso anterior.

```
kriging_b <-
  krige(
    Rinde ~ 1,
    datos_3,
    grid ,
    nmin = 7,
    nmax = 25,
    model = modelos,
    block = c(40, 40)
  )
#> [using ordinary kriging]
```

```
predK_btm <- tm_shape(kriging_b) +
```

```
tm_dots("var1.pred", style = "cont",
         title = "Predicción Bloque")
varK_btm <- tm_shape(kriging_b) +
  tm_dots("var1.var", style = "cont",
          title = "Varianza Bloque")

tmap_arrange(predK_btm, varK_btm)
```



Las siguientes líneas de código muestran cómo obtener un `data.frame` contenido las predicciones realizadas sobre la grilla. En primer lugar, se extraen las coordenadas del objeto `kriging_b` y luego se transforma la clase del objeto a `data.frame`.

```
kriging_b$x <- st_coordinates(kriging_b) [, 1]
kriging_b$y <- st_coordinates(kriging_b) [, 2]
predRinde <-
  data.frame(kriging_b) [, c("x", "y",
                             "var1.pred")]
names(predRinde)[3] <- paste("Tg")
```

En el caso que exista una tendencia dada por las coordenadas, se utiliza kriging universal como método de interpolación espacial. Los comandos son similares a los anteriores, sólo debe cambiarse la fórmula. Por ejemplo, para una tendencia de primer grado en las coordenadas espaciales se escribiría `Rinde~x+y`. El modelo de semivariograma que se utiliza debe haberse ajustando contemplado el mismo tipo de tendencia. En situaciones donde otras covariables explícan la tendencia en la media, éstas deben especificarse en la fórmula. Para poder hacer la interpolación es necesario que los valores de las covariables también estén disponibles en la grilla de predicción. En este último caso la interpolación se denomina kriging con deriva externa.

La interpolación también puede hacerse utilizando los parámetros *partial sill*, rango y *nugget* del semivariograma estimado con REML. Para el ajuste de un MLM con errores correlacionados espacialmente vía REML, la base de datos no debe ser muy grande. Para poder realizar el ajuste de un MLM en el conjunto de datos de ilustración, se tomó una muestra aleatoria de $n=500$ sobre la base de datos original de $n=9009$, usando la función `sample()`. Las siguientes líneas de código muestran el ajuste del semivariograma y obtención de los parámetros del mismo usando la función `gls()` del paquete `nlme`. Se realiza el ajuste de dos modelos lineales, el primero sin correlación espacial (`null_model`) y el segundo con una estructura de correlación espacial del tipo exponencial (`esp_model`).

```
set.seed(7)
datos_4 <- datos_3[
  sample(1:nrow(datos_3),
  500, replace = FALSE), ]
null_model <-
  gls(Rinde ~ 1, datos_4,
```

```
method = "REML",
na.action = na.omit)

esp_model <- gls(
  Rinde ~ 1,
  data = datos_4,
  correlation = corExp(
    form = ~ x + y,
    metric = "euclidean",
    nugget = T
  ),
  method = "REML",
  na.action = na.omit
)
```

A continuación, se realiza la comparación de ambos modelos mediante la prueba del cociente de verosimilitud. Los resultados muestran que hay diferencias significativas ($p<0.005$) entre ambos modelos, con lo cual se elige el modelo espacial. Los valores de AIC y BIC también mostraban un mejor ajuste de este último.

```
anova(null_model, esp_model)
#> Model df AIC BIC logLik Test L.Ratio p-value
#> null_model 1 2 1526 1535 -761
#> esp_model 2 4 1230 1247 -611 1 vs 2 300
<.0001
```

Con el modelo ajustado se obtienen los parámetros del semivariograma y se construye el objeto `m` que contendrá estos para su posterior interpolación vía kriging.

```
summary(esp_model)
#> Generalized least squares fit by REML
#> Model: Rinde ~ 1
#> Data: datos_4
#> AIC BIC logLik
#> 1230 1247 -611
#>
#> Correlation Structure: Exponential spatial
correlation
#> Formula: ~x + y
#> Parameter estimate(s):
#> range nugget
#> 102,547 0,134
#>
#> Coefficients:
#> Value Std.Error t-value p-value
#> (Intercept) 4,22 0,234 18 0
#>
#> Standardized residuals:
#> Min Q1 Med Q3 Max
#> -1,848 -0,371 0,378 1,097 2,634
#>
#> Residual standard error: 1,18
#> Degrees of freedom: 500 total; 499 residual
esp_model$sigma
#> [1] 1,18
nugget <- 0.1344285 * esp_model$sigma ^ 2
psill <- esp_model$sigma ^ 2 - nugget
range <- 102.5470883

m <- vgm(psill, "Exp", range, nugget)
kriging_mlm <-
  krig(
    Rinde ~ 1,
```

```
    datos_4,  
    grid ,  
    nmin = 7,  
    nmax = 25,  
    model = m  
)  
#> [using ordinary kriging]
```

El mismo ajuste de semivariograma puede realizarse utilizando la función `likfit()` del paquete `geoR`. En este caso es necesario generar un objeto del tipo `geodata`. La función de ajuste solicita valores iniciales de *sill* parcial y rango y modelo teórico (para esta aplicación se usa el exponencial). En el caso del efecto *nugget* el mismo es estimado por defecto. Es posible también incorporar tendencias utilizando el argumento `trend`. Los resultados del modelo muestran valores estimados de los parámetros similares a los obtenidos con la función `gls()`.

```
datos_geor <- as.data.frame(datos_4)  
datos_geor <-  
  as.geodata(datos_geor,  
             coords.col = c("x", "y"),  
             data.col = "Rinde")  
  
mlm_geor <-  
  likfit(  
    datos_geor,  
    ini = c(0.7, 90),  
    cov.model = "exponential",  
    lik.method = "REML",  
    messages = FALSE  
)
```

```
summary(mlm_geor)
#> Summary of the parameter estimation
#> -----
#> Estimation method: restricted maximum
likelihood
#>
#> Parameters of the mean component (trend):
#> beta
#> 4,22
#>
#> Parameters of the spatial component:
#> correlation function: exponential
#> (estimated) variance parameter sigmasq
(partial sill) = 1,2
#> (estimated) cor. fct. parameter phi (range
parameter) = 103
#> anisotropy parameters:
#> (fixed) anisotropy angle = 0 ( 0 degrees )
#> (fixed) anisotropy ratio = 1
#>
#> Parameter of the error component:
#> (estimated) nugget = 0,186
#>
#> Transformation parameter:
#> (fixed) Box-Cox parameter = 1 (no
transformation)
#>
#> Practical Range with cor=0.05 for asymptotic
range: 307
#>
#> Maximised Likelihood:
#> log.L n.params AIC BIC
#> "-608" "4" "1224" "1241"
#>
```

```
#> non spatial model:  
#> log.L n.params AIC BIC  
#> "-758" "2" "1520" "1528"  
#>  
#> Call:  
#> likfit(geodata = datos_geor, ini.cov.pars =  
c(0.7, 90), cov.model = "exponential",  
#> lik.method = "REML", messages = FALSE)
```

1.5.2 Validación cruzada

A continuación, se ilustra el proceso de validación cruzada *k-fold*. En este caso la función utilizada es `krige.cv()` del paquete `gstat`. Aquí como en el ajuste del semivariogram empírico y la interpolación, se tienen los argumentos `fórmula` (`Rinde~1`, lo cual especifica que es un proceso estacionario), la base de datos (`datos_3`) y el modelo de semivariograma teórico ajustado (`modelos`). Aquí también se realiza la validación usando kriging en un contexto local por lo que se colocan los argumentos `nmin` y `nmax`. El argumento `nfold` determina el número de grupos (`k`) en los que se divide la base de datos para realizar la validación cruzada *k-fold*. Para obtener repetibilidad en los resultados se sugiere fijar la semilla mediante la función `set.seed()`.

```
set.seed(17)  
valcru <-  
  krige.cv(  
    Rinde ~ 1,  
    datos_3,  
    modelos,  
    nfold = 10,
```

```

nmin = 7,
nmax = 25
)

```

Realizada la validación es posible calcular estadísticos resumen como el error medio (ME), error cuadrático medio (MSE), media del cociente de la desviación cuadrática (*mean squared deviation ratio*, MSDR), raíz del error cuadrático medio (RMSE), la RMSE relativa a la media de los observados (RMSE_rel) y la correlación lineal entre los observados vs. Predichos. Un gráfico de estos últimos se muestra al final.

```

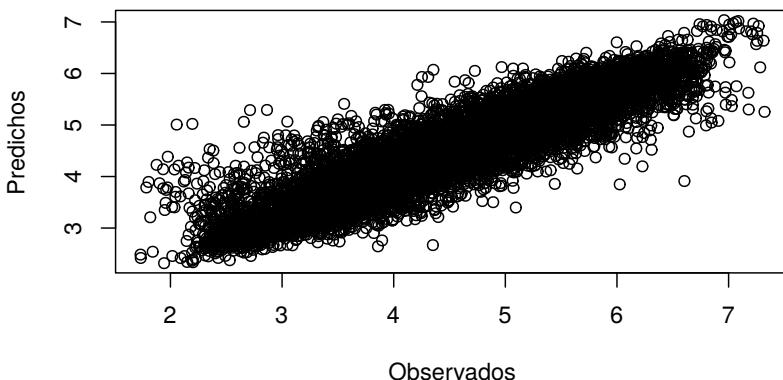
ME <- mean(valcru$residual)
MSE <- mean(valcru$residual ^ 2)
MSDR <- mean(valcru$zscore ^ 2)
RMSE <- sqrt(mean(valcru$residual ^ 2))
RMSE_rel <-
  sqrt(mean(valcru$residual ^ 2)) /
  mean(valcru$observed) * 100
r <- cor(valcru$observed,
          valcru$observed - valcru$residual)

tabla <- data.frame(ME, MSE, RMSE,
                     RMSE_rel, MSDR, r)
tabla
#>      ME    MSE   RMSE RMSE_rel   MSDR      r
#> 1 -0,00018 0,228 0,478      10,3 0,665 0,899
plot(
  valcru$observed,
  valcru$observed - valcru$residual,
  xlab = "Observados",

```

1.6 Caracterización de variabilidad espacial con múltiples capas de datos 109

```
    ylab = "Predichos"  
)
```



1.6 Caracterización de variabilidad espacial con múltiples capas de datos

1.6.1 Análisis de componentes principales

Para implementar el análisis multivariado es necesario contar con información de cada variable en los mismos sitios georreferenciados. En esta sección se usa la base de datos **Pred2.txt** que contiene mediciones de conductividad eléctrica aparente en dos profundidades 0-30 cm (CE30) y 0-90 cm (CE90), elevación (Elev), profundidad de suelo (Pe) y rendimiento de trigo (Tg). Para generar esta base debido a las diferentes resoluciones espaciales de las variables medidas, se calculó una zona buffer de 15 m de radio para la variable Pe y sobre cada punto buffer se calculó la mediana de las restantes variables. La matriz de datos resultante

quedó conformada por $n=482$ sitios (filas) y $p=7$ variables (columnas).

Para realizar el Análisis de Componentes Principales espacial (MULTISPATI-PCA) se utiliza los paquetes **ade4** y **adespatial**. Primero se necesita calcular la matriz de ponderación espacial en forma similar a la realizada para el cálculo del índice de Moran. Luego, se realiza un Análisis de Componentes Principales (PCA) clásico y posteriormente sobre las componentes generadas por PCA, se aplica el MULTISPATI-PCA.

Carga de base de datos multivariada.

```
pred <- read.table("datos/Pred2.txt",
                      header = TRUE)
head(pred)
#> x y Pe Elev CE30 CE90 Tg
#> 1 312283 5800205 80 161 21,8 30,6 4,19
#> 2 312257 5800229 40 161 30,3 17,9 4,00
#> [ reached 'max'
    / getOption("max.print") --
omitted 4 rows ]
```

La función **dudi.pca()** del paquete **ade4**, permite realizar un PCA sobre objetos de clase **data.frame**. Sus argumentos indican, las variables con las que se realizará el PCA, un valor lógico (**TRUE** o **FALSE**) indicando si debe o no centrarse por la media (**center**) y normalizarse (**scale**), un valor lógico para la realización o no del gráfico (**scannf**) y la cantidad de ejes guardados (**nf**), que coincide con la cantidad de variables utilizadas en el análisis.

```
pca <-
  dudi.pca(
```

```

pred[, 3:7],
center = TRUE,
scale = TRUE,
scannf = FALSE,
nf = 5
)

```

Para transformar un PCA en un PCA espacial (MULTISPATI-PCA) se calcula la red de vecindarios y la matriz de ponderación espacial. La distancia máxima para definir los sitios vecinos de cada dato fue de 45 m. Además, se adiciona el argumento `zero.policy=T` para poder generar la matriz de pesos espaciales contemplando que algunos puntos no tengan datos vecinos. La función `multispati()` permite realizar el MULTISPATI-PCA. Para ello es necesario colocar en la función el objeto que surge de realizar el ACP (`pca`) y la matriz de pesos espaciales (`pesos_sp`). El argumento `nfposi` hace referencia al número de ejes con autocorrelación positiva que es retenido en el análisis. También pueden guardarse ejes con autocorrelación negativa mediante el argumento `nfnega`. En general los ejes con autocorrelación negativa son aquellos de menor contribución a la variabilidad total.

```

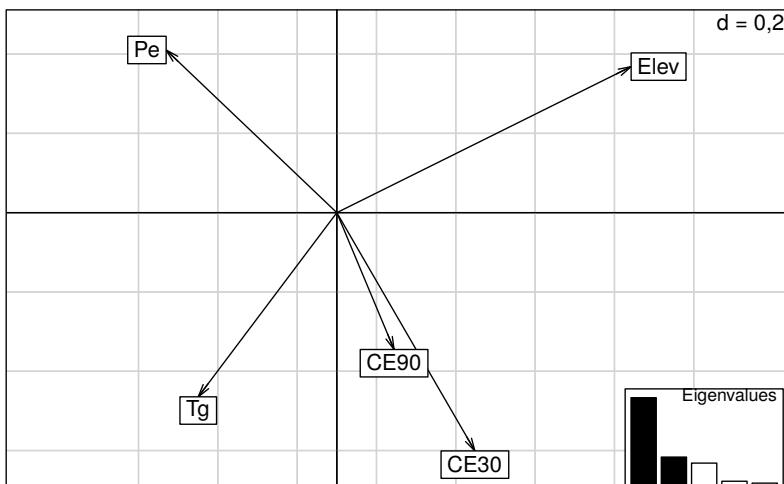
cord <- coordinates(pred[, 1:2])
vecindarios <- dnearneigh(cord, 0, 45)
pesos_sp <- nb2listw(vecindarios,
                      style = "W",
                      zero.policy = T)

pca_esp <-
adespatial::multispati(pca, pesos_sp,
                       scannf = F, nfposi = 5)

```

Para realizar un gráfico que muestre las correlaciones entre las variables se puede usar la función `s.arrow()`. En este gráfico de traza un vector para cada variable en el espacio definido por las componentes principales que se seleccionen. En este caso de estudio, la función utiliza la primera componente para graficar el eje horizontal y la segunda componente para el eje vertical. Para adicionar un gráfico de barras con los autovalores puede usarse el argumento `add.scatter.eig()`.

```
s.arrow(pca_esp$c1,
        xax = 1,
        yax = 2,
        clabel = 1)
add.scatter.eig(
  pca_esp$eig,
  xax = 1,
  yax = 2,
  posi = "bottomright",
  ratio = 0.2
)
```



1.6 Caracterización de variabilidad espacial con múltiples capas de datos 113

El gráfico obtenido del MULTISPATI-PCA muestra que las variables Elev y Pe son las más importantes en la explicación de la variabilidad espacial a nivel del primer eje (sPC1, eje horizontal). Mientras que la CE30 y Tg presentan mayor importancia en la SPC2. Además, se observa una correlación positiva entre CE30 y CE90, y negativa entre estas dos y la Pe. También la Elev y Tg se correlacionan en forma negativa. El gráfico de autovalores (barras) sugiere dos estructuras principales a nivel de sPC1 y sPC2, siempre la sPC1 explica la mayor parte de la variabilidad de los datos seguida por sPC2, sPC3, y así sucesivamente.

```
summary(pca_esp)
#>
#> Multivariate Spatial Analysis
#> Call: adespatial::multispaci(dudi = pca,
listw = pesos_sp, scannf = F,
#> nfposi = 5)
#>
#> Scores from the initial duality diagram:
#> var cum ratio moran
#> RS1 1,936 1,94 0,387 0,676
#> RS2 1,078 3,01 0,603 0,405
#> RS3 0,878 3,89 0,778 0,250
#> RS4 0,628 4,52 0,904 0,552
#> [ reached 'max'
    /getOption("max.print") --
omitted 1 rows ]
#>
#> Multispaci eigenvalues decomposition:
#> eig var moran
#> CS1 1,4254 1,808 0,7884
#> CS2 0,4684 1,095 0,4277
#> CS3 0,3712 0,748 0,4963
```

```
#> CS4 0,0790 0,689 0,1147  
#> CS5 0,0484 0,660 0,0734
```

Como se establece en la literatura, MULTISPATI-PCA maximiza el producto entre la varianza espacial y la autocorrelación mientras que PCA maximiza la varianza total. Los resultados muestran que con MULTISPATIPCA se explica una menor proporción de la varianza acumulada en el primer eje, respecto de PCA (1,81 vs. 1,94). Las dos primeras CP del PCA explican 60 % de la variabilidad total mientras que la CS1 y CS2 del MULTISPATI el 58 %. No obstante, los valores del índice de Moran calculados para las tres primeras CPs sugieren que la estimación de autocorrelación aumentó cuando se usó MULTISPATIPCA respecto de la contenida en las CPs del PCA (0,79 vs. 0,68 para el eje 1, 0,43 vs. 0,40 para el eje 2, 0,50 vs. 0,25 para el eje 3). Este resultado permitiría una visualización mejor de la variabilidad espacial. Por el contrario, a nivel de las CPs 4 y 5 este comportamiento fue inverso.

1.6.2 Análisis de conglomerados

Para implementar este análisis también es necesario contar con información de cada variable en los mismos sitios georreferenciados. Otra forma de lograr esto es interpolar cada una de ellas con la misma grilla de predicción. Es decir, que cada punto de la grilla tendrá un dato para cada variable predicha. Para el siguiente caso de estudio se realizó el procedimiento de interpolación con mediciones de conductividad eléctrica aparente en dos profundidades 0-30 cm (CE30) y 0-90 cm (CE90), elevación (Elev), profundidad de suelo (Pe) y rendimiento de trigo (Tg) (archivo **Pred.txt**). Para cada variable se realizó un análisis explo-

ratorio y la predicción espacial para el re-escalado usando una grilla común a todas ellas de 10×10 m. Una vez que se realiza el re-escalado de cada variable, se tiene un objeto para cada variable con igual número de filas y columnas que pueden unirse en un único objeto usando la función `cbind()`. Para `predRinde` se extraen las 3 primeras columnas correspondiente a las coordenadas y valores predichos, mientras que para las restantes sólo se extraen los valores predichos de cada variable (columna 3) considerando que, si se utilizó la misma grilla de predicción, las coordenadas de cada `data.frame` deberían ser las mismas. Se recomienda mantener clara la nomenclatura de cada variable, teniendo en cuenta que el software es *case-sensitive* (sensible a mayúsculas y minúsculas). A tal efecto, se renombraron las columnas. A continuación, se muestran los códigos de R para hacer el procedimiento de concatenación, pero para la exemplificación se carga y utiliza una base de datos que previamente fue concatenada.

```
pred <- read.table("datos/Pred.txt",
                      header = T)
```

Posteriormente, se implementará el análisis de clúster espacial KM-sPC (Córdoba et al., 2013). Para ello primero se realiza un análisis de componentes principales espaciales (MULTISPATI-PCA) sobre las variables originales. Luego las variables sintéticas (componentes principales espaciales, sPC) son utilizadas como input del análisis de *cluster fuzzy k-means*.

```
pca <-
  dudi.pca(
    pred[, 3:7],
    center = TRUE,
```

```

    scale = TRUE,
    scannf = FALSE,
    nf = 5
)

cord <- coordinates(pred[, 1:2])
vecindarios <- dnearneigh(cord, 0, 10)
pesos_sp <- nb2listw(vecindarios,
                      style = "W",
                      zero.policy = T)

pca_esp <-
  adespatial::multispati(pca, pesos_sp,
                         scannf = F, nfposi = 5)

```

La función `multispati()` almacena las sPC en la posición `li` dentro de los objetos creados. La siguiente sentencia crea un nuevo objeto con la unión de las columnas con las coordenadas y las sPC.

```

cs <- pca_esp$li[, 1:5]
pred_am <- cbind(pred[, c("x", "y")], cs)
head(pred_am)
#> x y CS1 CS2 CS3 CS4 CS5
#> 1 312433 5800234 1,92 0,543 -0,118 -0,321
0,0257
#> 2 312423 5800244 1,93 0,448 -0,215 -0,256
0,1732
#> [ reached 'max'
  / getOption("max.print") --
omitted 4 rows ]

```

Para realizar el análisis de *cluster fuzzy k-means* se utiliza la función `cmeans()` del paquete `e1071` (Meyer et al., 2019).

1.6 Caracterización de variabilidad espacial con múltiples capas de datos 117

Para ello se necesita determinar las sPC que se utilizarán como *input*. En este caso se seleccionaron las columnas que corresponden a la sPC1, sPC2 y sPC3, de esta forma una gran cantidad de la variabilidad total es contemplada ($\geq 70\%$) en el análisis. En este ejemplo se utilizaron 2, 3 y 4 clústers. Otras opciones de configuración son el número de iteraciones=100; método=cmeans (opción para usar el algoritmo fuzzy) y exponente difuso $m=1.3$.

```
clases2 <- cmeans(pred_am[, 3:5], 2,
                    100, method = "cmeans",
                    m = 1.3)
clases3 <- cmeans(pred_am[, 3:5], 3,
                    100, method = "cmeans",
                    m = 1.3)
clases4 <- cmeans(pred_am[, 3:5], 4,
                    100, method = "cmeans",
                    m = 1.3)
```

En el ejemplo de ilustración se debe seleccionar entre dos, tres y cuatro clases. Para ello se utilizaron los siguientes índices: Xie-Beni, coeficiente de partición, entropía de clasificación y Fukuyama-Sugeno. Estos índices serán calculados para 2, 3 y 4 clases o clúster, utilizando la función `fclustIndex()`. En todos los índices, excepto el coeficiente de partición, el número de clases óptimo se obtiene cuando los índices tienen el menor valor. Para hacer que la interpretación del coeficiente de partición sea igual a los otros índices, se utiliza el valor inverso del índice. Luego se confeccionó una tabla con los índices obtenidos.

```
indices <- c(
  "xie.beni",
  "fukuyama.sugeno",
```

```

"partition.coefficient",
"partition.entropy",
"proportion.exponent"
)

ind_2clases <-
fclustIndex(clases2,
            pred_am[, 3:5],
            index = indices)

ind_3clases <-
fclustIndex(clases3,
            pred_am[, 3:5],
            index = indices)

ind_4clases <-
fclustIndex(clases4,
            pred_am[, 3:5],
            index = indices)

indices <- cbind(ind_2clases,
                  ind_3clases,
                  ind_4clases)

indices
#>      ind_2clases ind_3clases ind_4clases
#> xb      4,83e-05   9,37e-05   1,09e-04
#> fs     -1,20e+04   -1,32e+04   -1,44e+04
#> pc      9,26e-01   8,63e-01   8,34e-01
#> pe      1,26e-01   2,43e-01   3,02e-01
#> pre        Inf         Inf         Inf

```

En este ejemplo la mayoría de los índices, excepto Fukuyama-Sugeno, muestran que el número de clases a seleccionar es dos. Puede suceder que ninguno de los índi-

ces coincida con otro en el número óptimo de clases. Para facilitar la toma de decisiones se recomienda calcular un índice resumen para cada clasificación. Este nuevo índice puede ser la distancia Euclídea de los valores de los índices previamente normalizados por su valor máximo a través de las diferentes clasificaciones.

```
XieBeniN <- indices[1,] / max(indices[1,])
FukSugN <- indices[2,] / max(indices[2,])
CoefPartN <- indices[3,] / max(indices[3,])
EntrPartN <- indices[4,] / max(indices[4,])

indicesN <-
  data.frame(rbind(XieBeniN, FukSugN,
                    CoefPartN, EntrPartN))
indicesN <- (indicesN) ^ 2

indices2N <- sqrt(sum(indicesN[, 1]))
indices3N <- sqrt(sum(indicesN[, 2]))
indices4N <- sqrt(sum(indicesN[, 3]))

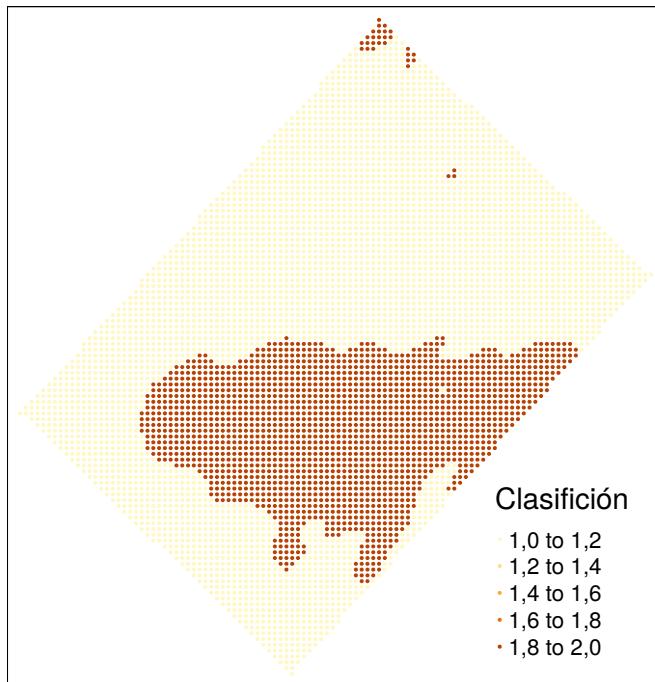
indices2N
#> [1] 1,54
indices3N
#> [1] 1,86
indices4N
#> [1] 2,06
```

El índice resumen se optimiza para la estructura conformada por dos clúster. Para realizar mapas de la clasificación, primero se debe extraer los datos de las clases delimitadas con el algoritmo *fuzzy k-means* y combinar con la base de datos inicial en la cual sólo se dejan las coordenadas. Posteriormente se transforma la base (`base_am`) a un objeto de clase `sf` y luego se grafican con el paquete `tmap`.

```
base_am <-
  cbind(
    pred_am[, 1:2],
    "2clases" = clases2$cluster,
    "3clases" = clases3$cluster,
    "4clases" = clases4$cluster
  )
head(base_am)
#> x y 2clases 3clases 4clases
#> 1 312433 5800234 1 1 1
#> 2 312423 5800244 1 1 1
#> 3 312433 5800244 1 1 1
#> [ reached 'max'
  / getOption("max.print") --
omitted 3 rows ]
```

```
base_am <- st_as_sf(base_am,
                      coords = c("x", "y"),
                      crs = 32721)
```

```
tm_shape(base_am) +
  tm_dots("2clases", title = "Clasificación")
```



1.7 Predicción con múltiples capas de datos

En esta sección también se usa la base de datos **Pred2.txt** que contiene mediciones de conductividad eléctrica aparente en dos profundidades 0-30 cm (CE30) y 0-90 cm (CE90), elevación (Elev), profundidad de suelo (Pe) y rendimiento de trigo (Tg). La matriz de datos está conformada por $n=482$ sitios (filas) y $p=7$ variables (columnas).

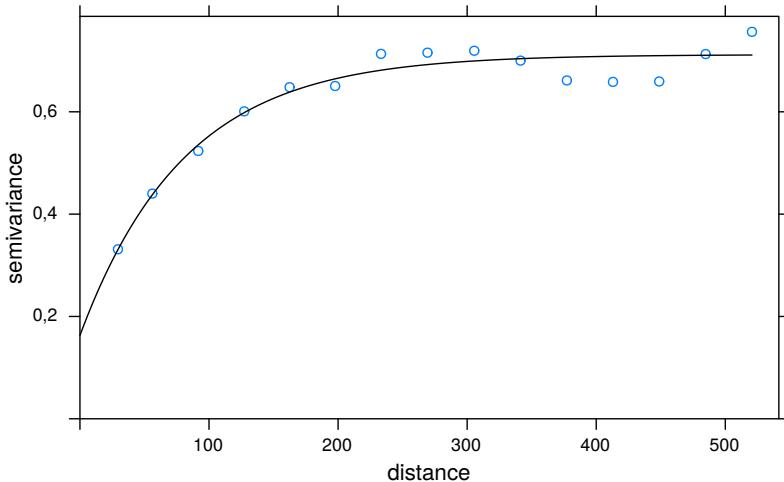
Adicionalmente, para realizar la interpolación utilizando información de las covariables es necesario que la grilla de predicción contenga los valores de las coordenadas y de las covariables. El archivo **grilla_am.txt** contiene estos datos. A continuación, se cargan ambas bases de datos y transforma a objetos de clase **sf**.

```
pred <- read.table("datos/Pred2.txt",
                      header = TRUE)
pred <- st_as_sf(pred, coords = c("x", "y"),
                   crs = 32721)
head(pred)
#> Simple feature collection with 6 features
and 5 fields
#> geometry type: POINT
#> dimension: XY
#> bbox: xmin: 312000 ymin: 5800000 xmax:
312000 ymax: 5800000
#> epsg (SRID): 32721
#> proj4string:
+ proj=utm
+ zone=21
+ south
+datum=WGS84 +units=m +no_defs
#> First 3 features:
#> Pe Elev CE30 CE90 Tg geometry
#> 1 80 161 21,8 30,6 4,19 POINT (312283
5800205)
#> 2 40 161 30,3 17,9 4,00 POINT (312257
5800229)
#> [ reached 'max'
/ getOption("max.print") --
omitted 1 rows ]
grilla <- read.table("datos/grilla_am.txt",
                      header = TRUE)
grilla <- st_as_sf(pred, coords = c("x", "y"),
                   crs = 32721)
```

1.7.1 Kriging con deriva externa

En las siguientes líneas de se realiza el ajuste del semivarioograma empírico contemplando una tendencia dada por las covariables CE30, CE90, Elev y Pe y se realiza el ajuste de semivariogramas teóricos. Finalmente, se grafican ambos semivariogramas. Los resultados muestran que el modelo de mejor ajuste fue el exponencial. Los parámetros obtenidos fueron nugget ($C_0 = 0,16$), sill parcial ($C = 0,55$) y rango (80 m). Nota: bajo la columna “psill”, para la fila Nugget, se debe leer el valor C_0 .

```
semiv_ked <- variogram(
  Tg ~ CE30 + CE90 + Elev + Pe, pred
)
v.fit_vut_ked <-
  fit.variogram(semiv_ked ,
    vgm(c("Exp", "Sph", "Gau")))
v.fit_vut_ked
#>   model psill range
#> 1   Nug 0,163  0,0
#> 2   Exp 0,549  80,7
plot(semiv_ked , v.fit_vut_ked)
```



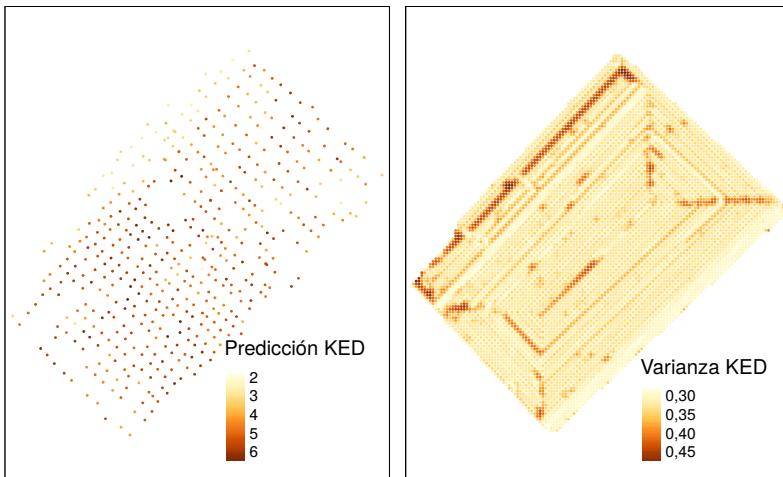
Para realizar la interpolación espacial se utiliza también la función `krige()` y se especifica la tendencia con las covariables en la fórmula. En este caso la predicción se realiza también en un contexto local (argumentos `nmin` y `nmax`).

```
kriging_ed <- krige(
  Tg ~ CE30 + CE90 + Elev + Pe,
  pred,
  grilla,
  model = v.fit_vut_ked,
  nmin = 7,
  nmax = 25
)
#> [using universal kriging]
```

```
prediccionKED <-
  tm_shape(kriging_ed) +
  tm_dots("var1.pred", style = "cont",
          title = "Predicción KED")
```

```
varianzaKED <-
  tm_shape(kriging_o) +
  tm_dots("var1.var", style = "cont",
          title = "Varianza KED")

tmap_arrange(prediccionKED, varianzaKED)
```



1.7.2 Kriging desde modelo de regresión

Para realizar la interpolación por el método kriging regresión primero se ajusta un modelo lineal de regresión entre la variable Tg y las covariables CE30, CE90, Elev y Pe.

```
mlr <- lm(Tg ~ CE30 + CE90 + Elev + Pe, pred)
```

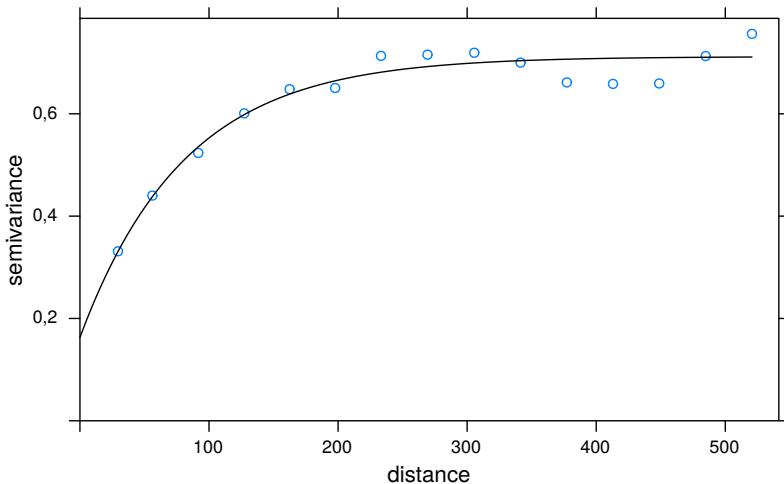
A partir del modelo ajustado se obtienen los residuos que son incorporados al objeto `pred`. Luego sobre estos residuos se modela el semivariograma empírico y teórico. Los resultados muestran que los valores estimados de los parámetros

del semivariograma son similares a los obtenidos en el caso anterior. Esto es esperable ya que ambas aproximaciones son equivalentes.

```
pred$residuos <- mlr$residuals  
names(pred)  
#> [1] "Pe" "Elev" "CE30" "CE90" "Tg"  
"geometry" "residuos"
```

```
semiv_rk <- variogram(residuos ~ 1 , pred)  
  
v.fit_vut_rk <-  
  fit.variogram(semiv_rk ,  
    vgm(c("Exp", "Sph", "Gau")))  
v.fit_vut_rk  
#>   model psill range  
#> 1   Nug 0,163  0,0  
#> 2   Exp 0,549  80,7
```

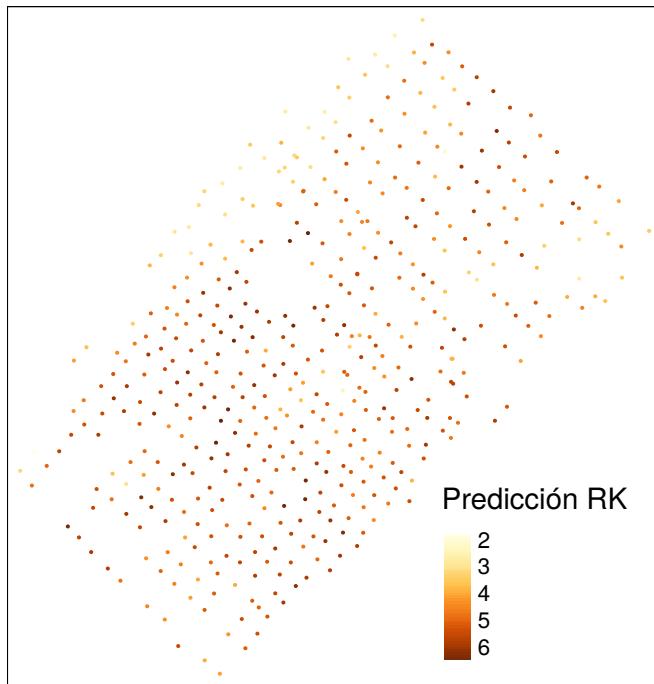
```
plot(semiv_rk , v.fit_vut_rk)
```



Finalmente se realiza la predicción en la grilla de los residuos y esta es sumada a la predicción del modelo de regresión ajustado inicialmente para obtener los valores predichos finales.

```
kgres <- krig(residuos ~ 1, pred,
                 grilla, model = v.fit_vut_rk)
#> [using ordinary kriging]
grilla$RK_pred <-
  predict(mlr, newdata = grilla) +
  kgres$var1.pred
```

```
tm_shape(grilla) +
  tm_dots("RK_pred", style = "cont",
          title = "Predicción RK")
```



1.7.3 Árboles aleatorios

Para este método primero se ajusta el algoritmo *random forest* utilizando para ello el paquete **caret**. Para ello se optimizará el parámetro **mtry** mediante un proceso de validación cruzada. Los valores probados de **mtry** se especifican en el objeto que lleva el mismo nombre. Con la función **fitControl()** se establece el tipo de validación cruzada, en este caso k-fold con $k=10$. Además, se permite el paralelizado del proceso en caso de ser necesario (**allowParallel=T**).

```
mtry <- expand.grid(mtry = seq(1, 4, 1))
fitControl <- trainControl(method = "cv",
                           number = 10,
                           allowParallel = T)
```

Las siguientes son las opciones de paralelizado que involucra los paquetes `parallel` y `doParallel`

```
library(parallel)
library(doParallel)
cluster <- makeCluster(max(1, detectCores() - 1))
registerDoParallel(cluster)
```

El ajuste del *random forest* se realiza con la función `train()` la cual requiere entre otros especifica la fórmula o modelo, la base de datos a utilizar, el algoritmo (`rf`, random forest), grilla de valores de hiperparámetros a evaluar (objeto `mtry`) y opciones de la validación.

```
set.seed(7)
train_rf <- train(
  Tg ~ CE30 + CE90 + Elev + Pe,
  data = pred,
  method = "rf",
  tuneGrid = mtry,
  trControl = fitControl
)
```

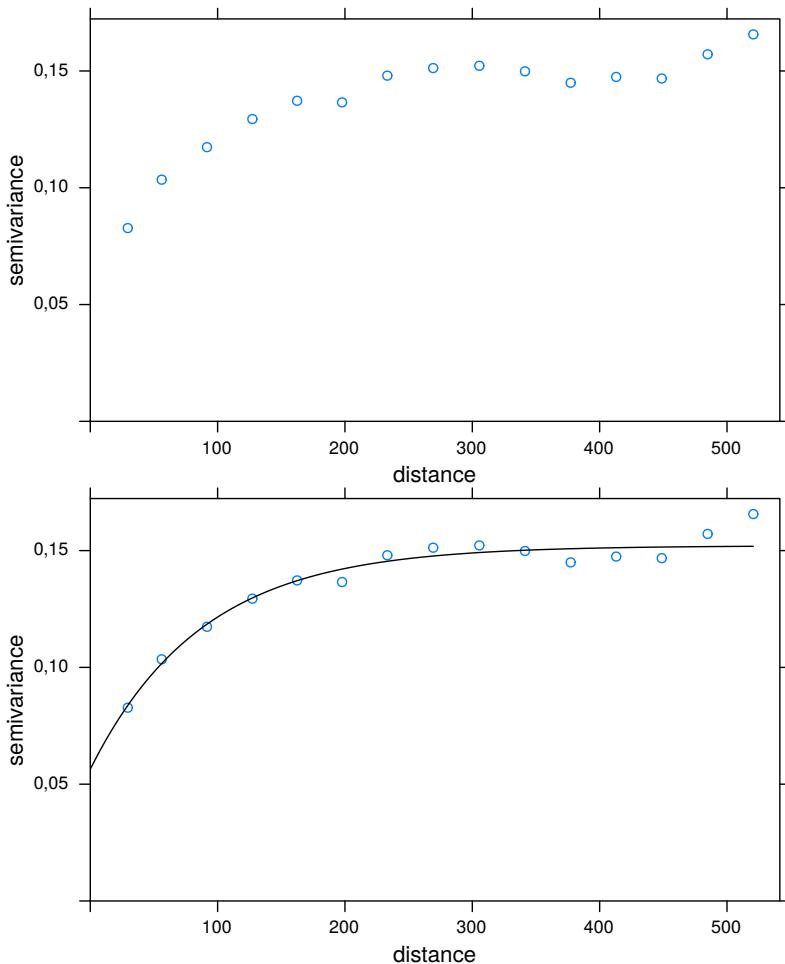
Luego de ajustar el modelo de RF se procede a obtener los residuos y el ajuste de los semivariogramas.

```
pred$residuosRF <-
  pred$Tg - predict(train_rf, newdata = pred)

semiv_RFk <- variogram(residuosRF ~ 1 , pred)
plot(semiv_RFk)

v.fit_vut_RFk <-
  fit.variogram(semiv_RFk ,
```

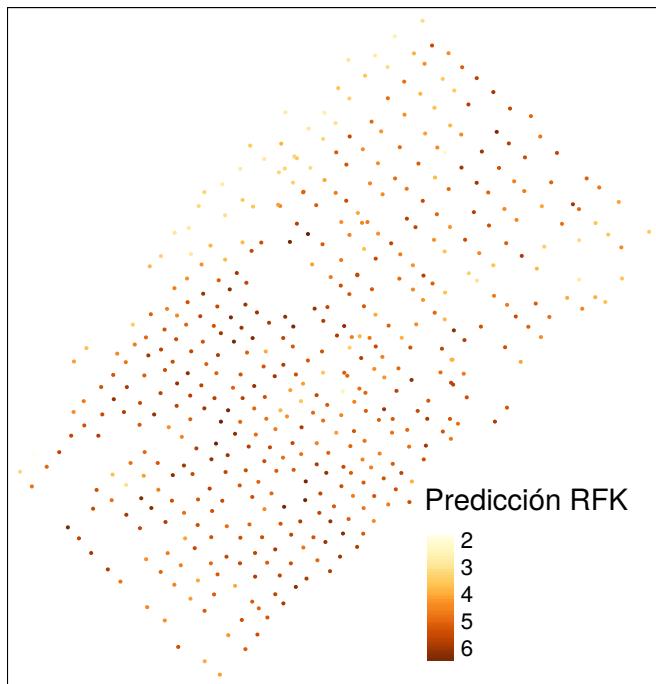
```
vgm(c("Exp", "Sph", "Gau")))
plot(semiv_RFk , v.fit_vut_RFk)
```



Finalmente se realiza la predicción de los residuos sobre la grilla y se suman a la predicción del modelo de random forest ajustado inicialmente.

```
kgresRF <- kriging(residuosRF ~ 1, pred,  
                     grilla, model = v.fit_vut_RFk)  
#> [using ordinary kriging]
```

```
tm_shape(grilla) +  
  tm_dots("RK_pred", style = "cont",  
          title = "Predicción RFK")
```





2

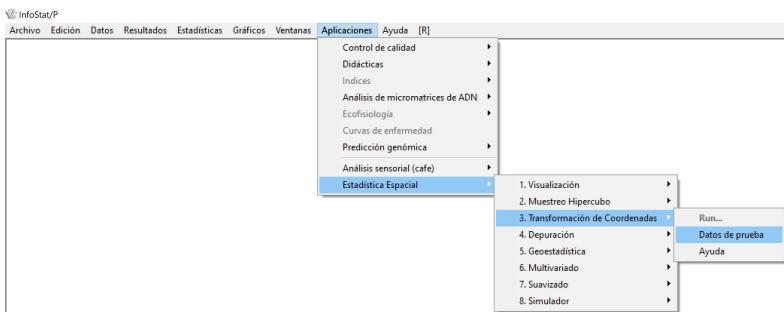
Implementación con InfoStat

El módulo “Estadística Espacial” de InfoStat permite realizar, a través de su conexión con el software R, la implementación de algoritmos univariados y multivariados para el análisis de datos georreferenciados. En esta sección se ilustrará el uso de este módulo para la transformación de coordenadas, la depuración de datos georreferenciados, el cálculo de índices de autocorrelación espacial, ajuste de semivariogramas, y la obtención de un mapa de variabilidad espacial mediante interpolación kriging. Complementariamente, se ilustra su aplicación con bases de datos de naturaleza multivariada para el cálculo de correlación bivariada, la implementación del análisis de componentes principales espaciales (MULTISPATI-PCA), la clasificación KM-sPC y el uso de métodos de regresión basados en árboles (*Random Forest*).

2.1 Conversión de coordenadas espaciales

Este menú permite la conversión entre distintos sistemas de coordenadas de referencias basándose en el uso de los códigos EPSG (<http://www.epsg.org/>). El archivo de datos que se abre desde el menú Aplicaciones → Estadística Espacial → Transformación de Coordenadas → Datos de prueba. El archivo se denomina **datosRinde.idb2** y con-

tiene 9810 observaciones (registros o filas del archivo) sobre datos de rendimiento de trigo (Rinde, $t ha^{-1}$) de un lote agrícola que fueron recolectados con un monitor de rendimiento. Cada valor registrado posee su ubicación geográfica (x e y) en unidades grado decimal que son expresadas en longitud (x) y latitud (y).



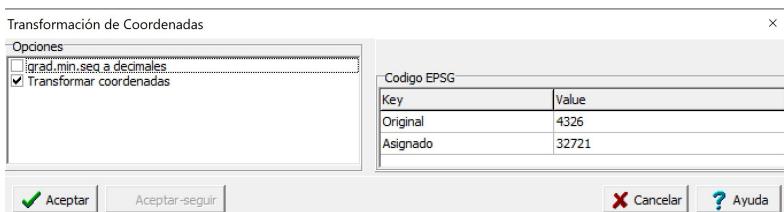
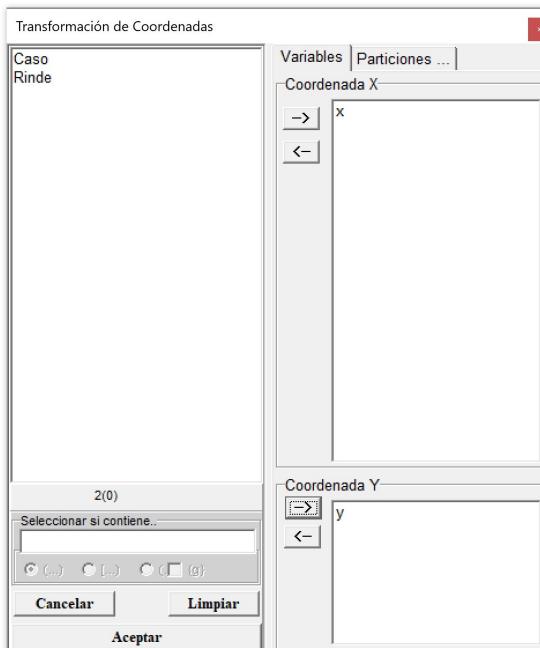
The screenshot shows a data table titled 'datosRinde'. The columns are labeled 'Caso', 'x', 'y', and 'Rinde'. The data rows are as follows:

Caso	x	y	Rinde
1	-59.13	-37.92	0.35
2	-59.14	-37.92	0.36
3	-59.13	-37.92	0.37
4	-59.13	-37.92	0.00
5	-59.13	-37.92	0.38
6	-59.14	-37.92	0.41
7	-59.13	-37.92	0.41
8	-59.13	-37.92	0.41
9	-59.14	-37.93	0.43
10	-59.14	-37.92	0.43
11	-59.14	-37.92	0.43
12	-59.13	-37.92	0.43
13	-59.14	-37.92	0.43
14	-59.13	-37.92	0.44
15	59.12	-37.92	0.44

At the bottom of the table, there is a summary row: 'Real | Registros: 9810*3 | n=1 Suma = -59.13 Media = -59.126 D.E. = 0.00 Min = -59.13 Max = -59.13 P05 = -59.13 P95 = -59.13'.

En el menú Aplicaciones → Estadística Espacial → Transformación de Coordenadas → Run, se abrirá una ventana de selector de variables donde debe indicarse la columna del archivo de datos que contiene la información correspondiente a la Coordenada X y la columna del archivo que contiene la información de la Coordenada Y. Luego, aparecerá una ventana para seleccionar el código EPSG original de los datos y código asignado. En este ejemplo el código

original para las coordenadas geográficas WGS84 es 4326 y el asignado para el sistema UTM zona 21 es 32721. En caso de ser necesario previo a esta transformación permite también el pasaje de datos que estén expresados en grados, minutos y segundos a grados decimales. Para ello sólo se debe tildar esta opción en el cuadro de dialogo. Los valores de las coordenadas deben presentarse con un espacio entre subunidades. Por ejemplo “33 1 1” expresan los grados minutos y segundos respectivamente.



Como resultado, en la tabla de datos original se agregarán dos nuevas columnas, al final del archivo, correspondiente a las coordenadas x e y transformadas, las cuales se

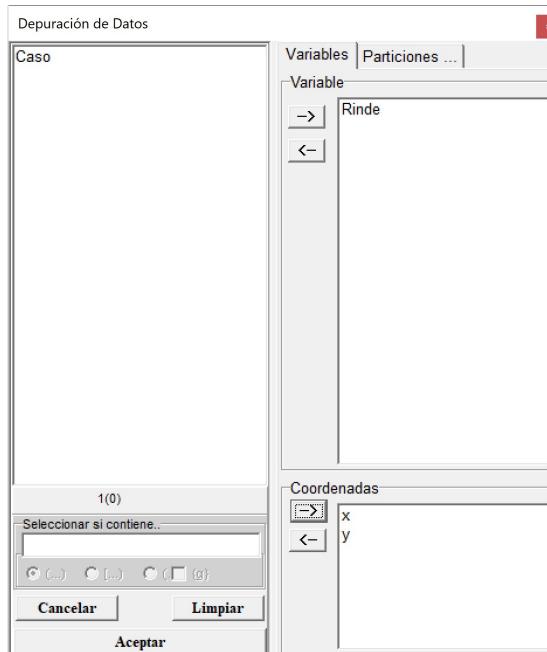
denominarán Xt e Yt, haciendo referencia a la variable x transformada (Xt) y a la variable y transformada (Yt). Estas nuevas variables (coordenadas transformadas) permiten interpretar la distancia entre puntos de Rinde en metros.

Caso	x	y	Rinde	Xt	Yt
1	-59.13	-37.92	0.35	313087.76	5800921.47
2	-59.14	-37.92	0.36	311983.13	5800810.92
3	-59.13	-37.92	0.37	312932.51	5800910.49
4	-59.13	-37.92	0.00	312685.33	5801277.33
5	-59.13	-37.92	0.38	312856.35	5800906.19
6	-59.14	-37.92	0.41	312227.25	5801004.49
7	-59.13	-37.92	0.41	313078.16	5800914.03
8	-59.13	-37.92	0.41	313109.16	5800916.18
9	-59.14	-37.93	0.43	312304.46	5800138.02
10	-59.14	-37.92	0.43	312045.70	5800529.79
11	-59.14	-37.92	0.43	312013.74	5800803.40
12	-59.13	-37.92	0.43	313044.15	5800912.92

Real Registros: 9810*5 n=1 Suma = -59.13 Media = -59.128 D.E. = 0.00 Min = -59.13 Max = -59.13 P05 = -59.13 P95 = -5

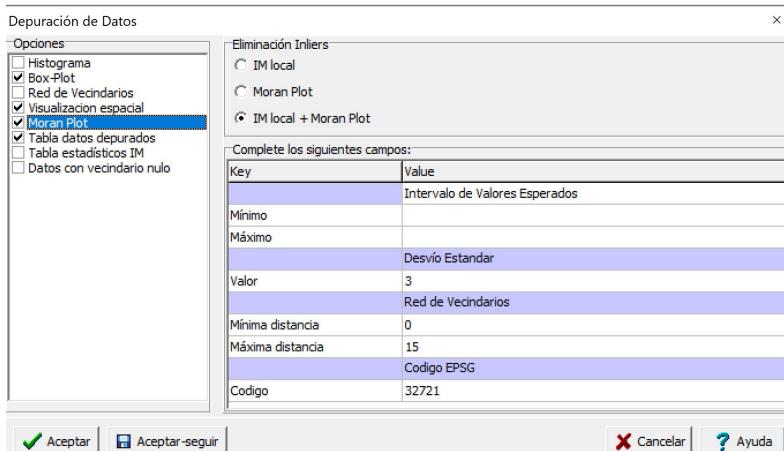
2.2 Eliminación de *outliers* e *inliers*

En el menú Aplicaciones → Estadística Espacial → Depuración → Datos de prueba se abrirá un archivo denominado **datosRinde_t.idb2** con 9810 registros y tres columnas correspondientes a las coordenadas planas UTM (x e y) de cada observación y los valores de la variable rendimiento de trigo ($t \text{ ha}^{-1}$). Luego, en el menú Aplicaciones → Estadística Espacial → Depuración → Run, se selecciona la variable respuesta, en este ejemplo Rinde y las columnas que indican la ubicación espacial de cada observación en el cuadro de Coordenadas.

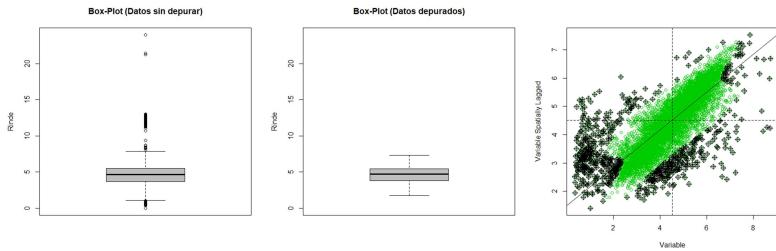


Al accionar el botón *Aceptar*, una nueva ventana permitirá seleccionar si desea visualizar los gráficos *Histograma*, *Box-Plot*, *Red de Vecindarios*, *Visualización espacial* y *Moran Plot*. También existe la opción de tildar que se genere una tabla nueva con los datos depurados, otra tabla con los estadísticos del Índice de Moran y una opción que permite que en la definición de los vecindarios existan algunos puntos que no presenten vecinos. Los valores *inliers* serán identificados a través del Índice de Moran local (*IM local*), *Moran Plot* o ambos (*IM local+Moran Plot*). En la misma ventana hay una opción de seleccionar el valor de desviación estándar (DE) de acuerdo al criterio del usuario. Considerando como valores *outliers* a los que se encuentren fuera del rango estimado como el valor medio \pm el valor de DE seleccionado. Por defecto presenta un valor de 3 DE. También el usuario previamente puede limitar los valores de los datos por un valor mínimo y/o máximo. Por defecto este filtro no se aplica.

En este ejemplo se seleccionaron las Opciones *Histograma*, *Box-Plot*, *Visualización espacial*, *Moran plot* y *Tabla datos depurados*. Además, se seleccionó que la eliminación de *inliers* se haga mediante el cálculo del índice de local de Moran y el Moran Plot (*IM local+Moran Plot*). Se utilizó una distancia de 15 para definir los vecindarios y un criterio de 3 DE para la eliminación de *outliers*. Por defecto el código EPSG es 32721 que correspondiente a la base de datos de prueba. En esta rutina el sistema de referencia es importante sólo para realizar la visualización espacial.



Como resultado se obtendrá el Box-Plot de la variable respuesta (Rinde) previo a la eliminación de los *outliers* e *inliers* (datos sin depurar), y luego de su eliminación (datos depurados). El gráfico de dispersión de Moran (Moran Plot) muestra en el eje horizontal los valores de la variable rendimiento mientras que en el vertical se representa el retardo espacial de la variable. Los puntos negros con forma romboidal son identificados como influyentes y se los considera como *inliers*.

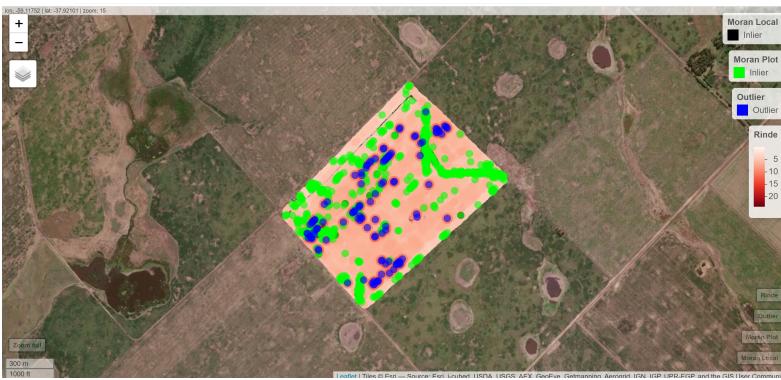


Como resultado de la depuración de datos en la base de datos original se crearán nuevas columnas al final del archivo con la clasificación de *outliers* y/o *inliers* para cada observación. Así, se adicionarán tres columnas denominadas *Outliers*, *Inliers_ML* e *Inliers_MP* según el método seleccionado. Estas variables son de tipo categóricas y contendrán la clasificación *Outliers* o *Normal* indicando si es considerado un valor extremo o no. En la columna de *Inliers_ML*, las categorías serán *Inliers* o *Normal* para indicar que no es un valor atípico respecto a sus vecinos cercanos y de la misma manera se indicarán los valores clasificados por el método de Moran Plot. En este ejemplo, 120 observaciones fueron clasificadas como *outliers*, 43 observaciones como *inliers* y 681 clasificadas como *inliers* según Moran Plot (en los que se encuentran incluidos los 43 casos considerados Inliers por el Índice de Moran Local). El procedimiento primero elimina los *outliers* por lo cual no serán considerados para su identificación como posibles *inliers*.

Caso	x	y	Rinde	MinMax	Outliers	Inliers_ML	Inliers_MP
1	313087.76	5800921.47	0.35	Normal	Normal	Normal	Inlier
2	311983.13	5800810.92	0.36	Normal	Normal	Normal	Inlier
3	312932.51	5800910.49	0.37	Normal	Normal	Normal	Inlier
4	312685.33	5801277.33	0.00	Normal	Outlier		
5	312856.35	5800906.19	0.38	Normal	Normal	Normal	Inlier
6	312227.25	5801004.49	0.41	Normal	Normal	Normal	Inlier
7	313078.16	5800914.03	0.41	Normal	Normal	Normal	Inlier
8	313109.16	5800916.18	0.41	Normal	Normal	Normal	Inlier
9	312304.46	5800138.02	0.43	Normal	Normal	Normal	Inlier
10	312045.70	5800529.79	0.43	Normal	Normal	Inlier	Inlier
11	312013.74	5800803.40	0.43	Normal	Normal	Normal	Inlier
12	313044.15	5800912.92	0.43	Normal	Normal	Normal	Inlier
13	311922.26	5800531.73	0.43	Normal	Normal	Normal	Inlier
14	313054.70	5800913.17	0.44	Normal	Normal	Normal	Inlier
15	313000.07	5800040.40	0.44	Normal	Normal	Normal	Inlier

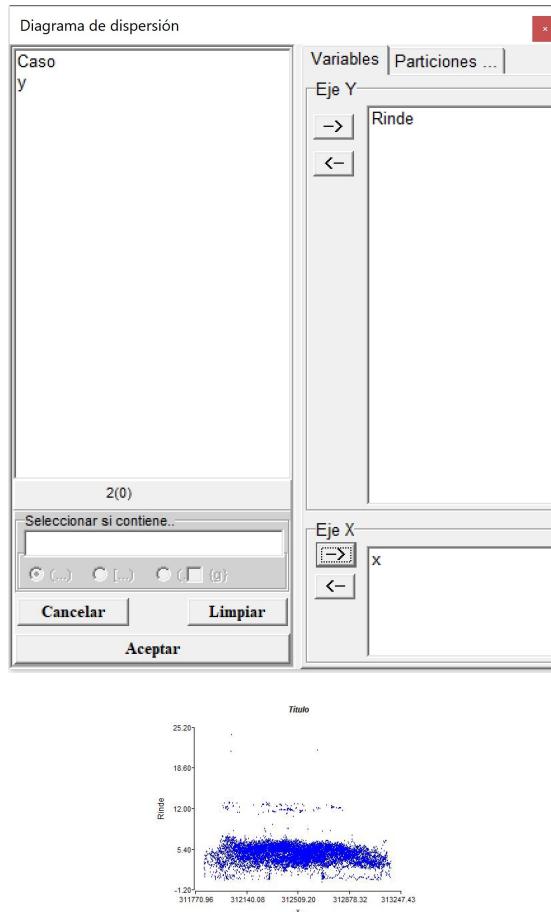
También se generará una nueva tabla de datos que contendrá las observaciones seleccionadas después de la depuración por *outliers* e *inliers*, que para este ejemplo contiene 9009 registros. Los valores seleccionados se muestran en la tabla denominada **Datos Depurados.idb2**. En caso que el usuario seleccione *Tabla estadísticos IM* se genera otra tabla **Estadísticos IM.idb2**, en la cual sus columnas tendrán los valores de índice local observado (li), el valor esperado (E.li), la varianza (Var.li), el estadístico (Z.li) y la significancia estadística a través del valor-p (Pr.z...0). La información del Moran Plot de los puntos influentes de la regresión proviene de diferentes estadísticos de diagnóstico como DFBETAS (dfb.1_ para la ordenada al origen y dfb.x para la pendiente), DFFITS (dffit), Covratio (cov.r), distancia de Cook (cook.d) y leverage (hat). Para cada uno de los estadísticos aparecerá una nueva columna del archivo con valores TRUE o FALSE según como ha sido clasificada la observación. En la ventana de Resultados de InfoStat se indica las tablas donde se encuentran estos resultados descriptos. Las nuevas tablas generadas son archivos temporarios, es decir, si desea guardar esta información, deberá ir al menú Archivo → Guardar tabla y seleccionar el directorio donde desea guardarlo. La opción de visualización espacial muestra un gráfico el cual se ejecuta como un ar-

chivo html en el navegador web. En este se podrá visualizar los datos detectados como *outliers* e *inliers*. Así mismo es posible cargar un mapa base como *esri.WorldImagery*, *OpenStreetMap*, *OpenTopoMap*.



2.3 Detección de tendencias espaciales

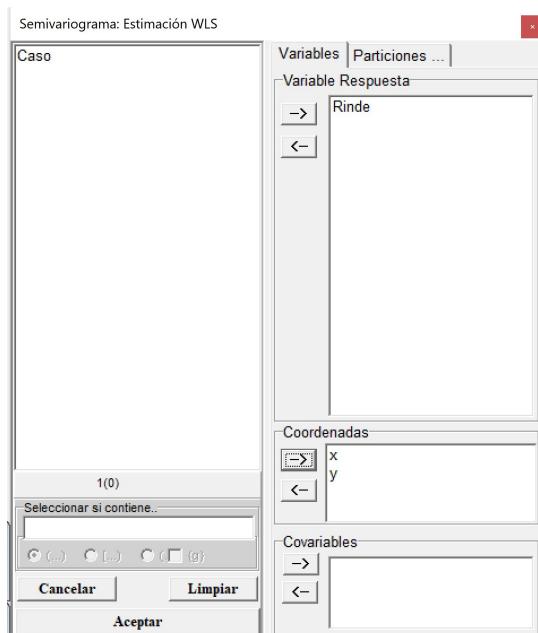
Un análisis gráfico para identificar de la presencia de tendencias se puede hacer con un gráfico de dispersión. Para ello ir al menú Gráficos → Diagrama de dispersión. En el casillero *Eje Y* colocar la variable *Rinde* y en *Eje X* la coordenada *x*. Al accionar Aceptar se generará el gráfico que muestra que no existe una tendencia entre ambas variables. El mismo procedimiento deber realizarse con la coordenada *y*.



También es posible ajustar de un modelo de regresión lineal entre la variable Rendimiento y las coordenadas x e y, se puede realizar yendo al menú Estadísticas → Regresión Lineal. En el selector de variables se coloca al rendimiento en el casillero de *Variable Dependiente* y las coordenadas x e y en el casillero *Regresoras*.

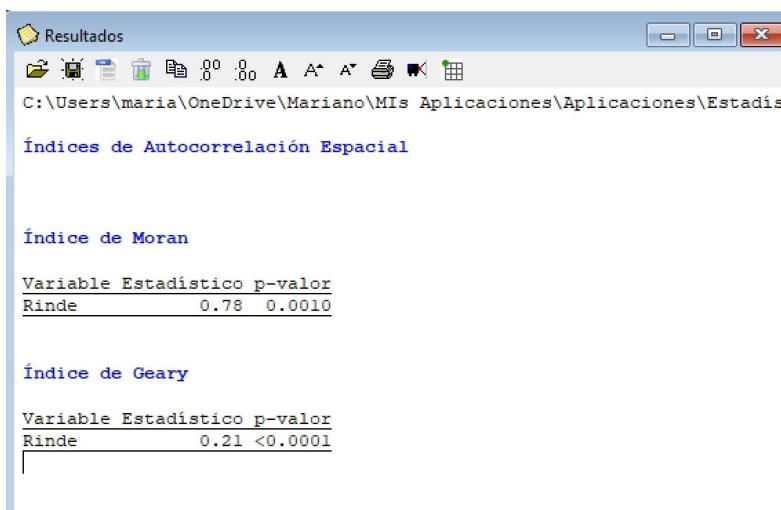
2.4 Cálculo del índice de Moran

Con la base de datos ya depurada en el punto 2.2 se procederá a calcular índices de autocorrelación espacial. El archivo de ejemplo **datosRinde_dep.idb2** también se encuentra disponible en el menú Aplicaciones → Estadística Espacial → Geoestadísticas → Semivariograma → WLS → Datos de prueba. Para cuantificar la magnitud de la estructuración espacial se estiman en este menú el índice de Moran y el índice de Geary. Para ello ir al menú Aplicaciones → Estadística Espacial → Geoestadísticas → Índices de Autocorrelación → Run. En el selector de variables colocar *Rinde* en el casillero *Variables* (aquí es posible colocar más de una) y en el casillero *Coordenadas* las columnas x e y.



Al accionar *Aceptar* en el siguiente cuadro se puede modificar el número de permutaciones que son utilizadas para

evaluar la significancia estadística de los índices a partir de simulación por Monte Carlo. Las ubicaciones son permutadas para obtener la distribución de los índices bajo hipótesis nula de distribución aleatoria. El cálculo también requiere la definición de una matriz de ponderación espacial, para este paso la red de vecindarios es definida usando la distancia Euclídea indicando el rango de distancia dentro de la cual dos observaciones serán consideradas colindantes o vecinas. El rango es estimado a partir de un valor de distancia máximo y uno mínimo. En este ejemplo se utiliza como distancia máxima 15 m. Además, se seleccionó la opción *Datos con vecindario nulo* la cual permite generar la matriz de pesos espaciales sin la restricción de que todos los puntos tengan al menos un dato vecino.



The screenshot shows the 'Resultados' window of the InfoStat application. The title bar reads 'Resultados'. The menu bar includes 'Archivo', 'Edición', 'Analisis', 'Visualizar', 'Ayuda'. The toolbar contains icons for file operations like Open, Save, Print, and zoom. The main area displays the following output:

```
C:\Users\maria\OneDrive\Mariano\MIS Aplicaciones\Aplicaciones\Estadisticas\Autocorrelacion Espacial\Resultados.txt
```

Índices de Autocorrelación Espacial

Índice de Moran

Variable	Estadístico	p-valor
Rinde	0.78	0.00010

Índice de Geary

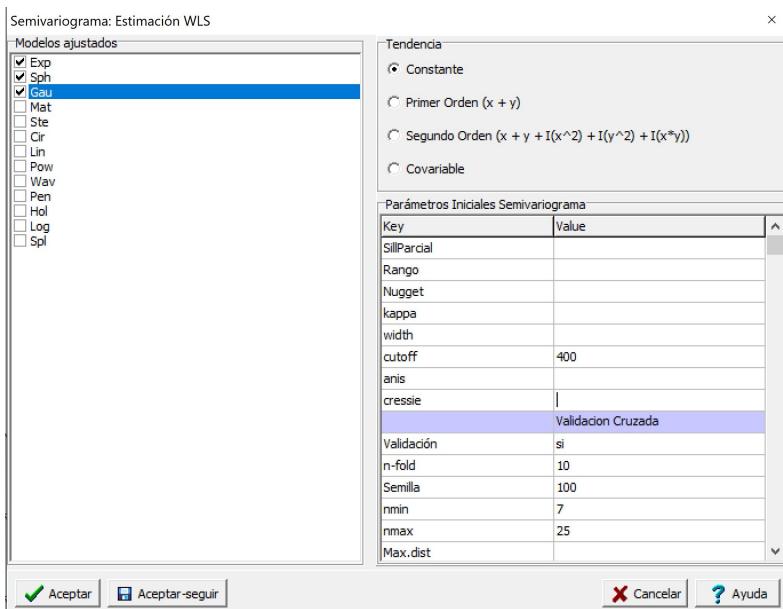
Variable	Estadístico	p-valor
Rinde	0.21	<0.0001

Para este ejemplo, tanto el índice de Moran como el índice de Geary indican una autocorrelación estadísticamente significativa (valor $p < 0.05$), es decir, no hay una distribución aleatoria de las observaciones en el espacio. La variable Rinde presentó una autocorrelación espacial positiva y alta (valor más cercano a 1 en el caso del índice de Moran y

mas cercano a 0 en el de Geary sugieren autocorrelaciones positivas más fuerte).

2.5 Análisis basado en semivariogramas

Para acceder al archivo de ejemplo **datosRinde_dep.idb2** se debe ir al menú Aplicaciones → Estadística Espacial → Geoestadísticas → Semivariograma → WLS → Datos de prueba. Luego se acciona la opción *Run* siguiendo la misma ruta de acceso. Esta opción realizará el ajuste de un semivariograma empírico y teórico usando el método de mínimos cuadrados ponderados (WLS). En el selector de variables se debe colocar la variable *Rinde* en el casillero *Variables* y en el casillero *Coordenadas* las columnas x e y.



Al accionar el botón *Aceptar*, aparecerá una nueva ventana

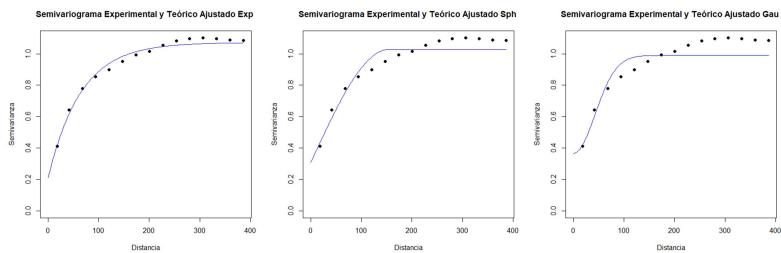
donde se podrá seleccionar la función de semivariograma: Exp (exponencial), Sph (esférico), Gau (gaussiano), Mat (Matern), Ste (parametrización de Matern Stein), Cir (circular), Lin (lineal), Pow (potencia o power), Wav (ondulado o wave), Pen (pentaesférico), Hol (holístico o hole), Log (logarítmico) y Spl (spline). Como opciones se puede seleccionar ajustar un semivariograma sin tendencia (*Constante*), con tendencia de primer orden, de segundo orden y con covariable. También es posible indicar los valores iniciales para los parámetros del semivariograma a ajustar (SillParcial, Rango y Nugget). En este ejemplo no se colocaron estos por lo cual la función estima valores iniciales razonable para realizar el ajuste. El parámetro *kappa* es opcional para los modelos Matern y Ste (parametrización de Matern Stein). También, puede ser fijado ingresando un valor en la opción *Kappa*. Si en lugar de colocar un valor numérico se coloca un carácter, se ajustará un *kappa* óptimo en un rango entre 0.05 y 5. La opción de parámetro *width* refiere a la amplitud del intervalo de distancia sobre la cual los pares de puntos agrupados para la estimación de la semivarianza. Por ejemplo, si la distancia máxima entre los pares de puntos es de 1000 y se selecciona un valor de *width*=100, se conformarán 10 grupos de una amplitud de 100 m. Para cada uno de esos 10 grupos se estimará la semivarianza. El *cutoff* es la máxima distancia de separación espacial hasta la cual los pares de puntos son tenidos en cuenta para la estimación de la semivarianza. Si la opción queda vacía, es decir no se coloca ningún valor, por defecto estima el *cutoff* como el tercio de la línea diagonal de una caja que contiene los datos. La función por defecto ajusta un semivariograma isotrópico. Para evaluar si el proceso es anisotrópico es posible el ajuste de semivariogramas direccionales. Esto puede hacer desde el menú Aplicaciones → Estadística Espacial → Geoestadísticas → Semivariograma → Direccionales. Para incorporar la anisotropía al modelo,

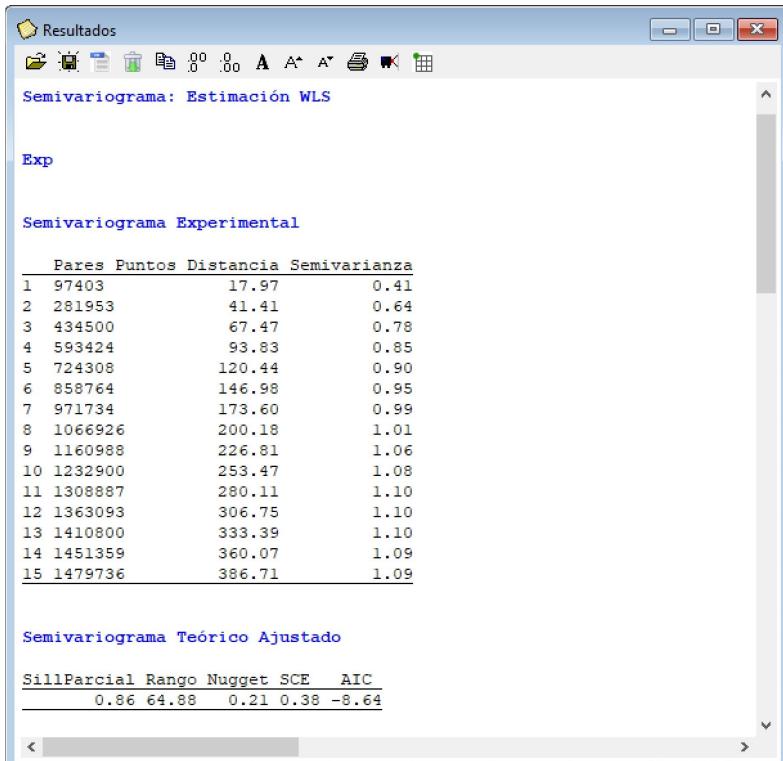
en la opción *anis*, se debe colocar la dirección de mayor correlación espacial i.e mayor rango (valor entre 0° y 360° , medidos en el sentido de las agujas del reloj, donde el Norte es 0° .) y el cociente de anisotropía (cociente entre el mayor y el menor rango que se producen en las direcciones evaluadas, valor entre 0 y 1). La dirección y el cociente de anisotropía deben ir separados por coma. El separador de decimales es el punto. Por ejemplo *anis=c(90, 0.2)* indica que el mayor rango se produce a las 3 en punto (dirección este) con una diferencia de 5 veces en el rango.

La opción *cressie* permite el ajuste del semivariograma teórico mediante el estimador robusto de Cressie- Hawkins. Para seleccionar este estimador debe colocarse un *si* a esta opción, caso contrario utilizará el estimador de los momentos de Matheron. Esta función permite realizar una validación cruzada el tipo k-fold. Para ello se debe colocar *si* en la opción *Validación*. El número de grupos *k* se coloca en la opción *n-fold*. Además, es posible fijar la semilla para que la asignación de cada dato a los grupos *k* no cambie en caso de repetir el proceso. Las opciones *nmin*, *nmax* y *Max.dist* se utilizan para que la función kriging, usada en el proceso de validación, se realice en un contexto local. Por defecto se definen vecindarios con un número mínimo y máximo de vecinos de cada punto a predecir siendo estos de 7 y 25, respectivamente. En caso de omitir estos últimos argumentos la interpolación se realiza en un contexto global.

En este ejemplo, se ajustó un semivariograma experimental a partir de los datos observados y se probaron los modelos exponencial, esférico y gaussiano como modelos de semivariograma teóricos. Sus parámetros iniciales fueron estimados por defecto por la función. El valor del *cutoff* se fijó en 400 m. Además, se realizó una validación cruzada con *k=10*. Todas las otras opciones se dejaron por defecto.

Al accionar *Aceptar* se generan los gráficos con los tres modelos ajustados. La ventana resultados muestra también la información del semivariograma empírico y cada uno de los modelos teóricos ajustados. Además, se proporcionan los valores de los parámetros estimados, criterios de bondad de ajuste (SCE y AIC), y error de predicción de los modelos ajustados. En este ejemplo siguiendo los criterios de SCE y AIC el modelo de mejor ajuste (valores más bajos de estos indicadores) fue el exponencial. Los valores de los parámetros fueron: Sill Parcial=0,86, rango=64,88, efecto Nugget=0,21. Los resultados de la validación cruzada muestran pequeñas diferencias entre los errores de predicción de los modelos ajustados. Para el caso del modelo exponencial la RMSE relativa a la media de los valores observados (nRMSE) fue del 11,82 % mientras cociente de la desviación cuadrática media (MSDR) fue de 0,60 (más cercano a 1 mejor modelo).





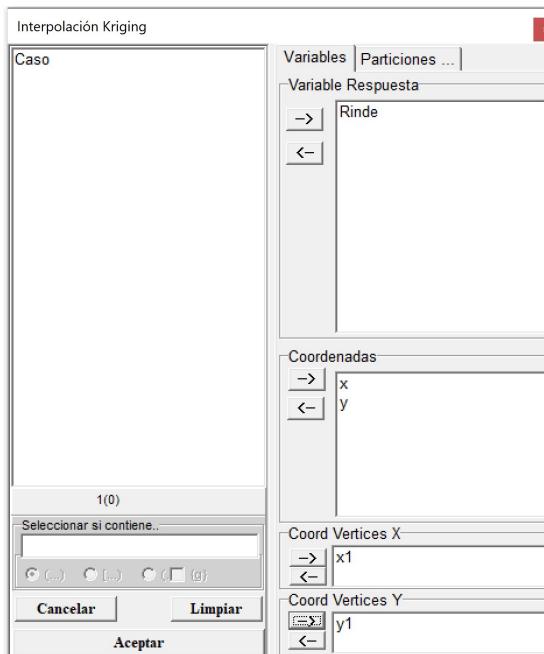
2.5.1 Mapeo de variabilidad espacial

Para realizar la interpolación espacial mediante kriging se utilizará la base de datos **datosRinde_limites.idb2** que se encuentra en el menú Aplicaciones → Estadística Espacial → Geoestadísticas → Interpolación → Kriging → Datos de prueba. El archivo de datos contiene además de las coordenadas (x e y), la variable respuesta (Rinde) y dos nuevas columnas (x1 e y1) que contienen los vértices del polígono donde se desea realizar la interpolación. Estas últimas no son obligatorias para realizar la interpolación. En caso de omitirse la función tomara los puntos más externos de la base de datos y a partir de este formara un polígono que definirán los límites del área a interpolar.

Caso	x	y	Rinde	x1	y1
1	312412.94	5801084.58	2.90	311841.92	5800614.28
2	313076.61	5800781.65	2.90	311950.32	5800727.67
3	312102.37	5800915.58	2.90	311997.97	5800788.24
4	312573.63	5800296.63	2.91	312005.54	5800834.60
5	312715.69	5800476.30	2.91	312566.04	5801430.92
6	312089.18	5800923.27	2.91	312589.86	5801434.79
7	312964.91	5800666.41	2.91	313161.62	5800903.11
8	313008.43	5800878.80	2.91	313166.60	5800857.50
9	312004.74	5800819.85	2.91	312353.47	5800088.10
10	312687.03	5800425.79	2.91	312330.94	5800089.06
11	312481.94	5800210.70	2.91		
12	312470.98	5800213.11	2.91		
13	312927.89	5800631.15	2.91		
14	312500.27	5800228.22	2.91		
15	311000.10	5800707.05	2.91		

Real | Registros: 9009*5 | n=1 Suma = 312413 Media = 312413 D.E. = 0 Min = 312412.94 Max = 312412.94 P05 = 312412.9

Si se vuelve a la misma ruta de accesos y se acciona *Run* se abrirá el selector de variables donde se colocará la variable *Rinde* en el casillero *Variable Respuesta*, x e y en *Coordenadas* y las columnas x1 e y1 en los casilleros *Coord. Vértices X* y *Coord. Vértices Y*, respectivamente.

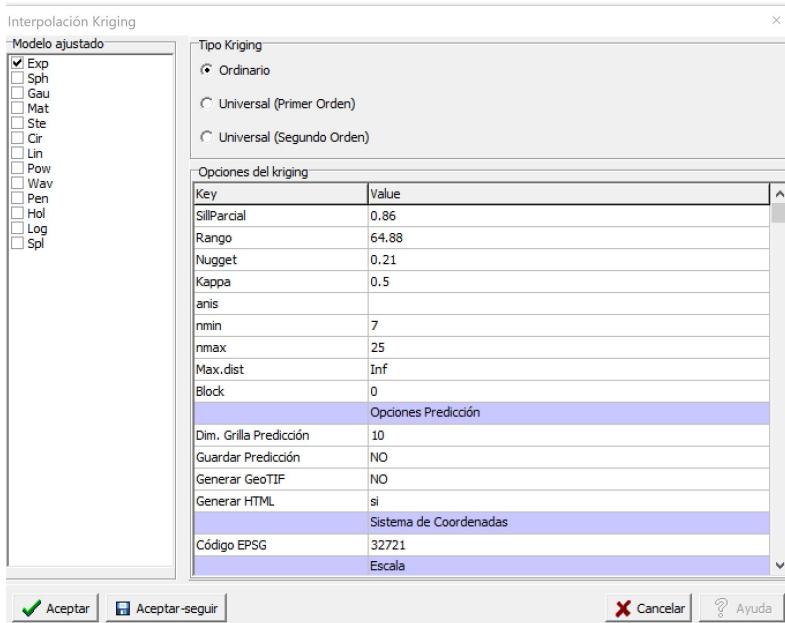


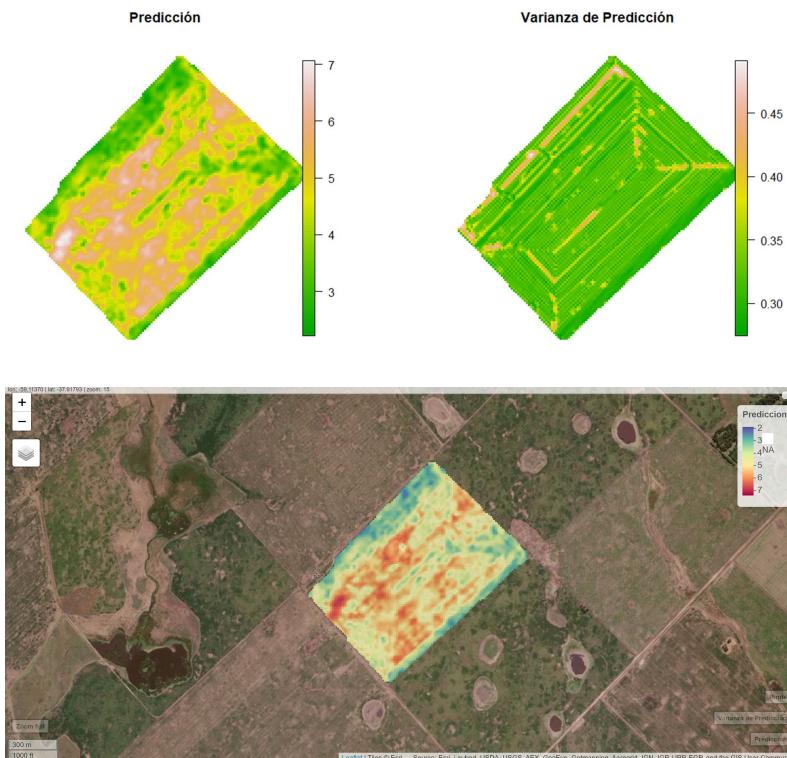
Luego de accionar *Aceptar*, se genera una ventana con diferentes opciones para seleccionar el tipo de modelo ajustado

(Exp, Sph, Gau, Ste, Cir, Lin, Pow, Wav, Pen, Hol, Log, y Spl), el tipo de kriging, ordinario sin tendencia o universal (incorporando la tendencia en primer o segundo orden) y los valores para realizar la predicción espacial. En este caso se utilizará el modelo seleccionado en el punto anterior. Así, se selecciona el modelo exponencial y se fijan los valores de los parámetros Sill Parcial=0.86, rango=64,88, efecto Nugget=0.21. En este caso la predicción se realiza en un contexto local (opción por defecto). Para ello se fija un número mínimo (*nmin*) y máximo (*nmax*) de puntos que son utilizados para realizar la predicción en cada uno de los sitios de la grilla de predicción. Otra opción para determinar cuáles son los puntos que aportan información para la predicción de un sitio determinado es usando una medida de distancia (*Max.dist*). Puntos que se ubican más allá de la distancia máxima determinada por el usuario respecto a la posición sobre la que se quiere predecir el valor de la respuesta, no serán utilizados para la predicción. En caso de requerir que la predicción se realice utilizando toda la información disponible (kriging global), las opciones *nmin*, *nmax* y *Max.dist* se dejan vacías. La predicción también puede realizarse de manera puntual o en bloque. En este ejemplo donde bloque (Block) es cero, la predicción será local.

Para realizar la predicción se requiere definir los puntos donde se realiza la interpolación en *Opciones de Predicción*. Con *Dim. Grilla Predicción* se fija la dimensión de la grilla de predicción, en este caso, 10 indica que tiene una dimensión de 10×10 m. También puede fijarse el sistema de coordenadas que tendrán los mapas generados usando el código EPSG. Para estos datos el código es 32721 que corresponde al sistema de coordenadas UTM, zona 21, hemisferio sur. Esto es importante para proyectar correctamente los mapas generados. Es posible visualizar estos

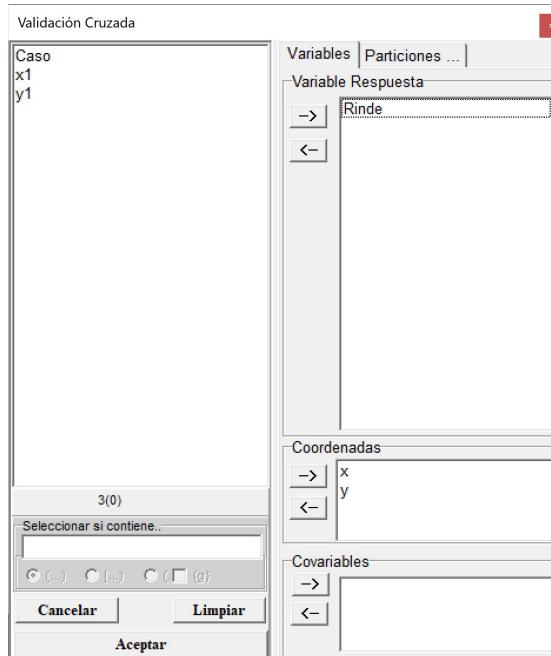
mapas en el navegador web colocando *si* en la opción *Generar HTML* o exportarlos como geotiff (opción *Generar GeoTIF*). En caso de colocar *si* en esta última opción el software abrirá una ventana para elegir el directorio en el cual se guardará el archivo generado. Los valores de la predicción también pueden ser colocados en forma de tabla mediante la opción *Guardar Predicción*. Finalmente, también es posible cambiar los valores mínimos y máximos de las escalas de valores de los mapas que se generan (Predicción y Varianza de Predicción).



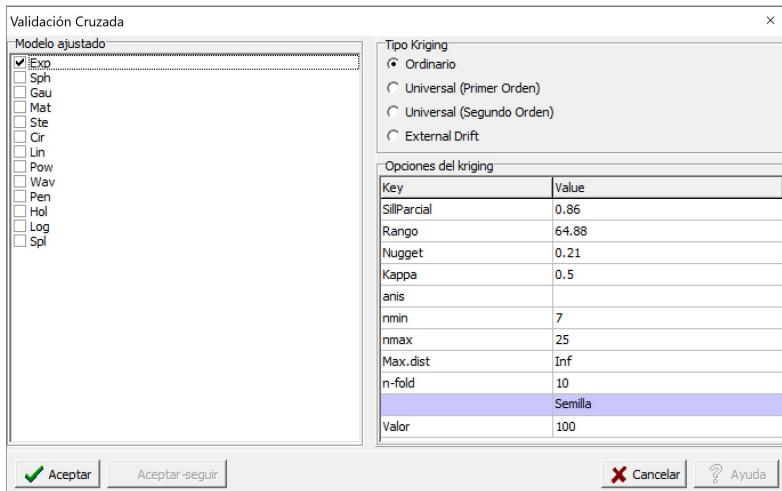


2.5.2 Validación cruzada

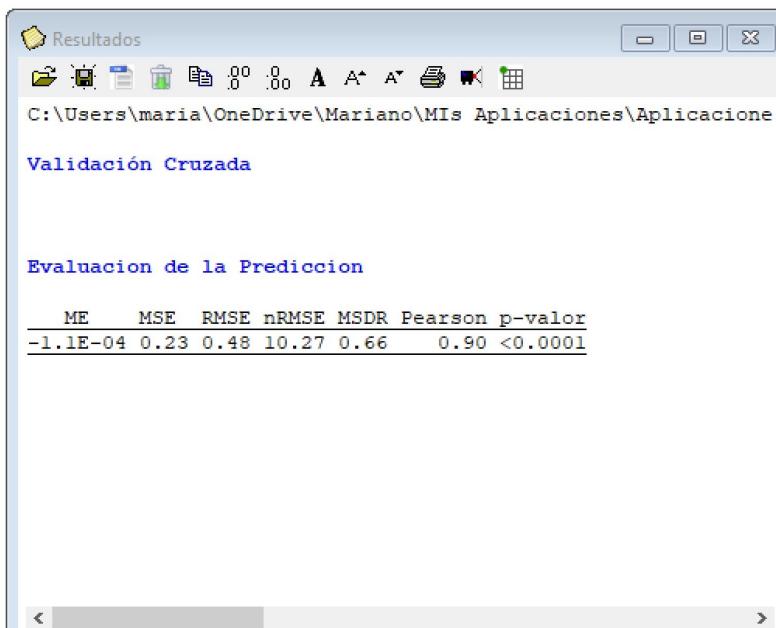
La evaluación de la capacidad predictiva de los modelos ajustados que se realizó en el punto 2.5, también puede realizarse desde el menú Aplicaciones → Estadística Espacial → Geoestadísticas → Interpolación → Validación Cruzada. En el selector de variables se coloca la variable *Rinde* en el casillero *Variable Respuesta* y las coordenadas en el cuadro *Coordenadas*.



En la siguiente ventana se coloca la información del modelo a evaluar, en este caso exponencial con parámetros Sill Parcial= 0.86, rango=64.88 y efecto Nugget=0.21. La predicción kriging se realiza en un contexto local (opción por defecto) con $nmin=7$ y $nmax=25$. El número de grupos de la validación cruzada es de $k=10$. La función permite también calcular el error de predicción para kriging universal (tendencias de primer y segundo orden) y kriging con deriva externa.



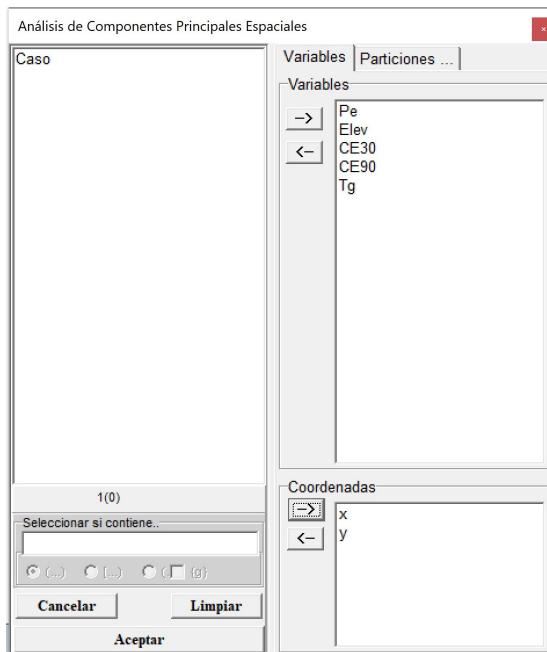
Para este ejemplo la RMSE relativa a la media de los valores observados (nRMSE) fue del 10,27 % mientras cociente de la desviación cuadrática media (MSDR) fue de 0.66 (más cercano a 1 mejor modelo).



2.6 Caracterización de variabilidad espacial con múltiples capas de datos

2.6.1 Análisis de componentes principales

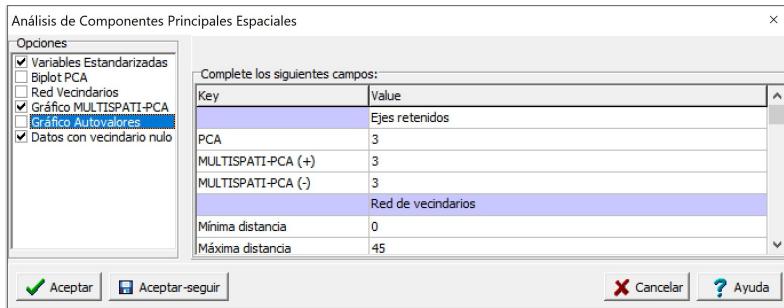
La base de datos **Pred2.idb2** se encuentra disponible en el menú Aplicaciones → Estadística Espacial → Geoestadísticas → Multivariado → a. MULTISPATI-PCA → Datos de prueba. En el selector de variables se colocan las variables Pe, Elev, CE30, CE90 y Tg en el cuadro *Variables* y las coordenadas x e y en *Coordenadas*.



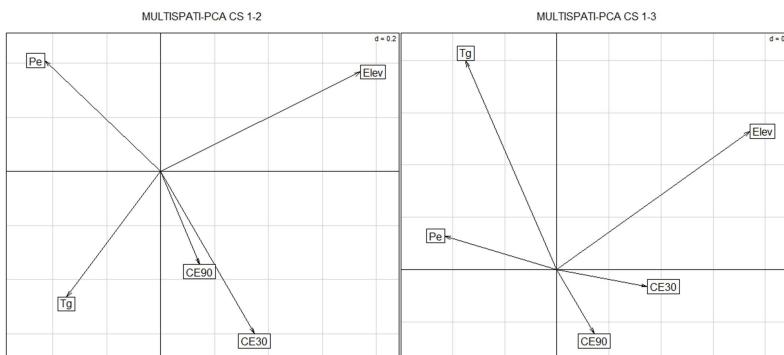
En la siguiente ventana se presentan las opciones del MULTISPATI-PCA. Aquí se puede seleccionar estandarizar las variables (opción por defecto), generara gráficos (biplot del PCA, gráfico de la red de vecindarios, gráficos del MULTISPATI-PCA y gráfico de los autovalores) y solicitar

2.6 Caracterización de variabilidad espacial con múltiples capas de datos 157

que la conformación de la matriz de pesos espaciales admita datos con vecindarios nulo. Las otras opciones del análisis implican fijar el número de ejes retenidos por el PCA y por el MULTISPATI-PCA. Para este último caso pueden ser aquellos que presenten una autocorrelación positiva (*MULTISPATI-PCA (+)*) o negativa (*MULTISPATI-PCA (-)*). La parametrización para este ejemplo se muestra en la siguiente figura:



El gráfico obtenido del MULTISPATI-PCA muestra que las variables Elev y Pe son las más importantes en la explicación de la variabilidad espacial a nivel del primer eje (sPC1, eje horizontal). Mientras que la CE30 y Tg presentan mayor importancia en la SPC2. Además, se observa una correlación positiva entre CE30 y CE90, y negativa entre estas dos y la Pe. También la Elev y Tg se correlacionan en forma negativa.



Los resultados muestran que con MULTISPATIPCA se explica una menor proporción de la varianza acumulada en el primer eje, respecto de PCA (1,81 vs. 1,94). Las tres primeras CP del PCA explican 78 % de la variabilidad total mientras que la CS1, CS2 CS3 del MULTISPATI el 73 %. No obstante, los valores del índice de Moran calculados para las tres primeras CPs sugieren que la estimación de autocorrelación aumentó cuando se usó MULTISPATIPCA respecto de la contenida en las CPs del PCA (0,79 vs. 0,68 para el eje 1, 0,43 vs. 0,40 para el eje 2, 0,50 vs. 0,25 para el eje 3).

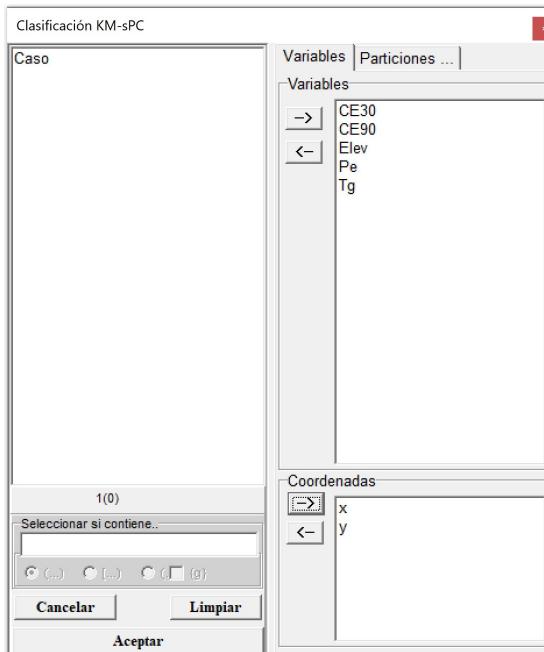
Resultados					
CPre.	CPe(-)ret.	CPe(+)ret.	Dist.min	Dist.max	
3.00	3.00	3.00	0.00	45.00	
Resultados del Análisis de Componentes Principales					
Eje Autovalores Proporción Prop. Acum. Índice de Moran					
1	1.94	0.39	0.39	0.68	
2	1.08	0.22	0.60	0.40	
3	0.88	0.18	0.78	0.25	
Resultados del MULTISPATI-PCA					
Eje Autovalores Varianza Espacial Proporción Prop. Acum. Índice de Moran					
1	1.43	1.81	0.36	0.36	0.79
2	0.47	1.10	0.22	0.58	0.43
3	0.37	0.75	0.15	0.73	0.50
Autovectores					
Variable CS1 CS2 CS3					
Pe	-0.43	0.41	0.13		
Elev	0.74	0.37	0.53		
CE30	0.35	-0.60	-0.06		
CE90	0.14	-0.35	-0.25		
Tg	-0.35	-0.46	0.80		

2.6.2 Análisis de conglomerados

Para realizar la ilustración se utilizará la base de datos **Pred.idb2** que se encuentra en el menú Aplicaciones → Estadística Espacial → Geoestadísticas → Multivariado → c. Clasificación KMsPC → Datos de prueba. El archivo de datos contiene además de las coordenadas (x e y), valores

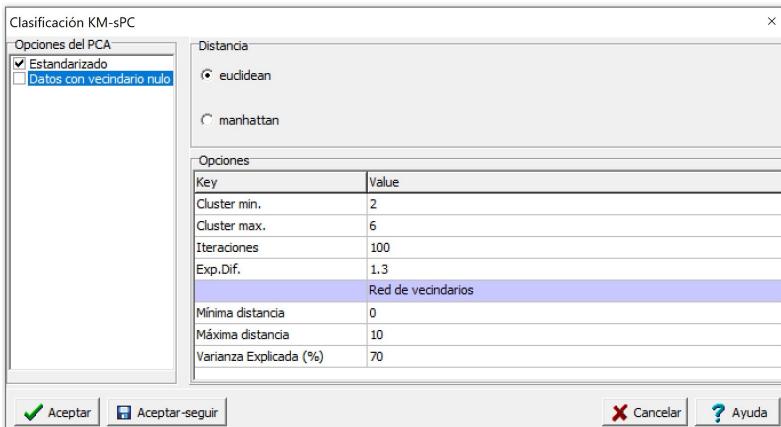
2.6 Caracterización de variabilidad espacial con múltiples capas de datos 159

de mediciones de conductividad eléctrica aparente en dos profundidades 0-30 cm (CE30) y 0-90 cm (CE90), elevación (Elev), profundidad de suelo (Pe) y rendimiento de trigo (Tg). Las variables CE30, CE90, Elev Pe y Tg se colocan en el casillero *Variables* y las coordenadas x e y en el cuadro *Coordenadas*.



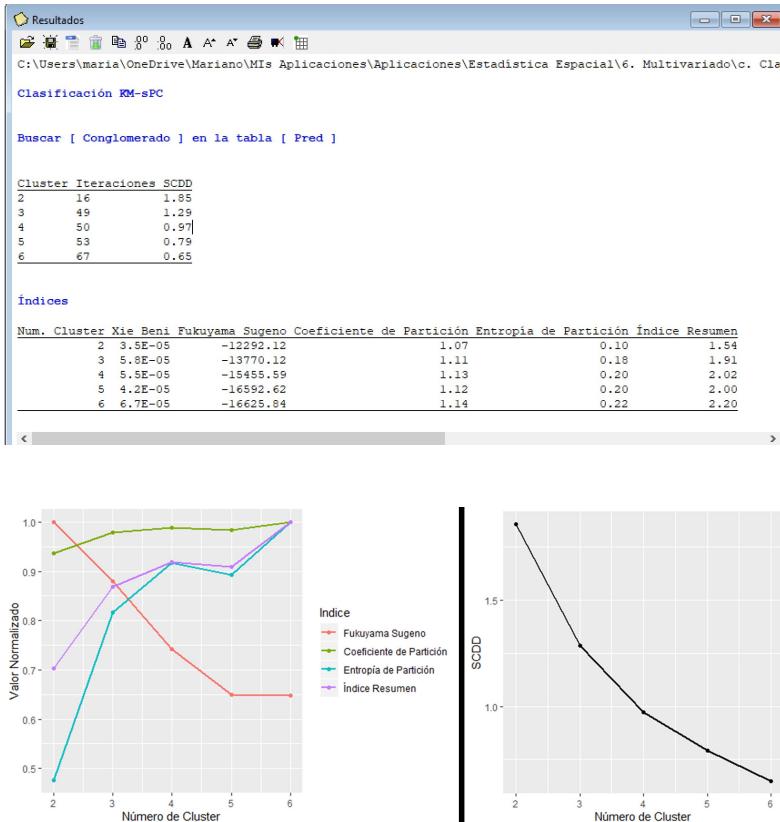
Al accionar *Aceptar*, aparecerá la ventana para seleccionar opciones del método de análisis. El método KM-sPC primero realiza un análisis de componentes principales espaciales (MULTISPATI-PCA) sobre las variables originales. Luego las variables sintéticas (componentes principales espaciales, sPC) son utilizadas como input del análisis de clúster fuzzy k-means. Por ello, es posible estandarizar las variables para realizar el MULTISPATI-PCA y elegir la opción que permite conformar una matriz de pesos espaciales en presencia de datos con vecindarios nulo. Otras opciones del análisis incluyen la distancia (euclídea o manhattan) utilizada en el método de cluster, el número mínimo y máximo de

clúster a generar, el número de iteraciones y el exponente difuso. En este ejemplo se usaron las siguientes opciones: 2 hasta 6 cluster, 100 iteraciones y un valor de 1.3 para el exponente difuso. Para el cálculo de la red de vecindarios, necesario para realizar el MULTISPATI-PCA, la distancia mínima y máxima fue de 0 y 10 m, respectivamente. La opción de varianza explicada (%) fue del 70, lo que indica que seleccione la cantidad de ejes (componentes principales espaciales) necesarios tal que la varianza total explicada sea mayor o igual a 70 %.



En la ventana resultados se muestra la suma de cuadrados de distancias dentro (SCDD) la cual puede usarse para determinar el número de clúster óptimo. Para esto otros índices son calculados como Xie-Beni, Fukuyama Sugeno, Coeficiente de Partición, Entropía de Partición. Para todos ellos un menor valor del índice implica mejor clasificación. Dado que muchas veces los índices no coinciden se adiciona el cálculo de un índice resumen. Los resultados muestran que, para la mayoría de los índices, incluyendo el resumen, el número de clúster optimo es 2. Cuando se ejecuta el análisis los índices también son graficados en forma conjunta usando una escala normalizada. En este caso también un menor

valor del índice en la escala normalizada implica una mejor clasificación.



2.7 Predicción con múltiples capas de datos

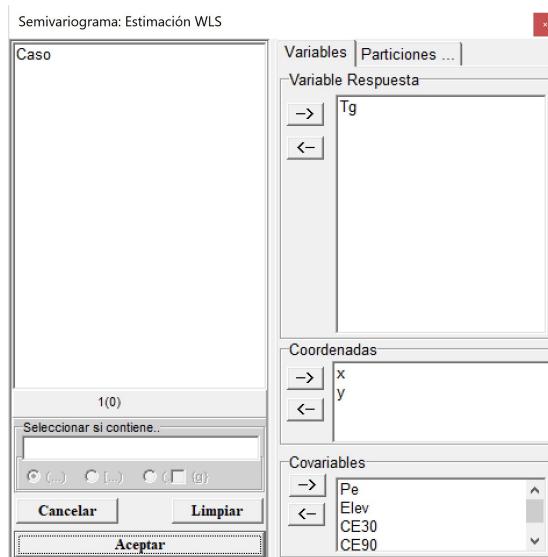
Los datos de ilustración **Pred2.idb2** serán utilizados para cada de las alternativas de predicción disponibles en el menú. El archivo de datos contiene además de las coordenadas (x e y), valores de mediciones de conductividad eléctrica aparente en dos profundidades 0-30 cm (CE30)

y 0-90 cm (CE90), elevación (Elev), profundidad de suelo (Pe) y rendimiento de trigo (Tg).

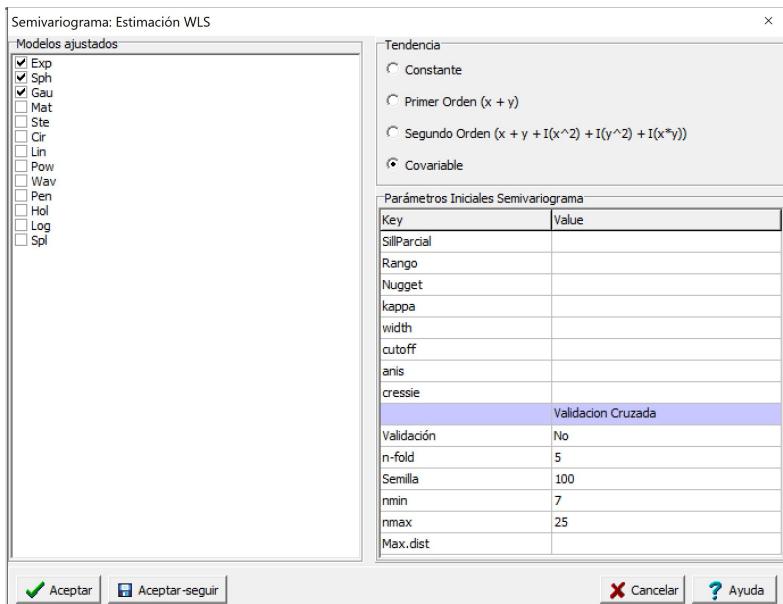
2.7.1 Kriging con deriva externa

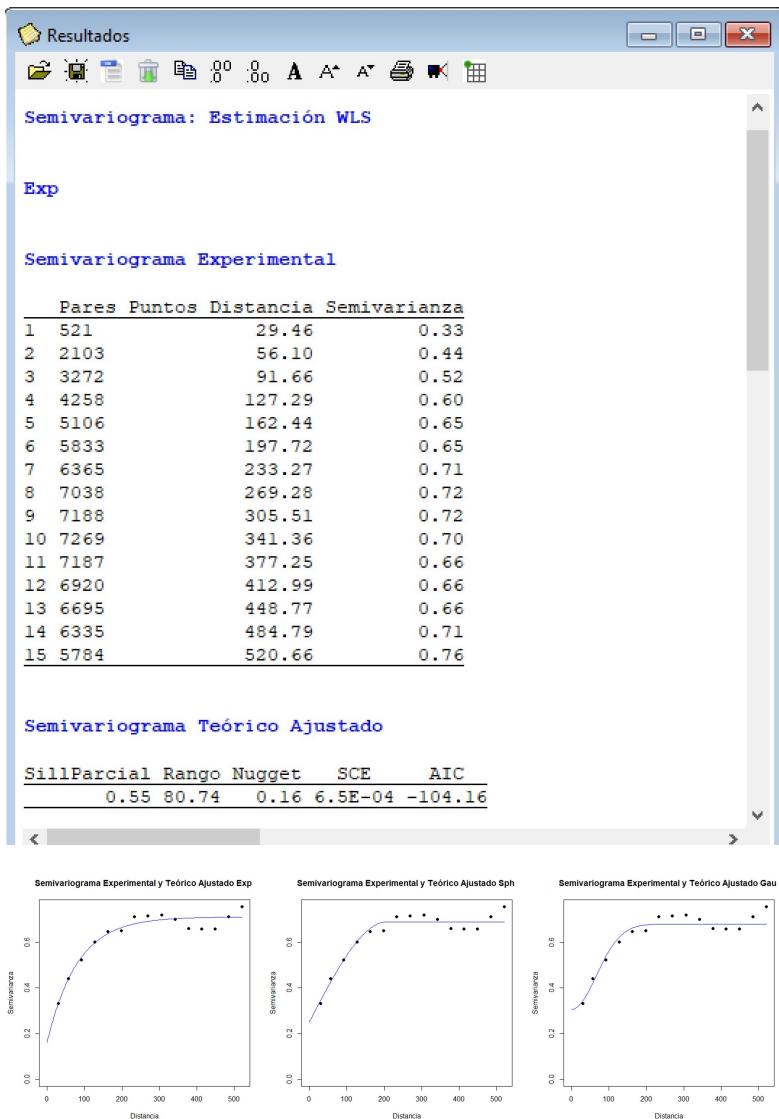
El acceso al menú es a partir de las opciones: Aplicaciones → Estadística Espacial → Geoestadísticas → Interpolación → c. Kriging (KED) → Datos de prueba. Para realizar la interpolación se necesita de una grilla de predicción previamente generada en formato de archivo .txt que cuente con la información de las coordenadas (x e y) en las dos primeras columnas y de cada una de las covariables que se usaron en el ajuste del modelo. Se requiere que los nombres de las covariables sean los mismos tanto en la grilla de predicción como en la tabla de observaciones. El archivo **grilla_am.txt** contiene esta información.

En primer lugar se procede al ajuste de los semivariograma experimental teórico. Para ello se accede al menú Aplicaciones → Estadística Espacial → Geoestadísticas → b. Semivariograma → WLS → Run. Las variables CE30, CE90, Elev Pe y Tg se colocan en el casillero *Variables*, las coordenadas x e y en el cuadro *Coordenadas* y Pe, Elev, CE30 y CE90 en el cuadro *Covariables*.



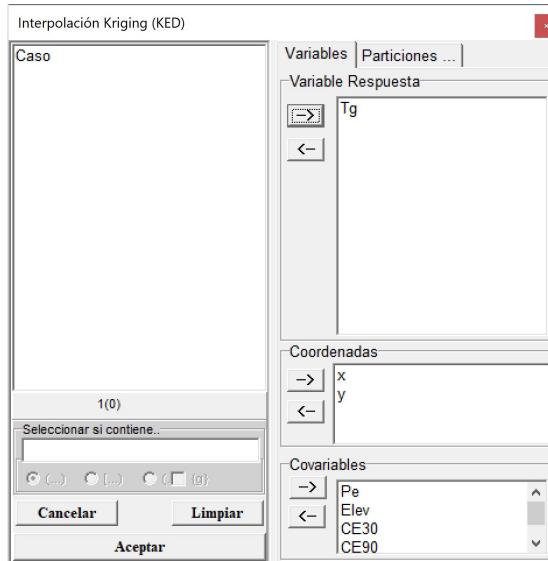
En la siguiente ventana se seleccionan los modelos teóricos a ajustar, en este ejemplo exponencial esférico y gaussiano. En la opción *Tendencia* se selecciona *Covariables*. Las otras opciones de ajuste de los semivariogramas se dejan por defecto. Los resultados muestran que el modelo de mejor ajuste según los valores de SCE y AIC es el exponencial. Los semivariogramas teóricos ajustados se despliegan en forma automática.



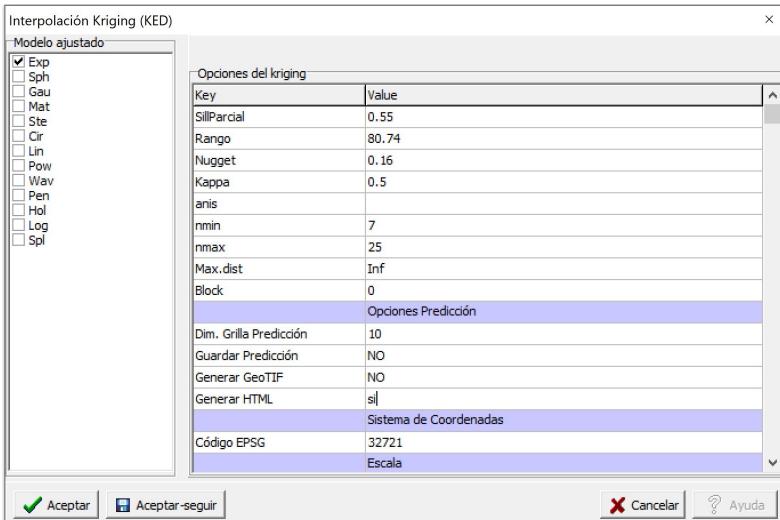


Con los valores de los parámetros estimados del semivariograma exponencial se procede a realizar la interpolación espacial. Para ello se accede al menú Aplicaciones → Estadística Espacial → Geoestadísticas → Interpolación → c. Kriging (KED) → Run. Las variables CE30, CE90, Elev Pe y Tg se colocan en el casillero *Variables*, las coordenadas x

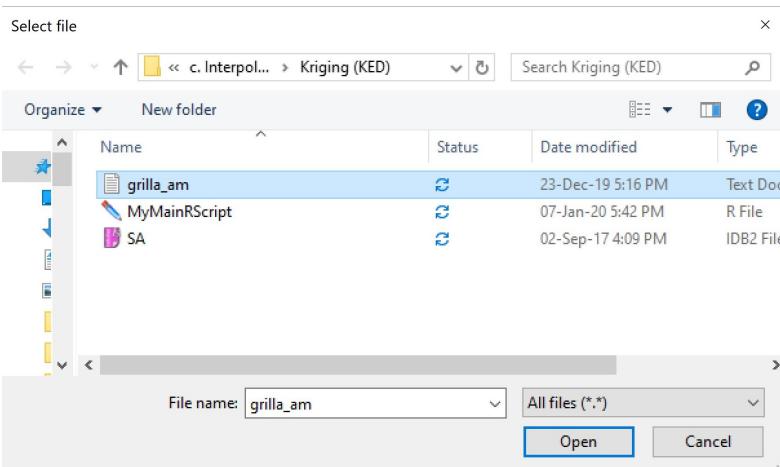
e y en el cuadro *Coordenadas* y Pe, Elev, CE30 y CE90 en el cuadro *Covariables*.

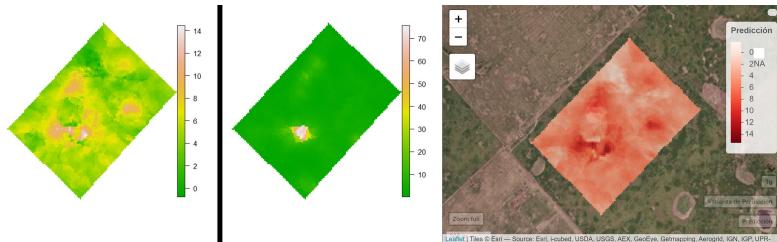


En la ventana siguiente se selecciona el modelo de semivariograma exponencial y se colocan los valores de los parámetros estimados en el paso anterior. En este ejemplo se selecciona la opción para mostrar el mapa de predicción en una ventana del navegador web (opción *Generar HTML*). Para ello es importante fijar el sistema de coordenada mediante el código EPSG.



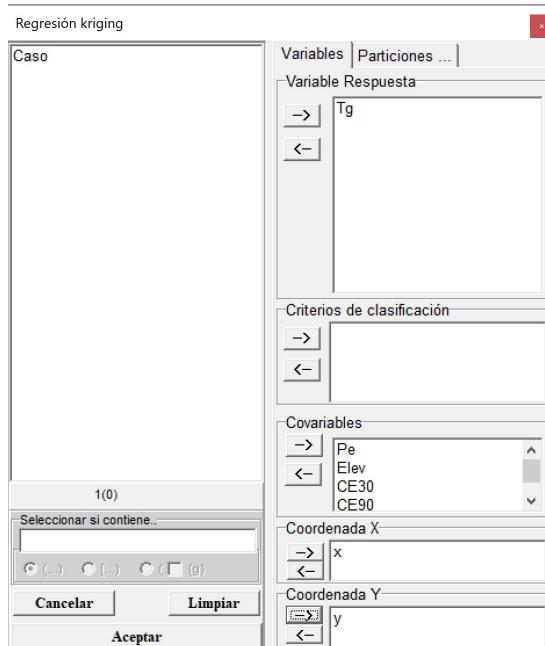
Al *Aceptar* se abrirá una ventana que permite seleccionar la grilla de predicción que en este ejemplo se denomina **grilla_am.txt**. Posteriormente se generarán los mapas de predicción y varianza de predicción.





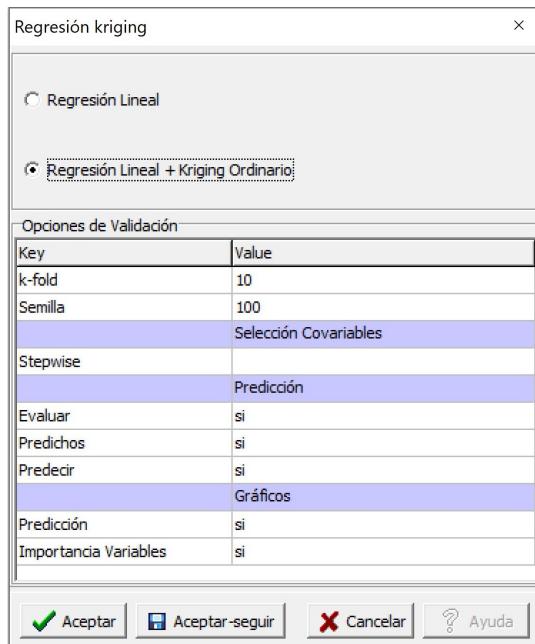
2.7.2 Kriging desde modelo de regresión

El acceso al menu es a partir de las opciones: Aplicaciones → Estadística Espacial → Geoestadísticas → Interpolación → c. Regression Kriging → Datos de prueba. Las variables CE30, CE90, Elev Pe y Tg se colocan en el casillero *Variables* y las coordenadas x e y en el cuadro *Coordenadas*.



En la ventana siguiente se puede seleccionar que el método sólo realice la predicción en base al ajuste de un modelo de regresión lineal o sumando a la predicción los valores

interpolados usando kriging ordinario de los residuos del modelo de regresión (*Regresión Lineal + kriging Ordinario*). La función ajusta en forma automática el semivariograma experimental y los modelos teóricos exponencial, esférico y gaussiano. Luego, selecciona el de mejor ajuste según el valor de SCE, y los parámetros de este son usados en la interpolación de los residuos. El método también permite realizar una selección de variables paso a paso (*stepwise*) en ambas direcciones usando el criterio de información de Akaike para evaluar el ajuste de los modelos. Otras opciones incluyen la posibilidad de realizar una validación cruzada k-fold, donde se debe especificar el valor de k y un valor para la semilla. Para ello es necesario colocar *si* en la opción *Evaluar*. También es posible guardar los valores predichos (opción *Predichos*) y obtener la predicción (opción *Predecir*) sobre una nueva base de datos. Para esto se necesita tener un archivo .txt con la información de las coordenadas (x e y) en las dos primeras columnas y de cada una de las covariables que se usaron en el ajuste del modelo. Se requiere que los nombres de las covariables sean los mismos tanto en la grilla como en la tabla de datos. La opción *Importancia* permite elaborar un ranking de la importancia relativa de cada covariable en el modelo ajustado. Esta se calcula en función a la influencia que tiene cada predictor en Error Cuadrático Medio (MSE) del modelo mediante un proceso de permutación de los valores de cada covariable.



Al ejecutar el análisis, dado que se solicitó realizar la predicción, el software mostrará una ventana para seleccionar el archivo .txt que corresponde a la grilla de predicción. En este ejemplo se denomina **grilla_am.txt**. Al accionar *Abrir* se procederá con el análisis. En la ventana *Resultados* se menciona que los valores *Predichos* se adicionan a la tabla de datos **Pred2.idb2**. Además, los valores de la predicción se adicionan a la grilla y se despliegan en una nueva tabla de datos denominada *Predictión*.

The screenshot shows two windows. The top window is a file selection dialog titled 'Select file' with the path 'c.Int... > Regression Kriging'. It lists three files: 'GrillaPred' (Text Doc), 'MyMainRScript' (R File), and 'Pred2' (IDB2 File). The bottom window is titled 'Resultados' and contains the following text:

```

Regresión kriging

Buscar [ Predichos ] en la tabla [ Pred2 ]
Buscar [ Predicción ] en la tabla [ Predicción ]
Resultados

Regresión Lineal

MAE RMSE nRMSE R2
0.69 0.85 17.20 0.11

Importancia de Variables

Variable General
Pe 23.91
Elev 100.00
CE30 0.00
CE90 9.78

Semivariograma teórico ajustado en residuos

Modelo Sill Parcial Rango Nugget
Exp 0.63 73.70 0.07

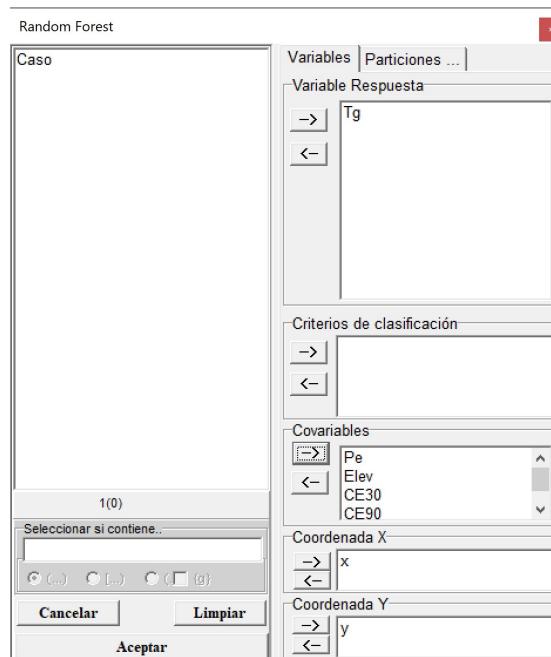
```

Los resultados muestran las medidas para cuantificar el error de predicción, entre estas el error medio absoluto (MAE), la raíz del error cuadrático medio (RMSE), la

RMSE relativa a la media de los observados (nRMSE) y un valor de R^2 . Los valores muestran un mejor desempeño del kriging regresión vs. el modelo de regresión lineal múltiple. En el ranking de importancia de las variables explicativas se observa que la elevación fue la variable que contribuyó en mayor medida a explicar la variabilidad del rendimiento de trigo. El semivariograma ajustado sobre los residuos del modelo de regresión lineal muestra la existencia de una estructura de correlación espacial (bajo valor del cociente nugget/sill).

2.7.3 Árboles aleatorios

Para realizar el análisis las variables CE30, CE90, Elev Pe y Tg se colocan en el casillero *Variables* y las coordenadas x e y en el cuadro *Coordenadas*.

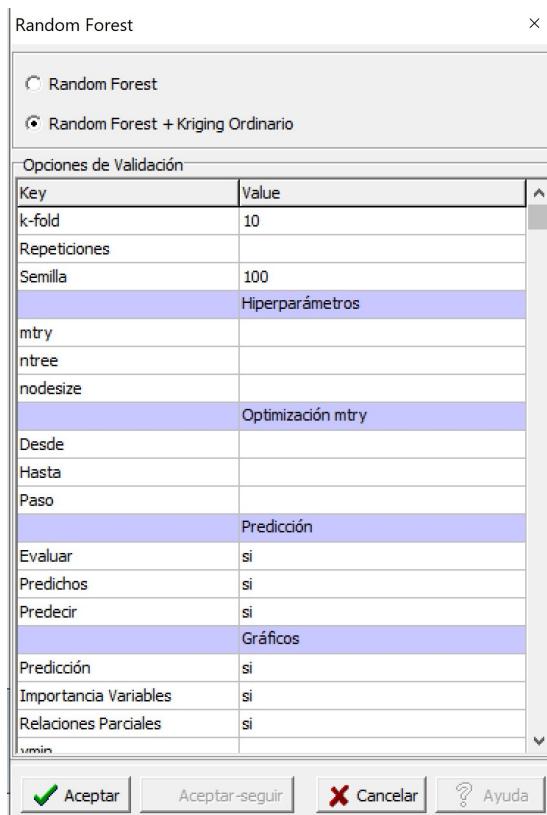


En la ventana siguiente se puede seleccionar que el méto-

do sólo realice la predicción en base al algoritmo *Random Forest* o sumando a la predicción de este los valores interpolados usando kriging ordinario de los residuos del Random Forest (*Random Forest + kriging Ordinario*). Luego, se selecciona el de mejor ajuste según el valor de SCE, y los parámetros de este son usados en la interpolación de los residuos. La función ajusta en forma automática el semivariograma experimental y los modelos teóricos exponencial, esférico y gaussiano. El método también permite fijar un valor del hiperparámetro *mtry* del random forest o también realizar una selección del *mtry* óptimo mediante un proceso de validación cruzada del tipo k-fold. Para la validación se puede especificar el valor de *k* y un valor de semilla. En caso de no fijar el valor de *mtry* utiliza el recomendado de $p/3$ para modelos de regresión o \sqrt{p} para modelos de clasificación. Si se quiere probar más de un valor se puede especificar un vector de valores que inician con el valor colocado en la opción *Desde* hasta el valor especificado en *Hasta* con un salto dado por la opción *Paso*. La opción *Evaluar* permite realizar una validación cruzada de la misma forma que la especificada inicialmente. Para ello es necesario colocar *si* en dicha opción.

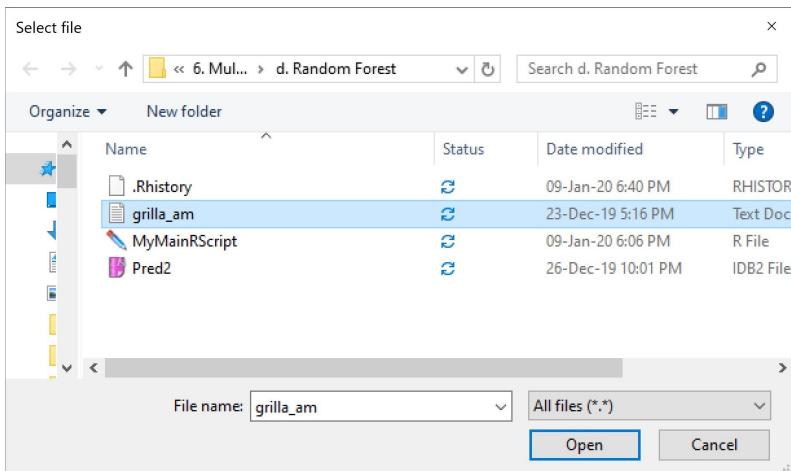
También, es posible guardar los valores predichos (opción *Predichos*) y obtener la predicción (opción *Predecir*) sobre una nueva base de datos. Para tal fin se necesita tener un archivo .txt de la grilla de predicción con la información de las coordenadas (x e y) en las dos primeras columnas y de cada una de las covariables que se usaron en el ajuste del modelo. Se requiere que los nombres de las covariables sean los mismos tanto en la grilla como en la tabla de datos. La opción *Importancia* devuelve un ranking de la importancia relativa de cada covariable en el modelo ajustado, que se calcula en función a la influencia que tiene cada predictor en Error Cuadrático Medio (MSE) del modelo mediante un

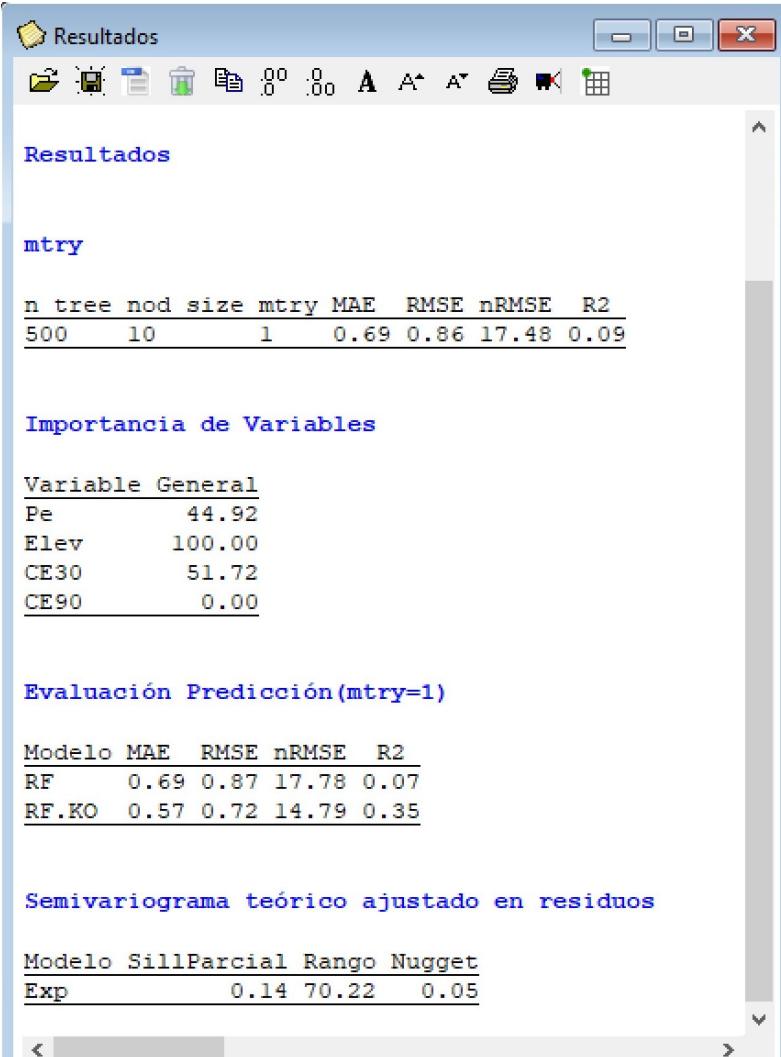
proceso de permutación de los valores de cada covariable. La opción *Relaciones Parciales* genera un gráfico que muestra el efecto marginal de cada una de las covariables sobre la variable respuesta. Para obtener este gráfico debe colocarse *si*. Esto genera un panel con los gráficos de cada una de las covariables del modelo. Si en lugar de *si* se coloca el carácter *m* el método genera un gráfico independiente para cada una de las regresoras. También, es posible editar los valores mínimos y máximos de la variable respuesta (eje y) en todos los gráficos mediante las opciones *ymin* e *ymax*.



Al ejecutar el análisis, dado que se solicitó realizar la predicción, el software mostrará una ventana para seleccionar el archivo .txt que corresponde a la grilla de predicción. En este ejemplo se denomina **grilla_am.txt**. Al accionar

Abrir se procederá con el análisis. En la ventana *Resultados* se menciona que los valores *Predichos* se adicionan a la tabla de datos **Pred2.idb2**. Además, los valores de la predicción se adicionan a la grilla y se despliegan en una nueva tabla de datos denominada *Predicción*.





Resultados

mtry

n	tree	nod	size	mtry	MAE	RMSE	nRMSE	R2
500		10		1	0.69	0.86	17.48	0.09

Importancia de Variables

Variable	General
Pe	44.92
Elev	100.00
CE30	51.72
CE90	0.00

Evaluación Predicción(mtry=1)

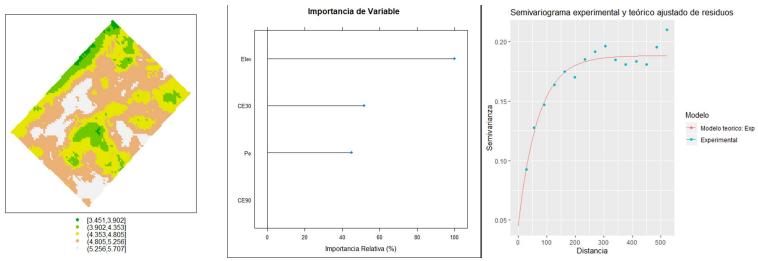
Modelo	MAE	RMSE	nRMSE	R2
RF	0.69	0.87	17.78	0.07
RF.KO	0.57	0.72	14.79	0.35

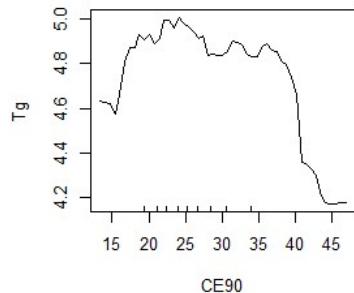
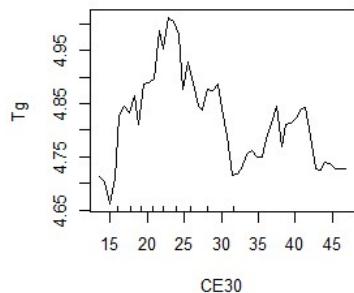
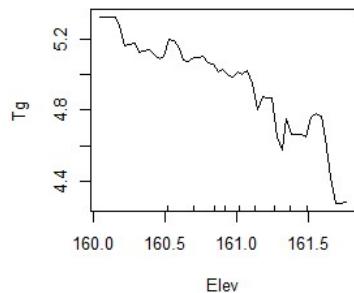
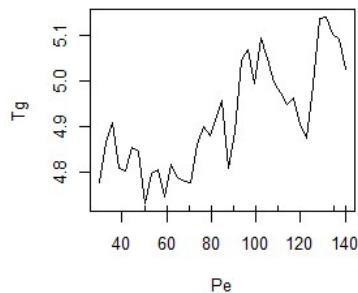
Semivariograma teórico ajustado en residuos

Modelo	Sill	Parcial	Rango	Nugget
Exp	0.14	70.22		0.05

Los resultados muestran las medidas para cuantificar el error de predicción, entre estas el error medio absoluto (MAE), la raíz del error cuadrático medio (RMSE), la RMSE relativa a la media de los observados (nRMSE) y un valor de R^2 . Los valores muestran un mejor desempeño del kriging regresión vs. el modelo de regresión lineal múltiple. En el ranking de importancia de las variables explicativas

se observa que la elevación fue la mayor contribución en la determinación del rendimiento de trigo. El gráfico de las relaciones parciales muestra que cuando aumenta la elevación disminuye el rendimiento de trigo. Con la variable Pe la correlación es negativa mientras que con la conductividad eléctrica presenta una relación no lineal. El semivariograma ajustado sobre los residuos del modelo de regresión lineal muestran la existencia de una estructura de correlación espacial (bajo valor del cociente nugget/sill).





Parte III

Análisis de datos a escala regional



1

Bases de datos regionales

Muchos procesos aleatorios varían de manera continua en el espacio a escala regional. Entonces, es de interés predecir el comportamiento de una variable en referencia a su ubicación geográfica. En algunas situaciones además de la distribución en el espacio otras capas de información (covariables de sitio) pueden ser usadas para mejorar la predicción en un sitio específico. En esta parte se ilustran el manejo de datos espaciales para la predicción a escala regional de una variable a partir de múltiples capas de información y la construcción de un modelo de predicción espacial vía estrategias metodológicas alternativas: regresión múltiple vía REML, regresión múltiple vía INLA y regresión no lineal vía modelo basado en árbol.

Como ejemplo se utiliza la base de datos **suelos_cba.txt**. Esta es una parte del SIG de los suelos del horizonte superficial de la provincia Córdoba (Hang et al., 2015), que contiene 350 sitios caracterizados por múltiples variables edáficas que describen los primeros 15 cm de profundidad. Las variables presentes en **suelos_cba.txt** son: COS (Carbono Orgánico de Suelo, g/kg) arcilla (%), pH, elevación (m.s.n.m.), twi (Índice Topográfico de Humedad). El objetivo del análisis es ajustar modelos que expliquen la variabilidad espacial de COS en función de las restantes variables en la base de datos.

Para seguir la ilustración, cargar los paquetes específicos

de R que albergan las funciones que se utilizarán tanto para el manejo como para la modelación.

```
library(sf)
library(raster)
library(tmap)
library(nlme)
library(INLA)
library(caret)
library(gstat)
```

1.1 Manejo de datos espaciales

Mediante la función `read.table()` se lee un archivo de texto que se guarda como un objeto denominado `suelos`, en el cual las columnas están separadas por tabuladores y la primera fila contiene los nombres de columnas. Mediante la función `head()` se visualizan las primeras filas del objeto `suelos` donde se observa que la primera columna corresponde a una identificación, las siguientes dos son las coordenadas X e Y las cuales corresponden al sistema de proyección UTM faja 20. Las columnas siguientes contienen las variables en estudio.

```
suelos <- read.table("datos/suelos_cba.txt",
                      sep = "\t", header = TRUE)
head(suelos)
#> ID_2 X Y elevacion twi arcilla pH COS
#> 1 2 603164 6576899 100 135 30,8 6,6 26,0
#> 2 3 596537 6390518 87 133 24,0 7,4 17,3
```

```
#> [ reached 'max'  
  / getOption("max.print") --  
 omitted 4 rows ]
```

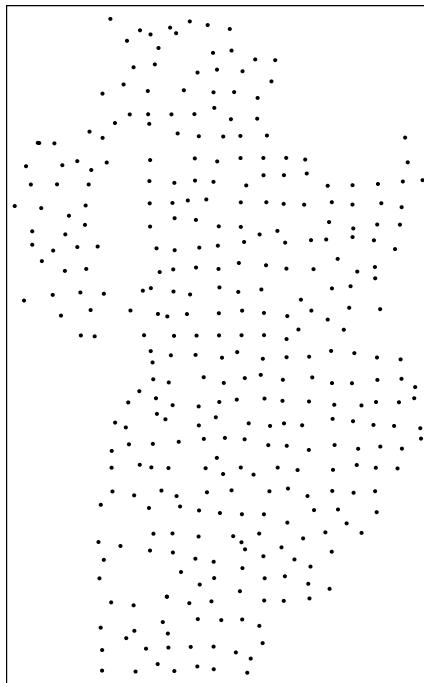
Para transformar este objeto en uno de clase espacial, se utilizará la función `st_as_sf()`, especificando que las coordenadas X e Y, se encuentran en las columnas “X” e “Y”, respectivamente. Todos los sistemas de coordenadas tienen asociados un código que los identifica y que a través del cual, se pueden conocer los parámetros asociados al mismo, este código se llama EPSG por su acrónimo en inglés. El código EPSG del sistema de referencia y proyección de la base de datos es 32720. El objeto `suelos_sf`, ahora es un objeto espacial de clase `sf`, donde cada observación corresponde a cada sitio de muestreo. Se muestra el sistema de referencia y proyección de las coordenadas y el tipo de geometría.

```
suelos_sf <- st_as_sf(suelos,  
                      coords = c("X", "Y"),  
                      crs = 32720)  
  
suelos_sf  
#> Simple feature collection with 350 features  
and 6 fields  
#> geometry type: POINT  
#> dimension: XY  
#> bbox: xmin: 236000 ymin: 6130000 xmax:  
603000 ymax: 6720000  
#> epsg (SRID): 32720  
#> proj4string:  
+ proj=utm  
+ zone=20  
+ south
```

```
+datum=WGS84 +units=m +no_defs
#> First 3 features:
#> ID_2 elevacion twi arcilla pH COS geometry
#> 1 2 100 135 30,8 6,6 26,0 POINT (603164
6576899)
#> 2 3 87 133 24,0 7,4 17,3 POINT (596537
6390518)
#> [ reached 'max'
/getOption("max.print") --
omitted 1 rows ]
```

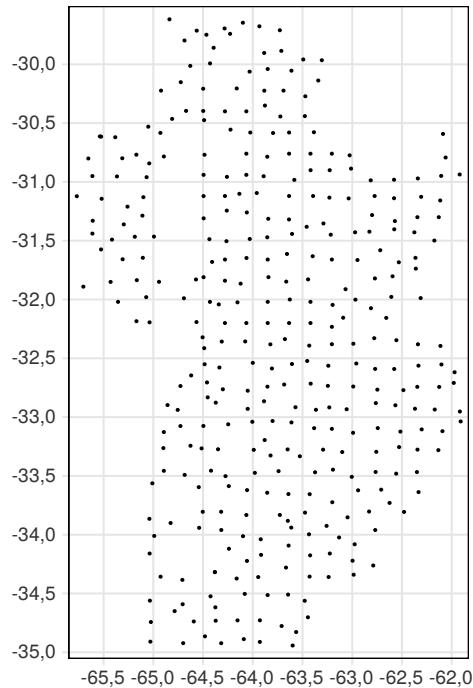
Para explorar los datos se usa el paquete `tmap` que permite realizar gráficos estáticos o dinámicos. Con la opción dinámica, se puede interactuar con el gráfico de manera análoga a un SIG. Para cada gráfico, se comienza utilizando la función `tm_shape()` especificando el objeto a graficar. Cada observación se grafica un punto mediante la función `tm_dots()`, cada nivel se agrega mediante el símbolo `+`.

```
tm_shape(suelos_sf) +
  tm_dots()
```



Para agregar latitud y longitud a esta figura se realiza una reprojeción. En la función `tm_shape()` se especifica el nuevo sistema de coordenadas con el que se desea graficar (argumento `projection`). Se agregar el nivel `tm_grid()` para visualizar una grilla que contiene las coordenadas latitud y longitud.

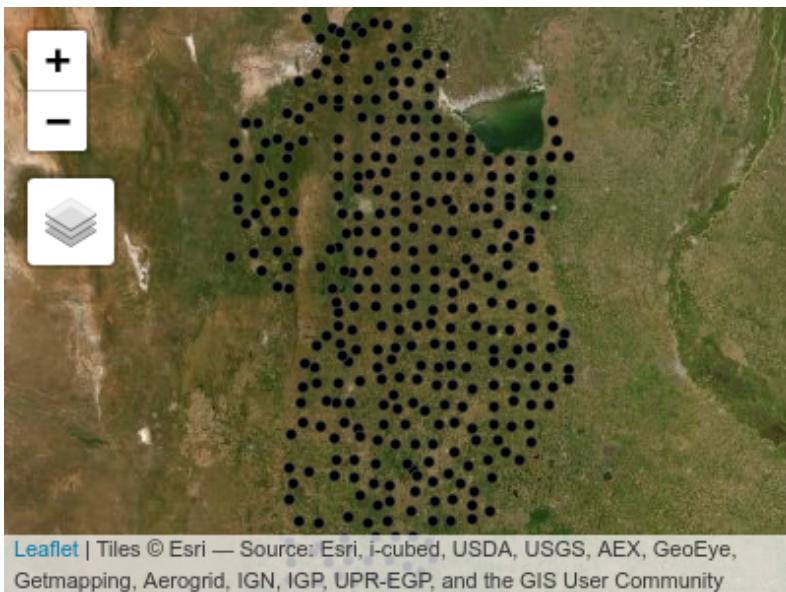
```
tm_shape(suelos_sf, projection = 4326) +  
  tm_grid(col="grey90") +  
  tm_dots()
```



Cualquiera de estos gráficos se puede convertir en un gráfico dinámico mediante la función `tmap_mode()` especificando como argumento "view". Para continuar con gráficos estáticos se debe especificar "plot" como argumento de esta función. Mediante la función `tm_basemap()`, se pueden incorporar distintas capas base. Las opciones disponibles para las capas base se pueden ver mediante el comando `names(leaflet::providers)`.

Cualquiera de estos gráficos se puede convertir en un gráfico dinámico utilizando la función `tmap_mode()` especificando como argumento "view". Para continuar con gráficos estáticos se debe especificar "plot" como argumento de esta función. Mediante la función `tm_basemap()`, se pueden incorporar distintas capas base (capas de fondo que ayudan a visualizar). Las opciones disponibles para las capas base se pueden ver mediante el comando `names(leaflet::providers)`.

```
tmap_mode("view")  
  
tm_shape(suelos_sf) +  
  tm_dots() +  
  tm_basemap("Esri.WorldImagery", "OpenTopoMap")
```



1.2 Confección de grillas de predicción

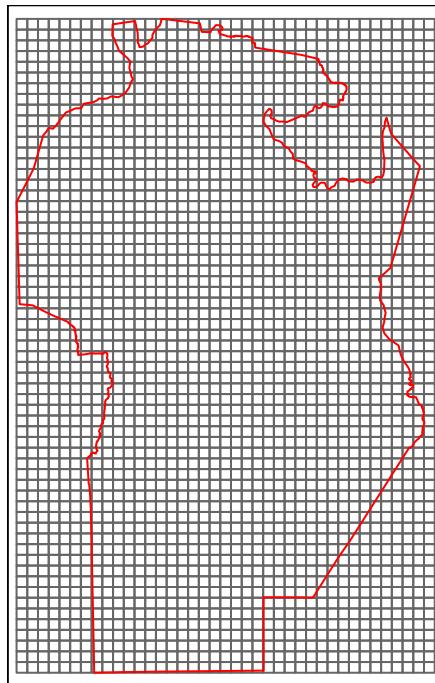
Para generar esta grilla es necesario definir una resolución espacial en el área de interés. Para este ejemplo, se utiliza un archivo vectorial, **limites_cba.shp**, el cual define el límite del territorio sobre el que se desea predecir.

```
limites_cba <- st_read("datos/limites_cba.shp",
                        quiet = TRUE)
limites_cba <- st_transform(limites_cba,
                            crs = 32720)
```

La función `st_make_grid()` genera una grilla rectangular conteniendo el área del objeto `limites_cba`. Para definir la resolución espacial de la grilla se utiliza el argumento `cellsize` definiendo un tamaño de grilla en relación con la unidad de medida del sistema de coordenadas, en este caso 10000 metros, dado que está en UTM.

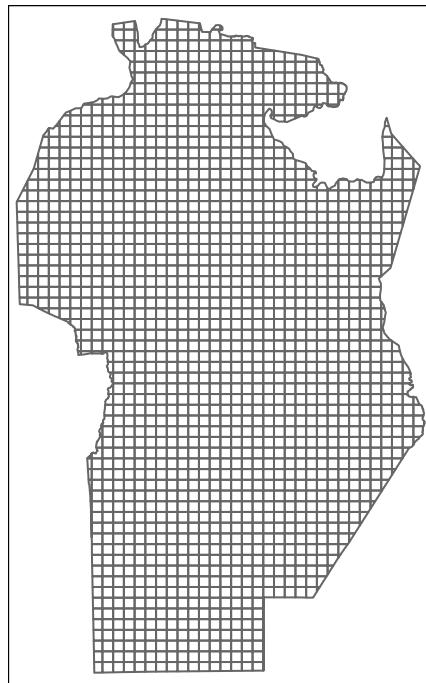
```
grilla_base <- st_make_grid(limites_cba,
                             cellsize = 10000)

tm_shape(grilla_base) +
  tm_borders() +
  tm_shape(limites_cba) +
  tm_borders(col = "red")
```



Dado que la grilla es rectangular, es necesario cortarla según los límites. Para esto se realiza una intersección entre los límites y la grilla utilizando la función `st_intersection()`.

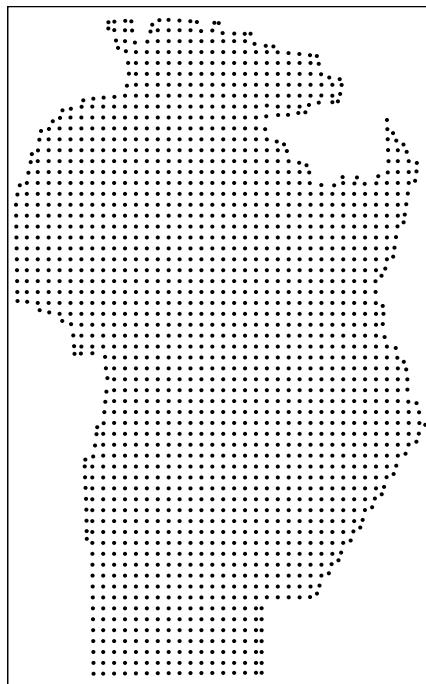
```
grilla_pred <- st_intersection(limites_cba,  
                                grilla_base)  
  
tm_shape(grilla_pred) +  
  tm_borders()
```



Los algoritmos de predicción implementados trabajan pre-diciendo sitios puntuales, por lo cual, a partir de la grilla, es necesario generar una grilla de puntos. Una alternativa es utilizar la función `st_centroid()` para obtener el centroide de cada celda.

```
centroide_pred <- st_centroid(grilla_pred)

tm_shape(centroide_pred) +
  tm_dots()
```



1.3 Agregado de capas de información

Se presenta los comandos necesarios para combinar múltiples capas de información en un mismo objeto. Las variables elevación y twi son extraídas desde modelos digitales de elevación, que se encuentran en formato raster. El paquete **raster** de R es específico para lectura y manipulación de este tipo de archivos. Para leer un archivo de este formato, se puede utilizar la función **raster()** mientras que para reproyectar se utiliza la función **projectRaster()**. El archivo **elevacion.tif** contiene datos de elevación para la provincia de Córdoba. Cuando se imprime el objeto, se muestra la cantidad de pixeles por fila, columna, pixeles totales, la resolución espacial, las coordenadas extremas en

latitud y longitud, el sistema de coordenadas de referencia, los valores mínimo y máximo de la variable observada.

```
elevacion <- raster("datos/elevacion.tif")
elevacion <-
  projectRaster(elevacion,
    crs = "+proj=utm +zone=20 +south
+datum=WGS84 +units=m +no_defs")
elevacion
#> class : RasterLayer
#> dimensions : 2867, 2373, 6803391 (nrow,
ncol, ncell)
#> resolution : 196, 231 (x, y)
#> extent : 195205, 660313, 6099410, 6761687
(xmin, xmax, ymin, ymax)
#> crs : +proj=utm +zone=20 +south +datum=WGS84
+units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
#> source : memory
#> names : elevacion
#> values : 38,6, 2746 (min, max)
```

Para obtener en los sitios de predicción el valor de la variable del objeto raster, se utiliza la función `extract()` definiendo como argumento el nombre del objeto raster y el nombre del objeto vectorial que contiene los sitios. Estos valores extraídos se adicionan en una columna llamada `elevacion` dentro del objeto `centroide_pred` utilizando el símbolo `$`.

```
centroide_pred$elevacion <-
raster::extract(elevacion, centroide_pred)
```

El archivo **twi.tif** contiene valores de un índice topográfico

de humedad también generado a partir de datos provenientes de un modelo digital de elevación..

```
twi <- raster("datos/twi.tif")
twi <-
  projectRaster(twi,
    crs = "+proj=utm +zone=20 +south
+datum=WGS84 +units=m +no_defs")
twi
#> class : RasterLayer
#> dimensions : 1438, 1189, 1709782 (nrow,
ncol, ncell)
#> resolution : 393, 462 (x, y)
#> extent : 194023, 661300, 6098486, 6762842
(xmin, xmax, ymin, ymax)
#> crs : +proj=utm +zone=20 +south +datum=WGS84
+units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
#> source : memory
#> names : twi
#> values : 54, 138 (min, max)
```

Utilizando la función `extract()` se extrae los valores de TWI para cada sitio de la grilla de predicción.

```
centroide_pred$twi <-
  raster::extract(twi,
    centroide_pred)
```

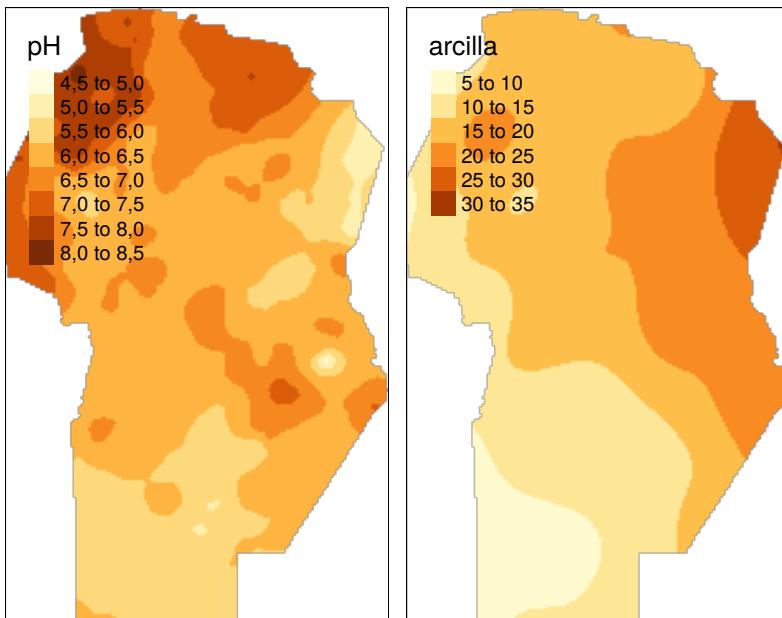
Se adiciona a la grilla variables procedentes de otras fuentes (SIG de muestreo de suelo). Estos datos se encuentran en los archivos raster llamados **arcilla.tif** y **pH.tif**, respectivamente. Estos raster tienen la misma resolución espacial y extensión, por lo que es posible superponerlos en un mismo objeto mediante la función `stack()`.

```

arcilla <- raster("datos/arcilla.tif")
pH <- raster("datos/pH.tif")
edaf <- stack(pH, arcilla)
projection(edaf) <-
  "+proj=utm +zone=20 +south
  +datum=WGS84 +units=m +no_defs"

tm_shape(edaf) +
  tm_raster()

```



Se seleccionan los valores de los sitios utilizando la función `extract()`. Como los valores extraídos mediante la función desde un `stack` de rasters genera un objeto de tipo `data.frame` con tantas columnas como capas contenga ese raster, se adicionan al objeto `centroide_pred` mediante la función `cbind()`, la cual une columnas de igual número

de filas. Ahora el objeto `centroide_pred` contiene todos los sitios de predicción con las variables auxiliares adicionadas.

```
edaf_pred <- raster::extract(edaf,
                               centroide_pred)
centroide_pred <- cbind(centroide_pred,
                        edaf_pred)
centroide_pred
#> Simple feature collection with 1668 features
and 8 fields
#> geometry type: POINT
#> dimension: XY
#> bbox: xmin: 239000 ymin: 6130000 xmax:
615000 ymax: 6730000
#> epsg (SRID): 32720
#> proj4string:
+ proj=utm
+ zone=20
+ south
+datum=WGS84 +units=m +no_defs
#> First 3 features:
#> UNION JURISDICCI CAPITAL FUENTE elevacion
twi pH arcilla
#> 1 -2,15e+09 CORDOBA CORDOBA IGN 275 118 6,39
8,21
#> geometry
#> 1 POINT (310499 6129968)
#> [ reached 'max'
/getOption("max.print") --
omitted 2 rows ]
```

Para identificar la variación de la variable de interés en un plano, se puede usar una escala de colores. La elección de la escala cambia según los colores y los puntos de corte (valores de la variable de interés en los cuales cambia el

color). Para definirla algunas opciones automáticamente identifican los valores para categorizar y asignar un color. Por defecto, `tmap` categoriza los valores en intervalos fijos. Utilizando el argumento `style`, se puede modificar el método a utilizar. Las opciones `style = "order"` y `style = "cont"` permiten representar variables numéricas en un gradiente de color. La opción “order” realiza una escala en función del ranking de los valores, permitiendo una mejor visualización de variables asimétricas. Para la visualización de más de un mapa generado con `tmap` en un mismo gráfico, se puede utilizar la función `tmap_arrange()`, utilizando como argumento los mapas que se quieren visualizar. El argumento `sync = TRUE` permite la visualización interactiva con la navegación (zoom y movimiento) sincronizada en ambos mapas.

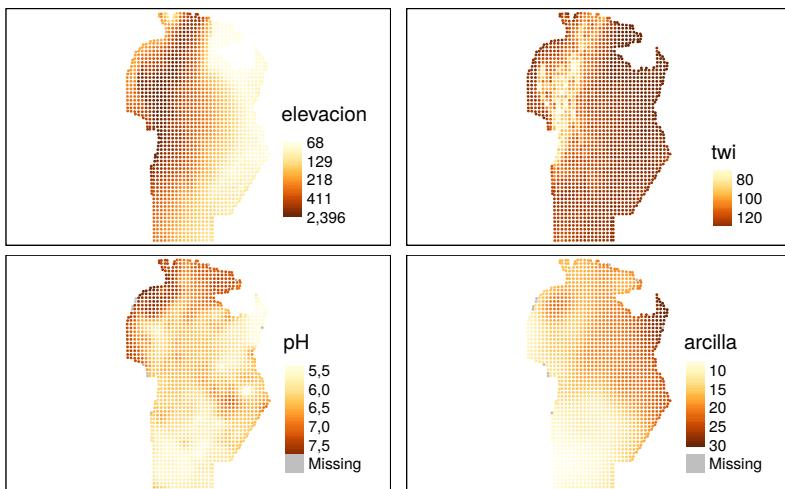
```
elevTm <- tm_shape(centroide_pred) +
  tm_dots("elevacion", style = "order")

twiTm <- tm_shape(centroide_pred) +
  tm_dots("twi", style = "cont")

pHTM <- tm_shape(centroide_pred) +
  tm_dots("pH", style = "cont")

arcillaTm <- tm_shape(centroide_pred) +
  tm_dots("arcilla", style = "cont")

tmap_arrange(elevTm, twiTm, pHTM, arcillaTm,
             ncol = 2)
```





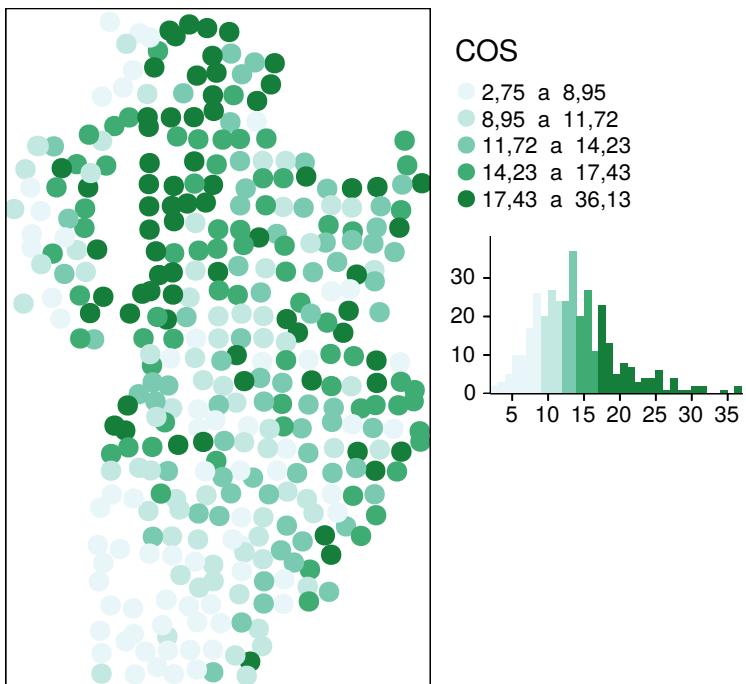
2

Predicción con múltiples capas de datos

Una vez que se ha confeccionado la grilla de predicción y se ha unificado el sistema de referencia espacial entre las distintas capas de información, se comienza con el ajuste de modelos que luego serán usados para la predicción espacial en sitios sin datos. El objeto `suelos`, se utilizará para el ajuste de los modelos predictivos, mientras que `cetroide_pred` se usará para obtener predicciones para cada celda de la grilla. La distribución espacial de la variable de interés (COS) puede visualizarse con funciones del paquete `tmap`. A través del argumento `palette` se modifica la paleta de colores, las opciones disponibles pueden buscarse ejecutando el comando `tmaptools::palette_explorer()`. También se pueden adicionar otras herramientas de estadística descriptiva, como por ejemplo un histograma de frecuencia mediante el argumento `legend.hist = TRUE`. Los estilos de los ejes y leyendas se pueden modificar con la función `tm_layout()`.

```
tm_shape(suelos_sf) +  
  tm_dots(  
    "COS",  
    style = "quantile",  
    size = 0.5,  
    palette = "BuGn",  
    legend.hist = TRUE  
  ) +
```

```
tm_layout(  
  legend.format = list(text.separator = " a "),  
  legend.outside = TRUE,  
  legend.hist.width = 1  
)
```



2.1 Regresión con errores correlacionados espacialmente vía REML

Se ajusta un modelo de regresión lineal con la función `gls()`, usando COS como variable dependiente y elevación, twi, arcilla y pH como variables predictoras. Primero, se ajusta suponiendo errores independientes (sin correlación

2.1 Regresión con errores correlacionados espacialmente vía REML 201

espacial). Los resultados se guardan en el objeto denominado `ajuste_ML`. Seguidamente, se ajusta otro modelo de regresión con igual estructura para la componente sistemática, pero suponiendo que los términos de error aleatorio no son independientes sino que se correlacionan a través de un modelo de covarianza espacial. En particular, se ajusta el modelo de correlación espacial esférico y se suponen varianza residual única (modelo homocedástico). El método de estimación del modelo es REML. Los resultados se guardan en el objeto `ajuste_err_corr`.

```
ajuste_ML <- gls(  
  COS ~ 1 + elevacion + twi + arcilla + pH,  
  data = suelos,  
  method = "REML")
```

```
ajuste_err_corr <- gls(  
  COS ~ 1 + elevacion + twi + arcilla + pH,  
  data = suelos,  
  correlation = corSpher(form = ~ X + Y),  
  method = "REML"  
)
```

Utilizando la función `summary()` se muestra a continuación el resultado del modelo sin correlación espacial (`objeto ajuste_ML`). Todos los términos del modelo, a excepción de `elevacion` resultaron significativos para un nivel de significación $\alpha = 0,05$. Se observó una correlación alta entre `elevacion` y `twi` (0,859), por esta colinealidad entre ambas variables, el término `elevacion` pudo no haber resultado significativo y podría sacarse del modelo. Se muestra también las características de la distribución de los residuos (mínimo, máximo valor y principales cuartiles). Es de esperar que los residuos estandarizados se encuentren en el intervalo

[-3, 3], los valores fuera de este rango se consideran valores atípicos y podrían ser eliminados para reajustar el modelo. La varianza residual es el cuadrado de 4.58, indicando que desviaciones de 4,58 g/kg pueden existir por azar y que no se relacionan a las fuentes de variación reconocidas a priori.

```
summary(ajuste_ML)
#> Generalized least squares fit by REML
#> Model: COS ~ 1 + elevacion + twi + arcilla +
pH
#> Data: suelos
#> AIC BIC logLik
#> 2088 2111 -1038
#>
#> Coefficients:
#> Value Std.Error t-value p-value
#> (Intercept) 37,2 6,65 5,59 0,0000
#> elevacion 0,0 0,00 1,64 0,1016
#> twi -0,2 0,05 -4,49 0,0000
#> arcilla 0,3 0,03 10,83 0,0000
#> [ reached 'max'
  / getOption("max.print") --
omitted 1 rows ]
#>
#> Correlation:
#> (Intr) elevcn twi arcill
#> elevacion -0,767
#> twi -0,919 0,859
#> arcilla -0,047 -0,001 -0,064
#> pH -0,347 -0,163 -0,035 0,040
#>
#> Standardized residuals:
#> Min Q1 Med Q3 Max
```

2.1 Regresión con errores correlacionados espacialmente vía REML 203

```
#> -3,427 -0,596 -0,118 0,472 4,787  
#>  
#> Residual standard error: 4,58  
#> Degrees of freedom: 350 total; 345 residual
```

Para el modelo ajustado suponiendo errores correlacionados, los criterios de información de AIC y BIC fueron menores que los obtenidos bajo el supuesto de errores independientes, indicando la conveniencia de considerar la correlación espacial. Los parámetros del modelo asociado a la componente aleatoria son rango = 17791,35 m y varianza residual igual al cuadrado de 4,56. Estos caracterizan la matriz de varianza y covarianza de los errores y proveen una estimación del semivariograma esférico que describe el proceso espacial subyacente, *i.e.* observaciones separadas por más de 17791,35 m no se encuentran correlacionadas y la varianza residual de las observaciones independientes o con distancias mayor al rango, expresada como desvío estándar, es 4,56.

```
summary(ajuste_err_corr)  
#> Generalized least squares fit by REML  
#> Model: COS ~ 1 + elevacion + twi + arcilla +  
pH  
#> Data: suelos  
#> AIC BIC logLik  
#> 2078 2105 -1032  
#>  
#> Correlation Structure: Spherical spatial  
correlation  
#> Formula: ~X + Y  
#> Parameter estimate(s):  
#> range
```

```

#> 17791
#>
#> Coefficients:
#> Value Std.Error t-value p-value
#> (Intercept) 37,0 6,44 5,74 0,0000
#> elevacion 0,0 0,00 1,71 0,0877
#> twi -0,2 0,05 -4,41 0,0000
#> arcilla 0,3 0,03 12,49 0,0000
#> [ reached 'max'
    / getOption("max.print") --
omitted 1 rows ]
#>
#> Correlation:
#> (Intr) elevcn twi arcill
#> elevacion -0,763
#> twi -0,916 0,853
#> arcilla -0,218 0,083 0,038
#> pH -0,285 -0,224 -0,108 0,260
#>
#> Standardized residuals:
#> Min Q1 Med Q3 Max
#> -3,415 -0,578 -0,117 0,476 4,843
#>
#> Residual standard error: 4,57
#> Degrees of freedom: 350 total; 345 residual

```

Las predicciones se realizaron utilizando la función `predict()` sobre los centroides de la grilla de predicción utilizando el mejor modelo entre los ajustados. Se convierte el objeto `centroide_pred` en un `data.frame`, eliminando del objeto `centroide_pred` la columna que contiene las características espaciales mediante la función `st_drop_geometry()` y extrayendo mediante la función `st_coordinates()`, las coordenadas sin los atributos espa-

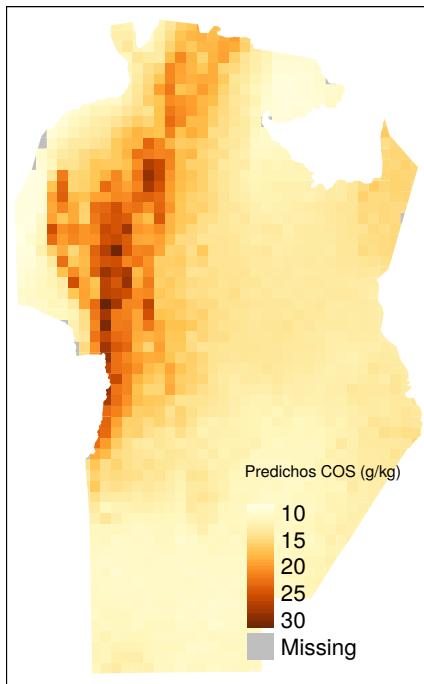
2.1 Regresión con errores correlacionados espacialmente vía REML 205

ciales. Estas partes se guardan en el objeto `suelos_pred` de clase `data.frame`.

```
suelos_pred <- data.frame(  
  st_drop_geometry(centroide_pred),  
  st_coordinates(centroide_pred))  
  
pred_ajuste_err_corr <- predict(  
  ajuste_err_corr,  
  newdata = suelos_pred,  
  na.action = na.pass)
```

Los predichos se adicionan al objeto `centroide_pred` utilizando la función `cbind()`. Para que la visualización de estos valores, se pueda realizar utilizando los polígonos de la grilla de predicción en vez de los centroides, se deben adicionar los predichos mediante la función `st_join()`.

```
pred_err_corr <- cbind(  
  centroide_pred,  
  "COS_pred" = pred_ajuste_err_corr)  
  
pred_err_corr <- st_join(  
  grilla_pred,  
  pred_err_corr)  
  
tm_shape(pred_err_corr) +  
  tm_fill("COS_pred", style = "cont",  
         title = "Predichos COS (g/kg)")
```



2.2 Regresión con efectos aleatorios de sitio vía INLA

Para abordar la regresión bayesiana de datos espaciales, primero se define el predictor lineal ajustando un modelo de regresión lineal con la función `inla()`. INLA representa una combinación de aproximaciones analíticas y esquemas de integración numérica eficiente para obtener una aproximación confiable de la distribución a posteriori de interés. En el ejemplo de ilustración, se usa COS como variable dependiente y elevación, twi, arcilla y pH como variables predictoras y no se ha contemplado la estructura de correlación espacial. Se especifica la distribución que se asume para la variable respuesta a través del argumento `family`.

El cómputo de las medidas para evaluación y comparación de modelos se realiza con el argumento `control.compute` especificando la medida que se pretende. Para explorar las opciones disponibles para la evaluación y comparación de modelos se ejecuta el comando `?control.compute` (en el ejemplo, se solicita el criterio DIC).

```
ajuste_INLA <- inla(  
  COS ~ 1 + elevacion + twi + arcilla + pH,  
  family = 'gaussian',  
  data = suelos,  
  control.compute = list(dic = TRUE))
```

El modelo ajustado es retornado como un objeto INLA. Este provee información sobre el tiempo de procesado y algunos estadísticos sobre las distribuciones a posteriori de los coeficientes de regresión (efectos fijos) y de los hiperparámetros. Para el modelo ajustado se observan los intervalos de credibilidad del 95 % para los coeficientes de regresión asociados a cada una de las variables predictoras (predictor lineal) y como hiperparámetro la precisión de las observaciones de COS. En este ajuste, no hubo efectos aleatorios ni especificaciones relacionadas a la espacialidad de los datos. El intervalo de credibilidad contiene al verdadero parámetro con un 95 % de probabilidad. Luego, el ajuste indica que todas las variables impactan a la respuesta, excepto la variable elevación para la cual el desvío estándar (sd) es alto relativo a la media de la distribución del coeficiente de regresión y el intervalo de credibilidad contiene al 0. Podría ser oportuno realizar un nuevo ajuste sin esta variable, que como se ha especificado anteriormente está altamente correlacionada con twi. La media a posteriori para el coeficiente de regresión que acompaña el pH es -0,766 con un intervalo de credibilidad del 95 % entre -1,516 y -0,017, por lo que se interpreta que a mayores valores

de pH se tendrán menores valores de COS. Se muestra también el intervalo de credibilidad

$$0,041; 0,055$$

para la precisión (inversa de la varianza $1/\sigma_e^2$), la estimación es 0,048 y por tanto la varianza residual es próxima a 20 o el error estándar residual cercano a 4,56. El valor de DIC, el cual es una función de la deviance del modelo y de una medida del número efectivo de parámetros del modelo, es 2065,79. El numero efectivo de parámetros es una cantidad que caracteriza la complejidad del modelo y que no solo depende de la cantidad de parámetros sino también de la dependencia entre ellos. Esta medida puede ser usada para comparar modelos, menores valores indican mejor ajuste del modelo a los datos. El mejor de los modelos ajustados, también tendrá menor diferencia entre el valor de DIC para ese modelo y el valor de DIC para el modelo saturado. La verosimilitud marginal es otro criterio usado en selección de modelos en estadística bayesiana, al reportarse en escala log menor valor indica mejor ajuste. R-INLA obtiene las distribuciones marginales a posteriori para todos los parámetros del modelo.

```
summary(ajuste_INLA)
#>
#> Call:
#> c("inla(formula = COS ~ 1 + elevacion + twi +
+ arcilla + pH, family =
#> \ "gaussian\\"", ", " data = suelos,
control.compute = list(dic = TRUE))")
#> Time used:
#> Pre = 0,417, Running = 0,138, Post = 0,0507,
Total = 0,606
#> Fixed effects:
#> mean sd 0.025quant 0.5quant 0.975quant mode
```

```
kld
#> (Intercept) 37,171 6,654 24,100 37,171
50,231 37,171 0
#> elevacion 0,003 0,002 -0,001 0,003 0,008
0,003 0
#> [ reached getOption("max.print") -- omitted
3 rows ]
#>
#> Model hyperparameters:
#> mean sd 0.025quant 0.5quant 0.975quant
#> Precision for the Gaussian observations
0,048 0,004 0,041 0,048 0,055
#> mode
#> Precision for the Gaussian observations
0,048
#>
#> Expected number of effective
parameters(stdev): 5,00(0,00)
#> Number of equivalent replicates : 70,00
#>
#> Deviance Information Criterion (DIC)
.....: 2065,80
#> Deviance Information Criterion (DIC,
saturated) ....: 359,50
#> Effective number of parameters
.....: 6,08
#>
#> Marginal log-Likelihood: -1070,11
#> Posterior marginals for the linear predictor
and
#> the fitted values are computed
```

Los efectos aleatorios en INLA se incluyen en la formula del predictor lineal usando la función `f()`. Para el ejemplo

de ilustración, más abajo se ajusta el modelo de regresión donde se adiciona un efecto aleatorio de sitio para caracterizar el proceso espacial subyacente a los datos. Dado que la función `f()` se valúa sobre un red de nodos conformada a partir de las observaciones, es primero necesario construir una malla que cubra el dominio espacial y definir un objeto que contiene la identificación de los nodos con observaciones. La malla se arma con la función `inla.mesh.2d()` cuyos argumentos o parámetros de la malla son: `cutoff` define la distancia mínima entre vértices de los triángulos que conforman la malla y `max.edge` que refiere a la longitud máxima del lado de cada triángulo. Por defecto, la malla se construye con el método de triangulación de Delaunay. Sobre esta malla se construye el objeto nodos.

```
sitios <- suelos[, c("X", "Y")]

malla <- inla.mesh.2d(sitios, cutoff = 200,
                      max.edge = 200000)

nodos <- malla$idx$loc
```

Para estimar la matriz de varianzas y covarianzas de los efectos de sitio por el método SPDE se utiliza la función `inla.spde2.matern()`. Un argumento a especificar es el parámetro α (que varía entre 0 y 2). Por defecto es 2 para aproxima una función de correlación espacial del tipo exponencial como modelo de correlación espacial entre los efectos de sitio.

```
spde <- inla.spde2.matern(mesh = malla,
                           alpha = 2)
```

Luego de identificar los nodos sobre la malla, se ajusta el

modelo de regresión con efecto aleatorio de sitio usando la función `inla()`. Los resultados del objeto INLA resultante incluyen las distribuciones a posteriori de los efectos latentes y de los hiperparámetros, así como estadísticos de resumen. Como se ejemplifica adelante, pueden obtenerse estimaciones a posteriori de parámetros del campo espacial latente.

```
ajuste_INLAspde <- inla(  
    COS ~ 1 + elevacion + twi + arcilla + pH +  
        f(nodos, model = spde, diagonal = 1e-6),  
    family = 'gaussian',  
    data = suelos,  
    control.compute = list(dic = TRUE),  
    control.predictor = list(compute = TRUE))  
  
summary(ajuste_INLAspde)  
#>  
#> Call:  
#> c("inla(formula = COS ~ 1 + elevacion + twi  
+ arcilla + pH + f(nodos, ", "  
#> model = spde, diagonal = 1e-06), family =  
\\"gaussian\\", data = suelos, ", "  
#> control.compute = list(dic = TRUE),  
control.predictor = list(compute =  
#> TRUE))" )  
#> Time used:  
#> Pre = 0,598, Running = 1,99, Post = 0,0821,  
Total = 2,67  
#> Fixed effects:  
#> mean sd 0.025quant 0.5quant 0.975quant mode  
kld  
#> (Intercept) 47,928 7,207 33,782 47,923  
62,089 47,913 0
```

```
#> elevacion 0,004 0,003 -0,001 0,004 0,009  
0,004 0  
#> [ reached getOption("max.print") -- omitted  
3 rows ]  
#>  
#> Random effects:  
#> Name Model  
#> nodos SPDE2 model  
#>  
#> Model hyperparameters:  
#> mean sd 0.025quant 0.5quant 0.975quant  
#> Precision for the Gaussian observations  
0,069 0,007 0,057 0,069 0,083  
#> Theta1 for nodos 8,579 0,154 8,291 8,574  
8,895  
#> mode  
#> Precision for the Gaussian observations  
0,069  
#> Theta1 for nodos 8,554  
#> [ reached 'max'  
    / getOption("max.print") --  
omitted 1 rows ]  
#>  
#> Expected number of effective  
parameters(stdev): 52,76(9,84)  
#> Number of equivalent replicates : 6,63  
#>  
#> Deviance Information Criterion (DIC)  
.....: 1988,52  
#> Deviance Information Criterion (DIC,  
saturated) ....: 412,04  
#> Effective number of parameters  
.....: 54,42  
#>
```

```
#> Marginal log-Likelihood: -1049,41  
#> Posterior marginals for the linear predictor  
and  
#> the fitted values are computed
```

El objeto resultante provee información sobre los intervalos de credibilidad del 95 % de los coeficientes de regresión y de los hiperparámetros. Estos son además de la precisión `Theta1` y `Theta2` que definen la función de correlación espacial subyacente. Los parámetros `Theta1` y `Theta2` no son de interpretación directa, pero dependen de los parámetros que caracterizan el proceso espacial (rango y varianza estructural). Utilizando la función `inla.spde2.result()` se puede obtener la distribución a posteriori de los parámetros expresadas en términos de rango y varianza estructural.

```
resultados_spde <-  
  inla.spde2.result(inla = ajuste_INLAspde,  
                     name = "nodos", spde = spde)  
  
inla.emarginal(function(x)  
  x,  
  resultados_spde$marginals.range.nominal[[1]]  
)  
#> [1] 175820  
  
inla.emarginal(function(x)  
  x,  
  resultados_spde$marginals.variance.nominal[[1]])  
#> [1] 9,45
```

Para comparar los modelos de regresión ajustados con errores independientes y con correlación espacial se visuali-

zan medidas de bondad de ajuste como DIC para ambos modelos.

```
c(ajuste_INLA$dic$dic, ajuste_INLAspde$dic$dic)
#> [1] 2066 1989
```

Comparando los valores de DIC se deduce la conveniencia de usar un modelo con correlación espacial respecto a uno que supone los valores de COS independientes.

En R-INLA no existe una función `predict()` como en gls. Las predicciones deben ser obtenidas como parte del modelo ajustado. Dado que las predicciones pueden ser entendidas como el ajuste de un modelo con datos faltante simplemente se especificará, antes del ajuste, `y[i] = NA` para aquellos sitios donde se desea predecir. Las distribuciones de los valores predichos no son devueltas directamente, pero se pueden explorar. INLA retorna las a posteriori marginales para los efectos aleatorios y para el predictor linear en el sitio faltante. Adicionando el ruido de las observaciones a los valores ajustados se obtienen los valores predichos para el sitio. Para implementar este proceso en R, primero se debe generar un `data.frame` juntando los datos observados y la grilla de predicción. La función `bind_rows()` del paquete `dplyr` permite juntar dos data frames que contengan el mismo nombre de columnas colocando `NA` cuando no hay valor para un campo.

```
suelos_pred_INLA <- dplyr::bind_rows(suelos_pred,
                                         suelos)
head(suelos_pred_INLA)
#> UNION JURISDICCI CAPITAL FUENTE elevacion
twi pH arcilla X Y ID_2
#> 1 -2,15e+09 CORDOBA CORDOBA IGN 275 118 6,39
8,21 310499 6129968 NA
```

```
#> COS  
#> 1 NA  
#> [ reached 'max'  
/ getOption("max.print") --  
omitted 5 rows ]
```

Luego de identificar el predictor lineal, debe definirse la malla y el modelo espacial para la grilla de predicción asociada a los efectos aleatorios de sitios. Mediante el argumento `control.predictor` en la función `inla()` se indica que debe computarse el valor de la variable respuesta en el lugar del dato faltante.

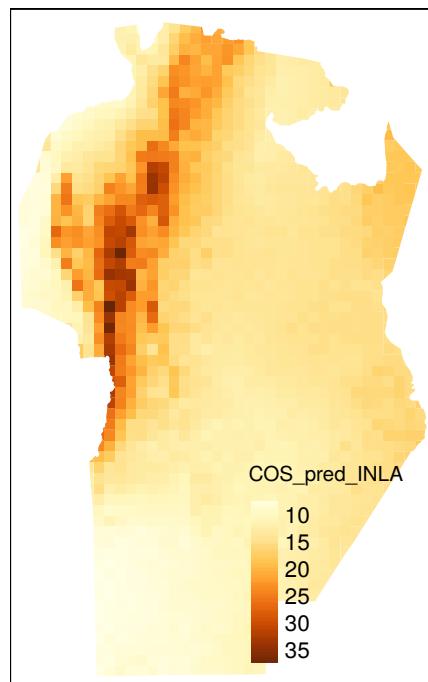
```
malla_pred <- inla.mesh.2d(sitios_pred,  
                           cutoff = 200,  
                           max.edge = 200000)
```

```
sitios_pred <- suelos_pred_INLA[, c("X", "Y")]  
  
nodos_pred <- malla_pred$idx$loc  
  
spde_pred <-  
  inla.spde2.matern(mesh = malla_pred, alpha = 2)  
  
pred_INLAspde <-  
  inla(  
    COS ~ -1 + elevacion + twi + arcilla + pH +  
    f(nodos_pred, model = spde_pred, diagonal = 1e-6),  
    family = 'gaussian',  
    data = suelos_pred_INLA,  
    control.predictor = list(link = 1, compute = TRUE)  
)
```

Se puede obtener la media de la distribución a posteriori de los valores predichos para cada sitio en la grilla de predicción, para mapear la distribución espacial de la variable respuesta.

```
pred_err_corr <-
  cbind(pred_err_corr,
        "COS_pred_INLA" =
          pred_INLAspde$summary.fitted.values$mean
          [is.na(suelos_pred_INLA$COS)])

tm_shape(pred_err_corr) +
  tm_fill("COS_pred_INLA", style = "cont")
```



2.3 Regresión vía modelos basados en árbol

Se ajusta un modelo GBR o *gradient boosting model* con errores correlacionados espacialmente en dos pasos, primero se optimiza la parametrización del predictor GBR usando datos de los sitios observados y se obtienen los residuos de este modelo. En segunda instancia, se ajusta un modelo de semivariaograma a los residuos que se usará para realizar predicción kriging de residuos sobre toda la grilla de predicción. Finalmente, los residuos predichos se adicional a la componente sistemática predicha con el modelo GBR sobre la misma grilla de predicción.

Para implementar GBR se utiliza el paquete `caret`. Para optimizar el modelo GBM. se genera una grilla de valores posibles para sus parámetros con la función `expand.grid()`. Esta función genera un `data.frame` que contiene en las filas cada una de las combinaciones posibles generadas a partir de los rangos de valores propuestos para cada parámetro del modelo GBM. Éstos son: `n.trees` que definen el número total de árboles ajustados, `shrinkage` que regula la extensión de cada árbol, `n.minobsinnode` que representa el mínimo de observaciones en cada nodo terminal y `bag.fraction` la proporción de observaciones del grupo de entrenamiento seleccionadas aleatoriamente para la expansión sucesiva del árbol. El tipo de validación cruzada para la optimización de los parámetros del modelo se realiza a través de la función `train.control`. Luego utilizando la función `train()` se especifica el modelo con el argumento `method`, en este caso `gbm`. La misma función `train()` genera un objeto con el modelo parametrizado con la configuración valores que arrojan el menor error predictivo, es decir con un modelo del tipo árbol optimizado.

```

param_gbm <- expand.grid(
  interaction.depth = c(2:4),
  n.trees = (1:30) * 100,
  shrinkage = c(0.001, 0.01),
  n.minobsinnode = c(7, 5)
)

control <- trainControl(method = "repeatedcv",
                        number = 5,
                        repeats = 5)

ajuste_gbm <- train(
  COS ~ elevacion + twi + arcilla + pH,
  data = suelos,
  method = "gbm",
  trControl = control,
  verbose = FALSE,
  metric = "RMSE",
  tuneGrid = param_gbm
)

```

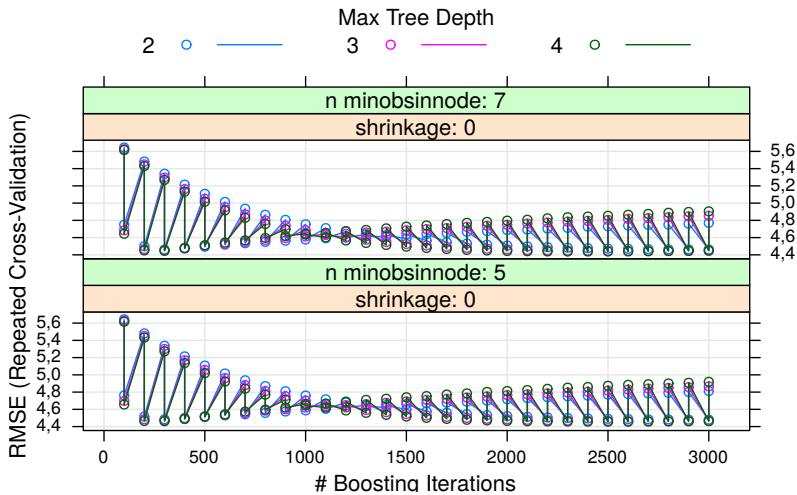
Pidiendo un gráfico del objeto `ajuste_gbm` se puede acceder al resumen del proceso de optimización de los parámetros. El rendimiento del modelo depende de estos parámetros, pero es posible identificar las combinaciones que generan el mejor desempeño predictivo. A su vez, a través del comando `ajuste_gbm$bestTune` podemos acceder a los parámetros que definen el modelo óptimo.

```

plot(ajuste_gbm)
ajuste_gbm$bestTune
#> n.trees interaction.depth shrinkage

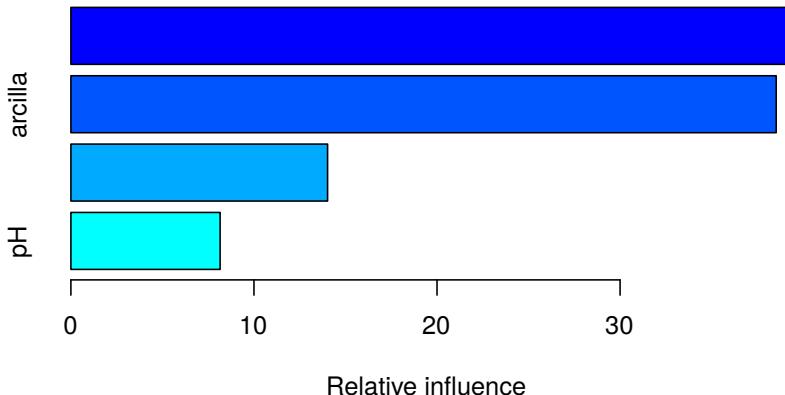
```

```
n.minobsinnode
#> 173 2300 4 0,001 7
```



El objeto resultante del ajuste GBM, provee un gráfico de la influencia relativa de cada variable predictora para explicar COS, y el árbol con el que se realizará la predicción. A partir de la función `predict()` sobre los datos observados, se obtienen predichos y consecuentemente los residuos del modelo GBR.

```
summary(ajuste_gbm)
#>                               var rel.inf
#> twi                           twi   39,30
#> arcilla           arcilla   38,53
#> elevacion         elevacion  14,02
#> pH                            pH    8,15
```



En la segunda etapa, se ajusta una función de semivarianza a los residuos del modelo GBM utilizando las funciones `variogram` y `fit.variogram` del paquete `gstat`.

```

suelos$residuosgbm <-
  suelos$COS = predict(ajuste_gbm,
                       newdata = suelos)

coordinates(suelos) <- c("X", "Y")

crs(suelos) <- CRS("+init=epsg:32720")

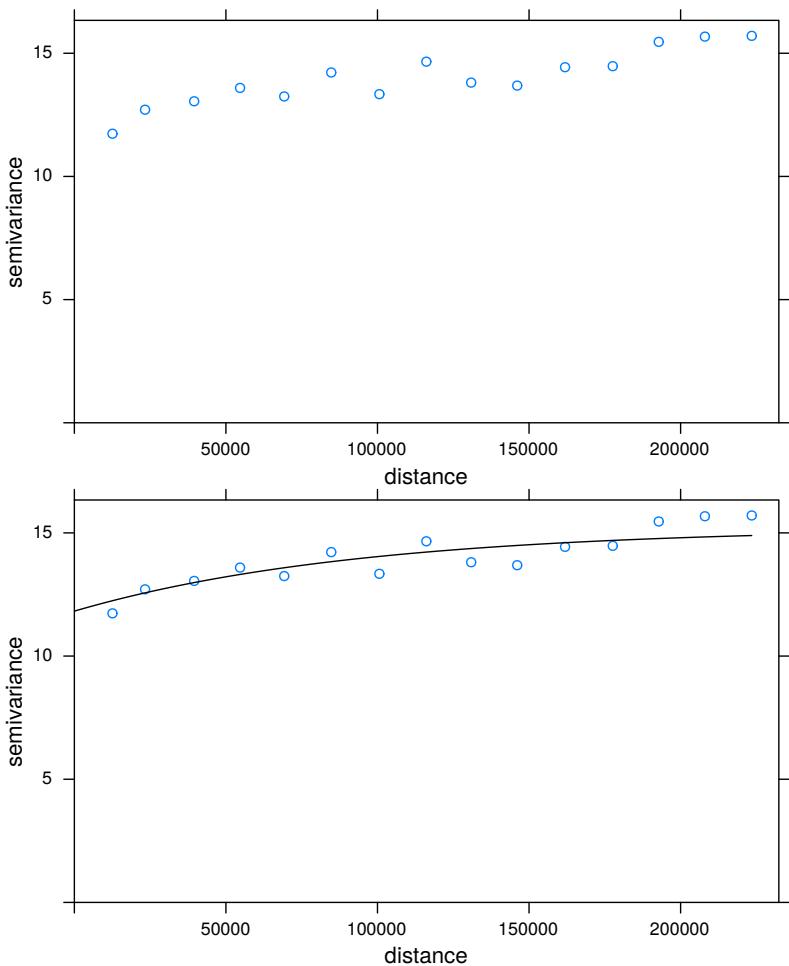
semiv_gbmk <- variogram(residuosgbm ~ 1, suelos)

plot(semiv_gbmk)

semiv_aj_gbmk <-
  fit.variogram(semiv_gbmk ,
                vgm(c("Exp", "Sph", "Gau")))
plot(semiv_gbmk , semiv_aj_gbmk)

coordinates(suelos_pred) <- c("X", "Y")
crs(suelos_pred) <- CRS("+init=epsg:32720")

```



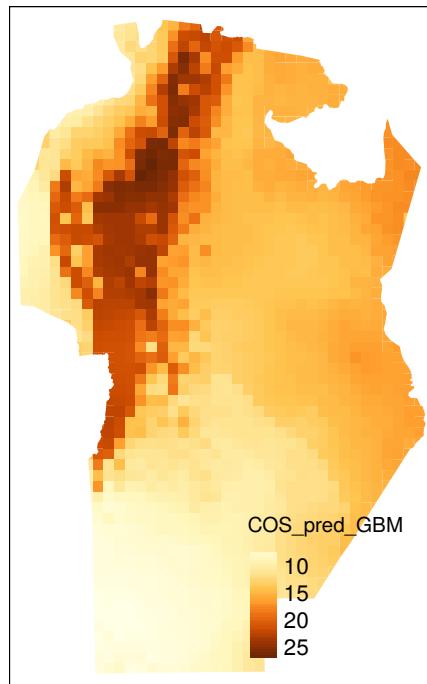
Para obtener un predicción de COS en los sitios no muestreados, se realiza la predicción kriging de los residuos sobre los sitios de la grilla de predicción utilizando el modelo ajustado en el paso anterior a partir de la función `krige()`. Luego se utiliza el modelo GBM (árbol optimo) para predecir COS sobre la grilla de predicción sin considerar la espacialidad. Finalmente, la predicción de COS en cada sitio se compone sumando la predicción del modelo GBM y la predicción kriging de los residuos para cada sitio de la grilla de predicción.

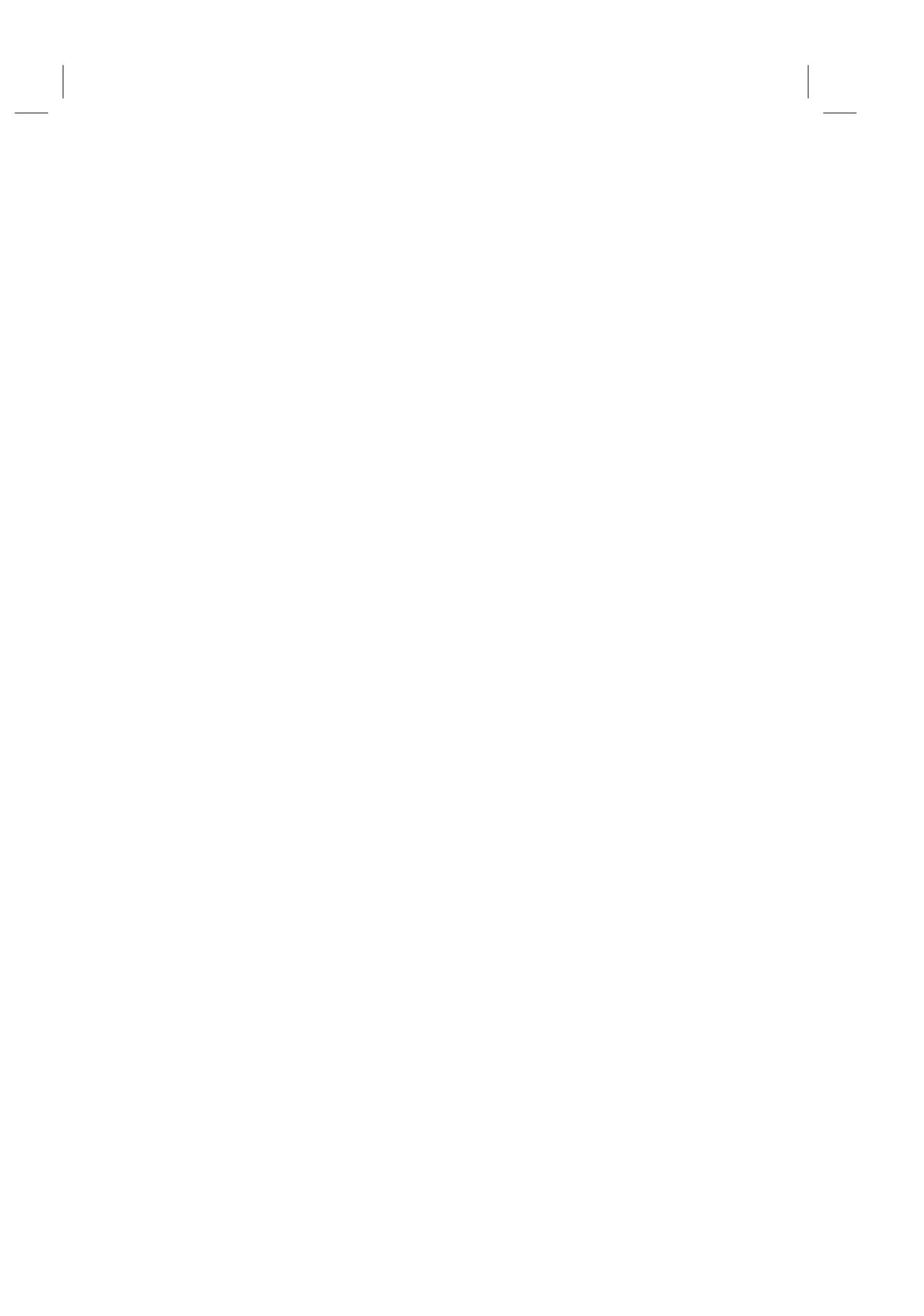
```
krig_res_gbm <-
  krig(
    residuosgbm ~ 1,
    location = suelos,
    newdata = suelos_pred,
    model = semiv_aj_gbmk
  )
#> [using ordinary kriging]

gbmk_pred <-
  predict(ajuste_gbm,
         newdata = suelos_pred,
         na.action = na.pass) +
  krig_res_gbm$var1.pred

pred_err_corr <-
  cbind(pred_err_corr,
        "COS_pred_GBM" = gbmk_pred)

tm_shape(pred_err_corr) +
  tm_fill("COS_pred_GBM", style = "cont")
```





A

Herramientas de software

R (R. C. Team, 2019) es un lenguaje de programación orientado a objetos (Ihaka y Gentleman, 1996). Es un software libre y de código abierto, lo que significa que puede ser usado, compartido y modificado libremente. Cualquier persona puede participar en el desarrollo de nuevas funciones y disponibilizarlas para la comunidad de los usuarios de R en forma de paquetes (packages), por lo que R se transformó en uno de los lenguajes de programación más utilizados en Estadística. Presenta potentes capacidades para el procesamiento y visualización, no solo de datos espaciales, sino también de otros tipos de dato. R puede ser instalado en plataformas Windows, Mac OS y en sistemas basados en Linux. Existen múltiples entornos de desarrollo integrado (Integrated Development Environment IDE) los cuales facilitan la programación. Ejemplos de este tipo de software es el intérprete de R que contiene InfoStat (Di Rienzo et al., 2019) y RStudio (R. Team, 2019). Para poder utilizar el intérprete de R en InfoStat es necesario tener instalados ambos programas. El instalador de InfoStat está disponible en <http://infostat.com.ar/>, mientras que R puede descargarse en <https://cran.r-project.org/>. El instalador de RStudio puede descargarse desde la página <https://rstudio.com/products/rstudio/download/#download>.

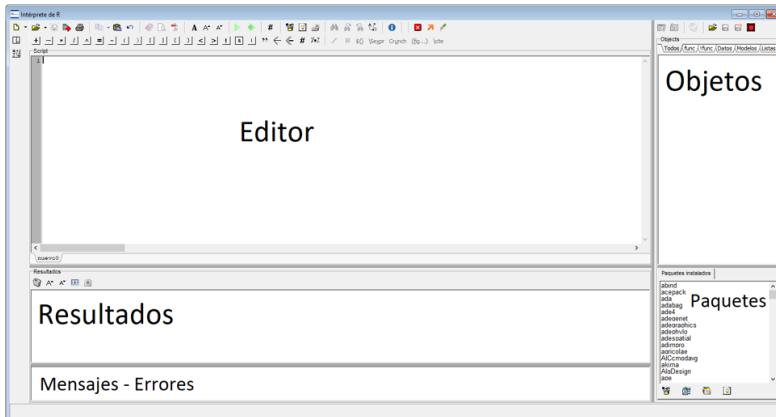
A.1 Introducción al manejo de datos espaciales con R

Numerosos paquetes para el manejo de datos espaciales se encuentran en repositorios digitales de R, ejemplos de ellos son `geoR`, `sp`, `gstat`, `rgdal`, `spdep`, `sf`, `stars` y `raster`. En los últimos años se han desarrollado paquetes especializados como `ggplot2`, `leaflet` y `tmap` que incrementaron considerablemente las capacidades para elaboración de gráficos y mapas, tanto estáticos como interactivos. La sintaxis de estos paquetes usa distintos niveles de información, *i.e.* individualmente se especifica cada nivel del gráfico y luego éstos se combinan para obtener el gráfico completo. Ejemplos de estos niveles son los datos, la estética, los objetos geométricos, las escalas, las particiones, entre otros. Es posible el análisis de datos espaciales utilizando R, sin necesidad de usar un software GIS.

A.2 Intérprete de R en InfoStat

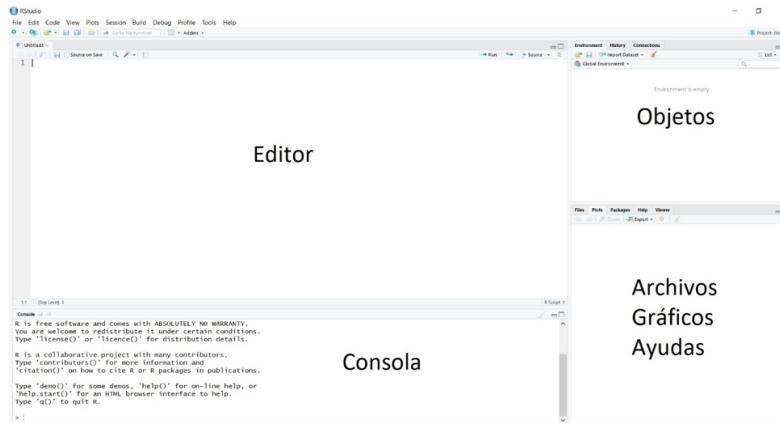
La interfaz del intérprete de R en InfoStat se divide en cinco paneles. El panel superior izquierdo permite al usuario visualizar scripts previamente escritos o escribir nuevos. En el panel Resultados se muestran las salidas y resultados en forma de texto, los resultados gráficos se mostrarán en una nueva ventana. Debajo de este panel, se muestran información ya sea detección de un error durante la ejecución de un comando, como la correcta finalización de ciertos procedimientos. En los paneles derechos se muestran los objetos cargados en el ambiente de trabajo, mientras

que en el panel inferior derecho se muestran los paquetes instalados y en rojo los paquetes cargados.



A.3 RStudio

La interfaz de RStudio se divide en cuatro paneles, a su vez, cada panel puede contener más de una pestaña. El panel superior izquierdo permite al usuario cargar scripts previamente escritos o escribir nuevos. En el panel consola se muestran las sentencias de código ejecutadas y los resultados. En los paneles derechos se muestran los objetos cargados en el ambiente de trabajo, mientras que en el panel inferior derecho se muestran archivos en el directorio de trabajo, gráficos generados, ayudas.



Referencias

- Anselin, L. (1995). Local indicators of spatial association—LISA. *Geographical analysis*, 27(2), 93-115.
- Babai, L. (1979). Monte-Carlo algorithms in graph isomorphism testing. *Université de Montréal Technical Report, DMS*, 79–10.
- Bakka, H., Rue, H., Fuglstad, G. A., Riebler, A., Bolin, D., Illian, J., Krainski, E., Simpson, D., y Lindgren, F. (2018). Spatial modeling with R-INLA: A review. *Wiley Interdisciplinary Reviews: Computational Statistics*, February, 1-24.
<https://doi.org/10.1002/wics.1443>
- Balzarini, M., Macchiavelli, R., y Casanoves, F. (2004). Aplicaciones de modelos mixtos en agricultura y forestería. *Curso de Capacitación Centro Agronómico Tropical de Investigación y Enseñanza-CATIE*.
- Bezdek, J. C., Coray, C., Gunderson, R., y Watson, J. (1981). Detection and characterization of cluster substructure i. linear structure: Fuzzy c-lines. *SIAM Journal on Applied Mathematics*, 40(2), 339-357.
- Blangiardo, M., y Cameletti, M. (2015). *Spatial and spatio-temporal Bayesian models with R-INLA*. John Wiley & Sons.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.
- Breiman, L., Friedman, J. H., Olshen, R. A., y Stone, C.

- J. (2017). Classification and regression trees. En *Classification and Regression Trees*. <https://doi.org/10.1201/9781315139470>
- Brenning, A. (2012). Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The R package sperrorest. *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, 5372-5375.
- Cameletti, M., Lindgren, F., Simpson, D., y Rue, H. (2013). Spatio-temporal modeling of particulate matter concentration through the SPDE approach. *AShA Advances in Statistical Analysis*, 97(2), 109-131. <https://doi.org/10.1007/s10182-012-0196-3>
- Clifford, P., Richardson, S., y Hemon, D. (1989). Assessing the Significance of the Correlation between Two Spatial Processes. *Biometrics*, 45(1), 123-134. <https://doi.org/10.2307/2532039>
- Córdoba, M., Bruno, C., Costa, J. L. J., y Balzarini, M. (2012). Principal component analysis with georeferenced data. An application in precision agriculture. *Rev. FCA UNCUYO*, 44(1), 27-39.
- Córdoba, M., Bruno, C., Costa, J. L., y Balzarini, M. (2013). Subfield management class delineation using cluster analysis from spatial principal components of soil variables. *Computers and Electronics in Agriculture*, 97, 6-14. <https://doi.org/10.1016/j.compag.2013.05.009>
- Correa Morales, J. C., Causil, B., y Javier, C. (2018). *Introducción a la estadística bayesiana: notas de clase*. Instituto Tecnológico Metropolitano.
- Cover, T., y Hart, P. (1967). Nearest neighbor pattern

- classification. *IEEE transactions on information theory*, 13(1), 21-27.
- Cressie, N., y Wikle, C. K. (2015). *Statistics for spatio-temporal data*. John Wiley & Sons.
- Di Rienzo, J. A., Casanoves, F., Balzarini, M. G., Gonzalez, L., Tablada, M., y Robledo, C. W. (2019). *InfoStat*.
- Dray, S., Chessel, D., y Thioulouse, J. (2003). Co-inertia analysis and the linking of ecological data tables. *Ecology*, 84(11), 3078-3089.
- Dray, S., Saïd, S., y Débias, F. (2008). Spatial ordination of vegetation data using a generalization of Wartenberg's multivariate spatial correlation. *Journal of vegetation science*, 19(1), 45-56.
- Dutilleul, P., Clifford, P., Richardson, S., y Hemon, D. (1993). Modifying the t Test for Assessing the Correlation Between Two Spatial Processes. *Biometrics*, 49(1), 305. <https://doi.org/10.2307/2532625>
- Efron, B., y Hastie, T. (2016). Computer age statistical inference: Algorithms, evidence, and data science. En *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. <https://doi.org/10.1017/CBO9781316576533>
- Efron, B., y Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438), 548-560.
- Frogbrook, Z. L., y Oliver, M. A. (2007). Identifying management zones in agricultural fields using spatially constrained classification of soil and ancillary data. *Soil Use and Management*, 23(1), 40-51. <https://doi.org/10.1111/j.1475-2743.2006.00065.x>
- Fukuyama, Y., y Sugeno, M. (1989). A new method of choo-

sing the number of clusters for the fuzzy c-mean method. *Proc. 5th Fuzzy Syst. Symp.*, 1989, 247-250.

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, 58(3), 453-467.

Gabriel, K. R., y Sokal, R. R. (1969). A new statistical approach to geographic variation analysis. *Systematic zoology*, 18(3), 259-278.

Galarza, R., Mastaglia, M. N., Albornoz, E. M., y Martínez, C. (2013). Identificación automática de zonas de manejo en lotes productivos agrícolas. *V Congreso Argentino de Agroinformática (CAI) e 42da. JAIIo*.

Hang, S., Negro, G., Becerra, A., y Rampoldi, A. E. (2015). *Suelos de Córdoba: Variabilidad de las propiedades del horizonte superficial*. Jorge Omar Editorial.

Hengl, T., Heuvelink, G. B., y Rossiter, D. G. (2007). About regression-kriging: From equations to case studies. *Computers and Geosciences*, 33(10), 1301-1315. <https://doi.org/10.1016/j.cageo.2007.05.001>

Ihaka, R., y Gentleman, R. (1996). R: a language for data analysis and graphics. *Journal of computational and graphical statistics*, 5(3), 299-314.

Kanevski, M., Timonin, V., Pozdnukhov, A., y Ritter, G. (2009). Machine learning for spatial environmental data: theory, applications, and software. En *Ssrn*. EPFL press. <https://doi.org/10.2139/ssrn.3015609>

Kuhn, M., y Johnson, K. (2013). *Applied predictive modeling* (Vol. 26). Springer.

Lee, D.-T., y Schachter, B. J. (1980). Two algorithms for

- constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3), 219-242.
- Li, J., Heap, A. D., Potter, A., y Daniell, J. J. (2011). Application of machine learning methods to spatial interpolation of environmental variables. *Environmental Modelling & Software*, 26(12), 1647-1659.
- Lindgren, F., y Rue, H. (2015). Bayesian Spatial Modelling with R - INLA. *Journal of Statistical Software*, 63(19). <https://doi.org/10.18637/jss.v063.i19>
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., y Leisch, F. (2019). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien. <https://CRAN.R-project.org/package=e1071>
- Milne, A. E., Webster, R., Ginsburg, D., y Kindred, D. (2012). Spatial multivariate classification of an arable field into compact management zones based on past crop yields. *Computers and Electronics in Agriculture*, 80, 17-30. <https://doi.org/10.1016/j.compag.2011.10.007>
- Morrell, C. H. (1998). Likelihood ratio testing of variance components in the linear mixed-effects model using restricted maximum likelihood. *Biometrics*, 1560-1568.
- Patterson, H. D., y Thompson, R. (1971). Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3), 545-554.
- Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559-572. <https://doi.org/10.1080/14786440109462720>
- Pejović, M., Nikolić, M., Heuvelink, G. B., Hengl, T.,

- Kilibarda, M., y Bajat, B. (2018). Sparse regression interaction models for spatial prediction of soil properties in 3D. *Computers and Geosciences*, 118, 1-13. <https://doi.org/10.1016/j.cageo.2018.05.008>
- Ping, J. L., y Dobermann, A. (2003). Creating Spatially Contiguous Yield Classes for Site-Specific Management. *Agronomy Journal*, 95(5), 1121. <https://doi.org/10.2134/agronj2003.1121>
- Rue, H., Martino, S., y Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2), 319-392.
- Schabenberger, O., y Gotway, C. A. (2005). *Statistical methods for spatial data analysis*. CRC press.
- Team, R. (2019). *RStudio: Integrated Development Environment for R*.
- Team, R. C. (2019). *R: A Language and Environment for Statistical Computing*.
- Vallejos, R., Mallea, A., Herrera, M., y Ojeda, S. (2015). A multivariate geostatistical approach for landscape classification from remotely sensed image data. *Stochastic Environmental Research and Risk Assessment*, 29(2), 369-378. <https://doi.org/10.1007/s00477-014-0884-5>
- Webster, R., y Oliver, M. A. (2007). Geostatistics for environmental scientists. En *Vadose Zone Journal* (Vol. 1). John Wiley & Sons. <https://doi.org/10.2136/vzj2002.0321>
- Xie, X. L., y Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis & Ma-*

chine Intelligence, 13(8), 841-847. <https://doi.org/10.1109/34.85677>

Zimback, C. R. L. (2001). *Analise espacial de atributos químicos de solos para fins de mapeamento da fertilidade do solo.* 2001. 114 f.

