

Trabajo Práctico final Estadística Espacial

2025-07-13

Análisis Calidad del aire en Estados Unidos

Maria Belén Alvariñas

```
suppressPackageStartupMessages({  
  library(tidyverse)  
  library(terra)  
  library(patchwork)  
  library(sf)  
  library(spdep)  
  library(viridis)  
  library(raster)  
})  
library(tidyverse)  
library(terra)  
library(sf)  
library("leaflet")  
library(tmap)  
library(OpenStreetMap)  
library(gstat)  
library(lattice)  
library(sp)  
library(patchwork)  
library(gstat)  
library(mapview)  
library(spdep)  
library(rnaturalearth)  
library(viridis)  
library(raster)  
library(whitebox)  
options(warn=-1)
```

Introducción

En este trabajo, se analizará el contaminante P.M 2.5, es decir, las partículas que se encuentran en el aire con diámetro de 2.5 micrómetros o menos. Estas partículas pueden ser dañinas para la salud humana y provienen de diversas fuentes como vehículos, fábricas y quemas. Debido a su pequeño tamaño, pueden permanecer suspendidas en el aire por períodos prolongados y pueden ser inhaladas profundamente en los pulmones, donde pueden causar problemas graves de salud.

Se analizará este contaminante a lo largo de Estados Unidos, con datos tomados en distintas estaciones que miden la concentración de este contaminante, entre otros. Se cuenta con 1430 mediciones, que luego de limpiarlas serán 923.

Estos datos provienen del Programa de Protección Ambiental de Estados Unidos (<https://www.epa.gov/>) y abarcan todo el año 2022.

```
usa <- read_csv("PM25USA2022.csv")
```

```
## Rows: 1429 Columns: 3
## -- Column specification -----
## Delimiter: ","
## dbl (3): longitude, latitude, value
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
head(usa)
```

```
## # A tibble: 6 x 3
##   longitude latitude value
##   <dbl>    <dbl> <dbl>
## 1    -87.9     30.5  7.95
## 2    -85.8     33.3  7.3
## 3    -86.0     34.3  7.24
## 4    -86.0     34.0  8.66
## 5    -86.8     33.6  9.75
## 6    -86.8     33.6 12.0
```

```
set.seed(344)
```

Limpio datos

```
usa <- na.omit(usa)

#mapa y coordenadas de estados unidos
mapa <- ne_countries(type = "countries",
                     country = "United States of America",
                     scale = "medium")
mapa <- st_crop(mapa, xmin = -130, xmax = -60, ymin = 18, ymax = 72)

#me quedo con los puntos que caen dentro del mapa
usa.sf.sinfiltro <- usa %>% st_as_sf(coords = c("longitude", "latitude"), crs = 4326)
usa.sf.conrep <- st_intersection(usa.sf.sinfiltro, mapa)

usa_data_no_limpia <- usa.sf.conrep %>%
  st_drop_geometry() %>%
  mutate(longitude = st_coordinates(usa.sf.conrep)[, "X"],
         latitude = st_coordinates(usa.sf.conrep)[, "Y"])

#saco repetidos
```

```
usa_data<- usa_data_no_limpia %>%
  group_by(longitude, latitude) %>%
  summarise(
    value = mean(value, na.rm = TRUE),
    .groups = 'drop'
  )

usa.sf <- usa_data %>% st_as_sf(coords = c("longitude", "latitude"), crs = 4326)
```

Para tener un análisis adecuado, filtro los datos, quedandome con los que se encuentran dentro de los límites de Estados Unidos. Elimino NA y repetidos, y convierto el data frame en un objeto espacial .sf.

Análisis exploratorio de los datos

Concentración P.M 2.5

```
mapview(usa.sf, zcol = "value", map.types = "CartoDB.Voyager", main = "Concentración de PM 2.5")
```

```
## file:///tmp/RtmpdboP05/file3b8924f62d99/widget3b893412bf9.html screenshot completed
```



Your file couldn't be accessed

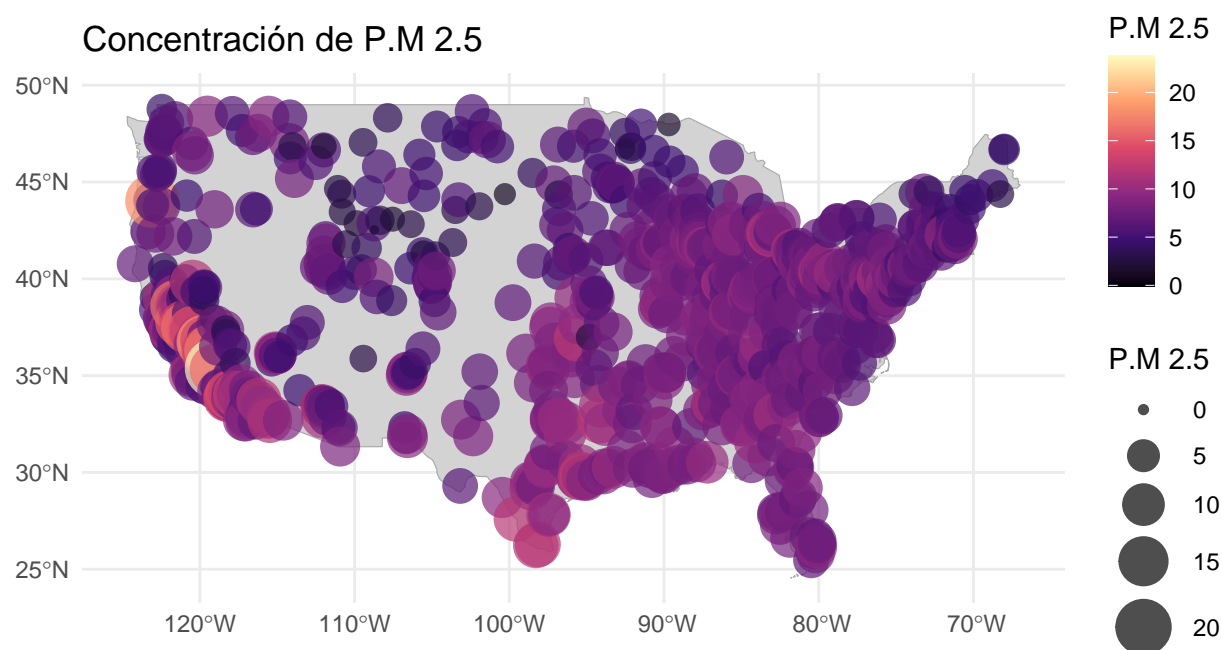
It may have been moved, edited, or deleted.

ERR_FILE_NOT_FOUND

Se puede observar como se distribuyen las diferentes mediciones a lo largo del mapa. Tenemos mayor cantidad de mediciones en la zonas este y oeste, a diferencia del centro del país.

Sobre las concentraciones se puede distinguir al este del país (zona california) la mayor concentración del contaminante, alejándose ligeramente de la costa. En contraste, en el norte del país se ven los valores más bajos de concentración. Mientras que, al este del país se observan, en general, valores medios.

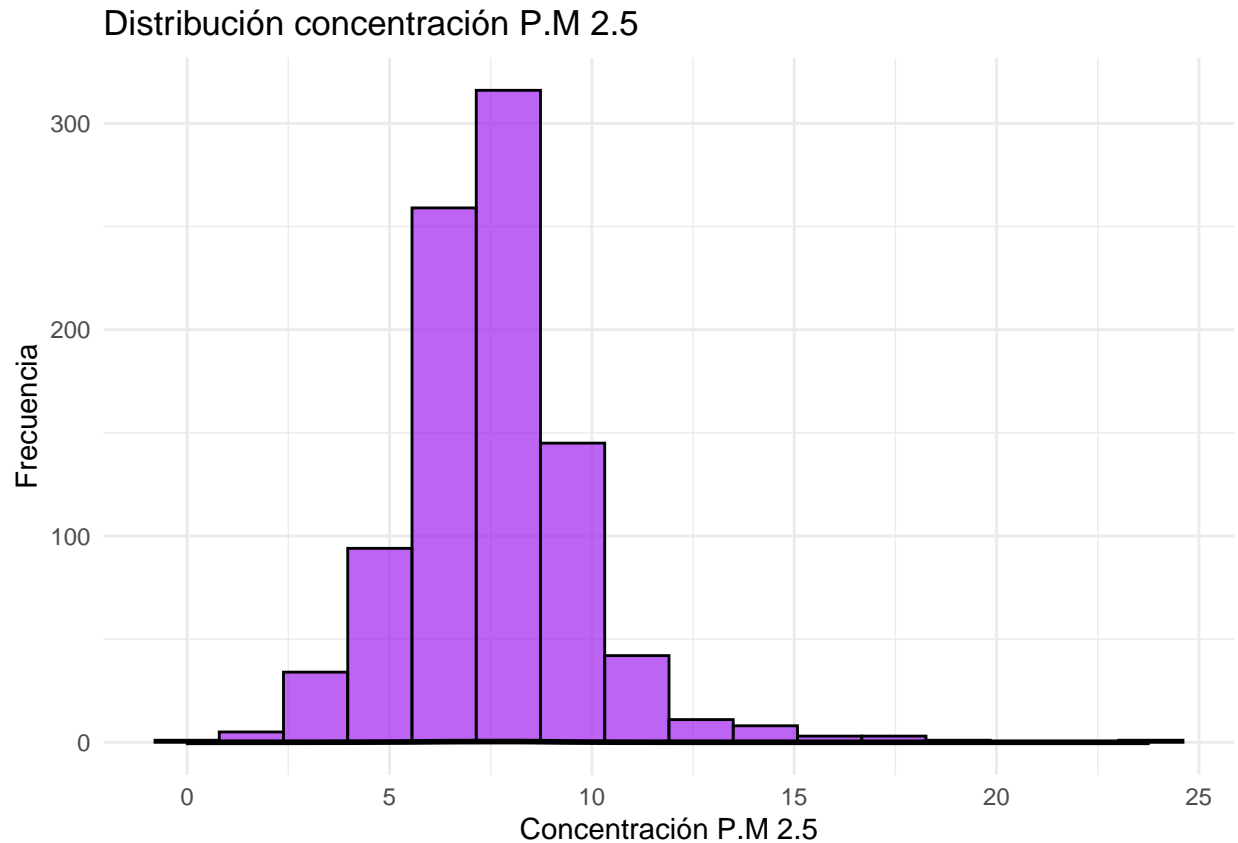
```
ggplot() +
  geom_sf(data = mapa, fill = "lightgray", color = "darkgray") +
  geom_sf(data = usa.sf, aes(size = value, color = value), alpha = 0.7) +
  scale_size_continuous(range = c(1, 10), name = "P.M 2.5") +
  scale_color_viridis_c(name = "P.M 2.5", option = "magma") +
  theme_minimal() +
  labs(title = "Concentración de P.M 2.5")
```



En este gráfico se confirma lo observado en el mapa anterior y se observa también presencia de valores elevados al sur del país (aunque no tanto como en el este).

Distribución del contaminante

```
ggplot(usa_data, aes_string(x = "value")) +
  geom_histogram(binwidth = (max(usa_data[["value"]], na.rm = TRUE) - min(usa_data[["value"]], na.rm = TRUE)) / 30,
  geom_density(aes(y = after_stat(density * (max(usa_data[["value"]], na.rm = TRUE) - min(usa_data[["value"]], na.rm = TRUE)) / 30),
  labs(title = "Distribución concentración P.M 2.5",
    x = "Concentración P.M 2.5", y = "Frecuencia") +
  theme_minimal()
```



Para seguir comprendiendo los datos, se puede observar que la mayoría de observaciones tienen una concentración entre 5 y 10, sin mucha variación entre los datos.

Indice de Moran para P.M 2.5

Este test estadístico permite analizar la autocorrelación espacial entre las distintas regiones del espacio, en este caso a lo largo de Estados Unidos. Para este test, se propone como Hipótesis Nula, que no existe una autocorrelación espacial, es decir, que el proceso no depende del espacio y se dió de forma homogénea a lo largo del espacio de interés.

Analizandolo, desde el sentido común, lo que se esperaría es que el proceso no sea homogéneo debido a que, la concentración de un contaminante, depende de las características del espacio. Es decir, lo que ocurriría sería rechazar la Hipótesis Nula.

```
#creo la matriz de pesos
cool<-st_coordinates(usa.sf)
distan<-(dist(cool))^0.5
w <- 1/as.matrix(distan)
w[w==Inf]<-0
diag(w) <- 0

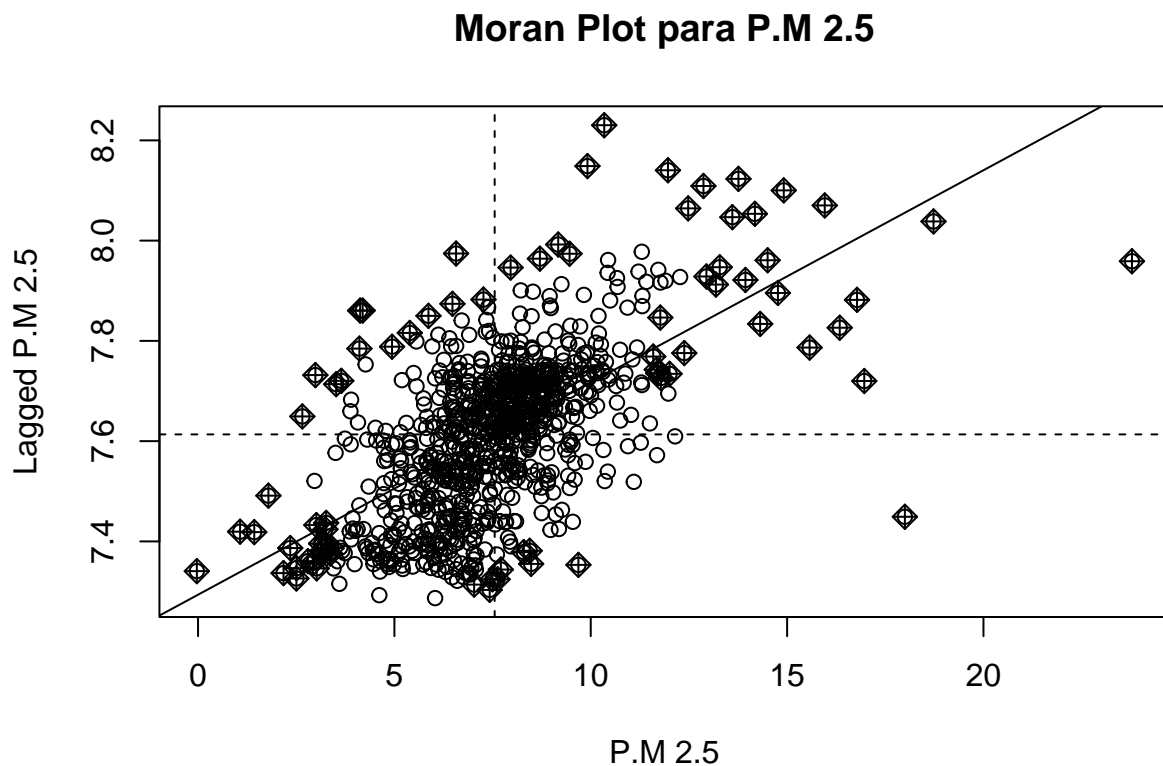
# test de moran
eltest<-moran.test(usa.sf$value,mat2listw(w,style = "W"),randomisation = TRUE)
eltest
```

```
##
```

```
## Moran I test under randomisation
##
## data: usa.sf$value
## weights: mat2listw(w, style = "W")
##
## Moran I statistic standard deviate = 40.396, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      4.233581e-02      -1.084599e-03      1.155356e-06
```

Sobre lo obtenido, tenemos un valor de estadístico positivo, junto a un p-valor muy pequeño, menor a $2.2e-16$. De estos dos valores, lo que se interpreta es que la probabilidad de observar los valores obtenidos, en el caso que el proceso fuera homogéneo, es muy pequeña. Por lo que, se rechaza la Hipótesis Nula, y se asume que existe una autocorrelación espacial.

```
moran.plot(usa.sf$value, mat2listw(w, style = "W"), labels = FALSE, xlab = "P.M 2.5", ylab = "Lagged P.M
```



El gráfico de Moran muestra la relación entre la concentración de P.M 2.5 y la de sus vecinos. Se observa que los puntos están alineados cerca de la diagonal, con una pendiente positiva, lo que indica una autocorrelación espacial positiva, es decir, que las concentraciones de P.M 2.5 son similares entre áreas cercanas.

Interpolaciones

Una de las mayores dificultades, a la hora de encontrar un dataset para el objetivo propuesto en este trabajo, fue encontrar uno que contenga una buena cantidad de datos distribuidos a lo largo del espacio. Al ser tomados por ciertas estaciones, dependiendo el país, pueden ser muy limitados. Por esto mismo, decidí realizar las técnicas de interpolación vistas en clase: Kriging, Inverse Distance Weighting y Natural Neighbour Interpolation.

Kriging

- Es una técnica de interpolación espacial que se utiliza para predecir valores en ubicaciones no muestreadas
- Se considera una extensión de los modelos lineales, pero los errores no son independientes; tienen autocorrelación espacial.
- Se utiliza el variograma para modelar cómo la correlación entre los datos cambia con la distancia y así asignar los pesos adecuados a cada punto de muestreo, estos pesos asignan más importancia a los datos más cercanos y correlacionados con el punto de predicción
- Universal Kriging (media no cte.)

$$Z(s) = X(s) * \beta + \epsilon(s)$$

```
usa_data$longitudo <- as.numeric(as.character(usa_data$longitudo))
usa_data$latitudo <- as.numeric(as.character(usa_data$latitudo))
```

```
#creo la grilla y la transformo en objeto espacial
```

```
grados <- 0.5
```

```
grid1 <- st_make_grid(mapa, cellsize = grados, what = "centers") %>%
  st_as_sf()
```

```
grid_coords <- st_intersection(grid1,mapa)
```

```
grid_coord <- st_coordinates(grid_coords) %>%
  as.data.frame()
colnames(grid_coord) <- c("longitudo", "latitudo")
```

```
grid2 <- grid_coord
coordinates(grid2) <- ~longitudo + latitudo
proj4string(grid2) <- CRS("+init=epsg:4326")
```

```
#variograma
```

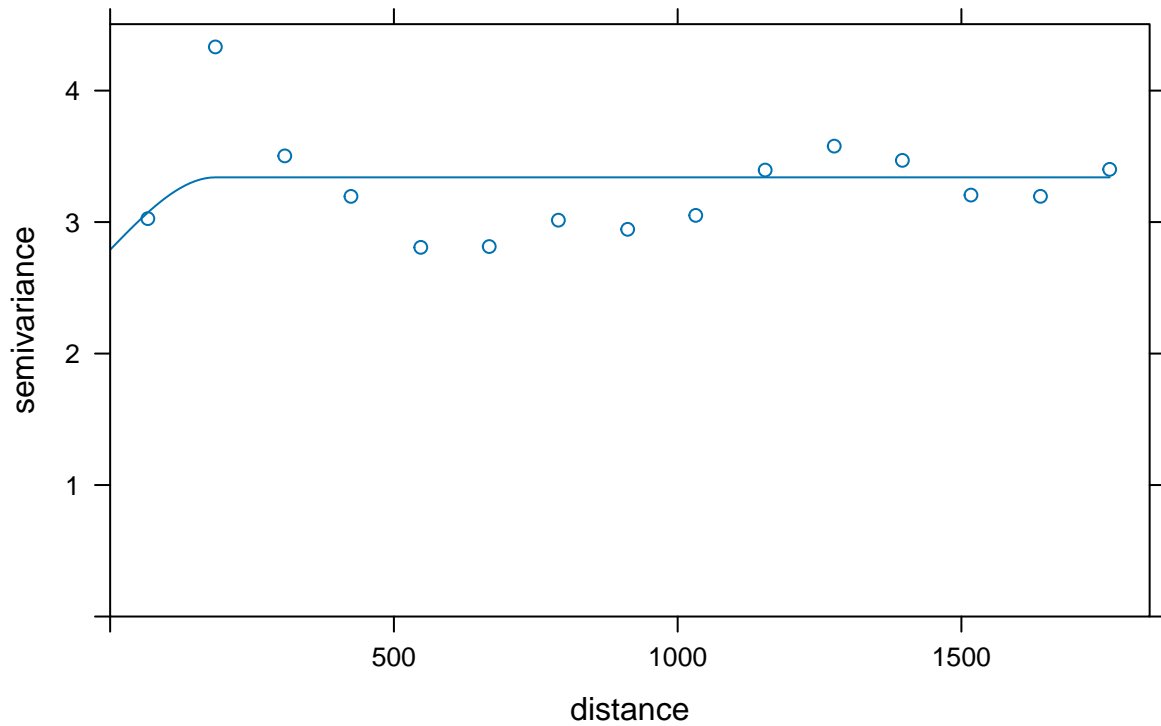
```
usa.spdf <- usa_data
coordinates(usa.spdf) <- ~longitudo +latitudo
proj4string(usa.spdf) <- CRS("+init=epsg:4326")
```

```
variograma <- variogram(value ~ poly(longitudo, 2) + poly(latitudo, 2), usa.spdf)
#plot(variograma, main = "variograma")
```

```
modelo_variograma <- fit.variogram(variograma, model = vgm(psill = 0.5, "Sph", range = 1000, nugget =3))
```

```
plot(variograma, modelo_variograma, main = "variograma ajustado")
```


variograma ajustado



Para encontrar un buen ajuste del variograma, probé con distintos modelos e intenté suavizar la componente espacial utilizando la función `poly`, usando un polinomio de grado 2. Sin embargo, no logré un gran ajuste.

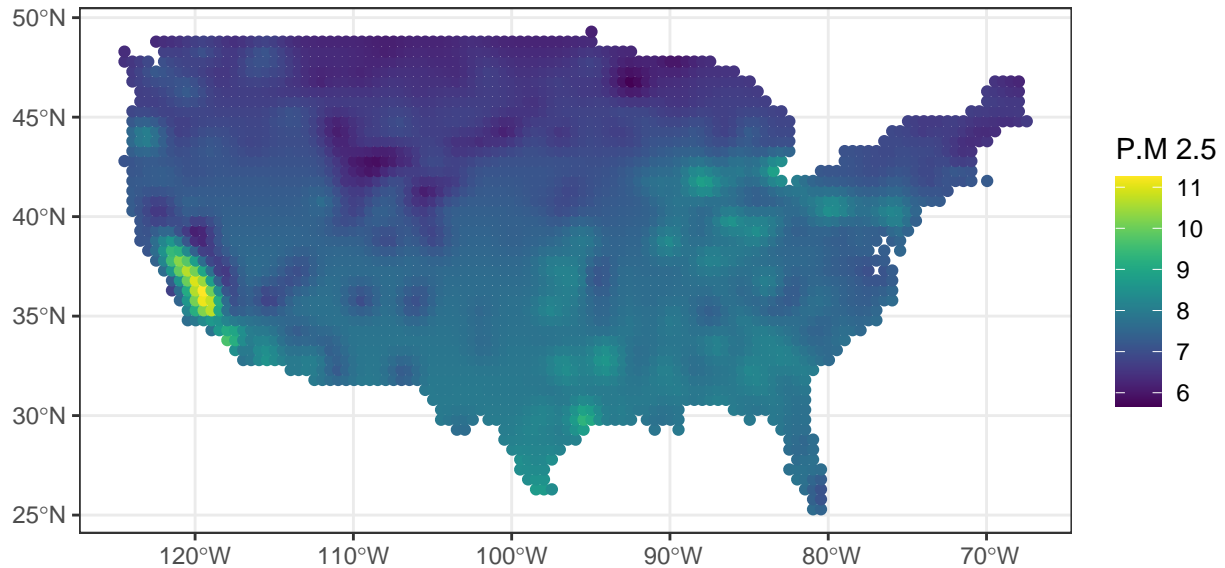
```
#interpolacion kriging
kriging_resultado <- krige(formula = value ~ poly(longitude, 2) + poly(latitude, 2), locations = usa.sp

## [using universal kriging]

kriging_sf <- as.data.frame(kriging_resultado) %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326)

#grafico kriging
ggplot() +
  geom_sf(data = kriging_sf, aes(color = var1.pred)) +
  #geom_sf(data = usa.sf) +
  scale_color_viridis(name = "P.M 2.5") + theme_bw() + labs(title="Interpolación Kriging P.M 2.5")
```

Interpolación Kriging P.M 2.5



```
#grafico kriging como raster
grid2$value_krige <- kriging_sf$var1.pred
r_template <- raster(ext = extent(grid2), resolution = 0.55, crs = proj4string(grid2))
raster_kriging <- rasterize(x = grid2, y = r_template, field = "value_krige", fun = mean)

n_colors <- 100
viridis_palette <- viridisLite::viridis(n_colors)

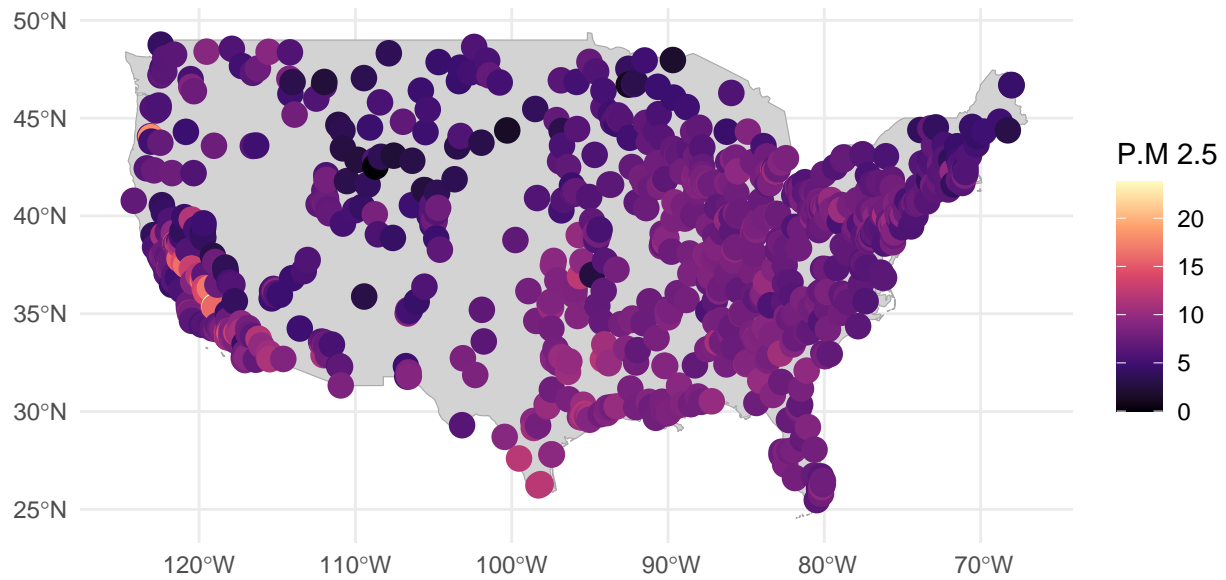
raster_krig <- plot(
  raster_kriging,
  col = viridis_palette,
  main = 'Concentración P.M 2.5 usando Kriging',
  axes = FALSE,
  box = FALSE,
  #axis.args = axis.ls,
  legend.args = list(text = "P.M 2.5", side = 3, font = 2, line = 0.2, cex = 0.8)
)
```

Concentración P.M 2.5 usando Kriging



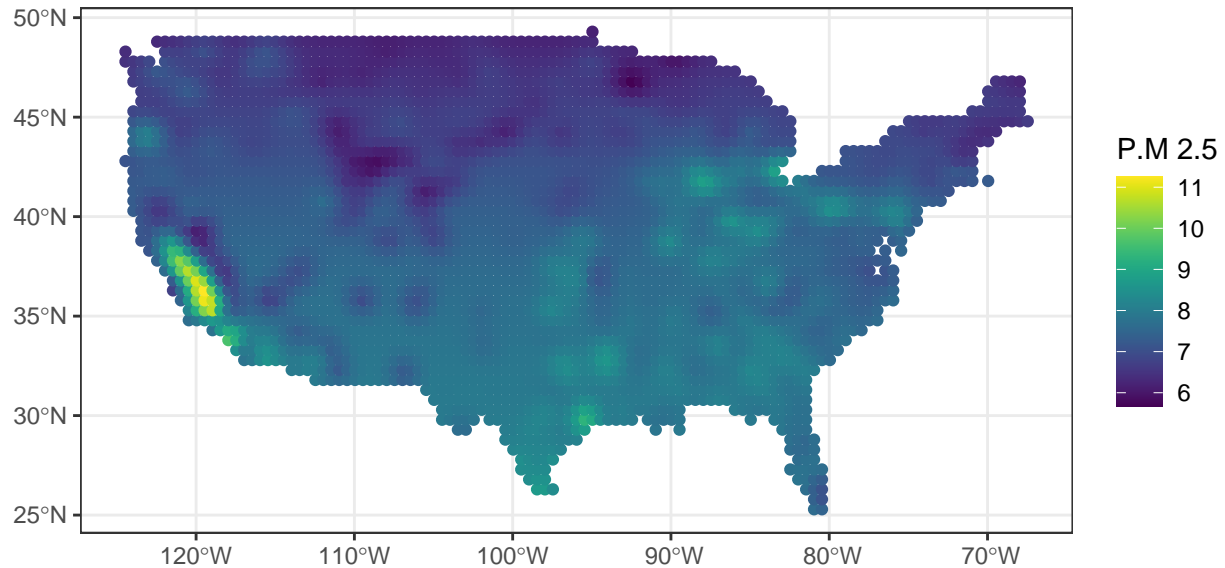
```
#grafico mis datos para comparar  
ggplot() +  
  geom_sf(data = mapa, fill = "lightgray", color = "darkgray") +  
  geom_sf(data = usa.sf, aes(color = value), size = 4, alpha = 3) +  
  scale_color_viridis_c(option = "magma", name = "P.M 2.5") +  
  theme_minimal() +  
  labs(title = "Datos originales P.M 2.5")
```

Datos originales P.M 2.5



```
ggplot() +
  geom_sf(data = kriging_sf, aes(color = var1.pred)) +
  #geom_sf(data = usa.sf) +
  scale_color_viridis(name = "P.M 2.5") + theme_bw() + labs(title="Interpolación Kriging P.M 2.5")
```

Interpolación Kriging P.M 2.5



Si bien el variograma no ajustó muy bien, observando este resultado y comparándolo con el análisis exploratorio inicial, se puede observar que en la zona este y sur parece estar captando, las concentraciones altas, al igual que al norte las bajas concentraciones, como se vió durante el análisis exploratorio de los datos. Sin embargo, se observa una variación de PM menor a los datos originales y algunos focos de altas concentraciones en la zona oeste, que no se captaban en el los datos originales. Para ver cuan buenos son los resultados, realizo cross-validation y miro R^2 , MSE y RMSE.

```
# hago cv
cv_resultado <- krige.cv(value ~ poly(longitude, 2) + poly(latitude, 2) , usa.spdf, model = modelo_vari

observado_cv <- cv_resultado$observed
predicho_cv <- cv_resultado$var1.pred

#MSE y RMSE
MSE_krig <- mean(cv_resultado$residual^2)
print(paste("MSE:", round(MSE_krig, 2)))

## [1] "MSE: 4.02"

RMSE_krig <- sqrt(mean(cv_resultado$residual^2))
print(paste("RMSE:", round(RMSE_krig, 3)))

## [1] "RMSE: 2.006"
```

```
RSQUARE = function(y_actual,y_predict){
  cor(y_actual,y_predict)^2
}

r2_krig <- RSQUARE(y_actual = observado_cv, y_predict = predicho_cv)
print(paste("R^2:", round(r2_krig, 4)))
```

```
## [1] "R^2: 0.2892"
```

El r^2 me da bastante bajo, 0.3 explica solo el 30% de la variabilidad en los datos de concentración de PM. A su vez, el MSE da cercano a 4, que teniendo en cuenta la variación de las concentraciones (entre 5-20) representa un valor bastante alto. Así mismo, el RMSE de 1.97 indica que mis predicciones se desvían de los valores reales en casi 2 unidades.

Inverse Distance Weighting (IDW)

- La estimación se basa en un promedio ponderado de los valores de los puntos cercanos.
- Los puntos más cercanos al lugar de predicción tienen mayor influencia.
- La fórmula asigna pesos inversamente proporcionales a la distancia. Lo que significa que los puntos más cercanos tienen una mayor influencia en la predicción.

$$\hat{Z}(s_0) = \frac{\sum_{i=1}^n Z(s_i) * \frac{1}{d_i^\beta}}{\sum_{i=1}^n \frac{1}{d_i^\beta}}$$

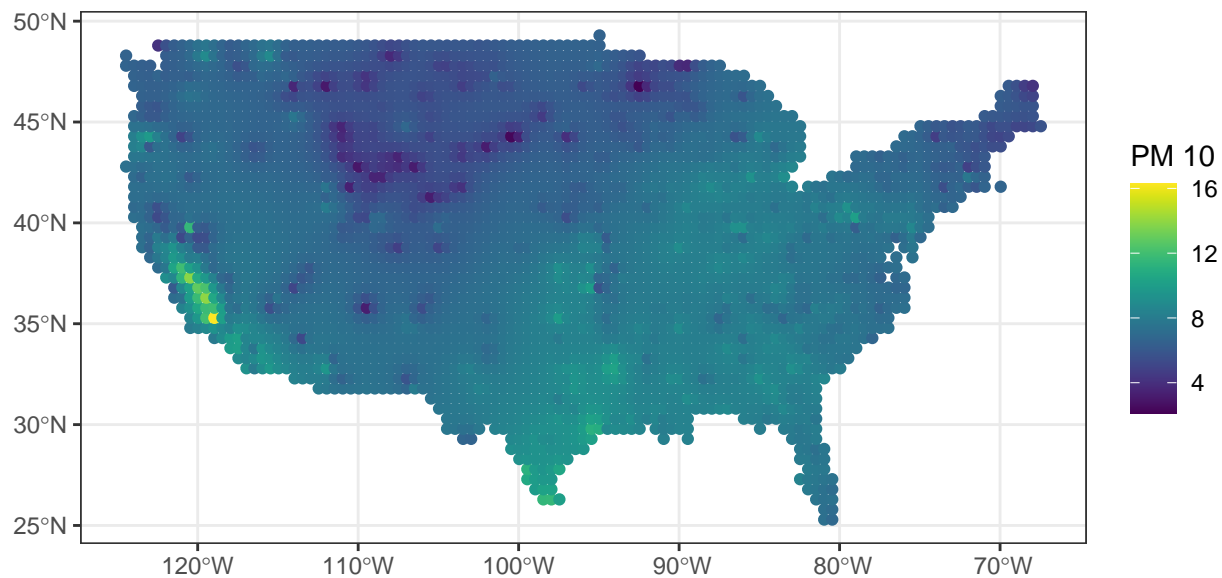
```
#uso la grilla de la interpolación anterior
#con value ~ 1
idw_resultado <- idw(value ~ 1, usa.spdf, newdata = grid2, idp = 2)
```

```
## [inverse distance weighted interpolation]
```

```
idw_sf <- as.data.frame(idw_resultado) %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326)

#grafico interpolación IDW
ggplot() + geom_sf(data = idw_sf, aes(option = "magma", color = var1.pred)) +
  #geom_sf(data = usa.sf) +
  scale_color_viridis(name = "PM 10") + theme_bw() +
  labs(title="Interpolación IDW para P.M 2.5")
```

Interpolación IDW para P.M 2.5



```
#plot raster

grid2$value_idw <- idw_sf$var1.pred
r_template_idw <- raster(ext = extent(grid2), resolution = 0.52, crs = proj4string(grid2))
raster_idw <- rasterize(x = grid2, y = r_template_idw, field = "value_idw", fun = mean)

n_colors <- 100
viridis_palette <- viridisLite::viridis(n_colors)

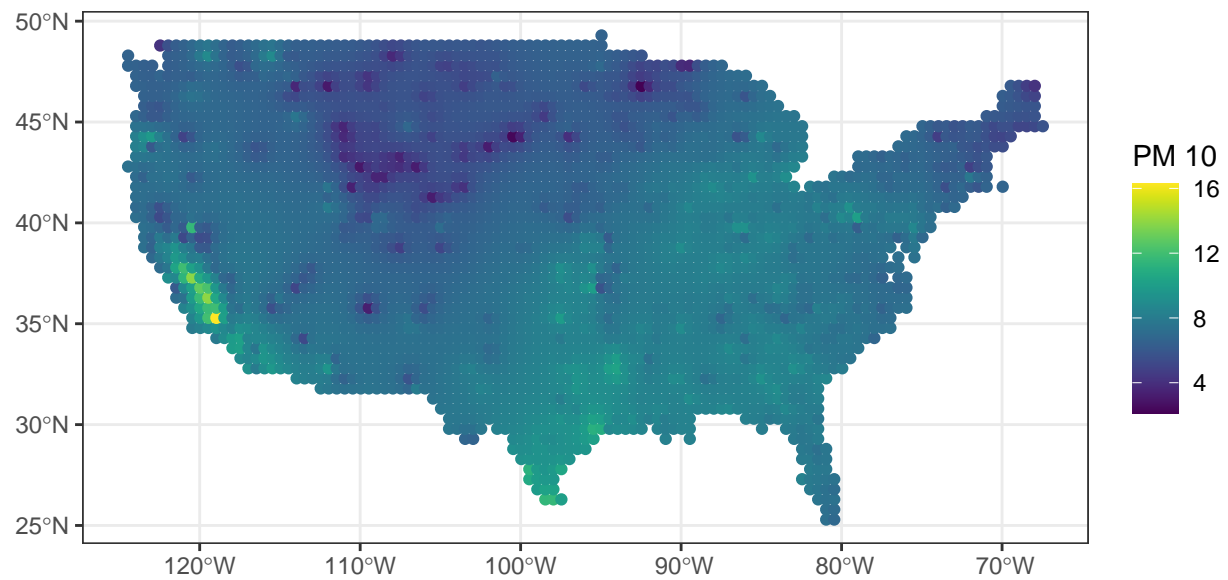
plot(
  raster_idw,
  col = viridis_palette,
  main = 'Concentración P.M 2.5 usando IDW',
  axes = FALSE,
  box = FALSE,
  #axis.args = axis.ls,
  legend.args = list(text = "P.M 2.5", side = 3, font = 2, line = 0.2, cex = 0.8)
)
```

Concentración P.M 2.5 usando IDW



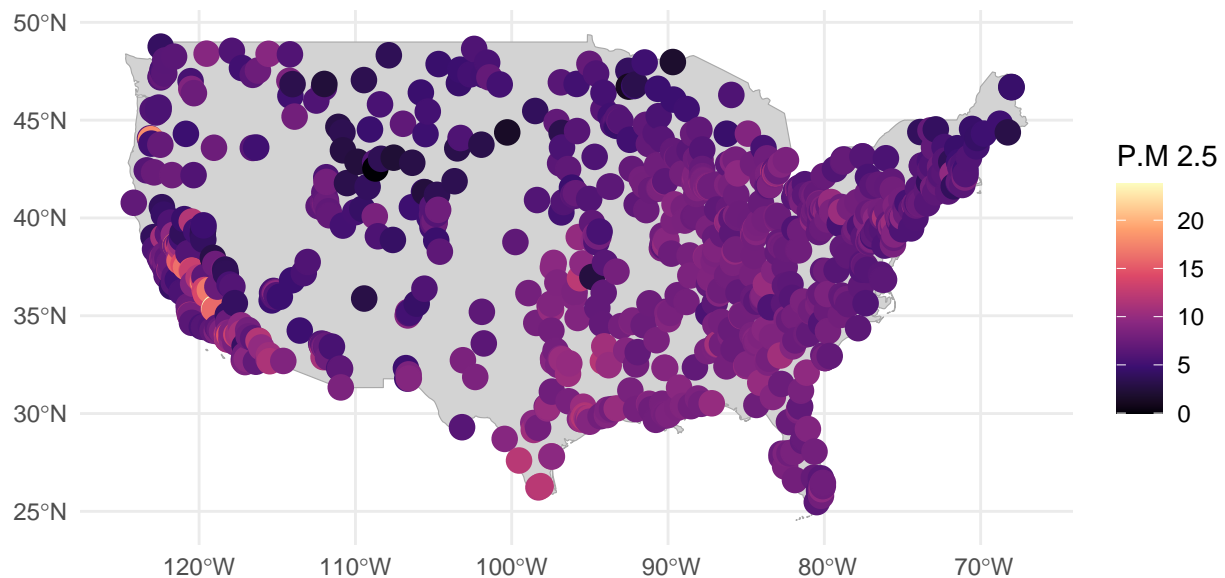
```
ggplot() + geom_sf(data = idw_sf, aes(color = vari.pred)) +  
  #geom_sf(data = usa.sf) +  
  scale_color_viridis(name = "PM 10") + theme_bw() +  
  labs(title="Interpolación IDW para P.M 2.5")
```


Interpolación IDW para P.M 2.5



```
#grafico mis datos para comparar  
ggplot() +  
  geom_sf(data = mapa, fill = "lightgray", color = "darkgray") +  
  geom_sf(data = usa.sf, aes(color = value), size = 4, alpha = 3) +  
  scale_color_viridis_c(option = "magma", name = "P.M 2.5") +  
  theme_minimal() +  
  labs(title = "Datos originales P.M 2.5")
```

Datos originales P.M 2.5



Se observa que en la zona este y sur parece estar captando, las concentraciones altas, al igual que al norte las bajas concentraciones, mejor que logró captarlo Kriging. Sin embargo, hay que calcular cv y evaluar los resultados obtenidos.

```
observed_cv_idw <- numeric(nrow(usa.spdf))
predicted_cv_idw <- numeric(nrow(usa.spdf))

#leave-One-Out Cross-Validation
for (i in 1:nrow(usa.spdf)) {

  data_train <- usa.spdf[-i, ]
  data_validate <- usa.spdf[i, ]

  captured_output <- capture.output({
    idw_local_pred <- idw(formula = value ~ 1, locations = data_train, newdata = data_validate, idp = 2)
  })

  observed_cv_idw[i] <- as.numeric(data_validate$value)
  predicted_cv_idw[i] <- as.numeric(idw_local_pred$var1.pred)
}

valid_indices_idw_cv <- !is.na(observed_cv_idw) & !is.na(predicted_cv_idw)
observed_cv_idw_clean <- observed_cv_idw[valid_indices_idw_cv]
predicted_cv_idw_clean <- predicted_cv_idw[valid_indices_idw_cv]

#MSE
```

```
mse_idw_cv <- mean((observed_cv_idw_clean - predicted_cv_idw_clean)^2)
message(paste0("MSE: ", round(mse_idw_cv, 4)))
```

```
## MSE: 2.9682
```

```
#RMSE
rmse_idw_cv <- sqrt(mse_idw_cv)
message(paste0("RMSE: ", round(rmse_idw_cv, 4)))
```

```
## RMSE: 1.7229
```

```
# R^2
r2_idw_cv <- RSQUARE(y_actual = observed_cv_idw_clean, y_predict = predicted_cv_idw_clean)
message(paste0("R^2: ", round(r2_idw_cv, 4)))
```

```
## R^2: 0.4231
```

Los resultados, son un poco mejores a Kriging, con un R^2 más alto y MSE y RMSE un poco más bajos.

Natural Neighbour Interpolation

- Tiene base en principios geométricos.
- Estima el valor de un punto utilizando un promedio ponderado de sus “vecinos naturales”, que no se eligen solo por proximidad, sino por su relación geométrica.
- Los pesos se calculan en función del “área prestada”, la porción del polígono de cada vecino que es absorbida por el polígono del nuevo punto a estimar.

$$G(x) = \sum_{i=1}^n w_i(x) f(x_i)$$

$$w_i(x) = \frac{A(x_i)}{A(x)}$$

```
library(whitebox)
wbt_init()

if (!dir.exists("./shapes")) {
  dir.create("./shapes")
}

target_crs <- "+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=37.5 +lon_0=-96 +x_0=0 +y_0=0 +ellps=GRS80 +datum=NAD83 +units=m +no_defs"

usa.sf_proj <- usa_data %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326) %>%
  st_transform(crs = target_crs)

mapa_proj <- st_transform(mapa, crs = target_crs)
```

```

usa_shp_path <- paste0(getwd(), '/shapes/', 'usa_data_points.shp')
st_write(usa.sf_proj, dsn = usa_shp_path, delete_layer = TRUE, quiet = TRUE)

output_raster_path <- './data/usa_nn_whitebox.tif'

wbt_natural_neighbour_interpolation( input = usa_shp_path, output = output_raster_path, field = 'value'

zn.wb <- raster(output_raster_path)
crs(zn.wb) <- target_crs

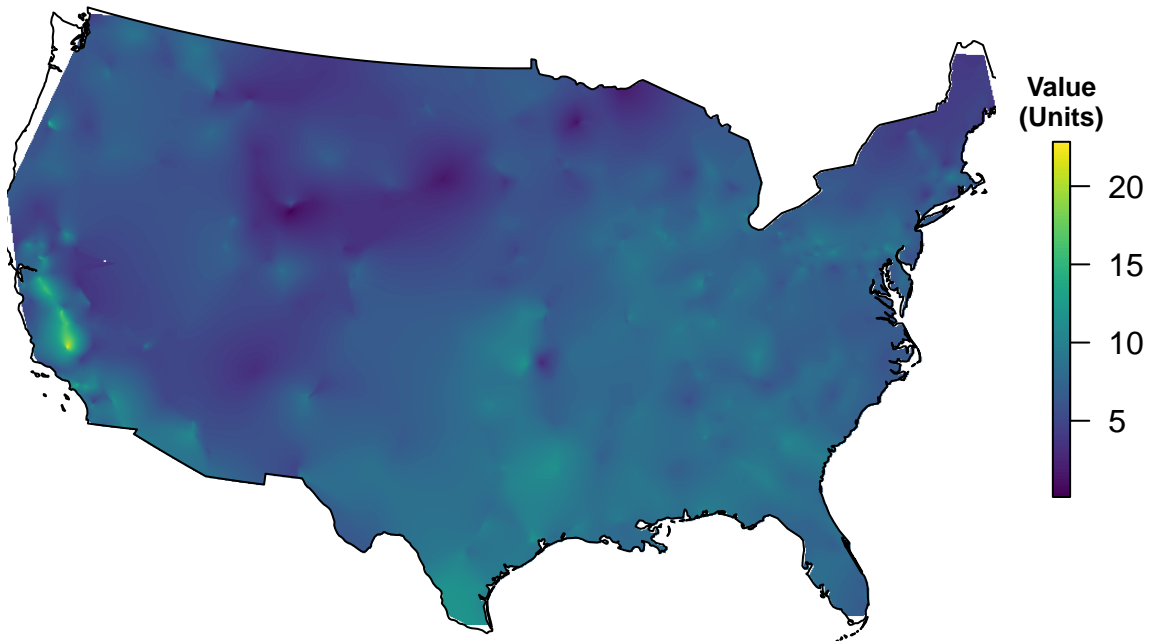
mapa_sp <- as(mapa_proj, "Spatial")
zn.wb_clipped <- mask(zn.wb, mapa_sp)

n_colors <- 100
viridis_palette <- viridisLite::viridis(n_colors)
mapa_proj <- st_transform(mapa, crs = target_crs)

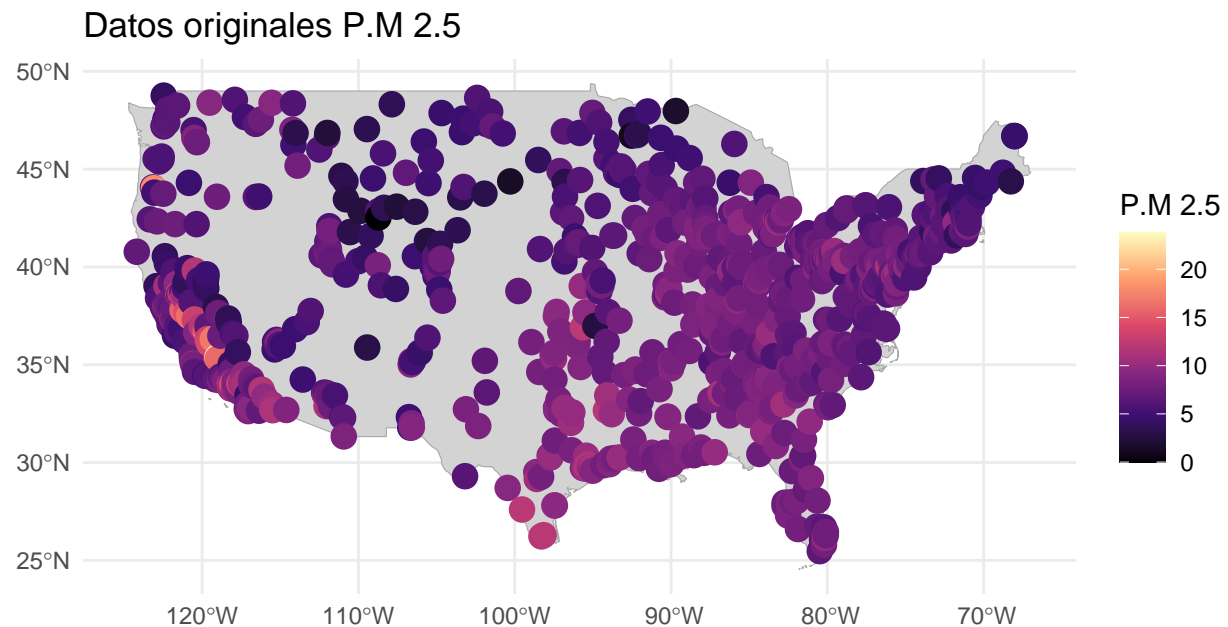
par(mar = c(1, 1, 3, 2))
plot(zn.wb_clipped,
     col = viridis_palette,
     main = 'Natural Neighbor Interpolation',
     axes = FALSE,
     box = FALSE,
     legend.args = list(text = "Value\n(Units)", side = 3, font = 2, line = 0.2, cex = 0.8)
)
plot(st_geometry(mapa_proj), add = TRUE, border = "black", lwd = 1)

```

Natural Neighbor Interpolation



```
#grafico mis datos para comparar  
ggplot() +  
  geom_sf(data = mapa, fill = "lightgray", color = "darkgray") +  
  geom_sf(data = usa.sf, aes(color = value), size = 4, alpha = 3) +  
  scale_color_viridis_c(option = "magma", name = "P.M 2.5") +  
  theme_minimal() +  
  labs(title = "Datos originales P.M 2.5")
```



Es el que mejor captó el rango de variación de las concentraciones del contaminante. Y parecería estar captando bien las zonas más altas y bajas que se veían en los datos originales. Hago cross-validation

```
k_folds <- 5

set.seed(34)
usa.sf_proj$fold <- sample(1:k_folds, nrow(usa.sf_proj), replace = TRUE)

observed_cv_nni <- numeric(0)
predicted_cv_nni <- numeric(0)

#k-Fold Cross-Validation
for (k in 1:k_folds) {

  data_train_sf <- usa.sf_proj[usa.sf_proj$fold != k, ]
  data_validate_sf <- usa.sf_proj[usa.sf_proj$fold == k, ]

  data_train_sp <- as(data_train_sf, "Spatial")
  train_shp_path <- paste0(getwd(), '/shapes/', 'nni_train_fold_', k, '.shp')
  st_write(data_train_sf, dsn = train_shp_path, delete_layer = TRUE, quiet = TRUE)

  fold_output_raster_path <- paste0('./data/nni_fold_', k, '.tif')

  wbt_natural_neighbour_interpolation(input = train_shp_path, output = fold_output_raster_path,
```

```

fold_zn_wb <- raster(fold_output_raster_path)
crs(fold_zn_wb) <- target_crs
fold_zn_wb_clipped <- mask(fold_zn_wb, mapa_sp)
predictions_k_fold <- raster::extract(x = fold_zn_wb_clipped, y = data_validate_sf)

observed_cv_nni <- c(observed_cv_nni, as.numeric(data_validate_sf$value))
predicted_cv_nni <- c(predicted_cv_nni, as.numeric(predictions_k_fold))

file.remove(train_shp_path)
file.remove(fold_output_raster_path)
}

indices_validos_nni <- !is.na(observed_cv_nni) & !is.na(predicted_cv_nni)
observed_cv_nni_clean <- observed_cv_nni[indices_validos_nni]
predicted_cv_nni_clean <- predicted_cv_nni[indices_validos_nni]

#MSE
mse_nni_cv <- mean((observed_cv_nni_clean - predicted_cv_nni_clean)^2)
message(paste0("MSE (NNI CV): ", round(mse_nni_cv, 4)))

## MSE (NNI CV): 2.5965

#RMSE
rmse_nni_cv <- sqrt(mse_nni_cv)
message(paste0("RMSE (NNI CV): ", round(rmse_nni_cv, 4)))

## RMSE (NNI CV): 1.6114

#R^2
r2_nni_cv <- RSQUARE(y_actual = observed_cv_nni_clean, y_predict = predicted_cv_nni_clean)
message(paste0("R^2 (NNI CV): ", round(r2_nni_cv, 4)))

## R^2 (NNI CV): 0.4859

```

Es el mejor R^2 de los 3, sigue sin ser alto. El MSE y RMSE, también son más bajos que kriging e IDW.

Conclusiones

- Se realizó un Test de Moran que resultó significativo, lo que nos habla de una autocorrelación espacial de los datos

```

resultados <- data.frame(
  Método = c("Kriging", "IDW", "NNI"),
  R2 = c(0.2974, 0.4231, 0.4859),
  MSE = c(3.99, 2.9682, 2.5965),
  RMSE = c(1.996, 1.7229, 1.6114)
)

resultados

```

##	Método	R2	MSE	RMSE
## 1	Kriging	0.2974	3.9900	1.9960
## 2	IDW	0.4231	2.9682	1.7229
## 3	NNI	0.4859	2.5965	1.6114

- Los mejores resultados de interpolaciones se obtuvieron con Natural Neighbour Interpolación con el mayor R^2 y menores valores de MSE y RMSE, le siguen los de IDW y por último Kriging.
- Explicación de los resultados: Este análisis únicamente está basado en dos variables, que son la latitud y longitud, es decir, se intentó explicar e interpolar la distribución de las concentraciones a lo largo del país con estas variables. En mi opinión, la concentración de este contaminante depende de otros factores, además de la latitud y longitud, que influyen en la misma, como: la temperatura, el viento, la humedad y la cantidad de habitantes, entre otros. Para un análisis más completo, certero y detallado, sería óptimo contar con más variables de análisis como estas.