Interpolación de Concentraciones de O_3 y $PM_{2.5}$ mediante *Kriging* en la Conurbación de Santiago

Estela Blanco

Ismael Bravo

FONDECYT Iniciación Nº11240322: Climate change and urban health. How air pollution, temperature, and city structure relate to preterm birth

Santiago, 21 de enero de 2025

Tabla de Contenido

Introducción

En el marco del FONDECYT de iniciación Nº11240322, titulado *Climate change and urban health: how air pollution, temperature, and city structure relate to preterm birth* dirigido por la profesora Estela Blanco, uno de sus desafíos consistió en la interpolación de los datos de la concentración en el aire de ciertos contaminantes (O_3 y $PM_{2.5}$) en la coordenadas específicas donde se emplazan los edificios de administración municipal de todas las comunas que componen la llamada conurbación de Santiago (que contempla todas las comunas de la Provincia de Santiago más la comuna de Puente Alto correspondiente a la Provincia Cordillera).

Así, para abordar dicho desafío se acordó el siguiente listado de tareas para ser desarrolladas, cuyos resultados se exponen en el presente reporte:

1. Revisión bibliográfica y documental sobre el método *kriging* de interpolación espacial de datos.
2. Preparación y procesamiento de las bases de datos requeridas para las tareas sucesivas.
3. Ajustar un modelo de *ordinary kriging* para cada uno de los contaminantes (O_3 y $PM_{2.5}$) y fechas contempladas en los datos, obteniendo las interpolaciones requeridas, además de la elaboración de gráficos para los resultados.
4. Codificar una función que permita asignar el promedio de exposición para cada contaminante a una base de nacimientos, según lugar de nacimiento y tres diferentes ventanas de exposición de las gestantes.

1. Revisión Bibliográfica

Si bien durante la revisión realizada se consultó una serie de fuentes bibliográficas tanto sobre la teoría que sustenta a la metodología *kriging* de interpolación espacial como su aplicación práctica en softwares de análisis, la fuente principal, de la cual se extrae la mayor parte de la información presentada en este apartado corresponde a la tesis de Andreas Lichtenstern del año 2013 del Departamento de Matemáticas de las Technische Universität München.

El método *kriging*, se utiliza para realizar interpolación o predicción espacial, es decir, utilizar los datos disponibles de los puntos muestreados para predecir los valores en las ubicaciones no observadas. El *kriging* consiste en una estimación lineal insesgada óptima o BLUP (por su sigla en inglés) y su objetivo general consiste en predecir el valor de una función aleatoria subyacente $Z(x)$ en un punto arbitrario x_0 a partir de observaciones $z(x_i)$ tomadas en n puntos de muestreo x_i dentro de una región geográfica D. La idea principal es asignar más peso a los puntos de muestreo cercanos para mejorar la precisión de la predicción. Esto se basa en el conocimiento de una estructura espacial, modelada mediante las propiedades de segundo orden, como el variograma o la covarianza de $Z(x)$.

El *kriging* utiliza un promedio ponderado de las observaciones para estimar el valor en un punto no muestreado. La cuestión clave radica en cómo definir los pesos "óptimos". En este contexto, los términos "mejor" u "óptimo" se refieren a que la estimación final debe ser insesgada y tener la mínima varianza de error entre todos los predictores lineales insesgados. Los pesos óptimos dependen de las suposiciones sobre el valor medio $\mu(X)$ y el variograma o la función de covarianza de $Z(x)$.

En la literatura se identifican diferentes tipos de *kriging*. Para este ejercicio particular se empleo el *ordinary kriging*, que asume que la media de la variable en estudio es constante dentro de la región de interés, aunque su valor específico es desconocido. Además, utiliza una función denominada semivariograma, que describe cómo varía la relación espacial entre puntos a diferentes distancias. Este método calcula el valor estimado como una combinación lineal ponderada de los valores conocidos, donde las ponderaciones se determinan para minimizar la varianza del error de predicción y garantizar que la estimación sea insesgada. Entre sus propiedades óptimas se encuentran la insesgadez, asegurando que el valor esperado de las estimaciones sea igual al valor verdadero, y la varianza mínima, que reduce al máximo el error de predicción.

Por su parte, el variograma permite estudiar cómo cambia la relación espacial entre datos de acuerdo con la distancia entre los puntos de muestra. La idea clave es que la relación espacial entre dos puntos no depende de su ubicación absoluta (es decir, su posición en el mapa), sino de la distancia relativa entre ellos. Esto significa que solo importa cuánto están separados, no dónde están específicamente.

Para entenderlo, supongamos que tenemos un conjunto de puntos de muestra x_i en una región espacial D, y que en cada punto observamos un valor $z(x_i)$. Estos valores son tomados como realizaciones de variables aleatorias $Z(x_i)$ de una función aleatoria $Z = Z(x)$, definida en el dominio espacial D. El variograma busca cuantificar esta dependencia espacial entre puntos de acuerdo a su distancia.

Dada la gran cantidad de información contenida en el variograma, metodológicamente se suele proceder con la obtención del variograma experimental a partir de los datos observados, lo cual implica los siguientes pasos:

1. Generar el *variogram cloud*, consistente en graficar las disimilitudes entre pares de puntos espaciales en función sus distancias *lags* h.

2. Construir el variograma experimental, basado en la agrupación de *lags* h similares.

3. Ajustar un modelo de variograma paramétrico, es decir, seleccionar un modelo teórico adecuado (en función del variograma experimental) y estimar sus parámetros, por ejemplo, mediante un ajuste de mínimos cuadrados.

Una vez concluidos estos pasos, se ha obtenido la función de variograma adecuada, la cual se utiliza en el *kriging* para predecir valores en ubicaciones donde no hay datos observados. La teoría detrás del variograma, conocida como variograma teórico, ayuda a restringir el conjunto de funciones válidas para el ajuste en el paso 3, ya que ciertas propiedades matemáticas deben cumplirse para que una función sea un variograma válido.

2. Preparación de Datos

A lo largo del trabajo desarrollado se requirió el procesamiento de una serie de bases de datos, tanto proporcionadas por la profesora Blanco como sets de datos producidos a partir de análisis propios. A continuación, se describen en grandes rasgos tanto la manipulación como procesamiento de los conjuntos de datos involucrados en esta etapa del proyecto de investigación, los cuales fueron realizados mediante el software `R`.

Datos de Contaminación

Este conjunto de datos fue proporcionado por el equipo de la profesora Blanco y consistió en 20 diferentes bases de datos, que correspondían a las mediciones de la concentración en el aire de los dos contaminantes analizados (O_3 y $PM_{2.5}$) para cada una de las 10 estaciones del Sistema de Información Nacional de Calidad del Aire (SINCA) de la Región Metropolitana.

Cabe señalar que los datos proporcionados no contaban con datos faltantes, puesto que previamente habían sido trabajadas por el equipo de investigación por medio del método de imputación múltiple por variables (MVI). Por lo que el manejo de *missings values* no representó un problema. Sin embargo, se solicitó la incorporación de una onceava estación de monitoreo correspondiente a la que se emplaza en la comuna de Talagante. Si bien dicha estación no se encuentra dentro del territorio cubierto por el estudio, cualquier observación adicional aporta al ajuste del *kriging*, por lo que el total de bases de datos ascendió a 22.

Posteriormente, los procesamientos a las bases consistieron básicamente en la unificación de las 22 bases de datos, la incorporación tanto de las coordenadas de cada estación como del vector del polígono respectivo. Puntualmente, se debió precisar un cambio de coordenadas para la estación de monitoreo de Cerrillos puesto que en determinada fecha cerró la estación I y las mediciones comenzaron a realizarse en la estación II. De este modo, se obtuvo una base consolidada que constaba de 7 variables y 46.752 observaciones, correspondientes a las mediciones tanto de ozono como de material particulado para una estación y fecha específicas (que van desde del 1 de enero de 2009 hasta el 31 de diciembre de 2020). Adicionalmente, se aseguró que los campos asociadas tanto a fechas como coordenadas fuesen tratados como objetos de dicha naturaleza, mediante los paquetes `lubridate`, `automap`, `sp` y `sf`.

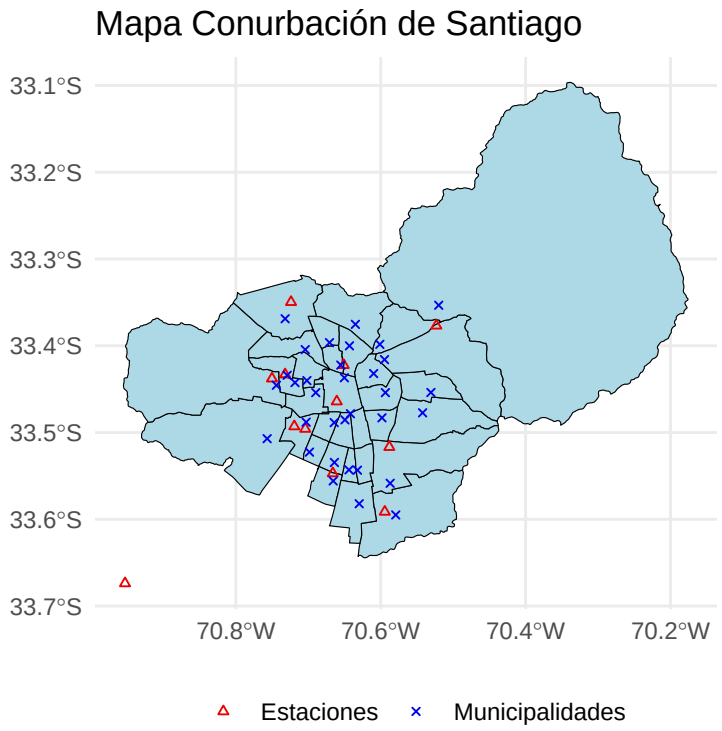
Datos de Nacimientos

Asimismo, se proporcionó una base de datos con 8 variables para 916.955 nacimientos que se corresponden con las delimitaciones temporales y geográficas establecidos en el diseño del estudio. Particularmente, este set de datos no requirió mayor procesamiento de mi parte, exceptuando la homologación de los nombres de las comunas, puesto que no compartían el mismo formato con el resto de las bases utilizadas para el análisis.

Datos Georreferenciados

Finalmente, se debió construiir una serie de conjuntos de datos con coordenadas de interés para el análisis. Particularmente, las coordenadas tanto de las estaciones de monitoreo como de los municipios se extrajeron desde *Google Maps*, las cuales se homologaron a un mismo sistema geodésico.

Asimismo, se extrajo desde el repositorio de la Biblioteca Nacional una base de datos que contiene vectores geométricos (polígonos) de las diferentes subdivisiones administrativas de Chile, de la cual se seleccionó el subconjunto de datos territoriales de interés para el estudio (conurbación de Santiago). En el siguiente gráfico se presenta un mapa del polígono seleccionada así como las ubicaciones tanto de las centrales de monitoreo donde se generaron las mediciones de contaminación como las posiciones de los municipios (puntos que se desean interpolar).



3. Ajuste de Kriging

El siguiente paso consistió en la tarea de ajustar el *ordinary kriging* para poder generar las interpolaciones necesarias. Ahora bien, uno de los desafíos más importantes consistió en gran volumen de datos disponibles. En efecto, se contaba con mediciones de contaminación para 4.383 fechas únicas. Es decir, se debían ajustar 8.766 modelos de *kriging* puesto que para cada fecha existían dos posibles contaminantes (O_3 y $PM_{2.5}$). Como se verá más adelante, esta enorme cantidad de modelos implicaba el ajuste del variograma teórico más adecuado para cada uno de los modelos por fecha y contaminante, lo que supuso serias dificultades en términos tanto teóricos como en la programación del análisis.

Asimismo, el segundo gran desafío que supuso ajustar un *kriging* para el conjunto de datos disponible consistió en interacción espacio-temporal de los datos, vale decir, dilucidar si para las observaciones no solamente existe un efecto de la variable espacial, sino que posiblemente también existe tanto un efecto temporal como la interacción espacio-temporal. Así, se exploró contemplar un análisis espacio-temporal, incluso se esbozó un ejercicio de periodograma. Sin embargo, finalmente se asumió independencia de las variables espacial y temporal, aunque los datos no se sometieron a un test de separabilidad que hubiese sido el procedimiento más riguroso. Este supuesto se sustenta en que el objetivo del análisis no consistía en entender el comportamiento subyacente de la concentración de ambos contaminantes, sino simplemente generar interpolaciones confiables para utilizarlas en análisis epidemiológicos del fenómeno de partos prematuros.

A continuación, se presenta el detalle del procesamiento de los datos para ajustar primero un *ordinary kriging* de una fecha puntual y posteriormente generar una función que produjese las interpolaciones de las coordenadas comunales definidas para el conjunto de los datos. Si bien se detallará más adelante, es importante señalar que inicialmente mayoritariamente mediante funciones de la librería `gstats`

Ajsute Puntual con `gstast`

Como se mencionó, para ajustar el *ordinary kriging* primero se procedió con ambos contaminantes para una única fecha puntual aleatoria. En las siguientes secciones se detalla el procesamiento paso a paso hasta obtener una interpolación para las coordenadas de los 33 municipios que componen la conurbación de Santiago.

Crear Objetos `gstast`

El primer paso del ajuste del *kriging* consiste en generar objetos de tipo `gstats` con un fecha puntual aleatoria, teniendo como fórmula un modelo de tendencia constante para la concentración de cada uno de los contaminantes, es decir, sin dependencia de otras variables. Asimismo, se asocia dicha fórmula con las variables georreferenciadas de longitud y latitud, previamente procesadas.

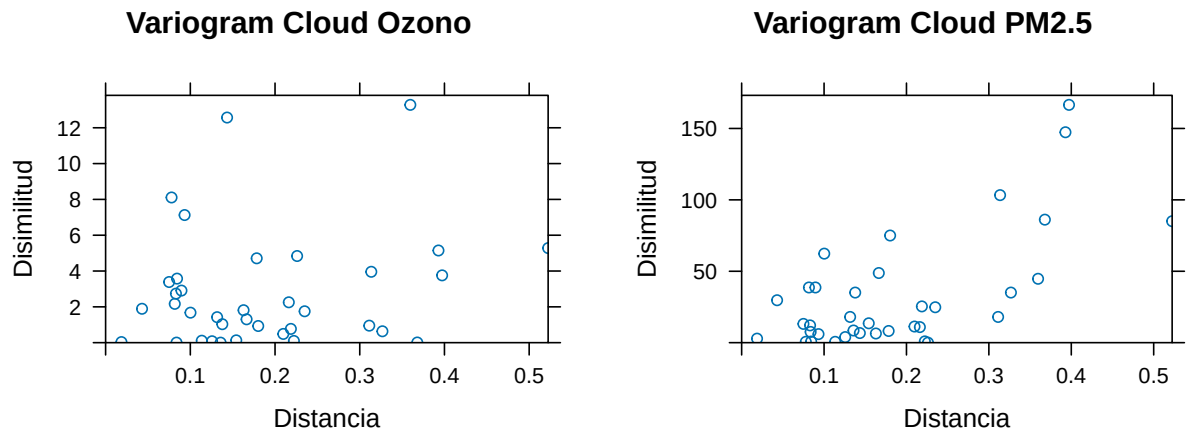
```
## Filtrar de datos ----
ejemplo <- combined_data %>%
  filter(fecha == as.Date("2018-07-10")) %>%
  filter(!is.na(o3) & !is.na(pm25))
```

```r
## Crear objetos gstats ----
g1 <- gstat(g = NULL, id = "o3", formula = o3 ~ 1,
            locations = ~longitud + latitud, data = ejemplo)
g2 <- gstat(g = NULL, id = "pm25", formula = pm25 ~ 1,
            locations = ~longitud + latitud, data = ejemplo)
```

Generar Nube de Variograma

A continuación, se calcula genera el variograma nube para ambos objetos `gstats` relacionados a los contaminates. Cabe señalar que el argumento `cutoff=Inf` indica que no hay límite en la distancia máxima entre pares de puntos considerados en el cálculo, mientras que el argumento `cloud=TRUE` especifica que se generará un variograma nube, mostrando la disimilitud para cada par de puntos en función de la distancia. Cabe señalar que el variograma nube consiste en una herramienta exploratoria para evaluar la calidad de los datos y comprender la estructura de dependencia espacial antes de ajustar un modelo de variograma más formal como los modelos teóricos que se verán más adelante. Finalmente, se presenta un par de gráficos con los variogramas nube para cada contaminante en la fecha aleatoria seleccionada.

```r
## Crear variogram cloud ----
vcloud_o3 <- variogram(object = g1, cutoff=Inf, cloud = TRUE)
vcloud_pm25 <- variogram(object = g2, cutoff=Inf, cloud = TRUE)
vco <- plot(vcloud_o3, main = "Variogram Cloud Ozono",
     xlab = "Distancia",
     ylab = "Disimilitud")
vcp <- plot(vcloud_pm25, main = "Variogram Cloud PM2.5",
     xlab = "Distancia",
     ylab = "Disimilitud")

grid.arrange(vco, vcp, ncol = 2)
```



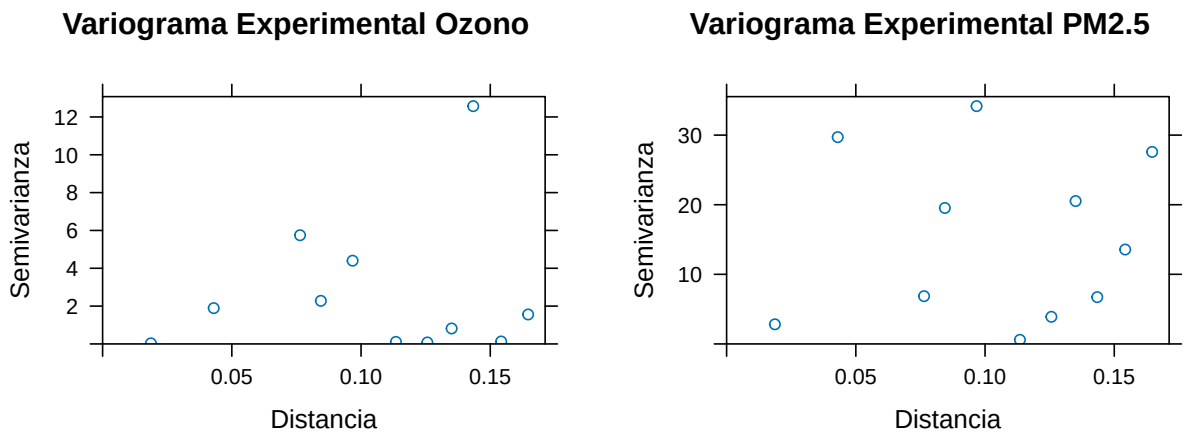
Calcular Variograma Experimental

Luego, se calculan los variogramas experimentales para ambos objetos. Es importante comprender que el variograma experimental agrupa los pares de puntos en intervalos de distancia (llamados *lags*) y calcula la semivarianza promedio dentro de cada intervalo. Básicamente, este cálculo simplifica la información del variograma nube, proporcionando una representación más general y manejable, sobre la cual se facilita el ajuste de un variograma teórico. En principio, en distancias pequeñas es común observar una semivarianza baja debido al supuesto de dependencia espacial (valores cercanos tienden a ser más similares). A medida que aumenta la distancia, la semivarianza suele crecer hasta estabilizarse en un valor llamado *sill* (representa la variabilidad total). De este modo, los parámetros que permiten modelar un variograma son los siguientes:

- *Range*: Distancia máxima en la que aún se observa dependencia espacial.
- *Nugget*: Variabilidad a distancias muy pequeñas, atribuida a errores de medición o variabilidad no capturada.
- *Sill*: Valor donde la semivarianza se estabiliza, representando la variabilidad total de los datos.

Asimismo, se generan dos gráficos de variograma experimental, los cuales resultan fundamentales para ajustar un modelo teórico de variograma, prerequisito para realizar una interpolación espacial mediante un *kriiging*.

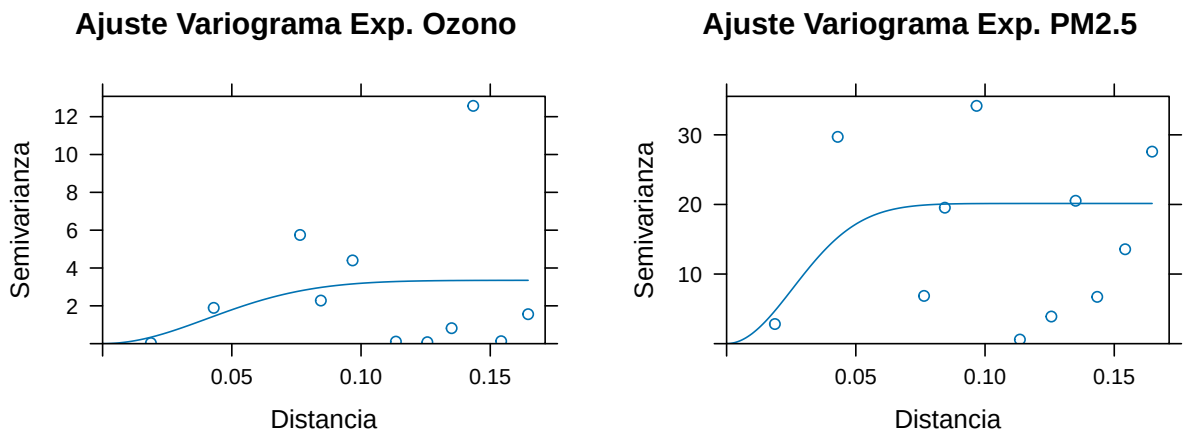
```
## Calcular variograma experimental ----
vemp_o3 <- variogram(object=g1)
vemp_pm25 <- variogram(object=g2)
veo <- plot(vemp_o3, main = "Variograma Experimental Ozono",
     xlab = "Distancia",
     ylab = "Semivarianza")
vep <- plot(vemp_pm25, main = "Variograma Experimental PM2.5",
     xlab = "Distancia",
     ylab = "Semivarianza")

grid.arrange(veo, vep, ncol = 2)
```



Ajustar Variograma Teórico

En este paso se ajusta un modelo de variograma teórico a los datos del variograma experimental de ambos contaminantes. Por su parte, el argumento `model = vgm(c("Mat", "Sph", "Exp", "Gau", "Lin"))` especifica los modelos que se probarán para ajustar el variograma. Los modelos propuestos son: Matérn (`Mat`), esférico (`Sph`), exponencial (`Exp`), gaussiano (`Gau`) y lineal (`Lin`). El argumento `fit.kappa = FALSE` indica que no se ajustará el parámetro K en el modelo Matérn, fijando su valor. Luego, se realiza un ajuste de los parámetros del modelo a los datos del variograma experimental para encontrar el mejor modelo que represente la variabilidad espacial. Adicionalmente, el código asegura que el *range* del variograma ajustado no sea negativo, lo cual no tiene sentido físico. Así, si el valor del *range* ajustado es negativo, se asigna un valor muy pequeño.

```
## Ajustar variograma teórico ----
vfit_o3 <- fit.variogram(vemp_o3,
                         model = vgm(c("Mat", "Sph", "Exp", "Gau", "Lin")),
                         fit.kappa = FALSE)
vfit_o3$range[vfit_o3$range < 0] <- 1e-6
vao <- plot(vemp_o3, model = vfit_o3, main = "Ajuste Variograma Exp. Ozono",
     xlab = "Distancia",
     ylab = "Semivarianza")
vfit_pm25 <- fit.variogram(vemp_pm25,
                           model = vgm(c("Mat", "Sph", "Exp", "Gau", "Lin")),
                           fit.kappa = FALSE)
vfit_pm25$range[vfit_pm25$range < 0] <- 1e-6
vap <- plot(vemp_pm25, model = vfit_pm25, main = "Ajuste Variograma Exp. PM2.5",
     xlab = "Distancia",
     ylab = "Semivarianza")

grid.arrange(vao, vap, ncol = 2)
```



Interpolación

Finalmente, el siguiente código realiza una interpolación puntual utilizando los modelos de variograma ajustados previamente para las variables `o3` y `pm25`, con el objetivo de estimar valores en ubicaciones no muestreadas basándose en los valores observados y en la estructura espacial de los datos.

Para ello, primero se actualizan los objetos `gstats` previos, los cuales ya contienen los datos de su respectivo contaminante y sus variogramas experimentales, incorporando el variograma ajustado de cada uno, el cual fue obtendo en el paso anterior. El método `predict(u_g1, newdata = interpol)` predice los valores de uno de los contaminantes en nuevas ubicaciones definidas en el conjunto de datos `interpol` (que contiene las coordenadas de los 33 edificaciones municipales de la conurbación de Santiago en el formato adecuado). Después de la predicción, los resultados se combinan con las columnas `longitud`, `latitud` y `municipio` del conjunto de datos `interpol` mediante `merge`.

```
## Actualizar objetos gstats ----
u_g1 <- gstat(g1, model = vfit_o3, id = "o3", formula = o3 ~ 1,
              locations = ~longitud + latitud, data = ejemplo)
u_g2 <- gstat(g2, model = vfit_pm25, id = "pm25", formula = pm25 ~ 1,
              locations = ~longitud + latitud, data = ejemplo)

## Interpolación puntual ----
p_g1 <- predict(u_g1, newdata=interpol)
```

```
## [using ordinary kriging]
```

```
p_g1 <- merge(p_g1, interpol[, c("longitud", "latitud", "municipio")],
              by = c("longitud", "latitud"))
head(p_g1)
```

```
##    longitud   latitud  o3.pred    o3.var    municipio
## 1 -70.51998 -33.35326 4.354890 0.9919222 Lo Barnechea
## 2 -70.53075 -33.45385 5.075897 3.5234545     La Reina
## 3 -70.54222 -33.47727 5.054350 3.2379087    Peñalolén
## 4 -70.57928 -33.59499 7.181804 0.4599008  Puente Alto
## 5 -70.58698 -33.55846 6.323020 0.9509661   La Florida
## 6 -70.59361 -33.45398 4.381669 2.8804960        Ñuñoa
```

```
p_g2 <- predict(u_g2, newdata=interpol)
```

```
## [using ordinary kriging]
```

```
p_g2 <- merge(p_g2, interpol[, c("longitud", "latitud", "municipio")],
              by = c("longitud", "latitud"))
head(p_g2)
```

```
##    longitud   latitud pm25.pred  pm25.var    municipio
## 1 -70.51998 -33.35326  32.08625 11.967565 Lo Barnechea
## 2 -70.53075 -33.45385  31.05464 22.689328     La Reina
## 3 -70.54222 -33.47727  31.42589 22.386223    Peñalolén
## 4 -70.57928 -33.59499  32.42791  6.328937  Puente Alto
## 5 -70.58698 -33.55846  33.51096 15.556552   La Florida
## 6 -70.59361 -33.45398  31.48020 22.180089        Ñuñoa
```

Problemas de Ajuste

Sin lugar a dudas, para ajuste de los modelos mediante `gstats` el mayor desafío de todo el análisis consistió en el ajuste de los variogramas experimentales. En primer lugar, hallar un único código que permitiese una forma unificada de encontrar el mejor ajuste para los más de 8.000 modelos necesarios para las interpolaciones diarias no resultó sencillo, recurriendo finalmente a la función `vgm` con la especficación de un vector con diferentes modelos teóricos de variograma. Particularmente existe la duda de si la función internamente ajusta un modelo teórico con sus diferentes configuraciones de parámetros o una combinación de éstos.

También se debe señalar que, posteriormente al ajuste y las interpolaciones, se evidenció que algunas fechas concretas presentan datos predichos de contaminación que varían mínimamente o que derechamente son constantes para todas las comunas consideradas. Esto se puede deber a múltiples razones:

- En primer lugar, puede explicarse por algún error el proceso de imputación de datos. Es completamente plausible que un error en el proceso de imputación afecte los resultados. Si hay datos faltantes o erróneos que se imputan, esto puede generar resultados que no tienen la variabilidad esperada, afectando la interpolación.
- Puede ser que la combinación de los modelos teóricos propuestos no sea la correcta o que la interacción de éstos provoque algunos resultados no esperados. Como hemos visto, el *kriging* depende fuertemente del modelo de variograma que se elija. Si el modelo teórico seleccionado no representa adecuadamente la estructura espacial de los datos, los resultados de la interpolación pueden no ser confiables, generando, por ejemplo, predicciones constantes o con poca variabilidad.
- Se podría deber a que el forzar la no negatividad de los valores del parámetro *range* pueda estar provocando problemas en la interpolación, aunque esta sería más bien la expresión superficial de un problema previo y no la razón última. El parámetro *range* en el contexto de un modelo de variograma es crucial porque define la distancia hasta la cual los puntos tienen una correlación significativa. Si se fuerza la no-negatividad de este parámetro, puede estar ocurriendo que el ajuste no sea adecuado para algunos datos, llevando a valores que no reflejan correctamente la estructura espacial de los datos. Este tipo de ajuste podría generar anomalías en las interpolaciones.
- Finalmente, otra posible explicación radica en que la misma naturaleza de los datos podría no ser consecuente con el principio básico de un *kriging* que se sustenta en que la función desconocida que se intenta predecir debiese respetar que entre dos puntos muy cercanos se debería obtener una diferencia menor, es decir, valores similares en la función subyacente. Cambios muy agudos en los valores de las concentraciones atmosféricas de los contaminantes

de dos coordenadas cercanas, sumado a menores diferencias con locaciones distantes, puede romper con el supuesto ya señalado, provocando que el parámetro *sill* se alcance a cortas distancias, pudiendo explicar tanto las interpolaciones constantes en el plano como valores negativos del *range*.

Sin lugar a dudas esta última opción puede llegar a ser la que tenga consecuencias más graves, puesto que si la naturaleza de los datos indica que no se cumple la suposición de que los valores en puntos cercanos son más similares que los valores en puntos lejanos, estaríamos ante un caso en que los datos presentan cambios abruptos en las concentraciones de contaminantes en distancias cortas, violándose el supuesto de *stationarity* (homogeneidad de los datos en el espacio). En este caso, la variabilidad de los datos no seguiría un patrón predecible según el modelo de variograma, lo que podría generar interpolaciones erróneas. Así, el ajuste del parámetro *sill* (el valor máximo de semivarianza) en distancias cortas, como se menciona, puede ser un indicio de que el modelo no está capturando correctamente la variabilidad espacial.

Solución de Ajuste con `automap`

Al explorar los resultados de las interpolaciones obtenidos por medio del procesamiento especificado en el apartado anterior de forma masiva a los datos de contaminación, se identificaron varios casos en los que las predicciones generadas por el modelo presentaban una muy baja variabilidad. Específicamente, se observó que el ajuste del variograma experimental daba lugar, en algunos casos, a una recta sin pendiente, lo que indicaba ausencia de variabilidad espacial y una interpolación idéntica para todos los puntos no observados.

Para dimensionar la frecuencia de estos casos problemáticos de ajuste, se identificaron las fechas en que el rango de las interpolaciones fuese igual a 0, es decir, donde no existía diferencia entre el valor máximo y mínimo predicho. Así, los resultados mostraron que estos casos no son aislados. De las 4.383 fechas diferentes, se observaron 824 fechas para O_3 y 499 fechas para $PM_{2.5}$ en las que ocurrió esta situación, vale decir, un 18,8% y 11,4% respectivamente.

Tras revisar aleatoriamente varios de estos casos, se corroboró que, aunque existe variabilidad en las mediciones. Sin embargo, en algunos casos se podría estar violando el principio de autocorrelación espacial, ya que los valores más cercanos no siempre son más similares, lo que podría producir problemas con ciertos supuestos y comportamiento esperado de los parámetros. Esta situación planteó la necesidad de realizar ajustes más detallados y de forma manual para cada caso, lo cual resultaba inviable debido al volumen de los casos.

Como ya se mencionó, inicialmente, se forzó un ajuste masivo por medio del paquete `gstat`. Ahora bien, este paquete no dispone de una función que permita un ajuste automático y masivo de modelos de para variogramas experimentales. La solución encontrada, consistió en proporcionar un listado de modelos teóricos para variograma a la función `fit.variogram`, suponiendo que seleccionaría el con mejor ajuste. Sin embargo, internamente la función no operaba así, sino que realizaba una integración de todos estos modelos teóricos, siendo relevante incluso el orden en que se enlistaban y sus posibles combinaciones, lo que limitó seriamente la capacidad de optimización automatizada.

Para superar esta limitación, se intentó desarrollar un código que evaluara múltiples modelos y seleccionara el mejor según la Suma de Cuadrados de los Residuos (SSR). Sin embargo, este enfoque se vio obstaculizado por fallos en la convergencia de algunos modelos, lo que interrumpía la ejecución del código.

Finamente, se exploró el paquete `automap`, que ofrece funciones optimizadas para seleccionar automáticamente el mejor modelo de variograma dentro de un conjunto predefinido y realizar interpolaciones mediante kriging, incluso optando por una modalidad que incorpora validación cruzada. Este paquete representa una ventaja significativa en términos de simplicidad y rapidez, especialmente cuando se requiere ajustar múltiples variogramas e interpolar valores de manera eficiente. Mientras que `gstat` sigue siendo preferible cuando se necesita un mayor control y flexibilidad sobre los parámetros para realizar análisis más complejos.

En vista de este escenario, se optó por proceder con `automap`, tomando el caso de una única fecha para luego escalarlo a la totalidad de los datos. Así, el primer paso fue el filtrado y preparación de los datos.

```
# Filtrar y preparar datos
ejemplo <- combined_data %>%
  filter(fecha == as.Date("2010-06-10")) %>%
  filter(!is.na(o3) & !is.na(pm25))
coordinates(ejemplo) <- ~longitud + latitud
coordinates(interpol) <- ~longitud + latitud
```

A continuación, y a diferencia de `gstats`, no se deben hacer manualmente los pasos de calcular la nube de variograma, el variograma experimental y su ajuste. Simplemente, con la función `autofitVariogram`, a la cual se le entrega la fórmula y los datos, además de señalar la distancia mínima (*lag*) con la que debe calcular el variograma experimental (argumento que se mantuvo en el valor por defecto).

Un aspecto muy interesante consiste en setear el argumento `verbose` en `TRUE`, permitiendo tener un detalle del proceso de selección del mejor ajuste posible para los datos proporcionados, como se puede apreciar a continuación. En la salida, primero se indican los modelos que se descartan por tener parámetros no razonables, luego se prueban los modelos razonables, se señala el modelo con mejor ajust (menor SSerror) y se resumen sus parámetros estimados:

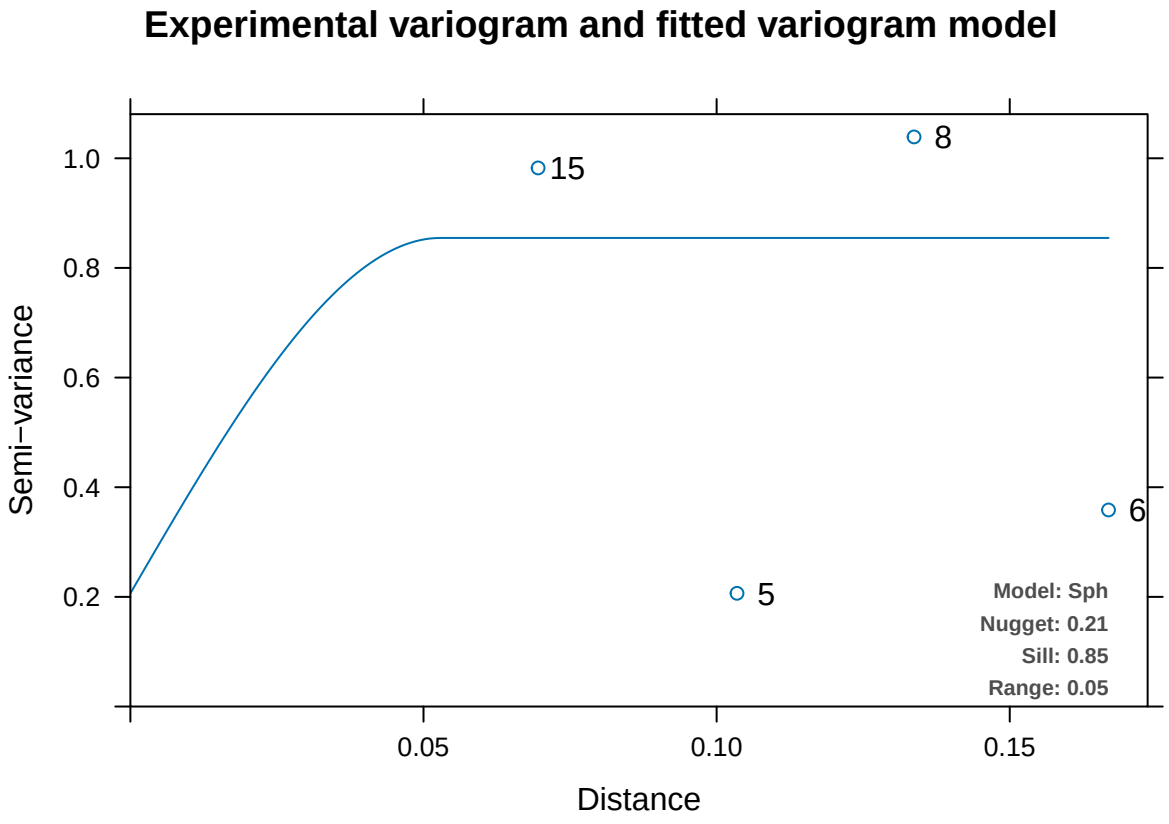
```
# Calcular y ajustar variograma para o3
variogram_o3 <- autofitVariogram(o3 ~ 1, ejemplo, verbose = TRUE)
```

```
## Checking if any bins have less than 5 points, merging bins when necessary...
##
## Selected:
##   model     psill      range
## 1   Nug 0.2065182 0.00000000
## 2   Sph 0.6481692 0.05286926
##
## Tested models, best first:
##    Tested.models kappa  SSerror
## 1            Sph     0 315.0117
## 4            Ste  0.05 341.6320
## 25           Ste    10 344.1809
## 24           Ste     5 347.9715
## 23           Ste     2 356.2282
```

```
## 22           Ste   1.9 356.8180
## 21           Ste   1.8 357.4249
## 20           Ste   1.7 358.1076
## 19           Ste   1.6 358.8270
## 18           Ste   1.5 359.6190
## 17           Ste   1.4 360.4731
## 16           Ste   1.3 361.4070
## 15           Ste   1.2 362.4217
## 14           Ste   1.1 363.5269
## 13           Ste     1 364.7235
## 12           Ste   0.9 366.0085
## 5            Ste   0.2 366.7260
## 11           Ste   0.8 367.3641
## 10           Ste   0.7 368.7475
## 9            Ste   0.6 370.0678
## 6            Ste   0.3 370.7202
## 8            Ste   0.5 371.1437
## 7            Ste   0.4 371.6132
## 3            Gau     0 418.0962
## 2            Exp     0 437.4684
```

Posteriormente, con la función `plot` se genera el gráfico del variograma experimental ajustado con el modelo seleccionado en el paso previo, especificándose (en la esquina inferior izquierda) toda la información referida al modelo:

```
plot(variogram_o3)
```



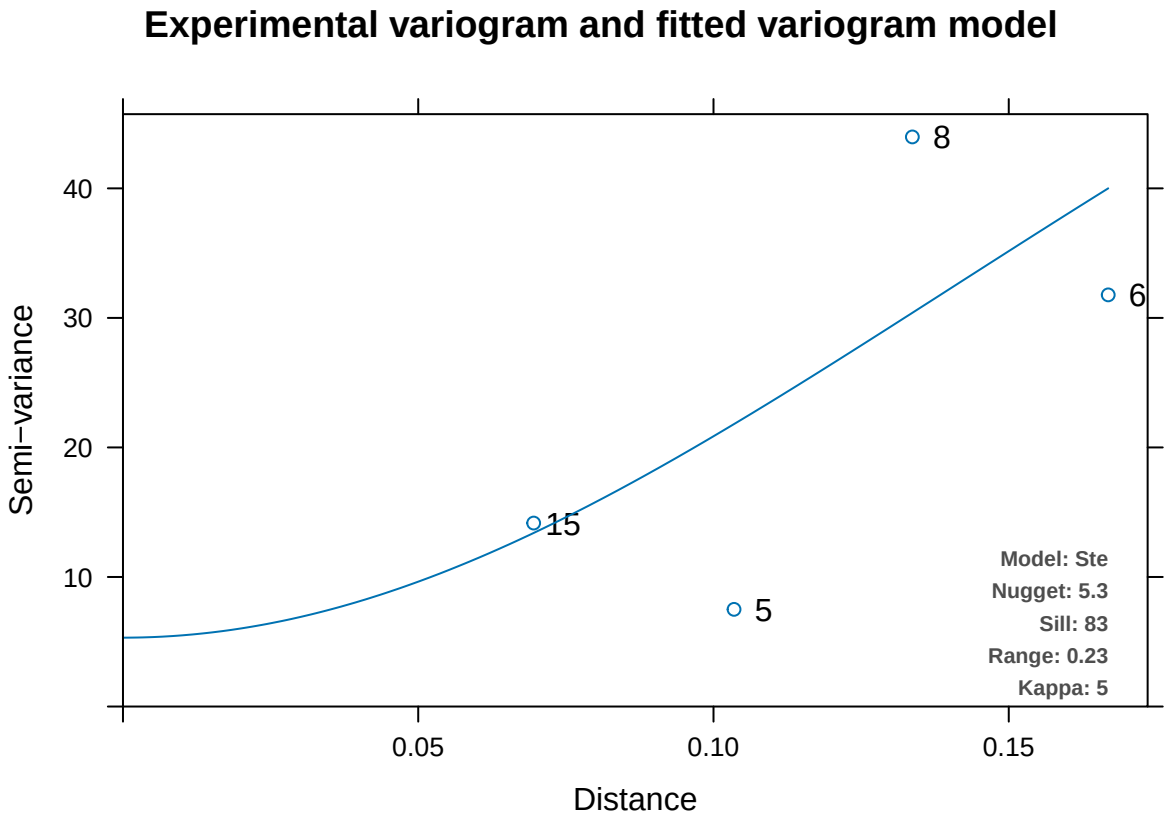
Luego de realizar el procesamiento para el caso de O_3, se realiza el mismo procedimiento con el ejemplo de $PM_{2.5}$.

```
# Calcular y ajustar variogramas para pm25
variogram_pm25 <- autofitVariogram(pm25 ~ 1, ejemplo, verbose = TRUE)
```

```
## Checking if any bins have less than 5 points, merging bins when necessary...

## [[1]]
##   model    psill        range kappa
## 1   Nug 21.71911  0.000000000     0
## 2   Ste 12.59695 -0.007648993    10
##
## ^^^ ABOVE MODELS WERE REMOVED ^^^

## Selected:
##   model     psill     range kappa
## 1   Nug  5.314901 0.0000000     0
## 2   Ste 77.572510 0.2323451     5
##
## Tested models, best first:
##    Tested.models kappa   SSerror
```

```
## 24           Ste     5  194454.8
## 21           Ste   1.8  195560.9
## 20           Ste   1.7  195656.8
## 22           Ste   1.9  195657.3
## 19           Ste   1.6  195762.0
## 18           Ste   1.5  195878.6
## 17           Ste   1.4  196008.0
## 16           Ste   1.3  196151.5
## 23           Ste     2  196278.3
## 15           Ste   1.2  196313.0
## 3            Gau     0  196425.4
## 14           Ste   1.1  196494.7
## 13           Ste     1  196700.5
## 12           Ste   0.9  196934.6
## 11           Ste   0.8  373115.4
## 2            Exp     0  589292.2
## 10           Ste   0.7  646729.6
## 9            Ste   0.6  760543.5
## 8            Ste   0.5  839429.6
## 7            Ste   0.4  909755.6
## 6            Ste   0.3  978412.7
## 5            Ste   0.2 1058026.5
## 4            Ste  0.05 1276184.4
## 1            Sph     0 1520944.8
```

```
plot(variogram_pm25)
```



Finalmente, mediante la función `krige`, se realiza la interpolación usando un *ordinary kriging*. Dicha función requiere de los siguientes argumentos: la fórmula (indicando el contaminante que operará como variable dependiente), las coordenadas (tanto de los puntos observados como los que se desea predecir) y el modelo (extraído del variograma ajustado previamente).

```
# Interpolación puntual para o3 y pm25
krige_o3 <- krige(
  formula = o3 ~ 1,
  locations = ejemplo,
  newdata = interpol,
  model = variogram_o3$var_model
)
```

```
## [using ordinary kriging]
```

```
head(krige_o3)
```

```
##              coordinates var1.pred  var1.var
## 1 (-70.70297, -33.48804)  5.999258 0.7437371
## 2 (-70.72919, -33.43401)  4.350880 0.4449001
## 3 (-70.67073, -33.39636)  5.873612 0.8966196
## 4 (-70.66561, -33.55579)  6.215800 0.5910811
```

```
## 5 (-70.68969, -33.45399)  5.902823 0.8841026
## 6   (-70.635, -33.37528)  6.000249 0.9363318
```

```
krige_pm25 <- krige(
  formula = pm25 ~ 1,
  locations = ejemplo,
  newdata = interpol,
  model = variogram_pm25$var_model
)
```

```
## [using ordinary kriging]
```

```
head(krige_pm25)
```

```
##              coordinates var1.pred  var1.var
## 1 (-70.70297, -33.48804)  36.37512  7.027968
## 2 (-70.72919, -33.43401)  32.66595  6.740236
## 3 (-70.67073, -33.39636)  34.35444  8.907970
## 4 (-70.66561, -33.55579)  43.25573  8.470353
## 5 (-70.68969, -33.45399)  35.25859  6.794951
## 6   (-70.635, -33.37528)  35.17493 11.773954
```

En vista del problema de ajuste anteriormente identificado al usar `gstats`, es importante señalar que aún utilizando las funciones indicadas de la librería `automap`, de todos modos persisten algunas fechas en las cuales, dado el comportamiento de los datos de contaminación, no se logra un ajuste adecuado del variograma experimental para obtener un buen kriging. Sin embargo, la reducción de estos casos en que se tiene una variabilidad nula en las interpolaciones se redujo considerablemente. En el caso de O_3 de 824 fechas con variabilidad nula, se redujo a 226 (5.16% del total de casos). Mientras que para $PM_{2.5}$ la reducción de casos problemáticos fue de 499 a 151 (3.44%).

Interpolación Comunal

Una vez obtenido un código que permite generar interpolaciones espaciales mediante un *ordinary kriging* para ambos contaminantes, mediante la librería `automap`, se extendió y replicó el código generado para la totalidad de las fechas contempladas en el set de datos de contaminación.

Código Interpolación

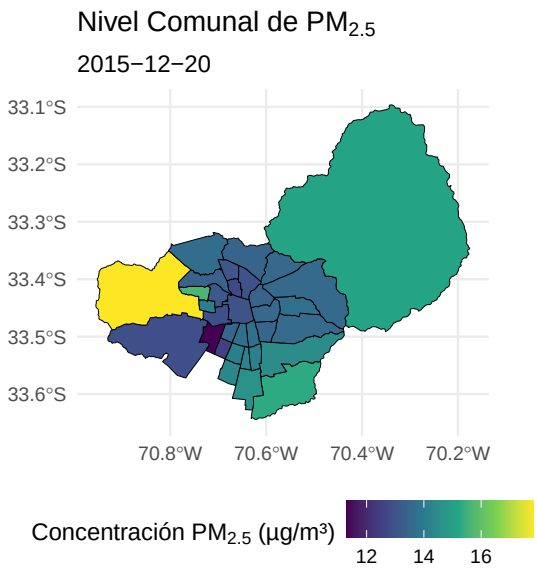
Prácticamente no existen grandes diferencias entre este código replicado y el que fue analizado en los apartados anteriores, salvo en el bucle `for` que se codifica para recorrer cada una de las fechas únicas para establecer todas las interpolaciones espaciales requeridas, además de unificar todos los resultados en dos bases de datos llamadas `interpolacion_o3` e `interpolacion_pm25`.

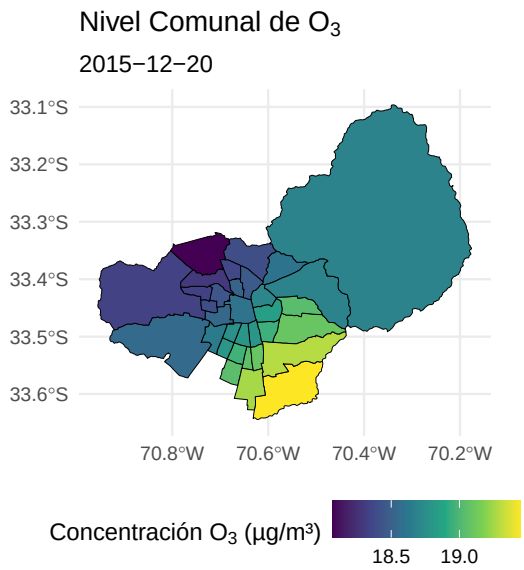
```
# Obtener las fechas únicas de mediciones
fechas_unicas <- unique(combined_data$fecha)

# Crear listas vacías para almacenar los resultados de las interpolaciones
resultados_o3 <- list()
resultados_pm25 <- list()

# Recorrer cada fecha única
for (fecha_actual in fechas_unicas) {

  # Filtrar datos para la fecha actual y remover valores NA
  ejemplo <- combined_data %>%
    filter(fecha == fecha_actual) %>%
    filter(!is.na(o3) & !is.na(pm25))

  # Asegurarse de que el objeto ejemplo tiene coordenadas
  coordinates(ejemplo) <- ~longitud + latitud

  # Calcular y ajustar variogramas usando automap
  variogram_o3 <- autofitVariogram(o3 ~ 1, ejemplo, verbose = FALSE)
  variogram_pm25 <- autofitVariogram(pm25 ~ 1, ejemplo, verbose = FALSE)

  # Interpolación para o3
  krige_o3 <- krige(
    formula = o3 ~ 1,
    locations = ejemplo,
    newdata = interpol,
    model = variogram_o3$var_model
  )

  # Interpolación para pm25
  krige_pm25 <- krige(
    formula = pm25 ~ 1,
    locations = ejemplo,
```

```
    newdata = interpol,
    model = variogram_pm25$var_model
  )

  # Convertir resultados a data.frame
  krige_o3_df <- as.data.frame(krige_o3)
  krige_pm25_df <- as.data.frame(krige_pm25)
  interpol_df <- as.data.frame(interpol)

  # Agregar la columna de fecha y realizar unión con municipios
  krige_o3 <- krige_o3_df %>%
    left_join(interpol_df, by = c("longitud", "latitud"))
  krige_o3$fecha <- fecha_actual

  krige_pm25 <- krige_pm25_df %>%
    left_join(interpol_df, by = c("longitud", "latitud"))
  krige_pm25$fecha <- fecha_actual

  # Agregar los resultados a las listas
  resultados_o3[[as.character(fecha_actual)]] <- krige_o3
  resultados_pm25[[as.character(fecha_actual)]] <- krige_pm25
}

# Combinar los resultados en tablas finales
final_o3 <- bind_rows(resultados_o3)
final_pm25 <- bind_rows(resultados_pm25)

# Cambiar formato de fecha
final_o3$fecha <- as.Date(final_o3$fecha, origin = "1970-01-01")
final_pm25$fecha <- as.Date(final_pm25$fecha, origin = "1970-01-01")

# Revisar las estructuras de los datos finales
str(final_o3)
str(final_pm25)

# Guardar los resultados
write.csv(final_o3, "interpolacion_o3.csv", row.names = FALSE)
write.csv(final_pm25, "interpolacion_pm25.csv", row.names = FALSE)
```

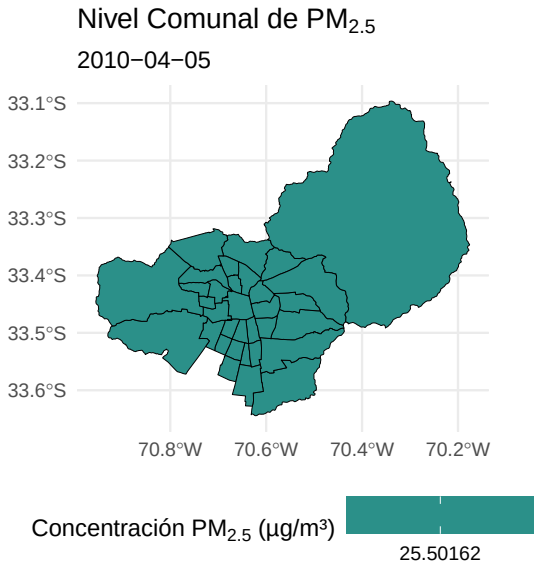
Gráficos Comunales

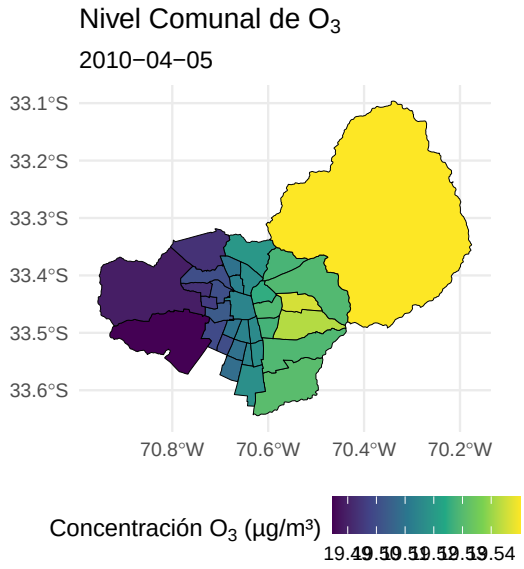
En base a las interpolaciones comunales que se generaron, se elaboró un código para graficar en una escala continua un mapa de calor de la conurbación de Santiago para cada uno de los contaminantes en una fecha específica. Así, se puede apreciar, a modo de ejemplo, una serie de dos pares de gráficos para las fechas 20 de diciembre de 2015 y 5 de abril de 2010 respectivamente.





En el par de gráficos anteriores se aprecia un comportamiento que se esperaría normal de las concentraciones de contaminantes en la conurbación de Santigo, es decir, con interpolaciones que presentan variabilidad en las comunas. Sin embargo, para ejemplificar la dificultad en el ajuste de los variogramas teóricos que se expuso anteriormente, se incorpora un segundo par de gráficos en que las concentraciones de $PM_{2.5}$ no solo no presenta variabilidad entre las comunas, sino que posee un valor constante, posiblemente por las razones previamente esbozadas.





4. Estimación de Exposición

Tal como se señaló anteriormente, el objetivo final de este proyecto no consitía simplemente en generar la interpolación espacial de los datos de contaminación del aire, sino que precisamente en utilizar dichas interpolaciones o predicciones para estimar la exposición promedio a la que cada una de las gestantes de la base de datos de nacimientos se vio expuesta durante diferentes ventanas (cuatro días antes del parto, el mes previo al parto y la duración completa de la gestación) en determinada comuna.

De este modo, se codificó una función en `R` que tomando la base de datos que contiene más de 916.000 nacimientos, asigna una exposición promedio tanto a ozono como material particulado según cada una de las tres ventanas de exposición señaladas, lo cual se puede apreciar en el código expuesto a continuación. Es importante precisar que para los casos en que los datos de contaminación no alcanzan a cubrir las ventanas gestacionales de exposición definidas, la función codificada toma únicamente las mediciones de concentración de los contaminantes para las fechas disponibles. Por lo tanto, queda pendiente eliminar aquellos casos de nacimiento cuyas ventanas de exposición no alcanzan a ser debidamente cubiertos por el periodo comprendido en el set de datos de contaminación.

```r
# Crear función de cálculo de exposición por ventana
calculate_exposure <- function(pm25, o3, births) {
  pm25$fecha <- as.Date(pm25$fecha)
  o3$fecha <- as.Date(o3$fecha)
  births$date_start_week_gest <- as.Date(births$date_start_week_gest)
  births$date_ends_week_gest <- as.Date(births$date_ends_week_gest)

  # Crear columnas para exposición de cada ventana
  births$o3_gest <- NA
  births$pm25_gest <- NA
  births$o3_mes <- NA
  births$pm25_mes <- NA
  births$o3_4d <- NA
  births$pm25_4d <- NA

  # Iterar por cada fila en births
  for (i in 1:nrow(births)) {
    # Extraer información relevante
    comuna <- births$name_com[i]
    start_date <- births$date_start_week_gest[i]
    end_date <- births$date_ends_week_gest[i]

    # Filtrar pm25 y o3 por comuna y rango de fechas
    pm25_comuna <- pm25[pm25$municipio == comuna, ]
    o3_comuna <- o3[o3$municipio == comuna, ]

    pm25_gest <- pm25_comuna[pm25_comuna$fecha >= start_date &
                               pm25_comuna$fecha <= end_date, "var1.pred"]
    o3_gest <- o3_comuna[o3_comuna$fecha >= start_date &
```

```
                          o3_comuna$fecha <= end_date, "var1.pred"]

    pm25_mes <- pm25_comuna[pm25_comuna$fecha > (end_date - 30) &
                              pm25_comuna$fecha <= end_date, "var1.pred"]
    o3_mes <- o3_comuna[o3_comuna$fecha > (end_date - 30) &
                          o3_comuna$fecha <= end_date, "var1.pred"]

    pm25_4d <- pm25_comuna[pm25_comuna$fecha > (end_date - 4) &
                             pm25_comuna$fecha <= end_date, "var1.pred"]
    o3_4d <- o3_comuna[o3_comuna$fecha > (end_date - 4) &
                         o3_comuna$fecha <= end_date, "var1.pred"]

    # Calcular promedios y asignarlos a la fila correspondiente
    births$o3_gest[i] <- mean(o3_gest, na.rm = TRUE)
    births$pm25_gest[i] <- mean(pm25_gest, na.rm = TRUE)

    births$o3_mes[i] <- mean(o3_mes, na.rm = TRUE)
    births$pm25_mes[i] <- mean(pm25_mes, na.rm = TRUE)

    births$o3_4d[i] <- mean(o3_4d, na.rm = TRUE)
    births$pm25_4d[i] <- mean(pm25_4d, na.rm = TRUE)
  }

  return(births)
}

# Inicializar una lista para almacenar los resultados
expo_list <- list()

# Iterar por cada año desde 2009 hasta 2020
for (year in 2009:2020) {
  # Filtrar nacimientos por año
  births_year <- births %>% filter(year_nac == year)

  # Calcular exposición para el año
  expo_year <- calculate_exposure(pm25, o3, births_year)

  # Almacenar los resultados en la lista
  expo_list[[as.character(year)]] <- expo_year
}

# Combinar todos los años en un solo dataframe
expo_combined <- bind_rows(expo_list, .id = "year")
write.csv(expo_combined, "exposure.csv")
```

La base obtenida, `exposure`, contiene tanto la totalidad la información que ya existía previamente en el set de datos `births` como seis nuevos campos, correspondientes a las tres ventanas de exposi-

ción señaladas para cada uno de los dos contaminantes contemplados en el estudio. Sin embargo, a continuación, a modo de ejemplo, se presenta una selección de variables del set `exposure` para dos muestras aleatorias de nacimientos para O_3 y $PM_{2.5}$ de tamaño 10.

```
##             name_com   date_nac weeks     o3_4d    o3_mes  o3_gest
## 162661     Quilicura 2010-09-02    40  9.538638 11.661597 15.39343
## 477598 Independencia 2014-12-28    39 23.118714 19.127080 11.62018
## 774731      Santiago 2018-08-27    39 13.079635 10.826956 15.44218
## 133140   Cerro Navia 2010-09-10    40 16.917982 13.581921 15.22396
## 338950      Recoleta 2013-10-17    39 24.022181 20.191327 13.74884
## 369042      La Reina 2013-10-27    39 21.750858 20.070581 13.60608
## 206280         Nunoa 2011-06-03    36  7.561926 10.133178 19.30097
## 681296      Santiago 2017-06-03    37  6.141746  6.559507 14.90902
## 26176     La Pintana 2009-09-06    38  9.757177 10.981929 15.80830
## 281943         Maipu 2012-07-04    38  9.353961  5.648899 16.14303
```

```
##                   name_com   date_nac weeks   pm25_4d pm25_mes pm25_gest
## 290071 Pedro Aguirre Cerda 2012-03-09    40 24.707974 18.63106  24.06715
## 664129          La Florida 2017-05-22    33 42.923914 29.83716  20.83439
## 550717         Puente Alto 2015-01-30    38 16.025597 20.43684  28.07683
## 237783    Estacion Central 2011-03-07    40 17.671600 19.43357  23.12978
## 776172         Puente Alto 2018-02-21    37  9.903228 16.86382  22.45074
## 106509            Recoleta 2010-09-15    40 19.541038 21.76145  28.19565
## 441927          Las Condes 2014-05-20    41 39.192585 36.52217  21.94265
## 698259            Pudahuel 2017-01-05    40 26.322941 23.16496  32.27448
## 557998       Quinta Normal 2015-03-12    38 30.715864 23.57301  24.98828
## 286843         Puente Alto 2012-12-04    40 20.597012 21.04011  27.20244
```