

Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Estimación de la radiación solar mediante técnicas de
geoestadística utilizando una red de sensores
distribuidos

Autor: Mario Fernández Rangel

Tutor: Amparo Núñez Reyes

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019



Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Estimación de la radiación solar mediante técnicas de geoestadística utilizando una red de sensores distribuidos

Autor:

Mario Fernández Rangel

Tutora:

Amparo Núñez Reyes

Profesora titular

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Grado: Estimación de la radiación solar mediante técnicas de geoestadística utilizando una red de sensores distribuidos

Autor: Mario Fernández Rangel

Tutora: Amparo Núñez Reyes

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

A mi familia

Agradecimientos

Quiero darle las gracias a mi tutora, Amparo Núñez Reyes, sin su ayuda este Trabajo de Fin de Grado no hubiera podido salir adelante.

También estoy muy agradecido a todas las personas que me han ayudado a lo largo de este grado, sus consejos han sido muy valiosos para llegar hasta este momento.

Darle las gracias a mis amigos que siempre han estado conmigo durante este tiempo y ayudarme a aliviar cuando la carga era demasiado excesiva para soportarla.

Finalmente, agradecer a mi familia todo el ánimo que he recibido.

Quiero acabar con una cita de mi escritor favorito:

“The most important step a person can take is always the next one.” Brandon Sanderson

Mario Fernández Rangel

Alumno del Grado en Ingeniería en Tecnologías Industriales

Sevilla, 2019

Resumen

Desde el Protocolo de Kioto en 1997 hasta nuestros días, se han llevado a cabo diferentes estrategias legales y obligatorias para hacer frente al cambio climático y minimizar sus impactos en la naturaleza y en la sociedad. La lucha contra el cambio climático es el mayor impulso a las energías renovables y las tecnologías de mayor eficiencia energética.

El presente trabajo se encuentra enmarcado dentro del contexto de la energía solar (térmica, fotovoltaica o termoeléctrica). Donde el conocimiento de la radiación solar es decisivo a la hora de mejorar la eficiencia de las plantas solares permitiendo optimizar el control posterior de los colectores solares.

Por ello, en el presente trabajo se han llevado a cabo investigaciones para el análisis, la variabilidad espacial y la estimación de la radiación solar utilizando técnicas de geoestadística. Para el estudio y utilizando datos provenientes de satélites de la Unión Europea, dentro del programa Copernicus, se ha simulado una red de sensores distribuidos a lo largo de una hipotética planta solar ubicada en el desierto de tabernas (Almería).

Agradecimientos	ix
Resumen	xi
Índice	xiii
Índice de Tablas	xv
Índice de Figuras	xvii
Notación	xix
1 Introducción	1
2 Redes de sensores inalámbricas	3
2.1. <i>Redes de sensores inalámbricas</i>	3
3 Geoestadística	6
3.1 <i>Introducción.</i>	6
3.2 <i>Variograma</i>	6
3.2.1 Modelos de variograma	8
3.3 <i>Kriging</i>	9
4 Origen y Análisis de los datos. Simulación de una red de sensores	10
4.1 <i>Ubicación geográfica de datos</i>	10
4.2 <i>Origen de los datos</i>	12
4.3 <i>Análisis exploratorio de datos para cuatro periodos de tiempo</i>	14
4.4 <i>Reproyectar</i>	19
5 Variabilidad espacial y estimación de la radiación solar	21
5.1 <i>Variabilidad espacial de la radiación solar</i>	21
5.2 <i>Predicción de la radiación solar mediante Kriging ordinario</i>	27
5.2.1 Ejemplo de Kriging ordinario en R.	27
5.2.2 Comparación de predicciones con distintos modelos de variograma.	31
6 Conclusiones	35
7 Anexo a: Código realizado	36
7.1 <i>Código realizado en R.</i>	36
7.2 <i>Código en Matlab</i>	47
Referencias	59

ÍNDICE DE TABLAS

Tabla 1. Distancia entre los puntos según la fórmula.	18
Tabla 2. Resultados de la validación cruzada de los datos.	30
Tabla 3. Media de los errores de la predicción de la hora (c) comparando con datos reales.	32
Tabla 4. Media de los errores de la predicción de los datos del mes (a) comparando con datos reales.	33
Tabla 5. Media de los errores de la predicción de los datos de días (b) comparando con datos reales.	33
Tabla 6. Media de los errores de la predicción de los datos de instantes (d) comparando con datos reales.	33

ÍNDICE DE FIGURAS

Figura 1. Distribución de la red de sensores (1).	3
Figura 2. Arquitectura del protocolo de comunicación de una WSN (1).	5
Figura 3. Ejemplo de variograma experimental (7).	7
Figura 4. Variograma experimental con modelo lineal.	9
Figura 5. Mapa del desierto de Tabernas (11).	11
Figura 6. Ubicación de los nodos de la red de sensores simulada.	12
Figura 7. Interfaz de usuario del CAMS Radiation Service.	13
Figura 8. Visualización de los datos descargados.	14
Figura 9. Nube de puntos.	15
Figura 10. Gráfica generada mediante Desc del mes de junio.	16
Figura 11. Gráfica q-q para el mes de junio.	17
Figura 12. Distribución de los valores de los datos de los días de junio	17
Figura 13. Gráfica q-q para los datos de días de junio	18
Figura 14. Histograma del mes de junio	19
Figura 15. Mapa con los valores de radiación de los nodos.	19
Figura 16. Zonas UTM de España (20).	20
Figura 17. Variograma experimental con la muestra de datos del mes de junio.	21
Figura 18. Direcciones para el variograma del mes de junio.	22
Figura 19. Variograma con modelo teórico del mes de junio.	23
Figura 20. Variogramas direccionales junto al modelo del variograma del mes de junio.	23
Figura 21. Variograma con el modelo de b)	24
Figura 22. Variograma con el modelo de c)	25
Figura 23. Variograma con el modelo de d)	25
Figura 24. Variogramas de los distintos meses del año 2015.	26
Figura 25. Predicción de las horas con el modelo de variograma del mes de junio.	27
Figura 26. Desviación típica de la predicción de horas con el modelo de variograma de junio.	28
Figura 27. Resultado de la validación cruzada con el modelo de junio del variograma.	29
Figura 28. Comparación de las estimaciones con datos normalizados y sin normalizar.	30
Figura 29. Comparación de las predicciones de la hora de los distintos modelos de variograma.	31
Figura 30. Puntos de validación sobre la predicción.	32

WSN	Wireless Sensor Network (Red inalámbrica de sensores)
ZEPA	Zona Especial de Protección para las Aves
PVGIS	Photovoltaic Geographical Information System
TOA	Radiación en plano horizontal por encima de la atmósfera
GHI	Radiación global en plano horizontal a nivel del suelo
BHI	Radiación directa en plano horizontal a nivel del suelo
DHI	Radiación difusa en plano horizontal a nivel del suelo
BNI	Radiación directa en plano móvil perpendicular a la incidencia solar
CPV	Concentrador fotovoltaico
CRS	Coordinates Reference System
UTM	Coordenadas universales transversal de Mercator
EPSG	European Petroleum Survey Group
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
R^2	Coefficiente de determinación.
IDW	Inverse Distance Weighting

1 INTRODUCCIÓN

La creciente preocupación por las graves consecuencias producidas por el calentamiento global, está conduciendo a la sociedad, a un progresivo endurecimiento sobre el control normativo de las emisiones de los gases de efecto invernadero a la atmósfera. Surgiendo cada año nuevas normas locales, nacionales, europeas e internacionales, como el protocolo de Kyoto que comprende dos períodos. El primero desde 2008 hasta 2012 y el segundo y actual desde 2013 hasta 2020. Siendo el principal objetivo de dicho acuerdo internacional, la reducción de las emisiones de seis gases de efecto invernadero relacionados directamente como causantes del calentamiento global, uno de los mayores problemas a los que se enfrenta el mundo globalizado actualmente.

Por todo ello se fomenta y justifica el uso de las energías renovables. Dentro de este contexto, se encuentra el presente trabajo, donde se estudia la variabilidad espacial de la radiación solar de una determinada zona del desierto de Tabernas para posteriormente realizar la interpolación espacial, estimando la radiación solar en los lugares donde no se ubiquen sensores, ya sea por motivos geográficos o por costes económicos.

El escenario hipotético en el que se realiza el estudio consiste en una planta solar ubicada en el desierto de Almería. Dicha planta puede ser de cualquier tipo y dispone de una red de sensores distribuidos irregularmente por todo el terreno. La planta presenta colectores solares que pueden ser controlados automáticamente mediante un controlador avanzado de procesos. Para ello es primordial conocer la radiación solar por toda la zona en la que se encuentran los paneles solares.

Los datos se han obtenido de satélites disponibles de forma libre y se ha simulado una red de sensores de 49 nodos distribuidos irregularmente. Con el uso de técnicas de geoestadística, el análisis estructural de los datos ha permitido obtener información sobre la variabilidad espacial de la radiación solar, utilizándose para la estimación posterior de la radiación solar en diferentes periodos de tiempo.

No se pretende entrar en profundidad en los desarrollos teóricos de los métodos y técnicas de la geoestadística, el principal objetivo consiste en lograr aplicar las técnicas mediante programas informáticos realizados mediante los entornos de programación R y Matlab, así como estudiar los resultados obtenidos para comprobar la viabilidad que puedan tener en entornos reales.

- **Organización de la memoria.**

El documento se ha dividido en las siguientes partes:

- **Redes de sensores inalámbricos.** En el primer capítulo, se hace una descripción de esta tecnología y las complejidades de su diseño, los problemas que presenta actualmente y las investigaciones actuales en curso para su mejora con el fin de resolverlos.
- **Geoestadística.** Se explica los conceptos importantes de la geoestadística necesarios para comprender su funcionamiento y realizar las estimaciones mediante kriging, el método elegido para estimar la radiación solar.
- **Origen y análisis de los datos.** Donde se explica en que consiste nuestro trabajo, elegimos el lugar y describimos la búsqueda de los datos de radiación solar, realizamos el análisis necesario de los datos y en último lugar, explicamos en que consiste la reproyección del sistema de coordenadas.
- **Variabilidad espacial y estimación de la radiación solar.** Dentro de este capítulo se muestran los resultados de las investigaciones realizadas en el trabajo que incluyen: Representación del variograma experimental, ajuste de los modelos teóricos para el variograma, la obtención de las estimaciones de los valores de radiación solar mediante kriging y los cálculos de los errores cometidos. Validaremos el resultado de nuestro modelo mediante validación cruzada y el grado de ajuste que tenemos a nuestros datos.

- **Conclusiones.** Aquí se exponen las conclusiones más importantes obtenidas sobre el trabajo realizado y proponemos líneas de investigación futuras para desarrollar este trabajo y lograr nuevos objetivos o mejoras posibles.

Finalmente, incluimos un anexo con todo el código desarrollado para el trabajo.

Disclaimer:

Contains modified Copernicus Atmosphere Monitoring Service Information 2019.

Neither the European Union nor ECMWF is responsible for the use of the information

2 REDES DE SENSORES INALÁMBRICAS

Las redes de sensores inalámbricas es un tema de actualidad en la investigación científica, todavía falta un tiempo para que su uso se extienda en diversas aplicaciones, pero probablemente sea una tecnología que tendrá mucha incidencia en el futuro.

2.1. Redes de sensores inalámbricas

Una red de sensores inalámbricos, conocida en inglés como Wireless sensor network (WSN), se define como un grupo de nodos distribuidos espacialmente conectados de forma inalámbrica, normalmente cuentan con un terminal común encargado de recoger toda la información que obtienen los sensores colocados en los nodos y trabajan de forma cooperativa para mandar la información obtenida. Su función suele ser monitorizar y/o controlar una variable ambiental como presión, temperatura, contaminantes, movimiento, niveles de ruido, etc.

Es una tecnología reciente y en investigación, por lo que sus aplicaciones todavía están por desarrollar, en (1 págs. 393-422), se repasan sus posibles aplicaciones, que pueden incluir:

- Aplicaciones militares.
- Aplicaciones en el medio ambiente: detección de incendios, mapeado del medio, control de especies animales, detección de inundaciones.
- Aplicaciones en la salud: Monitorización de datos fisiológicos, rastreo de pacientes y médicos.
- Aplicaciones en hogares: Monitorización de variables ambientales, automatización del hogar.
- Otros usos comerciales: Automatización de fábricas, control de inventarios

En (2 págs. 2292-2320) se clasifican las WSN en dos tipos, estructuradas y desestructuradas. Las desestructuradas contienen muchos nodos de sensores y se pueden colocar de manera aleatoria, se usan para funciones de monitorización, aunque su mantenimiento es costoso. Las redes estructuradas permiten colocar menos nodos y tienen menor coste de mantenimiento, aunque su colocación puede ser más costosa.

Para ilustrar de forma gráfica la WSN, podemos visualizar este esquema:

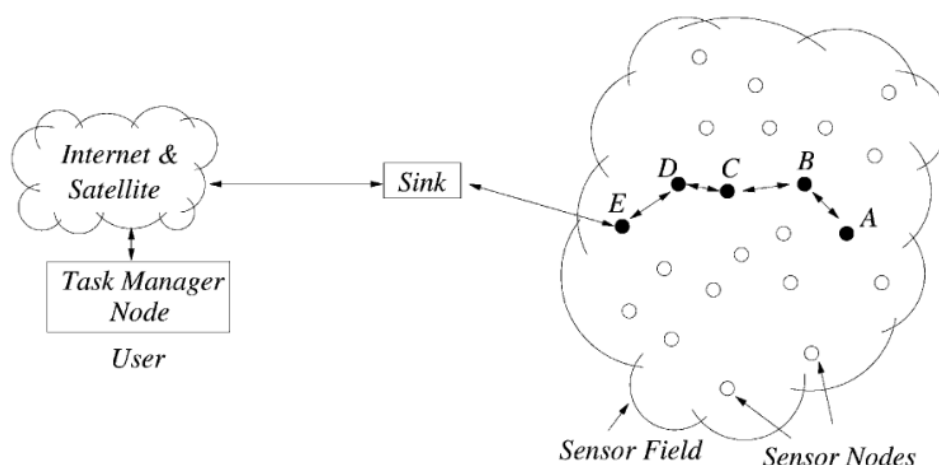


Figura 1. Distribución de la red de sensores (1).

El diseño de la WSN depende de factores como la tolerancia de la tolerancia, la escalabilidad, los costes de producción, las limitaciones del hardware, la topología de la red, el medio, el canal de transmisión y el consumo de energía.

En la actualidad, una de las mayores restricciones de las WSN puede ser el consumo de energía debido a que el método más usual de suministro ha sido mediante baterías que se agotan en el tiempo, por eso es necesario reducir el consumo en todos los pasos del diseño. Si fuera posible suministrar una energía renovable como la energía solar, no existiría esta limitación, por eso la instalación en una planta fotovoltaica, que además pueda suministrar energía a la WSN que permita mejorar la eficiencia en la obtención de energía sería ideal. Otro de los problemas es que los nodos de sensores sean capaces de organizarse a sí mismos en una red y ser capaz de manejarse de forma eficiente (2).

Normalmente, las WSN se colocan en la región objetivo sin una infraestructura de apoyo (3), los propios nodos toman la información que captan los sensores y se mandan al nodo “getaway” o “sink” en la que se vuelca toda la información y que suele estar conectado a Internet, mediante este nodo llegan los datos al terminal conectado mediante Internet, llamado estación base, desde la que un usuario puede controlar o monitorizar la información generada.

Debido a que los WSN suelen ser colocados sin una infraestructura de soporte, necesitan tener un protocolo de comunicación para transmitir la información de forma inalámbrica que suele incluir señales y detección de error como los protocolos usados habitualmente por los equipos informáticos, por ejemplo, el protocolo TCP. Este protocolo debe ser diseñado con el objetivo de reducir el consumo de energía de los nodos, transmitir de forma eficiente la información y promover el esfuerzo cooperativo de los nodos (1).

Las capas del protocolo se describen de la siguiente forma (1) y (2):

- Capa de aplicación: Dependiendo de la tarea, se pueden usar diferentes tipos de aplicaciones software, según las necesidades o requerimiento del uso que le demos a la red de sensores inalámbricos.
- Capa de transporte: Asegura la fiabilidad y calidad de los datos en el proceso. Podemos ver una lista de los protocolos de comunicación de la capa de transporte en (2).
- Capa de red: Enruta los datos que manda la capa de transporte a través de la red desde su fuente hasta el destino.
- Capa de enlace de datos: Maneja el acceso al canal de datos para evitar errores. Dado que el medio es inalámbrico, es necesario un control del acceso al medio.
- Capa física: Lleva a cabo las necesidades de métodos de modulación, recepción y envío.

Además, los planos de energía, movilidad y de gestión de tareas coordinan las tareas y reducen el consumo de energía de la red.

En la figura 2 se observa una representación del modelo de comunicación de las redes de sensores:

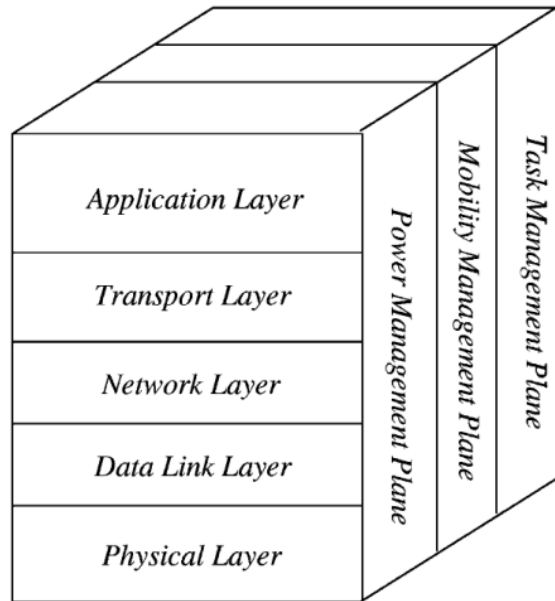


Figura 2. Arquitectura del protocolo de comunicación de una WSN (1).

Otra de las áreas de investigación actual es la eficiencia de la red en el diseño, además de tener en cuenta el coste y los requerimientos de la tarea a realizar, para ello hay que optimizar el hardware y el software utilizado. Esto incluye la elección de los sensores en la parte de hardware y el software se refiere a la seguridad, robustez, tolerancia de fallos, organización, la red utilizada para el intercambio de información (2).

Se dividen los problemas actuales de la tecnología en tres categorías que son (2):

- La plataforma de las distintas capas y el sistema operativo.
- La arquitectura del protocolo de comunicación.
- Los servicios de red, aprovisionamiento y problemas de despliegue.

También se analizan y comparan las posibles soluciones y las investigaciones actuales en cada categoría.

En este trabajo se va a suponer que disponemos de una red de sensores inalámbricos de tipo desestructurado, ya que permiten una colocación más sencilla en el espacio y resulta conveniente para nuestro método de predicción mediante kriging.

3 GEOESTADÍSTICA

3.1 Introducción.

La historia de la geoestadística comienza en los años 50, en el campo de la minería, con el fin de estimar las localizaciones de reservas de mineral en Sudáfrica, el doctor Krige, con ayuda de H. S. Sichel, tuvieron la idea de crear un nuevo método matemático para localizar los depósitos de oro de Sudáfrica. Posteriormente la formulación y expansión de las ideas de Krige fue desarrollada por el francés Matheron en 1962.

Aunque el inicio de la geoestadística fue en la minería, se ha extendido hacia otros campos de la geografía, y en otras áreas de la ciencia y la ingeniería: en la salud, se puede usar para predecir los niveles de contaminante del aire, en la geoquímica, en hidrología, climatología, etc.

El objetivo de la geoestadística es la caracterización de fenómenos naturales, estimando un parámetro de interés en función de un set de datos, además es capaz de estimar el error cometido. Estos métodos se basan en la existencia de una correlación espacial en el fenómeno sobre el que trabajamos, es decir, la noción de que los puntos más cercanos en el espacio deben tener valores más acordes que si nos alejamos, y mediante estas técnicas podemos llegar a identificarlo matemáticamente.

Consideramos que las variables tienen dos componentes, una determinista y otra aleatoria, la determinista representa influencias que no son afectadas por la aleatoriedad y la parte aleatoria tiene que ver con un proceso estocástico (también llamado proceso aleatorio), procesos que tienen incertidumbres en su modelo, evoluciona en el tiempo y puede cambiar. Esta parte aleatoria puede ser caracterizada mediante momentos estadísticos, como la varianza. Si conocemos esta correlación podemos estimar puntos que no tenemos datos mediante interpolación con los datos próximos. (3)

El variograma describe la relación espacial de los puntos, mientras que mediante la técnica de Kriging (nombre acuñado por Matheron en honor a su primer descubridor) usamos el modelo teórico que mejor se ajusta al variograma para obtener una predicción sobre una superficie a partir de datos aislados dentro de ella.

Las variables ambientales son medidas cuantitativas o descriptivas de características del medio ambiente, normalmente son recogidas mediante muestras de campo, pero debido a que tomar los datos es el mayor gasto de los sondeos, los equipos de sondeos visitan un número limitado de muestras, y la calidad de los mapas de las variables ambientales es limitada, por eso se está empezando a remplazar por otras formas de muestreo como redes de sensores (4).

3.2 Variograma

El variograma es una función matemática que representa el grado de relación en el espacio de un campo aleatorio, es decir, la relación entre los valores medidos dependiendo de la distancia que separa los puntos.

El término variograma, es usado por algunos autores como sinónimo de semivariograma, aunque según (5), el término adecuado debe ser variograma, ya que el otro proviene de una confusión en la literatura.

Las variables ambientales se pueden expresar en función de:

$$Z(s) = Z^*(s) + \epsilon'(s) + \epsilon''(s)$$

Siendo Z^* =Componente determinista ϵ' = Correlación espacial ϵ'' = Error de medición o ruido.

La hipótesis común es la estacionaridad, es decir, que las propiedades del conjunto de datos no dependen de su posición absoluta sino de la posición relativa del espacio, implica que la varianza y media sea constante en el espacio y que la covarianza depende solo de la diferencia de lugar entre los dos puntos (6), entonces se puede escribir lo siguiente:

$$\text{var}[z(s)] = \sigma^2, E[z(s)] = \mu$$

$$\text{cov}[z(s_1), z(s_2)] = f(s_1 - s_2)$$

Si se tienen los valores de dos puntos distintos, s_1 y s_2 , el variograma se define como:

$$2\gamma(s_1, s_2) = \text{var}(Z(s_1) - Z(s_2))$$

Si se tiene una media constante en el espacio de las medidas para todos los puntos, $\mu(s) = 0$, entonces se puede reescribir la ecuación:

$$2\gamma(s_1, s_2) = E[(Z(s_1) - Z(s_2))^2]$$

Luego se obtiene que:

$$\gamma(s_1, s_2) = \frac{1}{2} * E[(Z(s_1) - Z(s_2))^2]$$

Finalmente, realizando simplificaciones matemáticas (3):

$$\gamma(h) = \sigma^2 - \text{Cov}(h)$$

Siendo h la distancia entre dos puntos s_1 y s_2 , también llamado “lag distance”.

Si representamos los valores del variograma para dos puntos frente a la distancia de los pares de puntos para una serie de datos para todas las parejas de datos, obtenemos el variograma experimental.

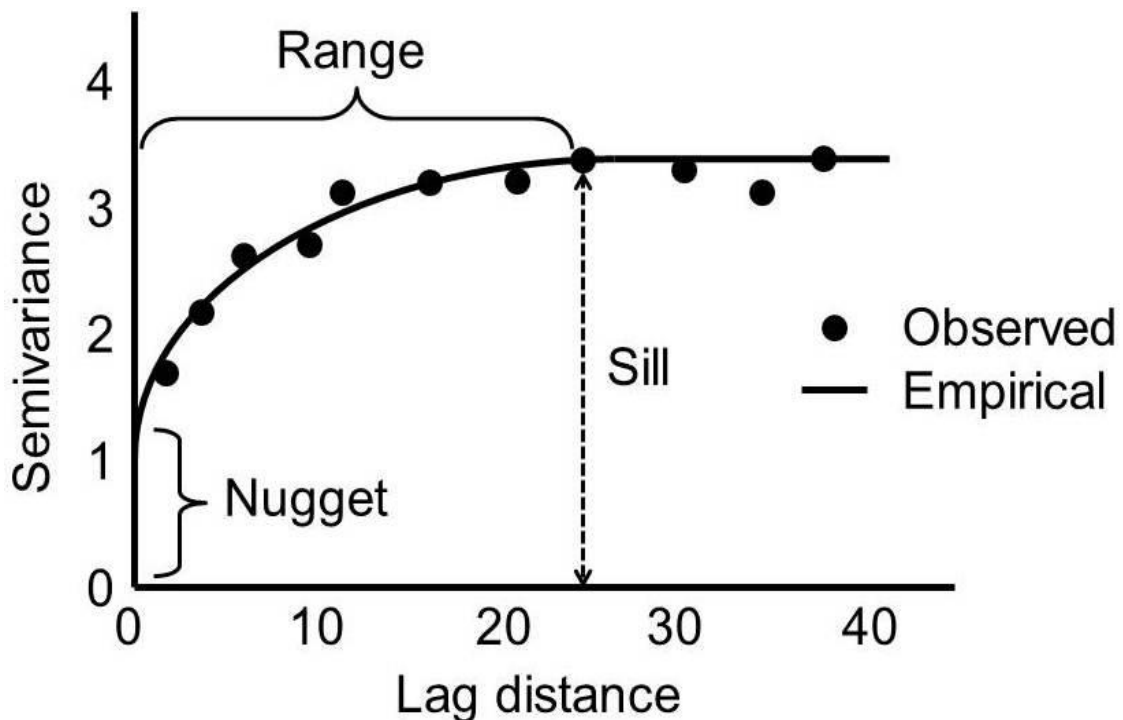


Figura 3. Ejemplo de variograma experimental (7).

Si la distancia aumenta, significa que tenemos puntos más alejados, por lo que la correlación espacial debería disminuir, entonces el valor del variograma aumenta. El gráfico se refiere al variograma como semivariance debido a la confusión en la literatura explicada anteriormente (5).

Podemos distinguir ciertas características del variograma:

- Sill o meseta: Valor en el que la curva del variograma se aplana, es decir el valor del variograma deja de aumentar y tiene forma horizontal o parecida. Indica que la correlación de los puntos deja de estar espacialmente afectada.
- Range o rango: Distancia en la que comienza la meseta, si la distancia entre un par de puntos es mayor que este valor, la relación entre los valores de la pareja se suele considerar despreciable.
- Nugget o pepita: Valor del variograma cuando la distancia es igual a 0, teóricamente debería ser cero ya que nos acercamos a un mismo punto (si distancia=0 es el mismo punto), pero puede aparecer una diferencia debido a errores de medición o pequeñas variaciones locales de los datos.

A partir de la disposición de los datos experimentales, obtenemos lo que se conoce como variograma experimental, según la forma del variograma experimental podemos distinguir distintos modelos teóricos, que son los que debemos utilizar para representar la relación de los datos.

El estimador tradicional del variograma viene dado por la fórmula siguiente:

$$2\hat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{N(h)} [z(s_i) - z(s_j)]^2$$

Donde

$N(h)$ son los puntos s_i, s_j tal que $\text{dist}(s_i, s_j) = h$

$|N(h)|$ son el total de pares de puntos que se encuentran a una distancia h entre sí.

3.2.1 Modelos de variograma

El variograma experimental está limitado en la realidad por los pares de datos disponibles, que al ser un número de puntos finito y generalmente no demasiados, no tendremos todas las distancias y todas las direcciones posibles que requiere Kriging para realizar las computaciones, por esta razón es necesario asignar un modelo teórico a nuestro variograma experimental que haga válido nuestro variograma para kriging.

Se va a representar el valor de la pepita por b , el rango por a y la meseta es c , la distancia entre los puntos es h .

Algunos modelos son:

- Modelo esférico: $\gamma(h) = \begin{cases} b + c * (1.5 * \frac{h}{a} - 0.5 * \frac{h^3}{a^3}) & \text{si } h < a \\ b + c & \text{si } h \geq a \end{cases}$
- Modelo lineal: $\gamma(h) = \begin{cases} b + c * (\frac{h}{a}) & \text{si } h < a \\ b + c & \text{si } h \geq a \end{cases}$
- Modelo exponencial: $\gamma(h) = b + c * (1 - \exp(-\frac{h}{a}))$

Las ecuaciones de algunos modelos más comunes para la radiación solar en una zona son el esférico y el lineal, en pocas ocasiones se ajusta con un modelo exponencial, de acuerdo con (8 págs. 1085-1094). Podemos ver un ejemplo de modelo lineal en la figura 4.

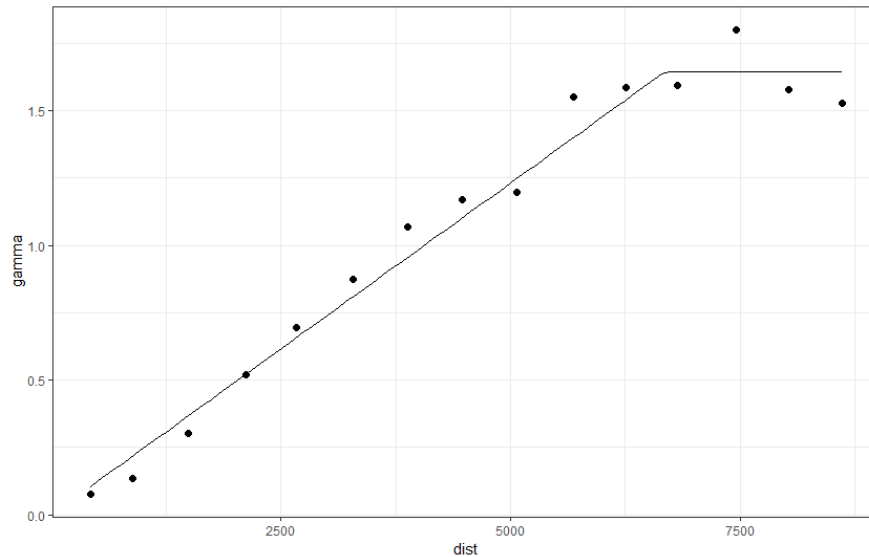


Figura 4. Variograma experimental con modelo lineal.

3.3 Kriging

Es una herramienta de interpolación basada en métodos geoestadísticos, respecto a otros métodos de interpolación tiene la ventaja de ser capaz de estimar la precisión de los resultados de la predicción. Kriging se considera el mejor predictor lineal insesgado (9).

Existen diversos métodos para aplicar kriging, nosotros vamos a utilizar el kriging ordinario.

La estimación de los valores se realiza mediante la siguiente fórmula:

$$\hat{Z}(s_0) = \sum_{i=1}^N w_i(s_0) * Z(s_i) = \lambda_0^T * \vec{Z}$$

Donde λ_0 es el vector de las cantidades de ponderación (w_i) y \vec{Z} es el vector de los valores medidos.

Las ponderaciones se calculan en base a la distancia de los puntos medidos, además, a diferencia de otras técnicas de interpolación, se tienen en cuenta las relaciones espaciales entre los puntos, mediante la covarianza de los datos, solucionando un sistema de ecuaciones. Para más detalles sobre la obtención del sistema (4), (9) o (3).

$$\begin{bmatrix} C_{11} & \cdots & C_{1n} & 1 \\ \vdots & \ddots & \vdots & \\ C_{n1} & \cdots & C_{nn} & 1 \\ 1 & & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ \lambda \end{bmatrix} = \begin{bmatrix} C_{01} \\ \vdots \\ C_{0n} \\ 1 \end{bmatrix}$$

Donde C_{ij} $i,j=1, \dots, n$ es la covarianza entre los puntos i y j , C_{0i} , con $i=1, \dots, n$ es la covarianza entre el punto a predecir y el punto i .

Por otro lado, λ es el multiplicador de Lagrange.

Para datos no gaussianos, el predictor óptimo suele ser no lineal, es conveniente transformar los datos en gaussianos, aunque existen otras formas de kriging no lineales (9).

4 ORIGEN Y ANÁLISIS DE LOS DATOS. SIMULACIÓN DE UNA RED DE SENSORES

En este capítulo se describe con detalle la extracción de los datos para la ubicación elegida, así como el análisis previo de los mismos. La implementación de los algoritmos relacionados con el análisis estructural y la estimación mediante Kriging se ha llevado a cabo con el lenguaje de programación R, dicho entorno de programación dispone de numerosas librerías especializadas en el análisis geoestadístico. También se ha utilizado el software Matlab para el tratamiento de los datos y archivos, complementando la programación realizada en R.

4.1 Ubicación geográfica de datos

No es sencillo encontrar bases de datos de radiación solar proveniente de sensores o estaciones que tengan una resolución temporal baja (de un minuto o inferior) con el que trabajar para detectar las variaciones de radiación solar que producen las nubes a nivel del suelo, por lo que se ha decidido trabajar con resoluciones mayores que enmascaran los cambios de radiación que queremos tener en cuenta a la hora de predecir los valores.

La mayor parte de las bases de datos encontradas usan datos de satélites como el Meteosat, con esto podemos comprobar la viabilidad de esta fuente de información para ver si son un sustituto válido para los datos recogidos mediante sensores, como piranómetros, para medir la radiación solar, lo que supondría una reducción importante de coste de instalación, mantenimiento y compra de estos. Aunque la pega es que los datos de satélites son más inexactos, por lo que habría que comprobar si esta diferencia afecta de manera significativa a los resultados.

También es necesario elegir una localización geográfica adecuada para nuestra planta fotovoltaica con el fin de mostrar la utilidad de nuestras predicciones en un entorno verosímil, de forma que podamos extraer conclusiones aplicables a la realidad del medio.

En cuanto a la zona geográfica en la que desarrollar el trabajo, se ha decidido trabajar hipotéticamente sobre la región del desierto de Tabernas, Almería, este recurso natural se encuentra a unos 30 km al norte de Almería, a una altitud de 400 metros sobre el nivel del mar situado cerca de la población de Taberna. (10 pág. 21), se ha representado en el mapa de la figura 5 la región elegida con un recuadro rojo.



Figura 5. Mapa del desierto de Tabernas (11).

Esta región se caracteriza por un clima árido, la vegetación de la zona es escasa, las precipitaciones son escasas durante el año, la nubosidad es baja, sobre todo en verano (12), y dispone de un número elevado de horas de sol al año, 3000 horas (10), por lo que es una localización tipo muy adecuada para la instalación de paneles fotovoltaicos. Sin entrar a fondo en cuestiones legislativas, el desierto de tabernas es una Zona de Especial Protección para las Aves (ZEPA), debido a su rica avifauna, por lo que tiene consideración de espacio protegido y una instalación de este tipo podría afectar a su fauna, por lo que probablemente no sea un proyecto viable.

Se ha seleccionado una zona de 50 km² rectangular dentro de los límites de la región de Tabernas, la razón de haber elegido este tamaño es debido a que se corresponde a las extensiones máximas de plantas fotovoltaicas en la actualidad (13), de esta forma se podrán observar los efectos de la climatología en las plantas fotovoltaicas del futuro. Seleccionando el área de forma rectangular hemos obtenido unos valores límites de coordenadas y con el fin de obtener unos sensores repartidos de forma no uniforme por esta superficie, los puntos se han generado aleatoriamente mediante Matlab, se han calculado los valores al azar en un plano 2D, de forma que se evite que los puntos se concentren en zonas pequeñas y que coincidan dos puntos, lo que generaría una duplicidad que no tendría mucho valor a la hora de predecir mediante kriging.

Las coordenadas 2D las pasamos a valores de latitud y longitud mediante una conversión lineal, de esta forma se tienen puntos que podemos localizar en el mapa y aseguramos de que se encuentren dentro de la zona deseada.

Por otro lado, se ha calculado mediante la ecuación de Haversine la distancia real entre los puntos y se ha creado una matriz con las distancias entre cada pareja de puntos seleccionada.

Se han guardado las coordenadas de estos puntos en un Excel para tenerlos disponibles y no perderlos en posteriores ejecuciones, además se ha almacenado la variable en Matlab. Google Maps dispone de una herramienta para seleccionar puntos en el mapa leídos a través de archivos con formato “xlsx” y se han representado en el mapa los puntos obtenidos.

En la figura 6 se han representado los puntos en un mapa físico de la zona para ver la disposición de nuestra red de sensores simulada, los puntos rojos son los extremos que limitan la zona seleccionada.



Figura 6. Ubicación de los nodos de la red de sensores simulada.

4.2 Origen de los datos

Una vez elegidos los puntos, el siguiente paso es buscar una base de datos disponible que se ajuste a las necesidades de nuestro problema. No hay datos disponibles para puntos específicos de la zona que se ha elegido, ya sean mediante estaciones meteorológicas o sensores colocados, por lo que se hace necesario usar datos obtenidos mediante imágenes de satélite y métodos, como el método Heliosat-2, que permite convertir observaciones de satélites en estimaciones de la radiación solar en el suelo (14).

Se ha optado por usar el [CAMS Radiation Service](#), que tiene disponible en el portal SoDa, este servicio se encuadra dentro del programa Copernicus, el programa de observación de la Tierra de la Unión Europea con el objetivo de tener datos sobre el estado de nuestro planeta y el medioambiente.

Otros servicios que se han encontrado y que podrían ser útiles son, el [PVGIS](#), servicio de la Unión Europea para obtener información sobre radiación y herramientas de comprobación de eficacia de paneles fotovoltaicos o el [CM SAF](#), la agencia de monitorización climática de la unión europea, dispone de muchos productos de datos de variables climatológicas, incluyendo series históricas de datos y a tiempo real. No se han utilizado porque son más complejos de descargar los datos y su posterior utilización, pero proporcionan alternativas aceptables, en caso de que no estuviera disponible el servicio utilizado.

Toda la información generada dentro del servicio CAMS Radiation es libre y gratuita para todos los usuarios, si se especifica al usar los datos lo indicado en el acuerdo de licencia (15).

La guía de usuario del servicio nos repasa los métodos utilizados con anterioridad para el cálculo de la radiación solar. Este servicio utilizó previamente el método Heliosat-2 con el que se construyeron las bases de datos HelioClim-3 y SOLEMI que proporcionan datos de radiación solar, usando distintas implementaciones para el cálculo de la radiación solar, es decir, las dos bases de datos se basan en modelos de cálculo de la radiación solar distintos. También se avisa de que el método Heliosat-2 tiene limitaciones, como que no tiene en cuenta la aparición de nieve repentina. (16).

En una versión posterior, actualizaron el método al Heliosat-4, que es el usado actualmente en el servicio CAMS. Se basa en variables como las propiedades de aerosoles, vapor de agua total en columna o el ozono. Es capaz de

estimar la radiación a nivel de suelo en todas las condiciones meteorológicas del cielo, siendo posible calcular otras variables como los componentes directo y difuso o la radiación en plano horizontal (16). Está disponible el informe de validación por trimestre sobre las estimaciones, comparando con datos de estaciones a nivel del suelo, y podemos leer las limitaciones existentes para validar los datos. Se ha leído parte de los informes y se concluye que existe un error en las medidas, tanto por sobreestimación como por subestimación y muestran distintos valores para indicar el grado de error cometido por estación, como el RMSE, por lo que concluimos que los datos no son del todo exactos.

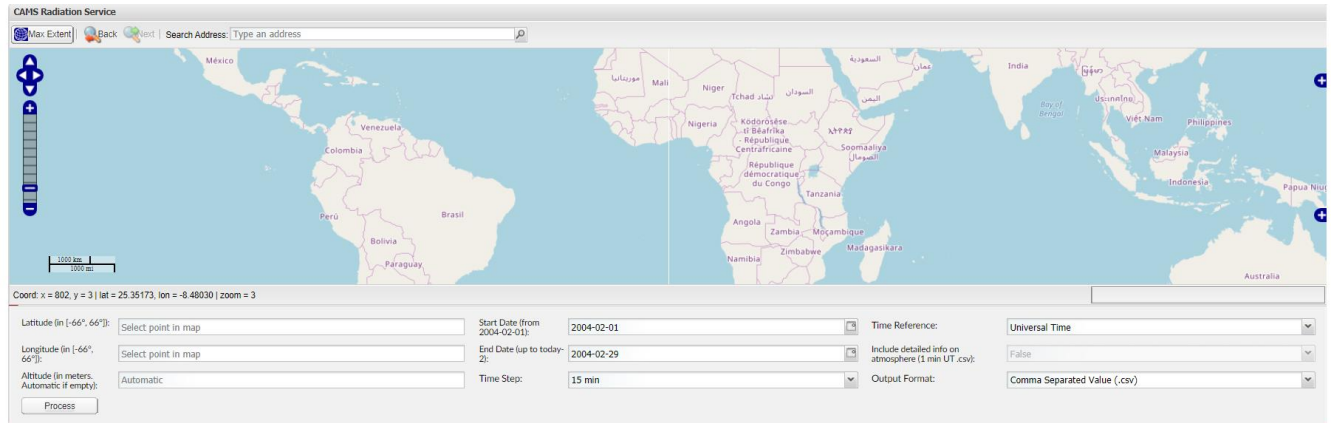


Figura 7. Interfaz de usuario del CAMS Radiation Service.

Hay datos disponibles desde febrero de 2004 hasta datos de hace dos días desde la fecha en que lo miremos, con resolución temporal de 1 min, 15 min, 1 hora y 1 día. Se ha descargado un periodo de cuatro años, por lo que se ha optado por resolución de 15 min, ya que el tamaño de los archivos generados provocaría problemas de espacio y las lecturas de datos serían muchos más lentos.

Si queremos un determinado punto, el servicio interpola para obtener el valor de la variable en el punto que le indiquemos, lo que equivaldría a disponer en la realidad de un sensor en ese punto.

Para descargar los datos, hay que ir al portal e introducir las coordenadas del punto deseado e introducir el rango de tiempo a calcular, y seleccionar la resolución temporal y el formato del documento, que puede ser “.nc” o “.cv”, se han descargado en formato “.cv” ya que el programa Excel puede manejarlos sin mucha dificultad.

Una vez descargado el fichero que se genera, si lo abrimos con Excel (puede ser necesario cambiar algunas opciones de la configuración del programa para leer los datos correctamente) podemos ver que tenemos en intervalos de 15 min los datos de las siguientes variables:

- TOA: Radiación en plano horizontal por encima de la atmósfera. [Wh/m^2].
- GHI: Radiación Global en plano Horizontal a nivel del suelo [Wh/m^2].
- BHI: Beam irradiance o Radiación directa en plano Horizontal a nivel del suelo [Wh/m^2].
- DHI: Radiación difusa en plano Horizontal a nivel del suelo [Wh/m^2].
- BNI: Radiación directa en plano móvil perpendicular a la incidencia solar. [Wh/m^2].

También están estas variables con cielo despejado (clear sky), que son los datos que se tendrían si no hubiera nubosidad sobre nuestro punto.

Reliability: Última columna, proporción fiable de datos (0-1).

Mostramos una captura de los datos descargados mediante el programa Excel:

# Observation period	TOA	Clear sky GHI	Clear sky BHI	Clear sky DHI	Clear sky BNI	GHI	BHI	DHI	BNI	Reliability
2014-06-21T00:00:00.0/2014-06-21T00:15:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T00:15:00.0/2014-06-21T00:30:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T00:30:00.0/2014-06-21T00:45:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T00:45:00.0/2014-06-21T01:00:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T01:00:00.0/2014-06-21T01:15:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T01:15:00.0/2014-06-21T01:30:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T01:30:00.0/2014-06-21T01:45:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T01:45:00.0/2014-06-21T02:00:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T02:00:00.0/2014-06-21T02:15:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T02:15:00.0/2014-06-21T02:30:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T02:30:00.0/2014-06-21T02:45:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T02:45:00.0/2014-06-21T03:00:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T03:00:00.0/2014-06-21T03:15:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T03:15:00.0/2014-06-21T03:30:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T03:30:00.0/2014-06-21T03:45:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T03:45:00.0/2014-06-21T04:00:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T04:00:00.0/2014-06-21T04:15:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T04:15:00.0/2014-06-21T04:30:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T04:30:00.0/2014-06-21T04:45:00.0	0	0	0	0	0	0	0	0	0	1
2014-06-21T04:45:00.0/2014-06-21T05:00:00.0	0.7203	0.1487	0.002	0.1467	0	0.1333	0	0.1333	0	0.7333
2014-06-21T05:00:00.0/2014-06-21T05:15:00.0	12.2619	3.3899	0.459	2.931	10.036	3.0089	0	3.0089	0.0006	0.8
2014-06-21T05:15:00.0/2014-06-21T05:30:00.0	27.565	9.8431	2.7232	7.1199	31.5192	7.7102	0.0002	7.71	0.0018	1
2014-06-21T05:30:00.0/2014-06-21T05:45:00.0	43.0883	18.3751	7.1135	11.2615	53.7561	10.9084	0.0531	10.8554	0.3773	1
2014-06-21T05:45:00.0/2014-06-21T06:00:00.0	58.7654	28.422	13.3576	15.0644	74.4783	17.1071	0.5937	16.5134	3.1942	1
2014-06-21T06:00:00.0/2014-06-21T06:15:00.0	74.5291	39.6098	21.1792	18.4306	93.3375	20.552	0.8998	19.6522	4.014	0.2
2014-06-21T06:15:00.0/2014-06-21T06:30:00.0	90.3121	51.6288	30.2833	21.3455	110.2836	23.0121	2.1716	20.8405	7.6182	0.8

Figura 8. Visualización de los datos descargados.

“Generated using Copernicus Atmosphere Monitoring Service Information 2019”

En las horas de noche, se observa que la radiación que tenemos es cero. También podemos ver el efecto de las nubes en las horas de día, en las que tendremos una caída del valor de la radiación global en la zona afectada.

Se trabaja con la GHI, o radiación global en plano horizontal a nivel del suelo, corresponde a la suma de las radiaciones que llegan por radiación difusa y directa ya que los paneles más usados absorben el total de la radiación que llega, aunque existe una excepción, los sistemas de energía solares de concentración o CPV, que usan paneles curvos que concentran la energía solar que llega a las células fotovoltaicas que utilizan principalmente la radiación directa, es una tecnología en desarrollo, en el futuro podrían sobrepasar la eficiencia de la tecnología actual.

Se han descargado 49 datos para predecir los valores de radiación solar en toda nuestra zona elegida, mediante kriging, para ello es necesario estudiar la variabilidad espacial entre los puntos de forma previa y tenemos 10 datos de validación, que usaremos para ver la diferencia entre la estimación y el valor “real” de la variable.

4.3 Análisis exploratorio de datos para cuatro periodos de tiempo

Los datos se han leído mediante Matlab, y se han creado tablas con datos para que la lectura sea sencilla mediante R. Se ha obtenido la media de todos los valores mayores que cero, dentro de los siguientes periodos:

- Cada mes de 2015
- 3 días de junio.
- Una hora de junio.
- Media de los valores de un mismo periodo de 15 min de todos los días de junio.

El objetivo de obtener datos de periodos distintos es comparar las diferencias entre los modelos de los variogramas generados y las discrepancias que se producen en las predicciones.

Se ha generado una nube de puntos, mediante la comparación del valor de la radiación de una pareja de puntos en cada instante, y se ha ajustado la recta de mayor proximidad a los puntos:

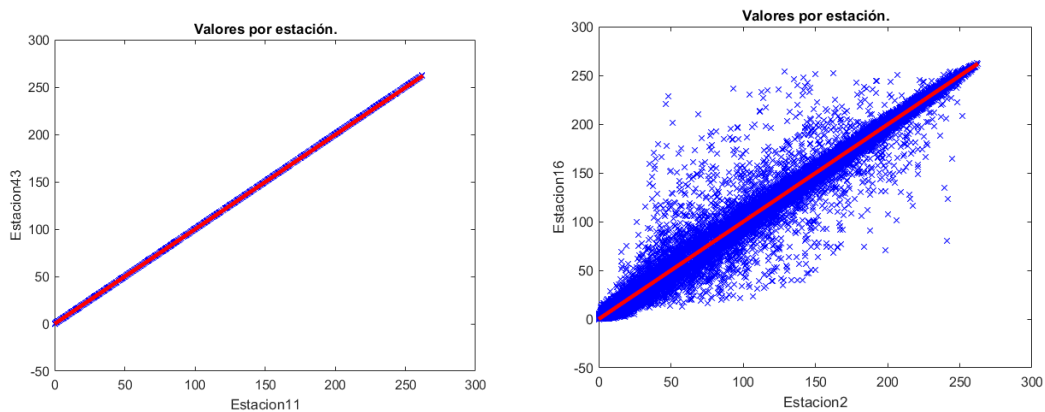


Figura 9. Nube de puntos.

Los puntos van en azul y en rojo se representa la recta con mejor ajuste a todos los puntos.

Los puntos 3 y 11 están muy cerca, mientras que el 16 y el 2 están más alejados, resulta obvio que cuanto mayor sea la distancia entre los nodos, mayor será la dispersión de los valores en un mismo instante.

El análisis de los datos es indispensable antes de trabajar con ellos con el fin de comprender los patrones que siguen y ayudar a analizarlos mediante la obtención de sus características.

Se dispone de librerías para realizar el análisis de forma sencilla mediante R, por lo que se van a utilizar.

Se guardan los ficheros .txt, generados mediante Matlab con los datos en forma de tabla, con los datos en tres columnas separados mediante comas. Es mejor colocarlos dentro del directorio, no es necesario, pero facilita la ruta a los ficheros, que se leen mediante la función `read.table` de R, de la siguiente forma:

```
datos <- read.table("Radiacion_media_mes.txt", header=T, sep = ",")
```

Recordamos que en el anexo se puede leer el código completo.

Si usamos la función `summary()`, se puede obtener información sobre los datos como la mediana, la media o los valores de los cuartiles.

Por ejemplo, con `summary(datos$junio)`

```
Min.    1st Qu.  Median    Mean    3rd Qu.    Max.
131.4  132.9     133.5    133.5    134.3     135.5
```

Mediante la función `Desc()`, obtenemos una descripción más extensa de la variable, con valores para la desviación estándar, asimetría entre otros, además realiza una representación gráfica de los datos, que mostramos a continuación:

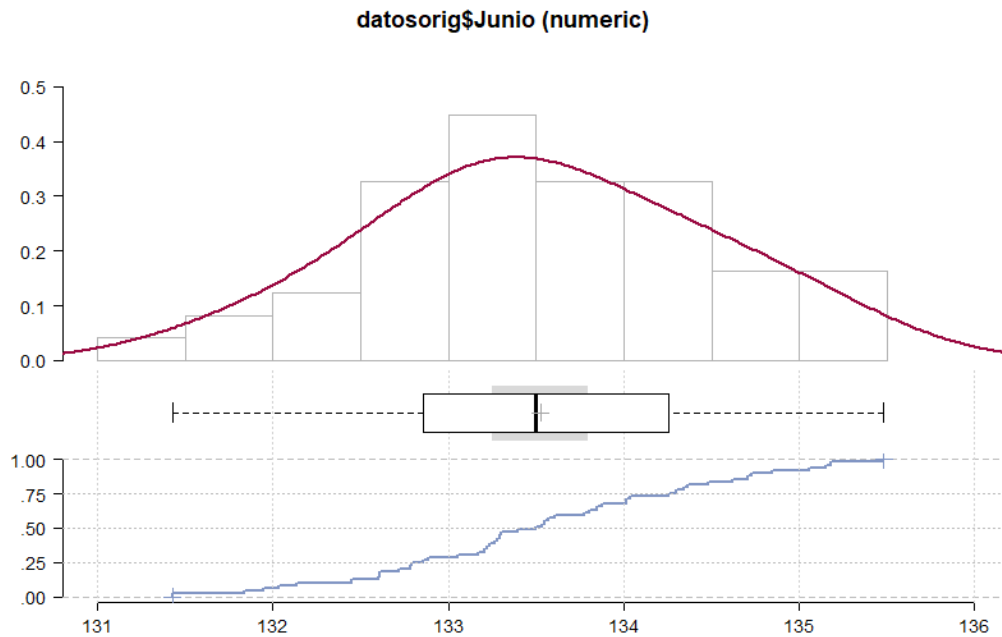


Figura 10. Gráfica generada mediante Desc del mes de junio.

El método de kriging genera las mejores estimaciones posibles si nuestros datos siguen una distribución normal, aunque no es un requisito fundamental, nos interesa obtener el menor error posible en las estimaciones, por lo que cuanto más cerca nos encontremos de una distribución normal, mejores resultados.

En (17), se realiza un estudio con los valores normalizados y no normalizar para aplicar kriging ordinario y no se encuentran diferencias substanciales.

En el caso del mes de junio (que tenemos en la figura 10), los datos presentan una distribución normal, pero otros meses no siguen esta distribución. Para comprobarlo, aplicamos el test de Shapiro-Wilk, si el resultado de p es menor que 0.05, la distribución de los datos no es normal, si es mayor es posible que sea normal. Estos tipos de tests se ven afectados por el tamaño de la muestra, por lo que se suelen complementar con un gráfico q-q, que es una comprobación subjetiva, de manera gráfica, de la distribución que siguen los datos, si los puntos en la gráfica siguen una distribución normal, deberían formar una línea recta diagonal (18).

```
Shapiro-Wilk normality test.
data:  datosorig$Junio
W = 0.98903, p-value = 0.9257
```

A continuación, representamos el gráfico q-q para el mes de junio:

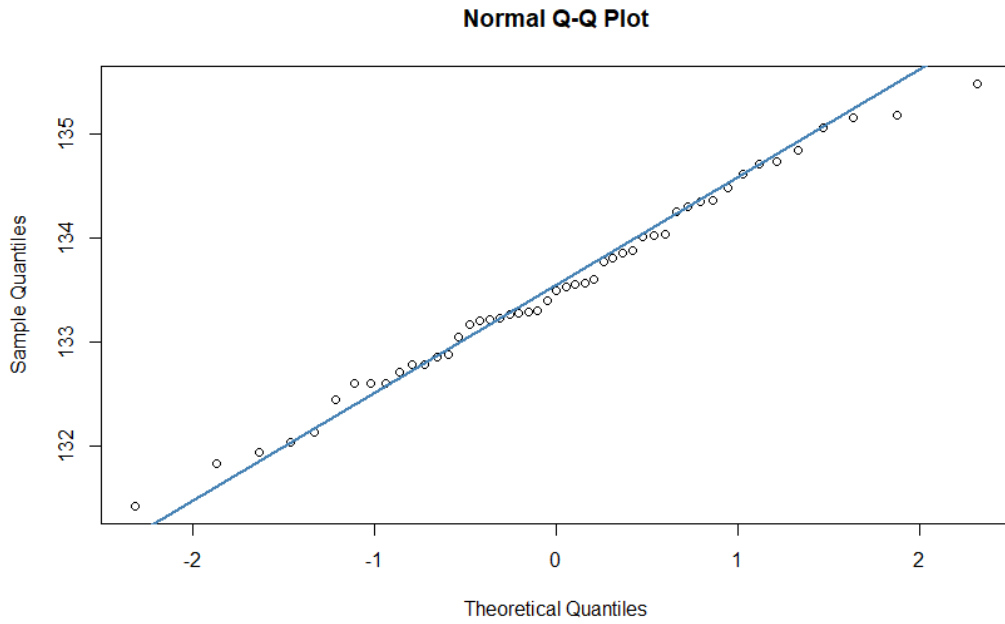


Figura 11. Gráfica q-q para el mes de junio.

En el caso de que no fuera normal, se realiza una normalización de los datos mediante métodos como aplicar el logaritmo, raíces cuadradas o un escalado a los datos. También se ha utilizado una librería, `bestNormalize` (19), que permite aplicar una función con el mismo nombre, cuyo objetivo es realizar posibles transformaciones de normalización con los datos. Incluso utilizando este recurso, algunos datos no han pasado el test de normalidad y no se aprecian diferencias significativas en las gráficas q-q, por lo que hemos decidido trabajar con datos que no se ajustan totalmente a la distribución normal, vemos un ejemplo con los datos de los días de junio:

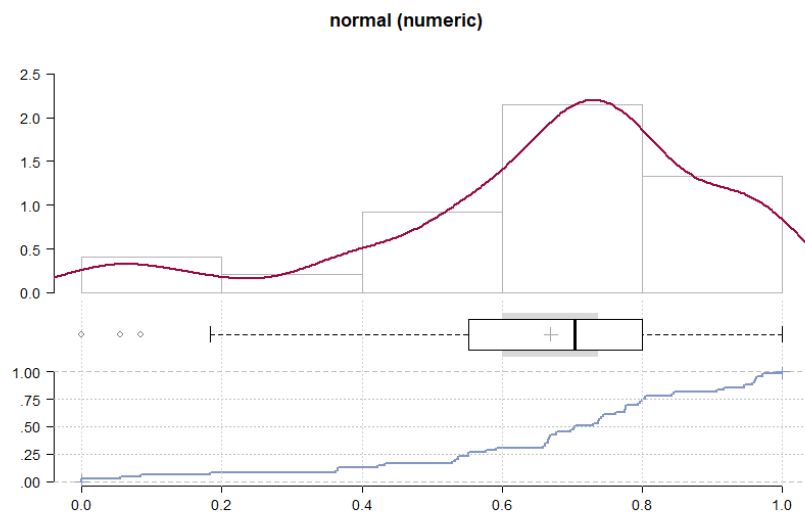


Figura 12. Distribución de los valores de los datos de los días de junio

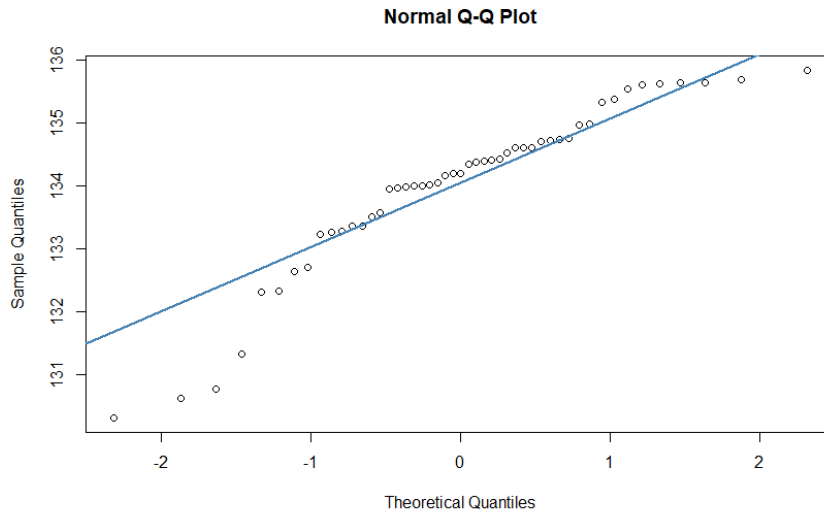


Figura 13. Gráfica q-q para los datos de días de junio

Se puede observar que una parte de los datos de la izquierda no se encuentran sobre la recta, si observamos el histograma, vemos que existe una asimetría negativa, esto quiere decir que la distribución de los datos se alarga en el lado izquierdo, la causa de esto es posiblemente debido a factores meteorológicos como que hubiera estado más nublado por la zona de esos puntos durante el tiempo en el que se han tomado los datos. Por otro lado, el resto de los puntos se ajustan de manera adecuada a la recta, por lo que no tendremos muchos problemas con el estimador.

La distancia entre los puntos se ha vuelto a calcular mediante dos fórmulas, Vincenty y Haversine, para comprobar la variación entre las distancias en los puntos, que resultan ser pequeñas.

Tabla 1. Distancia entre los puntos según la fórmula.

Fórmula	Distancia máxima [m]	Distancia media [m]
Haversine	8780	3426
Vincenty	8772	3424

A continuación, representamos un histograma de los datos, en rojo marcamos la media de los datos.

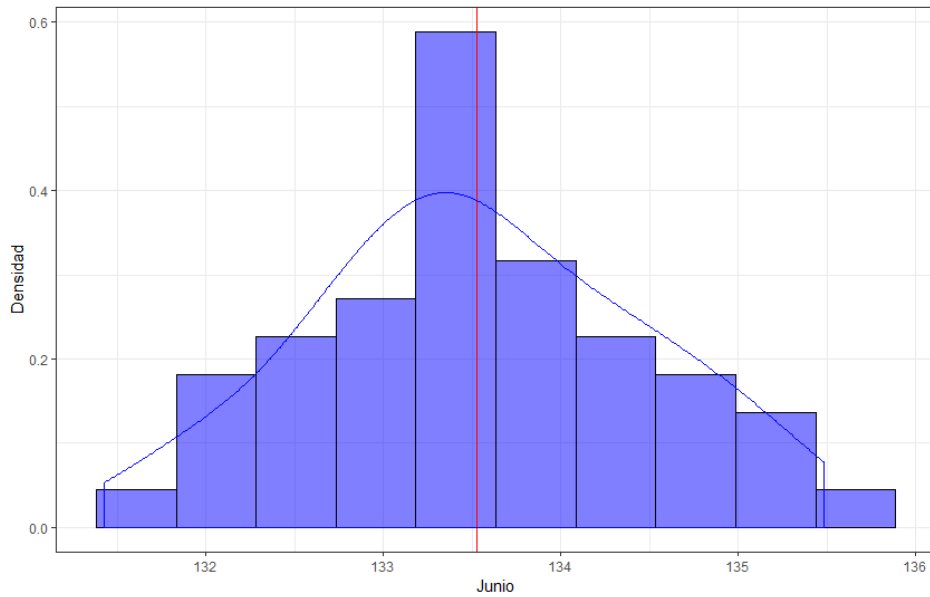


Figura 14. Histograma del mes de junio

Para visualizar gráficamente los datos, realizamos una representación espacial en forma de mapa, con un gradiente de colores para el valor de la radiación.

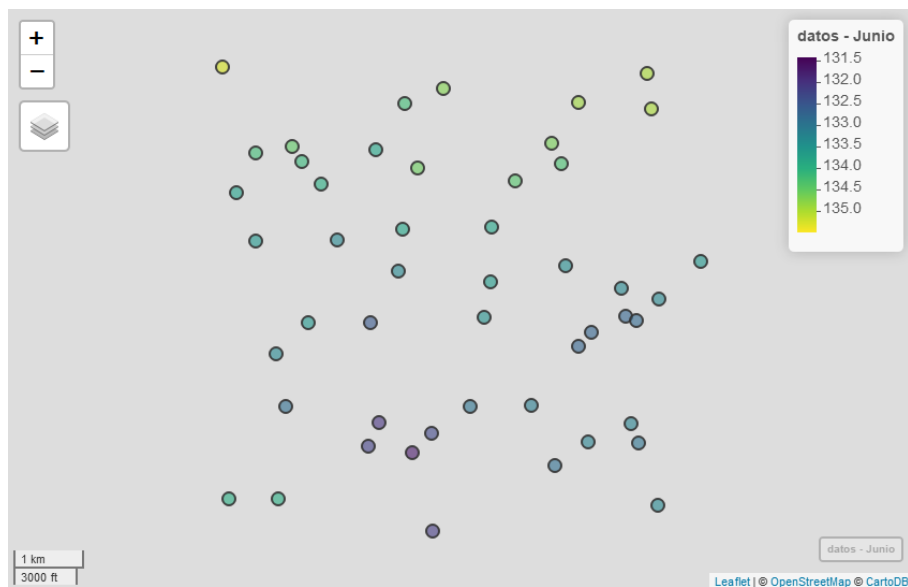


Figura 15. Mapa con los valores de radiación de los nodos.

4.4 Reproyectar

La reproyección consiste en cambiar los valores de coordenadas del conjunto de los datos de un sistema de coordenadas a otro.

En el momento en el que se calcularon las posiciones de los puntos en el espacio, se asignaron valores de coordenadas según el sistema de coordenadas geográficas de latitud y longitud, pero para calcular las distancias de forma correcta con las funciones que vamos a utilizar para calcular el variograma, hay que reproyectar las coordenadas a otro sistema de coordenadas de referencia (CRS) que permita calcular distancias de forma

sencilla.

Se utiliza el sistema de coordenadas universal transversal de Mercator (UTM por sus siglas en inglés), que se expresa únicamente en metros al nivel del mar. En el sistema UTM, se separa las latitudes de la tierra en 60 zonas, con 6° grados de anchura cada zona, también llamado huso y se proyecta cada zona mediante la proyección Mercator con una baja distorsión.

Los valores de los ejes X e Y tienen un centro de coordenadas en cada zona, que se identifica con un número, en España pasan las siguientes zonas por el territorio:



Figura 16. Zonas UTM de España (20).

A nuestra localización, en Almería, le corresponde la zona 30 del hemisferio norte.

Los sistemas de proyección se pueden identificar mediante un código EPSG, en este caso el 32630. Se puede comprobar en (21), usamos este código para transformar de forma sencilla.

Para re proyectar los nodos se instala la librería “rgdal”, con funciones que facilitan las operaciones a realizar, que mostramos a continuación:

```
coordinates(datos) <- ~lon+lat # necesita libreria sp
#proj4string(datos) = "+proj=longlat +datum=WGS84"
proj4string(datos) <- CRS("+init=epsg:4326") # sistema de coordenadas lat+long
#+proj=utm +zone=30 +ellps=WGS84 +datum=WGS84 +units=m +no_defs      epsg:32630
UTM zone 30N
datos=spTransform(datos, CRS("+init=epsg:32630"))
```

De esta manera se obtiene el objeto espacial datos con las coordenadas pasadas al sistema UTM, en la zona 30N, cuyas distancias entre nodos se pueden calcular de forma sencilla con nuestras funciones.

5 VARIABILIDAD ESPACIAL Y ESTIMACIÓN DE LA RADIACIÓN SOLAR

En este capítulo se realiza el análisis estructural de los datos en los cuatro periodos de tiempo elegidos, se observan las variaciones de los datos en las distintas direcciones del espacio y se representan los modelos teóricos de variograma que mejor se ajustan a nuestros datos en los distintos periodos, también se muestran los variogramas de los distintos meses del año.

Finalmente, se calculan las predicciones de la radiación solar en nuestra zona, siendo representadas de forma gráfica y se compara los resultados de la predicción obtenidos mediante los distintos modelos generados por los datos de los distintos periodos.

5.1 Variabilidad espacial de la radiación solar

El paquete gstat (22) contiene la mayor parte de las funciones de geoestadística utilizadas, un ejemplo sobre la puesta en práctica de estas se puede consultar en este documento (23).

Como trabajo previo a la predicción de la radiación solar se ha llevado a cabo el análisis estructural de los datos. En dicho análisis se estudia la variabilidad espacial de la radiación solar en los cuatro periodos de tiempo en los que estamos trabajando.

El único modelo que se ha normalizado ha sido el de la media de los instantes del mes de junio y se han comparado los modelos de variograma normalizados y sin normalizar.

El variograma se calcula mediante la función variogram, incluida en el paquete gstat, antes de poder usar esta función hay que crear un objeto estadístico con la función gstat, cuando ejecutamos variogram obtenemos una variable que podemos dibujar con el comando plot.

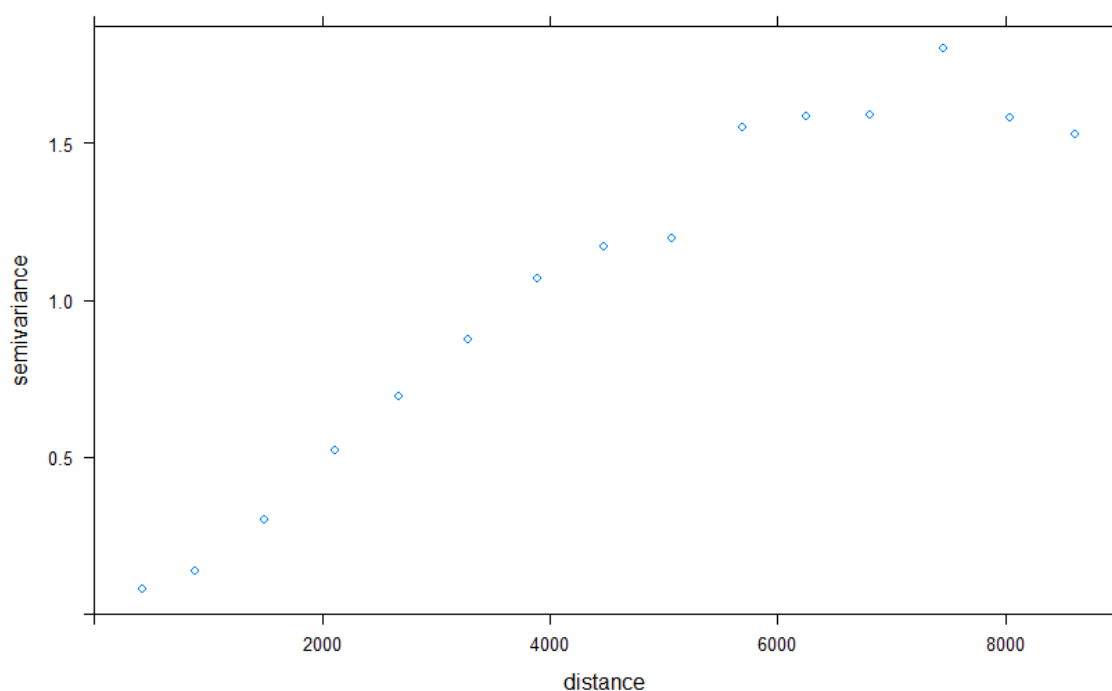


Figura 17. Variograma experimental con la muestra de datos del mes de junio.

Se recuerda que el variograma experimental se calcula con todas las parejas de puntos o nodos que se encuentren a esa distancia entre ellas, por lo que cuanto más nos acercamos a valores de distancia extremos, el número de parejas de puntos a esas distancias es menor, ya que los puntos se han generado de forma aleatoria dentro de una región limitada, por lo que los valores mayores y más bajos de distancia en el variograma tienen una muestra menor de puntos, entonces habrá mayor desviación debido al número limitado de puntos. La función por defecto limita hasta la mayor distancia existente entre los puntos dividida por tres, pero en este trabajo se ha utilizado hasta la máxima distancia de los puntos.

La figura 17, es el variograma experimental, se pueden cambiar parámetros como cutoff, que indica el punto de corte, o alpha que modifica el ángulo del variograma. Con la función `fit.variogram` es posible obtener un variograma ajustado por esta función, pero vamos a ajustarlo experimentalmente.

La dirección (si están hacia el este o hacia el norte de nuestro dato) en la que se mueve en el espacio puede afectar al valor del variograma, para comprobarlo se puede hacer lo siguiente, poner en la función `variogram`, `map=TRUE`, generamos un mapa que muestra los valores de la varianza según nos movemos en el espacio.

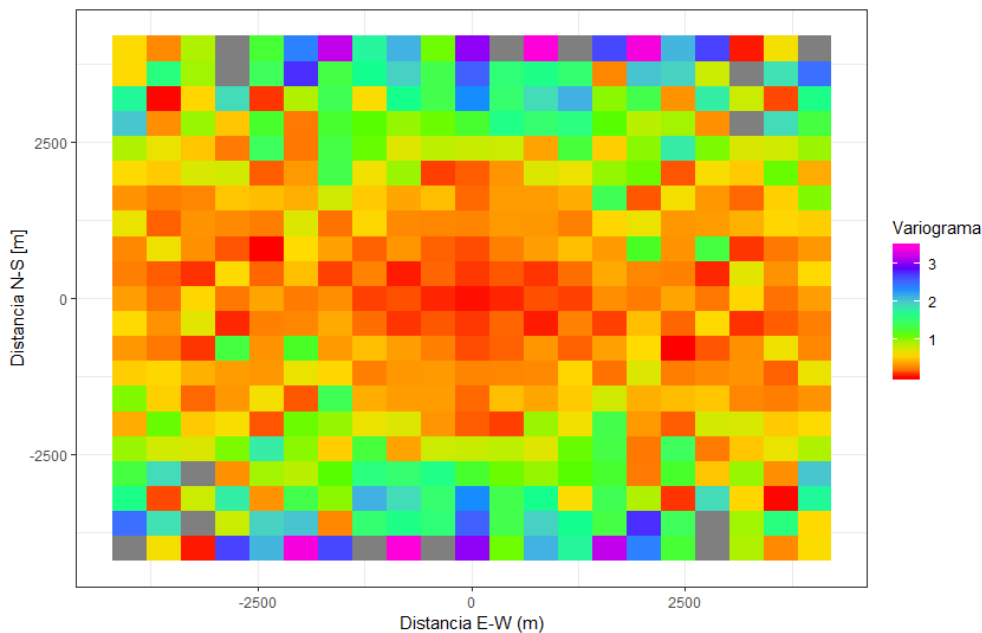


Figura 18. Direcciones para el variograma del mes de junio.

En este caso, los valores más cercanos a cero tienen tonos más cercanos al rojo, en la dirección horizontal, hacia este u oeste, los valores varían menos que si vamos hacia el norte o el sur, aunque esto puede ser muy limitado debido al bajo número de puntos y para algunos ángulos puede que dispongamos de pocos datos.

Para usar el modelo, es necesario calcular experimentalmente (de forma manual) los valores de la pepita, la meseta y el rango, para ello hay que fijarse en la distribución de los puntos del variograma experimental e intentado fijarse un poco la forma que tendría el modelo de variograma que mejor se ajusta a nuestros datos, se asignan estimaciones para los parámetros y se prueba si se ajustan con `fit.variogram` introduciendo en la variable "model" de la función, los parámetros y el tipo de parámetro como condiciones iniciales para ajustar.

Veamos el resultado en la figura 19.

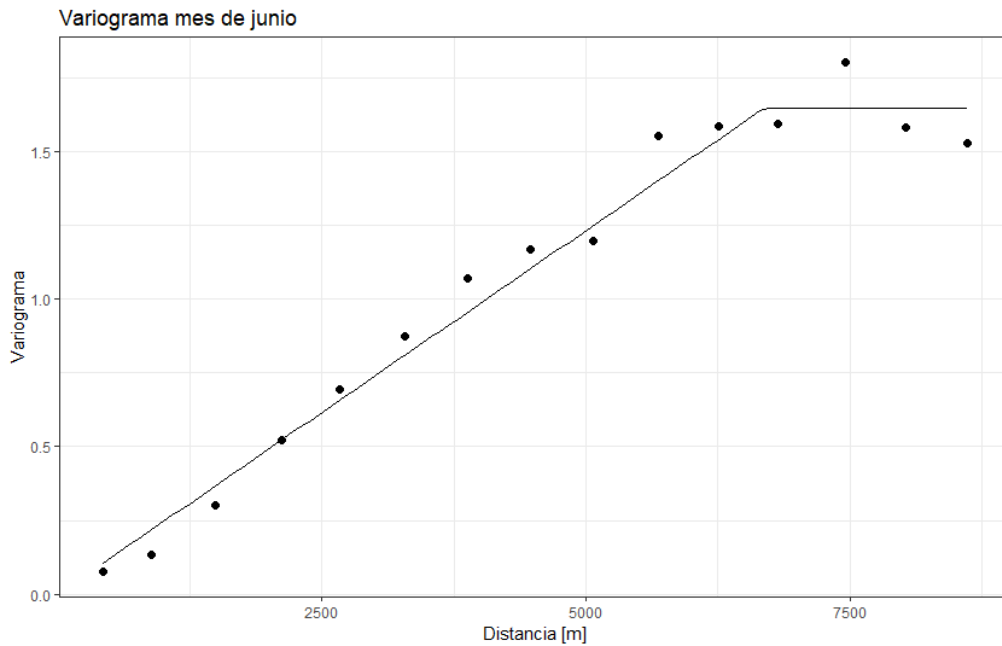


Figura 19. Variograma con modelo teórico del mes de junio.

Para representar el modelo se ha creado una función con la ecuación del modelo lineal, podemos ver que se ajusta muy bien a los puntos. Posiblemente sea posible usar un modelo esférico, pero el que mejor se ha ajustado ha sido el lineal. Si se ajusta el modelo de forma adecuada el modelo a los puntos, guardamos los parámetros del modelo en una variable para usarlos más tarde en el kriging.

Comprobemos como se ajusta este modelo en distintas direcciones del espacio.

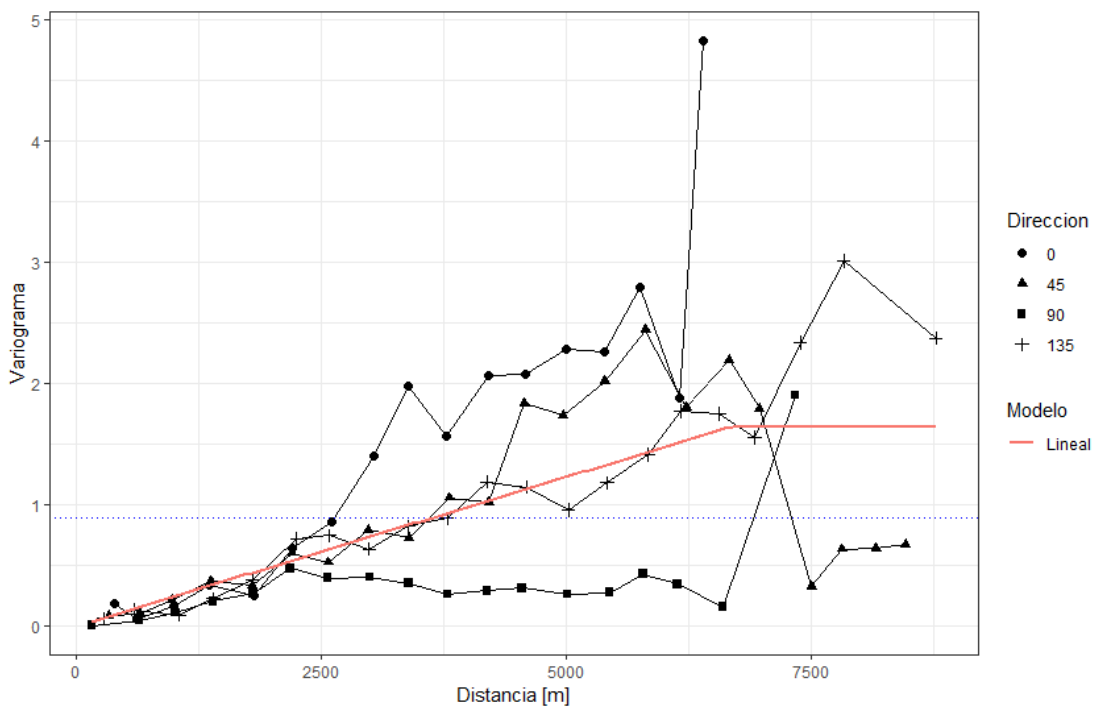


Figura 20. Variogramas direccionales junto al modelo del variograma del mes de junio.

Los ángulos siguen el sentido del reloj, un ángulo de 0° indica dirección vertical hacia arriba (norte), mientras que un ángulo de 90° señala hacia la dirección del eje x positiva (hacia el este en el mapa).

Vemos que el modelo se ajusta de forma adecuada en distancias cortas, que es donde disponemos más datos, conforme nos alejamos el ajuste es peor, sobre todo en distancias grandes donde deben existir pocos puntos que se encuentren en esas direcciones, por ejemplo, no hay puntos por encima de los 0° a distancias mayores de 6500

metros, por lo que el variograma en esa dirección se corta.

Si nos fijamos en el mapa, los valores del variograma en las distintas direcciones podemos ver la relación con esta imagen, por ejemplo, el ángulo de 90° tiene los valores más bajos, lo mismo ocurre en el mapa si avanzamos hacia la derecha desde el centro, donde vemos valores pequeños.

Se han realizado todos los variogramas con el modelo teórico de los siguientes periodos:

- a) Todos los meses del año 2015
- b) Los días del 14 al 17 de junio de 2015
- c) De 12:00 a 13:00 del 15 de junio de 2015
- d) La media de los valores de un mismo periodo de 15 minutos de las 9.45 a 10:00 de todos los días de junio.

Se va a referir a estos periodos de datos por las letras que lo identifican para disminuir la confusión generada.

Mostramos los variogramas de estos periodos junto al modelo teórico en las siguientes figuras (el variograma del mes de junio está en la figura 19).

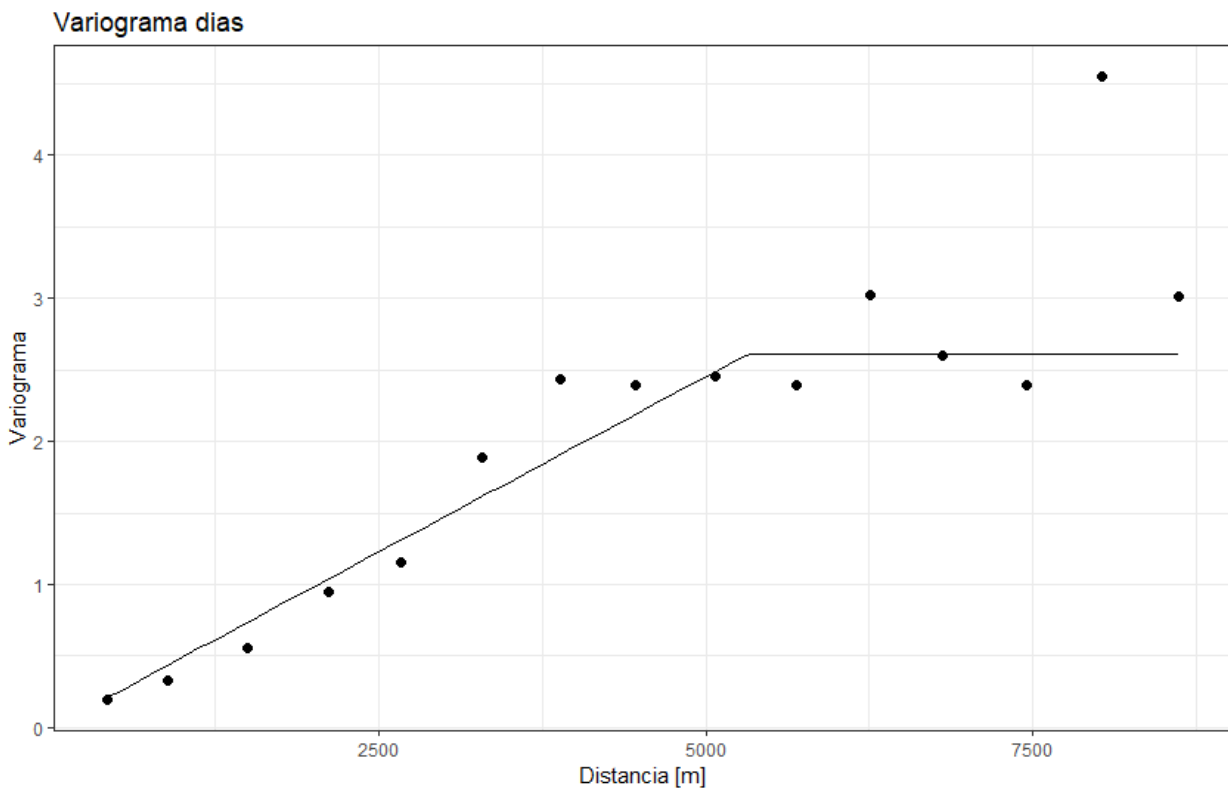


Figura 21. Variograma con el modelo de b)

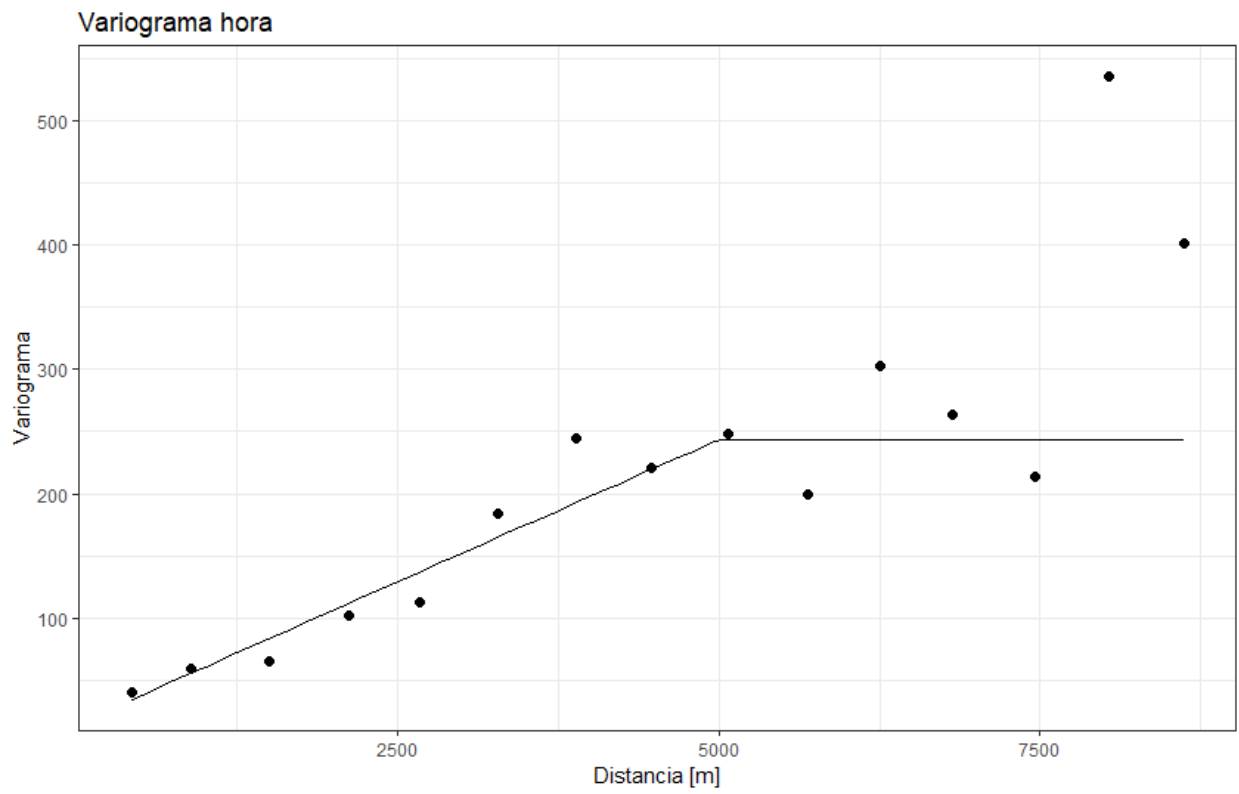


Figura 22. Variograma con el modelo de c)

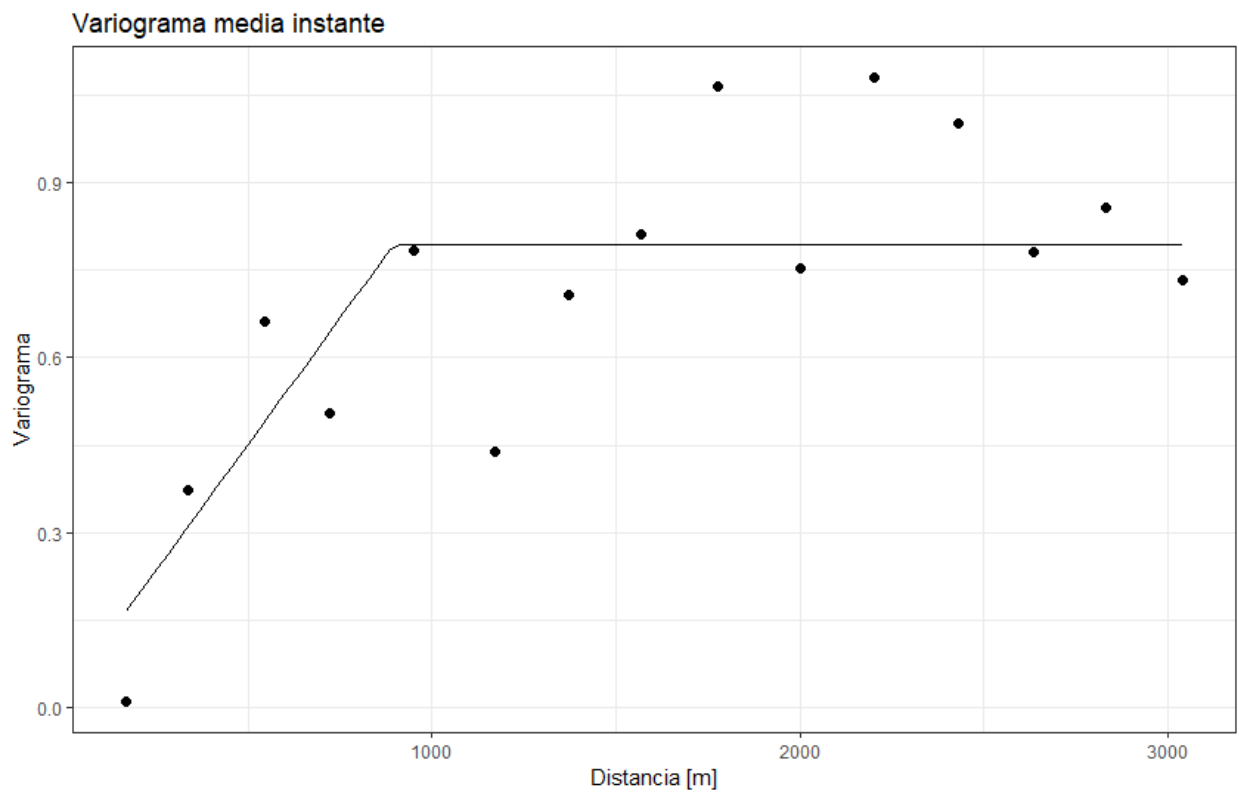


Figura 23. Variograma con el modelo de d)

Se han calculado los variogramas de todos los meses del año 2015, todos los meses se han ajustado a un modelo de variograma lineal y los mostramos en la figura 24.

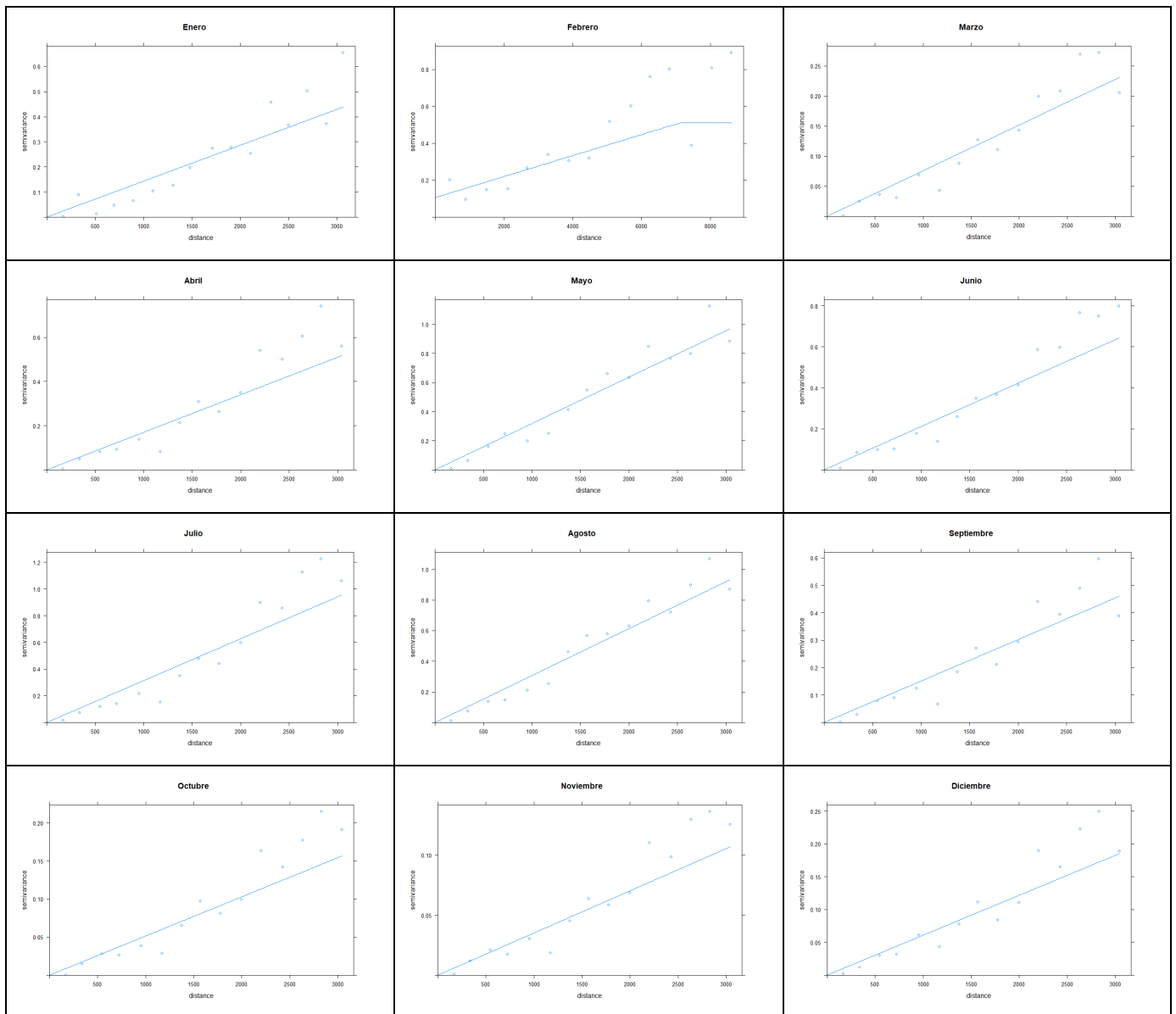


Figura 24. Variogramas de los distintos meses del año 2015.

Se puede observar que los variogramas tienen una forma parecida, aunque cuentan con variaciones en los valores del variograma obtenidos con los puntos experimentales por lo que el modelo no es exactamente igual en ningún mes.

5.2 Predicción de la radiación solar mediante Kriging ordinario

5.2.1 Ejemplo de Kriging ordinario en R.

Dentro de la función `krige` disponible en el paquete `gstat`, podemos cambiar diversos parámetros, se recomienda leer la ayuda para entenderlo mejor.

```
resultado=krige(mediahora~1,datosall,datosint,model=modelo_ajustado)
```

El primer parámetro es la variable que vamos a estimar, al poner `~1` indicamos que vamos a utilizar el kriging ordinario, `datosall` es una variable tipo `SpatialPointsDataFrame` que contiene las coordenadas transformadas a UTM y todos los datos.

`Datosint` son los lugares donde vamos a realizar las predicciones, se ha definido una cuadrícula dentro de nuestros límites espaciales, según el tamaño de la cuadrícula, se calculan más o menos estimaciones.

A la variable `Model` se le asignan los parámetros de nuestro modelo, calculado con anterioridad.

Esta función nos devuelve una matriz con la localización de los puntos en coordenadas UTM con las predicciones y la varianza para cada punto.

Es posible usar modelos de una distribución temporal distinta para estimar los datos de otra, por ejemplo, usar el modelo del mes de junio para estimar con los datos de los días de junio, ya que las relaciones espaciales de los datos no deberían cambiar de forma significativa, sobre todo si estamos dentro del mismo mes.

Vamos a calcular la predicción a partir de los datos tomados sobre una hora y con el modelo del mes de junio, se realiza la predicción y usamos `spplot` para mostrar los resultados, poniendo la gráfica de forma que se vean con colores iguales las zonas que tengan la radiación más alta y las líneas indican que los puntos sobre esa línea tienen el valor de radiación indicado en la línea.

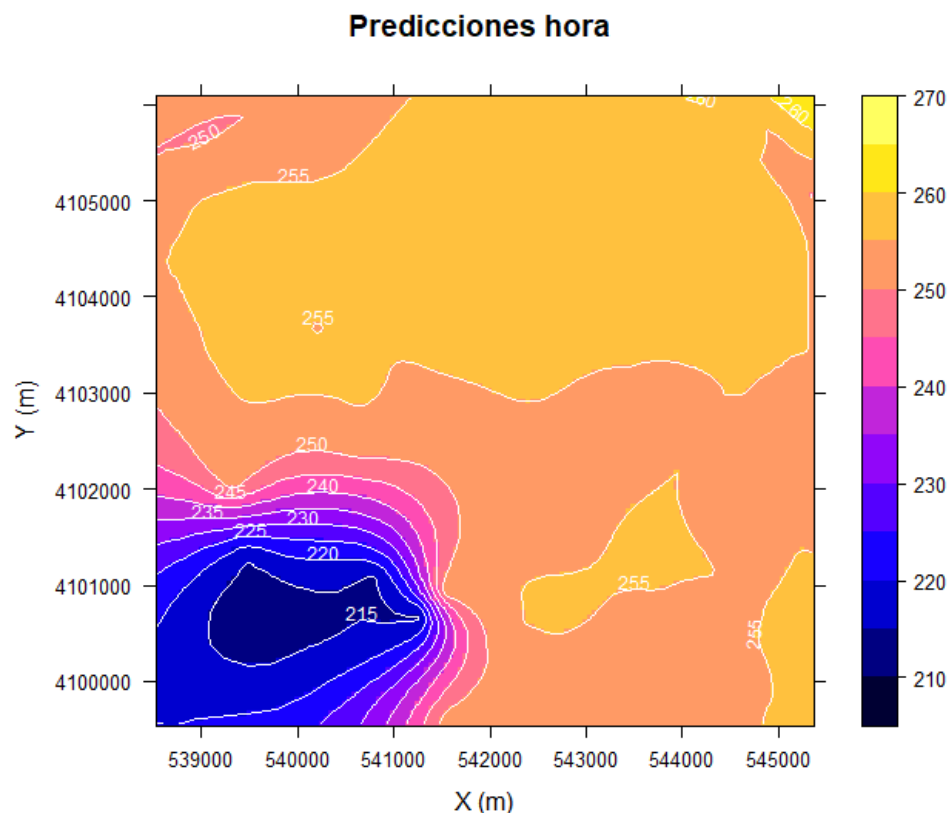


Figura 25. Predicción de las horas con el modelo de variograma del mes de junio.

Se observa una región con menos radiación en la parte inferior izquierda de la imagen, esto es debido a la presencia de nubes por esa zona, el resto del mapa presente una radiación uniforme sobre los 255 Wh/m^2 , según la predicción realizada con el modelo del mes de junio, en la figura 29, podemos ver las diferencias entre las predicciones según el modelo de variograma utilizado con kriging.

También es posible utilizar un parámetro obtenido de la función, una variable que nos indica la varianza de cada región calculada, si hacemos la raíz cuadrada para calcular la desviación típica, podemos mapear el error que pueden tener las predicciones de nuestro modelo de variograma.

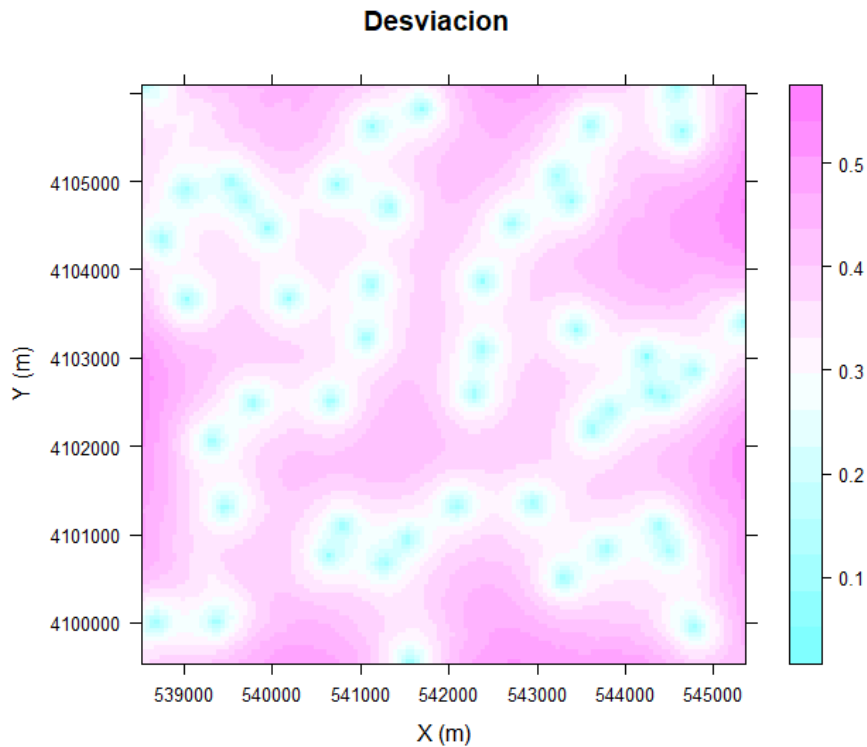


Figura 26. Desviación típica de la predicción de horas con el modelo de variograma de junio.

Las zonas azules corresponden a las zonas más cercanas a los puntos de los que disponemos el dato, por eso el error cometido debe ser menor, conforme nos vamos alejando de nuestros datos, aumenta el error que cometemos al estimar.

Para calcular el error cometido en las estimaciones de krige con nuestros propios datos, podemos utilizar la función `krige.cv`, que utiliza validación cruzada, de forma que dejando fuera del cálculo de krige un punto observado, predice el valor mediante krige ordinario en la localización del punto eliminado y lo compara con el observado, procediendo de esta forma con todos los puntos que tenemos como datos, y representamos el resultado:

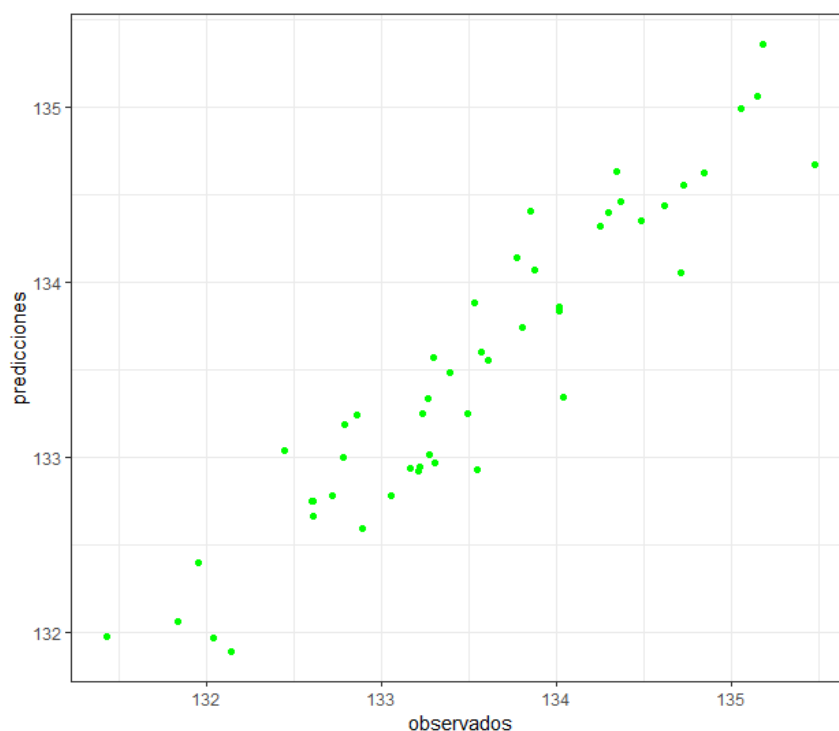


Figura 27. Resultado de la validación cruzada con el modelo de junio del variograma.

Los puntos verdes son los valores predichos frente al dato que tenemos, para comprobar la correlación se calcula el coeficiente R^2 .

La función nos devuelve una variable que contiene el residual que es la diferencia entre observado y estimación para cada dato observado de la muestra, y los valores estimados, procedemos a calcular las siguientes medidas:

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

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

Donde:

N= Número de datos.

\hat{Z} =Estimaciones.

Z= Valores observados.

Con los modelos que se han calculado para los periodos del mes de junio, los días, horas e instantes, es una buena idea utilizar la función `krige.cv` para estimar con los propios datos cada periodo (datos del mes y modelo del mes, datos de días y modelo de días, ...) y se procede a comparar los resultados de aplicar estas fórmulas.

Tabla 2. Resultados de la validación cruzada de los datos.

Periodo de datos	R ²	RMSE	MAE
Mes	0,884	0,318	0,255
Días	0,844	0,515	0,356
Hora	0,661	7,565	4,666
Media de instantes	0,349	0,758	0,577

Cuanto más próximo sea el valor de R² a 1, más exacto resultan las estimaciones generadas mediante el modelo y a menor RMSE o MAE menores son los errores cometidos.

Se observa que cuanto menor sea el periodo temporal en el que se realiza la media para obtener los datos (por ejemplo, la media en el tiempo del mes tiene más que la del día) para calcular el modelo del variograma, peor parece ser la predicción de nuestros datos, probablemente debido a que las nubes afectan a la radiación solar y cuantos menos datos se tengan en cuenta mayor sea el efecto perturbador de las nubes sobre el cálculo de los valores, que recordemos que se ha realizado la media de los valores mayores que cero dentro del periodo, por lo que si tenemos muchos datos las nubes que puedan haber afectado algunos días no tendrán mucho efecto sobre el total. Por esta razón, el valor de RMSE y MAE es mayor en la hora de junio, donde podemos encontrar nubes por una parte de nuestra zona como veremos mejor después.

La media de instantes de junio se ha calculado con datos normalizados por lo que el RMSE sale pequeño debido a que son valores normalizados, pero tiene el menor R² de todos.

Los cálculos de kriging se realizan con datos sin normalizar debido a que si se transforman los datos se pierde los valores reales de las variables y es más complicado realizar la transformación a los valores originales.

Se procede a comparar la diferencia entre usar datos normalizados y no normalizados en la predicción representando el mapa de la predicción en la figura 25.

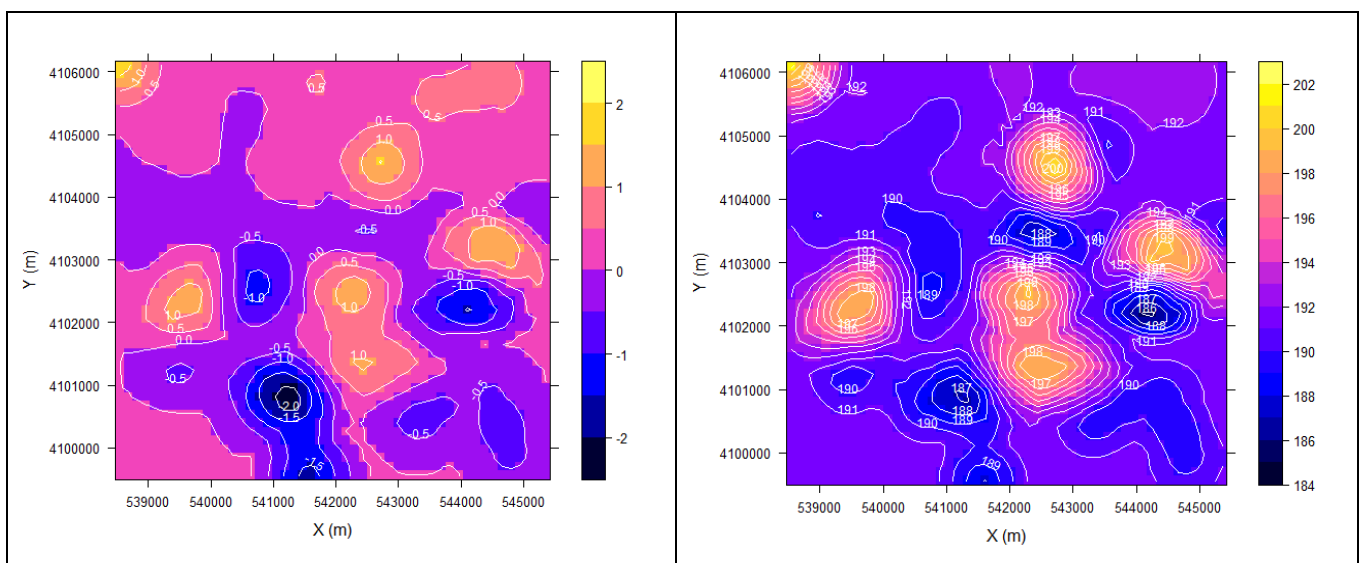


Figura 28. Comparación de las estimaciones con datos normalizados y sin normalizar.

Los datos normalizados están a la izquierda, aunque los resultados puedan ser mejores por ajuste del R^2 si calculamos el krige.cv (explicado a continuación), perdemos el significado físico de los datos ya que en vez de conocer la radiación conocemos un valor normalizado.

5.2.2 Comparación de predicciones con distintos modelos de variograma.

Primero vamos a representar las predicciones que realizan los distintos modelos, para comparar las diferencias existentes.

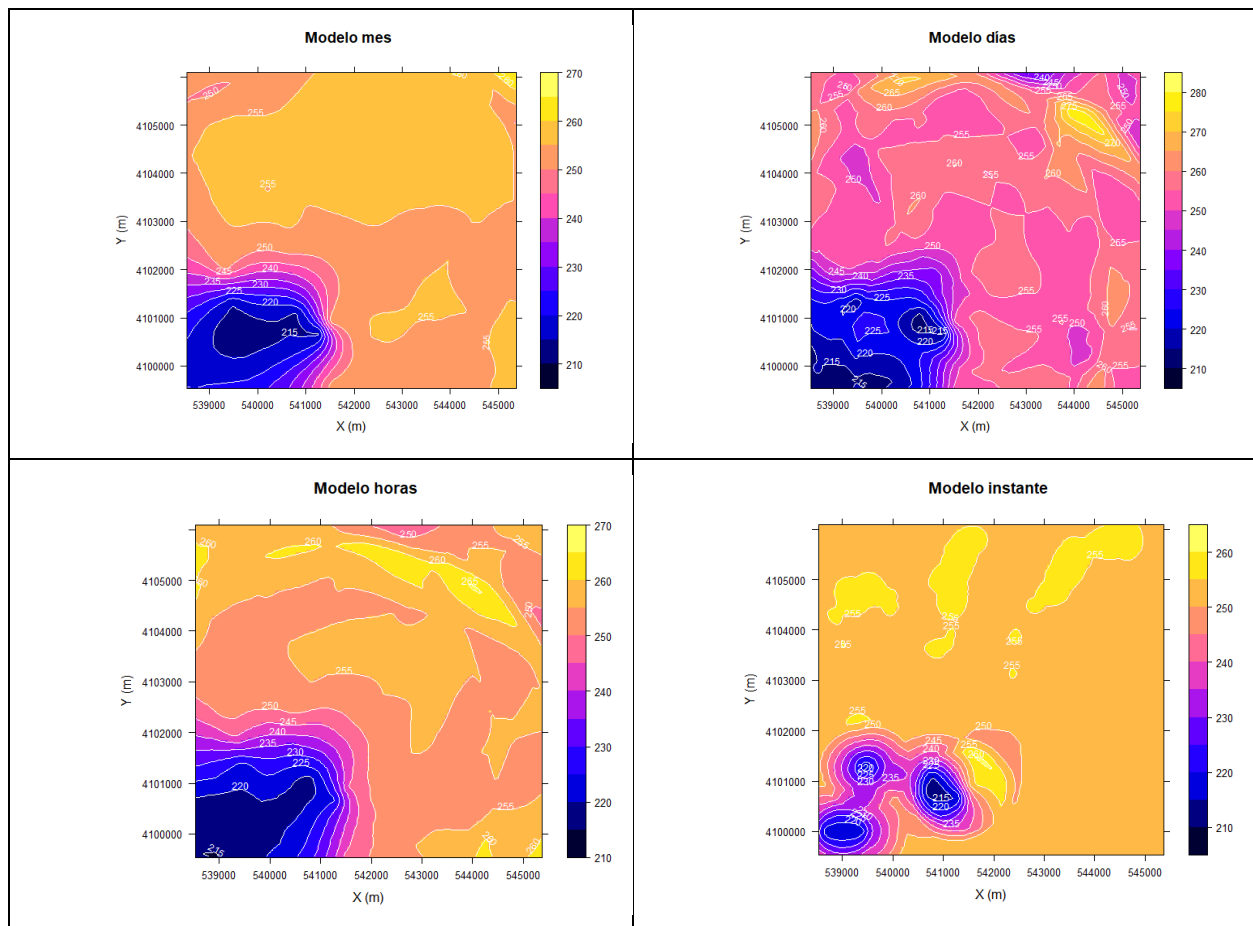


Figura 29. Comparación de las predicciones de la hora de los distintos modelos de variograma.

Se observan claras diferencias entre las predicciones realizadas por los distintos modelos de los variogramas, pero no se puede estar seguro sobre que predicción se ajusta mejor a los datos reales de radiación, para resolver esto, se comprueban las predicciones con valores reales, tenemos 10 coordenadas con datos que no se han usado para predecir, ahora los utilizamos para comprobar si las predicciones se ajustan a la realidad.

Se va a representar los puntos que vamos a añadir para comprobar los resultados de la predicción para poder localizarlos de forma gráfica en el mapa.

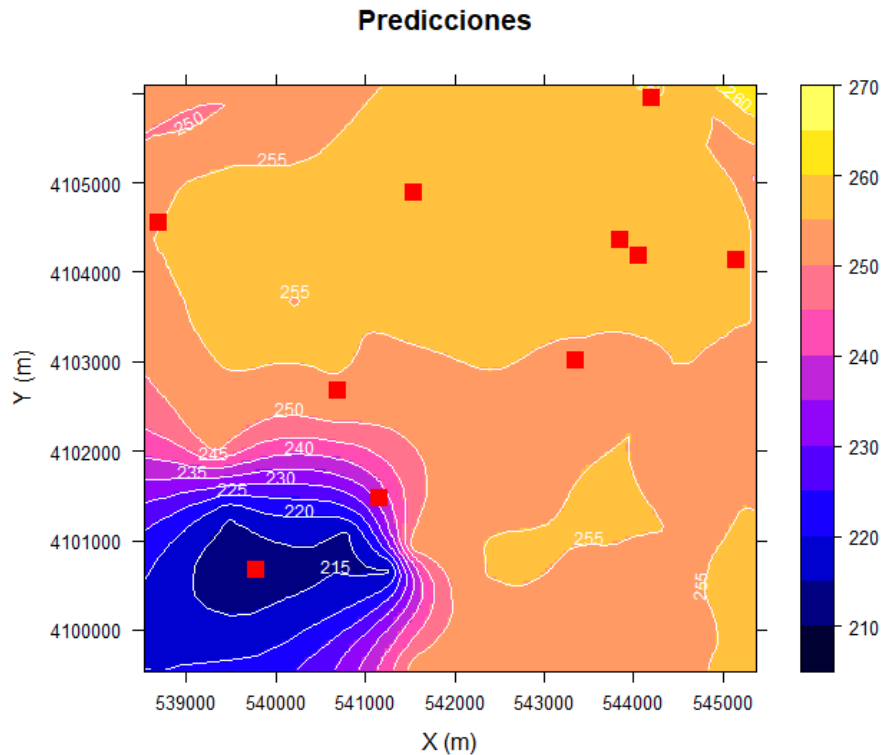


Figura 30. Puntos de validación sobre la predicción.

A continuación, se guardan las predicciones en un archivo tipo .csv, para ser utilizados para calcular el error en Matlab, en ese archivo se tienen los valores de las coordenadas y de la predicción para cada punto.

En Matlab se utilizan las coordenadas en el sistema latitud y longitud por lo que antes hemos de re proyectar a ese sistema de coordenadas (WGS84). Entonces, en Matlab se obtiene el punto de la predicción más cercano a los 10 que se usan para comprobar con datos reales (recordar que tenemos una cuadrícula de puntos con predicciones) y se calcula el error absoluto de la predicción, restando al valor real el estimado en valor absoluto y el error relativo se calcula igual, pero dividiendo entre el valor real del dato y se expresa en porcentaje multiplicando por 100.

Se ha calculado la media de los errores de los 10 puntos de validación y se muestra en la siguiente tabla para la estimación de las horas.

Tabla 3. Media de los errores de la predicción de la hora (c) comparando con datos reales.

Modelo	Error absoluto medio	Error relativo (%)
Mes	3,36	1,51
Días	6,50	2,83
Horas	4,20	1,83
Instantes	7,82	3,49

Se muestran los errores de las predicciones de los distintos modelos de variograma calculados realizadas con los propios datos con los que se han calculado los modelos, por ejemplo, los datos del mes que se han utilizado para generar el modelo del mes hacemos kriging usando esos datos con los modelos b), c) y d) de variograma calculados.

Esto se ha realizado con todos los datos de los modelos de variograma del mes de junio, b), c) y d) y los mostramos en las siguientes tablas con los errores cometidos.

Tabla 4. Media de los errores de la predicción de los datos del mes (a) comparando con datos reales.

Modelo	Error absoluto medio	Error relativo (%)
Mes	0,287	0,21
Días	0,404	0,30
Horas	0,424	0,32
Instantes	0,445	0,33

Tabla 5. Media de los errores de la predicción de los datos de días (b) comparando con datos reales.

Modelo	Error absoluto medio	Error relativo (%)
Mes	0,344	0,26
Días	0,465	0,35
Horas	0,369	0,28
Instantes	0,844	0,64

Tabla 6. Media de los errores de la predicción de los datos de instantes (d) comparando con datos reales.

Modelo	Error absoluto medio	Error relativo (%)
Mes	1,09	0,58
Días	1,8	0,95
Horas	1,98	1,05
Instantes	1,01	0,54

Se puede comprobar que, aunque el modelo coincide exactamente con los datos, el modelo del mes obtiene las mejores predicciones excepto en la media de los instantes del mes, aunque la diferencia no sea grande entre los errores cometidos.

También se observa que cuanto menor sea la cantidad de datos observados utilizados para generar la media que se usa como valor en los modelos, mayores son los errores cometidos, esto se debe a la influencia que tienen las nubes que existan en un determinado momento sobre cada punto, cuanto menor sea el número de datos, mayor efecto tienen la caída de la radiación provocadas por un cielo nublado sobre su valor.

En algunos estudios, observan que los modelos de las estaciones del año para la radiación solar, funcionan igual de bien que los modelos de días y horas, sería posible reducir la necesidad de cálculos al ser posible usar el modelo de variograma del mes para estimar los valores, por lo que no sería necesario calcular un modelo teórico de variograma para los propios datos antes de estimar, ya que el modelo del mes produce buenas estimaciones para datos horarios (8).

6 CONCLUSIONES

El objetivo de este trabajo era lograr obtener estimaciones de los valores de la radiación para nuestra planta solar hipotética en el desierto de Almería sin sensores colocados en el terreno, se ha investigado para obtener los datos que se ajusten a una red de sensores distribuidos irregularmente en nuestra planta y se han utilizado datos obtenidos mediante satélites de la Unión Europea, disponibles de forma gratuita en la red.

Entonces se ha utilizado la geoestadística, haciendo un análisis estructural de los datos para obtener la variabilidad espacial de la radiación solar por distintos periodos de tiempo y los tratamientos necesarios para, posteriormente, aplicar las técnicas de interpolación de kriging para obtener las predicciones de los valores de la radiación solar, por lo que se podría aplicar este proceso para mejorar la eficiencia de nuestros colectores solares ya que se conocen las zonas con la mayor radiación solar.

Al comparar las predicciones con otros valores reales vemos que los errores no son demasiado grandes, por lo que podemos decir que kriging es un método válido para predecir los valores de radiación solar en una zona geográfica de tamaño extenso.

Nuestros datos no se han obtenido de forma experimental mediante sensores sino mediante datos de satélite por lo que no es exactamente una aplicación a una situación real, pero puede servir de base a desarrollar para posteriores trabajos con datos reales.

Los datos obtenidos mediante satélite no son exactos, pero suponen un ahorro en los costes de los sensores piranómetros, su instalación y su mantenimiento, por lo que podría considerarse su uso en aplicaciones no críticas. Se ha realizado una búsqueda de las bases de datos disponibles de forma gratuita en Internet, se ha hecho una pequeña recopilación con las más accesibles y se ha explicado cómo funcionan.

Líneas de investigación futuras.

- Utilizar datos obtenidos de forma real mediante sensores, ya sea por redes de sensores u otros métodos como drones y comparar la diferencia entre los valores obtenidos mediante satélites y si las predicciones varían de forma significativa.
- Comparar distintos métodos de kriging y comparar las desviaciones de las predicciones.
- Comparar con distintos métodos de interpolación, por ejemplo, con la distancia inversa ponderada (IDW por sus siglas en inglés), red irregular triangulada o splines, y estudiar cómo varían los resultados de la predicción de la radiación solar.
- Aplicar las predicciones de los valores al control de los paneles solares para estudiar el posible aumento de la eficacia en la captación de energía solar.
- Estudiar el número de nodos de red óptimos dentro de una región y su disposición mediante kriging con el fin de reducir el coste en el número de sensores sin perder información de la radiación.

7 ANEXO A: CÓDIGO REALIZADO

7.1 Código realizado en R.

```
library(gstat)
library(sp)
library(mapview)
library(DescTools)
library(ggplot2)
library(rgeos)
library(automap) #autofitvariogram
library(rgdal) #reproyectar
library(latticeExtra) #AÑADIR cosas a spplot
library(magrittr, warn.conflicts = FALSE) #pipe operator
library(MASS)
library(bestNormalize)
library(rngtools)

rm(list = ls()) #borrar workspace como clear all en matlab

# opciones para ggplot

theme_set(theme_bw())
theme_update(legend.position='right')

#Cargamos todos los datos.

#datosmes<- read.table("Radiacion_media_mes.txt", header=T,sep = ",")
#datosdias<- read.table("Radiacion_media_dias.txt", header=T,sep = ",")
#datos horas <- read.table("Radiacion_media_hora.txt", header=T,sep = ",")
#datosinst<- read.table("Rad_media_instantes_junio.txt", header=T,sep = ",")

#Elegimos los datos a cargar.

datos_orig <- read.table("Radiacion_media_dias.txt", header=T,sep = ",")
```



```

datos <- datos_orig
coordinates(datos) <- ~lon+lat # necesita libreria sp
#proj4string(datos) = "+proj=longlat +datum=WGS84"
proj4string(datos) <- CRS("+init=epsg:4326") # sistema de coordenadas normal ->
lat+long

#necesario reprojectar los datos.

#+proj=utm +zone=30 +ellps=WGS84 +datum=WGS84 +units=m +no_defs          epsg:32630
UTM zone 30N

#EPSG:25830 PARA EUROPA. PROYECCION UTM HUSO 30 N

datos=spTransform(datos, CRS("+init=epsg:32630"))

#dev.off()

shapiro.test(datos_orig$mediadia) #>0.05 gaussiana

summary(datos_orig)
Desc(datos_orig$mediadia)

##Intentos de normalizar ##

pru1<- log(datos_orig$mediadia)
Desc(pru1)
shapiro.test(pru1)

pru2<- scale(datos_orig$mediadia)
#Desc(pru2)

pru4<- datos_orig$mediadia

normal=(pru4-min(pru4))/(max(pru4)-min(pru4))

Desc(normal)
shapiro.test(normal)

```

```

#Normalizamos con BestNormalize

pru2<- bestNormalize(datos$mediadia, allow_lambert_h = TRUE)
pru2

#Transformación.
medianorm<- predict(pru2)
Desc(medianorm)
shapiro.test(medianorm)

qqnorm(medianorm, pch = 1)
qqline(medianorm, col = "steelblue", lwd = 2)
#Comprobar el p-value, si sale <0.05 no es normal.

#Usaremos datos no normales (requisito no indispensable)

#crea objeto gstat, para hacer predicciones geoestadísticas
gjunio=gstat(id='Radiacion_media_dia', formula=mediadia~1, data=datos )

dat.vgmju=variogram(gjunio) #omnidireccional
dat.vgmju=variogram(gjunio, cutoff=9000) #omnidireccional
#dat.vgmju=variogram(gjunio, alpha=90, cutoff=3000)#variograma exp en direccion
este (hacia donde aumenta x)
head(dat.vgmju)
plot(dat.vgmju)
fitdat.vgme <- fit.variogram(dat.vgmju, vgm('Lin')) # Lin= modelo lineal del
semivariogram
plot(dat.vgmju, fitdat.vgme)

# Mapa de la superficie del variograma

#Lo estoy haciendo con dias de junio

#cutoff es el ancho del mapa
#width es el tamaño de los puntos

map.vgm=variogram(gjunio,width=400,cutoff=4000,map=TRUE)

plot(map.vgm)
ggplot(data.frame(map.vgm), aes(x=map.dx, y=map.dy, fill=map.Radiacion_media_dia)) +
  geom_raster() +

```

```

scale_fill_gradientn(colours=rainbow(7))+
labs(x='Distancia E-W (m)', y='Distancia N-S [m]', fill='Semivarianza')

#Variogramas direccionales

#direccion y(positiva) alpha=0 direccion x(positiva) alpha=90

dat.vgmdir=variogram(gjunio,alpha=c(0,45,90,135),tol.hor=22.5,width=400,cutoff=9000) #puede ser necesario poner la tolerancia angular con tol.hor
ggplot(dat.vgmdir,aes(x=dist,y=gamma,col=factor(dir.hor),shape=factor(dir.hor)))
+
  geom_point(size=2)+
  geom_line() + #conecta los puntos.
  labs(x='Distancia (m)', y='Semivarianza',col='Direccion',shape="")+
  geom_hline(yintercept = var(datos$mediadia),col=2,linetype=2) #añade líneas de referencia

#Modelo teórico, ajustar valores mediante prueba-error fijándonos en el generado con el fit.variogram

mod="Lin"
meseta=2.5
rango=4000
pepita=0.1
modelo3=vgm(meseta,mod,rango,pepita)

vgm_ajustado=fit.variogram(dat.vgmju,model=modelo3)
plot(dat.vgmju,vgm_ajustado)

(error_ajuste=sqrt(attributes(vgm_ajustado)$SSErr/nrow(datos_orig))) #parece que tiene poco error

plot(dat.vgmdir,vgm_ajustado)

# guardar los parámetros del modelo.

parametros=list(y0=vgm_ajustado$psill[1],s=vgm_ajustado$psill[2],a=vgm_ajustado$range[2])
#par_lineal=list(y0=vgm_ajustado$psill[1],s=vgm_ajustado$psill[2])

```

```

#Guardamos el modelo ajustado
modelo_ajustado=vgm(parametros[[2]],mod,parametros[[3]],parametros[[1]])

#ecuacion del modelo lineal.
lineal= function(x,y0,a,s) {ifelse(x<=a, y0+s*x/a, y0+s)}

#Representación gráfica.

ggplot(dat.vgmju,aes(x=dist,y=gamma)) +
  geom_point(size=2) +
  stat_function(fun=lineal,args=parametros)+
  labs(x='Distancia
[m]',y="Variograma",col="Modelo",shape="Direccion")+ggtitle("Variograma dias")

ggplot(dat.vgmdir,aes(x=dist,y=gamma,shape=factor(dir.hor)))+
  geom_point(size=2)+
  geom_line()+
  geom_hline(yintercept=var(datos$mediadia),col=4,linetype=3)+
  stat_function(fun=lineal,
               args=parametros,
               aes(col='Lineal'),size=1)+
  labs(x='Distancia [m]',y="Variograma",col="Modelo",shape="Direccion")

###Kriging ordinario, validación cruzada.###

krig_ord=krige.cv(mediadia~1, locations=datos, model=modelo_ajustado)

summary(krig_ord$residual)
ggplot(as.data.frame(krig_ord),aes(observed,var1.pred))+
  geom_point(col="green")+coord_fixed()+
  labs(x="observados",y="predicciones")
ggplot(as.data.frame(krig_ord),aes(residual))+
  geom_histogram(bins=15,col=1,fill='blue')+
  labs(x="Residuales",y='Frecuencia')+
  geom_vline(xintercept=0,col=2)

#error cuadrático medio

```

```

#Residual contiene la diferencia entre observado y predecido

(ok.rmse=sqrt(sum(krig_ord$residual^2)/(nrow(krig_ord))))
#Sale 0.5.
(ok.mae=sum(abs(krig_ord$residual))/(nrow(krig_ord)))

#Comparar ese error con las cifras significativas de los datos tomados-> menos
del 1%

preds <- krig_ord$var1.pred
actual <- krig_ord$observed
rss <- sum((preds - actual) ^ 2)
tss <- sum((actual - mean(actual)) ^ 2)
rsq <- 1 - rss/tss

#R^2 = 0.844

#library(automap) Para comprobar los valores
#compare.cv(list(krige.cv_output = krig_ord))

#Para kriging necesitamos una variable que indique donde calcular (newdata)

# preprocesamiento para analisis espacial #calculamos los grids, la distancia como
el mayor de la diferencia
# A menor dint más puntos de interpolacion
#signif es un truncamiento, nos quedamos con el número con tres C.S. sin aprox.

dint=signif(max(c(diff(range(datos$lon))/length(datos$lon),diff(range(datos$lat)
)/length(datos$lat))),3)
#dint= 50 #mejor resolucion

# generar secuencia de las x e y
xint= seq(min(datos$lon),max(datos$lon),dint) #coord. x del grid para la
interpolacion
yint= seq(min(datos$lat),max(datos$lat),dint)

datosint= expand.grid(x=xint,y=yint)
gridded(datosint)=~x+y # grid como objeto espacial

#Necesario poner los datos como el mismo sistema coordenado de referencia

```

```

#proj4string(datosint) <- CRS("+init=epsg:4326" ) #puede ser necesario indicar
eso antes
proj4string(datosint) <- CRS(proj4string(datos))

#Interpolación con kriging ordinario

#Cambiando datosint se podrian calcular mas puntos si se desea.

#Para hacer más fácil el cálculo se juntan todos los datos en una tabla.

datosall<- read.table("Radiacion_media_all.txt", header=T, sep = ",")

ogdatosall<- datosall

coordinates(datosall) <- ~lon+lat # necesita libreria sp

proj4string(datosall) <- CRS("+init=epsg:4326") # sistema de coordenadas normal
-> lat+long

datosall=spTransform(datosall, CRS("+init=epsg:32630"))

#He cambiado datos por datosall para tener todos los datos en la variable.
#Poner en la primera variablemediames, mediadia, mediahora o mediainst según la
predicción a calcular.

resultado_dias=krige(mediainst~1,datosall,datosint,model=modelo_ajustado)

#.var contiene la varianza var=sd^2
#Añadir desviaciones estandar
resultado_dias$var1.des=sqrt(resultado_dias$var1.var)

#Mapa de las predicciones y las desviaciones

(p1=spplot(resultado_dias,"var1.pred",asp=1, col.regions=bpy.colors(60),
main="Predicción mes de junio",
contour=T, col="white",labels=list(cex=0.75,col="white"), pretty=T,
scales=list(draw=TRUE), xlab="X (m)", ylab="Y (m)"))

res=cbind(resultado_dias@coords, resultado_dias$var1.pred)

```

```

#Guardamos la predicción.

#Cambiar el nombre a las columnas
colnames(res)<- c("x","y","pred")

pred<-res

pred<-as.data.frame(pred)
coordinates(pred)<- ~x+y

proj4string(pred) <- CRS("+init=epsg:32630")

pred=spTransform(pred, CRS("+init=epsg:4326"))

# Exportar el spatialpointsdataframe

pred<- as.data.frame(pred)
write.csv(pred, file = "datos_pred_inst_modelodias.csv")

#Añadir los puntos de validación al mapa

puntos_pred <- read.table("coordenadas_comprobaciones.txt", header=T, sep = ",")

coordinates(puntos_pred) <- ~lon+lat # necesita libreria sp

proj4string(puntos_pred) <- CRS("+init=epsg:4326") # sistema de coordenadas
normal -> lat+long

#Necesario reprojectar los datos.

puntos_pred=spTransform(puntos_pred, CRS("+init=epsg:32630"))

#Se creap la capa con las localizaciones de los puntos y se pone la forma y el
tamaño.

comprobacion.layer <- list("sp.points", puntos_pred, pch=15,col=2, cex=1.5)

(pl=spplot(resultado_dias,"var1.pred",asp=1, col.regions=bpy.colors(60),
main="Predicciones",
contour=T, col="white",labels=list(cex=0.75,col="white"), pretty=T,
scales=list(draw=TRUE), xlab="X (m)", ylab="Y (m)",
sp.layout=comprobacion.layer))

```

```
( p2=spplot(resultado_dias,"var1.des", asp=1, col.regions=cm.colors(80),
  main='Desviacion',
  scales=list(draw=TRUE), xlab="X (m)", ylab="Y (m)",
  sp.layout=comprobacion.layer ) )
```

- Análisis exploratorio de los datos.

```
#Estudio estructural de los datos.
```

```
library(gstat)
library(sp)
library(mapview)
library(DescTools)
library(ggplot2)
library(rgeos)
library(automap) #autofitvariogram
library(rgdal) #reproyectar
library(geosphere) #distHaversine

# opciones para ggplot

theme_set(theme_bw())
theme_update(legend.position='right')

# datos
#list.files()
#datos <- read.table("~/R/geostadistica proyecto prueba/proyecto/BroomsBarn.txt",
# sep="\t")
datos <- read.table("Radiacion_media_mes.txt", header=T, sep = ",")
datosorig=datos #copia seg

coordinates(datos) <- ~lon+lat # necesita libreria sp
#proj4string(datos) = "+proj=longlat +datum=WGS84"
proj4string(datos) <- CRS("+init=epsg:4326") # sistema de coordenadas normal ->
lat+long
```



```

#exploracion de datos
summary(datosorig)
Desc(datosorig$Junio)

#Test de normalidad
shapiro.test(datosorig$Enero) #Enero no tiene distribución normal.
shapiro.test(datosorig$Junio)

qqnorm(datos$Enero, pch = 1)
qqline(datos$Enero, col = "steelblue", lwd = 2)

apply(datosorig,2,var)
head(datosorig)
#quantile(dist(datosorig[,1:2])) Para ver las distancias, pero es para planos.

ma <- distm(datosorig[,1:2]) #Calcula la matriz de distancias

quantile(distm(datosorig[,1:2],datosorig[,1:2], fun=distVincentyEllipsoid))
#Distancia máxima en metros, media,..
quantile(distm(datosorig[,1:2],datosorig[,1:2], fun=distHaversine)) #Varía un
poco según el método usado
#pointDistance(datosorig,lonlat=TRUE)

ggplot(datosorig,aes(Enero))+
  geom_histogram(aes(y=..density..),bins=10,col=1,fill=4,alpha=.5)+
  geom_vline(xintercept = mean(datosorig$Enero),col=2)+
  geom_density(col=4)+
  labs(y='Densidad')

# procesamiento para analisis espacial
#calculamos los grids, la distancia
dint=
signif(max(c(diff(range(datos$lon))/length(datos$lon),diff(range(datos$lat))/len
gth(datos$lat))),3)

# generar secuencia de las x e y
xint= seq(min(datos$lon),max(datos$lon),dint) #coord. x del grid para la
interpolacion
yint= seq(min(datos$lat),max(datos$lat),dint)

datosint= expand.grid(x=xint,y=yint)

```

```
gridded(datosint)~x+y # grid como objeto espacial

mapview(datos, burst= T, hide=T)
mapview(datos, zcol='Junio', legend=T)

# contorno que contiene los datos, edge de la figura

#No funciona porque necesita planar coordinates
#q=min(c(diff(range(datos$lon)),diff(range(datos$lat))))
#outline= gBuffer(datos, byid=FALSE, id=NULL, width=dint*q,
#                 joinStyle = "ROUND", quadsegs=10)
#outline= gBuffer(outline, byid=FALSE, id=NULL, width=-dint*q,
#                 joinStyle = "ROUND", quadsegs=10)
#plot(outline)
```

7.2 Código en Matlab

Generación de los puntos aleatoriamente: generar_puntos.m

```
% Generar los puntos del mapa
clear, close

NPUNTO=10; %Numero estaciones

AREA=50; %Se ha hecho con 50 km2
TAM_SUPERF=sqrt(AREA); %AREA CUADRADA, LADO EN KM
%N números aleatorios en el intervalo (a,b) con la fórmula r=a+(b-a).*rand(N,1)
puntos=TAM_SUPERF.*rand(NPUNTO,2);

for i=1:NPUNTO
    hold on
    plot(puntos(i,1),puntos(i,2),'ro-')
end
axis([ 0 TAM_SUPERF 0 TAM_SUPERF])
title('Superficie')
xlabel('X')
ylabel('Y')

% Guardar una generacion



- Cálculo de coordenadas, plot de los puntos y cálculo de distancia entre los puntos.


%Aqui tenemos los datos de los puntos en un plano, los pasamos a coordenadas y se
mapean finalmente calculamos la distancia.

close

NPUNTO=50; %Numero estaciones

AREA=50; %Elegir area
TAM_SUPERF=sqrt(AREA); %AREA CUADRADA, LADO EN KM

lat1=37.105;
lat2=37.041;
lon1=-2.488;
lon2=-2.5675;
```

```

diflat=lat1-lat2;
diflon=lon1-lon2;

%Puntos elegidos de forma aleatoria y guardados, se cargan.
load estaciones1
%declaracion de vector de coord
coordenadas_est=ones(NPUNTO,2);

%Grafica de las estaciones en un plano
figure()
for i=2:NPUNTO

    hold on

    plot(estaciones(i,1),estaciones(i,2),'ro-')
end
axis([ 0 TAM_SUPERF 0 TAM_SUPERF])
title('Superficie')
xlabel('X')
ylabel('Y')

%Pasamos a coordenadas con una relación lineal.
for i=1:NPUNTO
coordenadas_est(i,1)=estaciones(i,1)*diflon/TAM_SUPERF+lon2;
coordenadas_est(i,2)=estaciones(i,2)*diflat/TAM_SUPERF+lat2;
end

filename = 'testdata.xlsx'; xlswrite(filename,coordenadas_est,1,'C2')

%El primer punto no lo tengo como estación, por eso lo salto.
figure()
for i=2:NPUNTO

    hold on

    plot(coordenadas_est(i,1),coordenadas_est(i,2),'go-')
end
%axis([ lon2 lon1 lat2 lat1])

```

```

title('Superficie')
xlabel('X')
ylabel('Y')

%calculo de la matriz de distancias.

%Las distancias se van a poner en una matriz de 49*49 simétrica, donde el
%componente i,j es la distancia de i hasta j.

%Para calcular las distancias usar haversine (o distance)

coor=[coordenadas_est(:,2),coordenadas_est(:,1)];
%coor(:,1) contiene latitudes,(:,2) longitudes

distancia=ones(49); %49*49

for i=2:50

    for j=2:50

distancia(i-1,j-1)=haversine(coor(i,1),coor(i,2),coor(j,1),coor(j,2));

    end
end

```

- Función Haversine para cálculo de distancias en la Tierra con coordenadas.

```
function dist= haversine(lat1,lon1,lat2,lon2)
%Calculo de la distancia usando haversine

R= earthRadius('km'); % 6371 (el radio es aprox)
dlat=deg2rad(lat2-lat1);
dlon=deg2rad(lon2-lon1);
lat1=deg2rad(lat1);
lat2=deg2rad(lat2);

a = (sin(dlat./2)).^2 + cos(lat1).*cos(lat2).*(sin(dlon./2)).^2;
    c = 2.*asin(sqrt(a));
    dist=c*R;

end
```

- Lectura de los datos de los ficheros, cálculos y transformaciones de los datos.

```
%Varias versiones de este programa, esta calcula la media de los dias %14 a 17 de junio.
```

```
%run puntoelegido.m
```

```
%clear all
```

```
close all
```

```
sheet=1;
```

```
%Elegir el rango de los datos con el que queremos calcular la media.
```

```
Rango= 'G34411:G34794';
```

```
% seleccionar carpeta de forma interactiva (datos buenos o datos comprobar)
```

```
ubicacion = uigetdir;
```

```
%Guardamos los nombres como vector
```

```
D=dir([ubicacion, '*.*csv']);
```

```
%Extraer los nombres
```

```
nombre_archivos={D(:).name}
```

```
datos = cell(length(D),1);
```

```
for i = length(D):-1:1
```

```
    % Crear el nombre completo y parcial
```

```
    nombrecompleto = [ubicacion filesep D(i).name]; %filesep en windows es '\'
```

```
    % Leer datos
```

```
    datos{i} = xlsread(nombrecompleto, sheet, Rango);
```

```
end
```

```
%Pasar a matriz
```

```
[L,N]=size(datos);
```

```
radiacion_globalmediainstantes=[];
```

```
M = max(cellfun(@numel, datos));
```

```
% maxEl=M ;%max number of elements of a cell element in your cell
```

```
% C=cell2mat(cellfun(@(x) [cell2mat(x) zeros(1,maxEl-numel(x))],
```

```

datos,'UniformOutput',0))
% for j=1:L
% datos(L)=[datos(L)

%Nota importante: Para hacer la matriz tiene que tener el mismo tamaño

for i=1:L
    dif=M-numel(datos{i});
    if dif>0
        radiacion_globalmediainstantes(:,i)=[cell2mat(datos(i)) ; zeros(1,dif)];
%todos tengan el mismo tamaño

    else
        radiacion_globalmediainstantes(:,i)=[cell2mat(datos(i))];

    end

end

aux2=radiacion_globalmediainstantes'; %Quiero la matriz de esta forma.

%Voy a eliminar los elementos cero para sumar la media sin ceros.
[tamf,tamc]=size(aux2);
cont=0;
total=0;

media_dia=zeros(tamf,1)

% dias_mes=[31,28,31,30,31,30,31,31,30,31,30,31];
% datos_dia=4*24; %4 datos cada hora (cada 15 min).

for i=1:tamf
    for j=1:tamc

        if aux2(i,j)>0
            total=total+aux2(i,j);
            cont=cont+1;

```



```

end

end

media_dia(i)=total/cont;

cont=0;
total=0;

end

media_dias_datos=media_dia;

datos2=media_dias_datos;

%Guardar datos de validación.
%save vali_dias datos2

%Ejecutar antes de esta parte puntoelegido.m
%run puntoelegido.m
%Tabla para usar en R.
tabla_media=table(coordenadas_est(2:50,1),          coordenadas_est(2:NPUNTO,2),
datos2,...
    'VariableNames',{'lon','lat','mediadia',})

writetable(tabla_media,'Radiacion_media_dias')

%Vamos a pintar todos los puntos para estudiar los datos.

% run nube_puntos.m

• Nube de puntos:

%Nube de puntos. Comparamos los valores en el mismo t entre estaciones.

%HAY QUE EJECUTAR Media_enero.m antes!
close all

```

```
%Elegir la estacion que representar cambiando el número. (1-49)
est1=2;
est2=16;
plot(aux1(est1,:),aux1(est2,:),'bx')

string1=strcat('Estacion ', num2str(est1));
string2=strcat('Estacion ', num2str(est2));
title('Valores por estación.')
xlabel(string1)
ylabel(string2)

% refline
%Otra opción.

%Representamos la recta que más se ajusta a la nube de puntos.

poli = polyfit(aux1(1,:), aux1(2,:), 1); %ajuste a un polinomio tipo ax+b grado
1

% Buscamos los puntos que se obtienen por poli
puntosx = linspace(min(aux1(1,:)), max(aux1(1,:)), 200); %se genera 200 puntos
equidistantes entre min y max
puntosy = polyval(poli, puntosx); %evalua el polinomio en los puntos x obtenidos

hold on;
plot(puntosx, puntosy, 'r-', 'LineWidth', 3);
```

- Cálculo del error de la predicción.

```

% Calculo del punto mas proximos de la prediccion y el error de la
% prediccion.

%%%% IMPORTANTE!   %%%

%DOS COSAS: EL FILE A ELEGIR DEBE ESTAR DENTRO DE LA CARPETA DE
%DATOS/PREDICCIONES

%Elegir los datos a cargar para ver el error.

%load datos\vali_mes.mat
%load datos\vali_dias.mat
  %load datos\vali_hora.mat
% load datos\vali_instant.mat

%Coordenadas de los puntos de validación.

coor_puntos=[-2.50255556126556          37.0987135144990;-2.51226112447342
37.0723449213304;-2.54229059133516      37.0695175168455;-2.49195734711735
37.0823640326471;-2.56476153660002      37.0863993491749;-2.53261982340732
37.0892999476469;-2.53716610266111      37.0586656049279;-2.50664141534215
37.0845009713186;-2.50428160785960      37.0829262722543;-2.55264362793793
37.0514071510525];

n=length(coor_puntos)

%csvread solo admite valores numericos.
%Prediccion de horas con modelo horas.

[archivo,ubicacion]=uigetfile('*.*csv');
predictions=readtable(fullfile(ubicacion,archivo));

%
% %Si se cambia la ubicacion seleccionar el archivo con
% % ubicacion = uigetdir;
% % pred=readtable(ubicacion+"\datos_pred_pru.csv")  ABRE TODO LOS
% ARCHIVOS! CUIDADO!
% %

```

```

estimacion=table2array(predictions(:,2));
puntos=table2array(predictions(:,3:4));

%Buscamos el punto más cercano en distancia para comparar nuestros datos de
%validación con el punto más cercano.
M=length(puntos);
distkm=zeros(1,M);
distmin=zeros(1,n);
puntomin=zeros(1,n);
for i=1:n

    for j=1:M
        dist=distance(coor_puntos(i,1),coor_puntos(i,2),puntos(j,1),puntos(j,2));

        distkm(j)=deg2km(dist); %Pasar de grados a km, por defecto asume el radio de
la Tierra
    end

    [distmin(i),puntomin(i)]=min(distkm);
end

%Quiero conocer el error de la prediccion, tengo que sacar los valores de
%la radiación para los instantes en los que se ha hecho la prediccion.

%Calculo del error.

error_abs=zeros(1,10);
error_rel=zeros(1,10);
for i=1:10

    error_abs(i)=abs(datos2(i)-estimacion(puntomin(i)));

    error_rel(i)=abs(datos2(i)-estimacion(puntomin(i)))/datos2(i);
end

%Mirar el error_rel y comparar con la desviación

```

```

% Las coordenadas de los puntos de validación
%a r, para dibujarlos en el mapa de la prediccion.

%     serie1=(1:10) '
%
tabla_comp=table(coor_puntos(:,1),coor_puntos(:,2),serie1,'VariableNames',{'lon'
,'lat','id',})
%     writetable(tabla_comp,'coordenadas_comprobaciones')
max(error_abs);

error_absoluto_medio=mean(error_abs)
error_relativo_medio=mean(error_rel)

```


REFERENCIAS

1. *Wireless Sensor networks: a survey*. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. 4, s.l. : Comput. Networks, 2002, Vol. 38.
2. *Wireless sensor network survey*. Yick, Jennifer, Mukherjee, Biswanath y Ghosal, Dipak. 12, s.l. : Computer Networks, 2008, Vol. 52.
3. **Poblete, Julio A. Valle**. *Wireless Sensor Networks and Kriging*. s.l. : Universidad de Valencia, 2010.
4. **Hengl, Tomislav**. *A Practical Guide to Geostatistical Mapping*. s.l. : University of Amsterdam, 2009.
5. **Bachmaier, M. y Backes, M.** [Online] <https://www.fca.unesp.br/Home/Instituicao/Departamentos/CienciadoSolo/gepag/variogram-or-semivariogram.pdf>.
6. **Emery, Xavier**. *Geoestadística*. [Online] 2013. https://www.u-cursos.cl/usuario/f697544fb48474f4153ef2aafbe4c963/mi_blog/r/Geoestadistica.pdf.
7. **Cheng Si, B. and Biswas, A.** *Model Averaging for Semivariogram Model Parameters, Advances in Agrophysical Research*. [Documento] 2013.
8. *Determination of Semivariogram Models to Krige Hourly and Daily Solar Irradiance in Western Nebraska*. **G. Merino, y et al.** 6, s.l. : Journal of Applied Meteorology, 2001, Vol. 40.
9. **Cressie, N.** *Statistics for spatial data*. New York : John Wiley & Sons, 1993.
10. **almeria.es**. [Online] [http://www.almeria.es/Servicios/Informacion/Informacion.nsf/d5eda4da81a73132c1256fcb005a3b1b/5d0fdb5dc132372c1257384004d0958/\\$FILE/Estudio.pdf](http://www.almeria.es/Servicios/Informacion/Informacion.nsf/d5eda4da81a73132c1256fcb005a3b1b/5d0fdb5dc132372c1257384004d0958/$FILE/Estudio.pdf).
11. [Online] <https://www.mapa.nom.es/espana/pueblos-provincia-de-almeria/4/1>.
12. **Weather Spark**. [Online] <https://es.weatherspark.com/y/38232/Clima-promedio-en-Tabernas-Espa%C3%B1a-durante-todo-el-a%C3%B1o>.
13. **Business standard**. [Online] 10 Abril 2019. https://www.business-standard.com/article/pti-stories/renew-power-commissions-300-mw-solar-plant-in-pavagada-119041000686_1.html.
14. *The method Heliosat-2 for deriving shortwave solar radiation from satellite images*. **Rigollier, C., Lefèvre, M., Wald, L.** 2, 2004, *Solar Energy*, Vol. 77, pp. 159-169.
15. **soda-pro**. [Online] <http://www.soda-pro.com/documents/10157/326300/CAMS-data-license.pdf/59241f5b-085d-4e79-bf42-1f19b0731d9f>.
16. **Marion Schroedter-Homscheidt, et al. soda-pro**. [Online] 14 12 2019. http://www.soda-pro.com/documents/10157/326332/CAMS72_2015SC3_D72.1.3.1_2018_UserGuide_v1_201812.pdf/95ca8325-71f6-49ea-b5a6-8ae4557242bd.
17. *Is it necessarily a normally distributed data for kriging? A case study: soil salinity map of Ghahab area, central Iran*. **Bodaghabadi, M.B.** 2, s.l. : Desert , Vol. 23.
18. *Normality Tests for Statistical Analysis: A Guide for Non-Statisticians*. **Zahediasl, A. and Ghasemi, S.** 2, s.l. : Int J Endocrinol Metab, 2012, Vol. 10. DOI: 10.5812/ijem.3505.
19. **Peterson, Ryan A.** *Estimating normalization transformations with bestNormalize*. [Online] 2017. <https://github.com/petersonR/bestNormalize>.
20. **Del Bosque González, Isabel, et al.** *Los Sistemas de Información Geográfica y la Investigación en Ciencias Humanas y Sociales*. Madrid : s.n., 2012.

21. Consejería de agricultura, ganadería, pesca y desarrollo sostenible. [Online] http://www.juntadeandalucia.es/medioambiente/site/rediam/menuitem.04dc44281e5d53cf8ca78ca731525ea0/?vgnnextoid=2a412abcb86a2210VgnVCM1000001325e50aRCRD&lr=lang_es.
22. *Multivariable geostatistics in S: the gstat package*. Pebesma, E.J. 7, s.l. : Computers & Geosciences, 2004, Vol. 30.
23. Pebesma, Edzer. [Online] 16 Mayo 2019. <https://cran.r-project.org/web/packages/gstat/vignettes/gstat.pdf>.

