Some Random Title Here *

Farah Chin *400229991*
Talat Hakim *400315290*
Nathan Nadeau *400342430*
Cindia Dao-Vu *400319161*
Ifra Awan *400261667*
Oliver

---

abstract goes here; some extra spiel added here

Keywords: opioids, education, income, spatial analysis

Introduction

Words go here

More text for introduction Add more words. For a review of land use regression models see Hoek et al. (2008).

```
rm(list = ls())
```

Data

```
# Load libraries for interpolation (might not need all of them)
library(deldir) # Delaunay Triangulation and Dirichlet (Voronoi) Tessellation
library(isdas) # Companion Package for Book An Introduction to Spatial Data Analysis and Statistics
#library(plotly) # Create Interactive Web Graphics via 'plotly.js'
library(gstat)
library(spatstat) # Spatial Point Pattern Analysis, Model-Fitting, Simulation, Tests
library(spatstat.explore) # Exploratory Data Analysis for the 'spatstat' Family
library(spdep) # Spatial Dependence: Weighting Schemes, Statistics
library(tidyverse) # Easily Install and Load the 'Tidyverse'
library(sf)
library(cancensus) #retrieve Canadian Census Data
library(geojsonsf)

library(stars)
library(readxl)
library(Metrics)
```

```
# Load the air quality data

AQ_data <- read_excel('data/Air_Quality_Data.xlsx')
```

```
## New names:
## * `` -> `...9`
## * `` -> `...10`
```

*Paper submitted to complete the requirements of ENVSOCTY 4GA3 Applied Spatial Statistics; with additional edits by Antonio Paez for this version.

```r
colnames(AQ_data)[4] <- c("P")
#
# AQ_data <- mutate(AQ_data,
#                   X3 = X^3, X2Y = X^2 * Y, X2 = X^2,
#                   XY = X * Y,
#                   Y2 = Y^2, XY2 = X * Y^2, Y3 = Y^3)
```

```r
summary(AQ_data)
```

```
##      Name                Lon              Lat              P
##  Length:33          Min.   :-79.61   Min.   :43.58   Min.   : 1.236
##  Class :character   1st Qu.:-79.47   1st Qu.:43.66   1st Qu.: 4.994
##  Mode  :character   Median :-79.40   Median :43.68   Median : 6.102
##                     Mean   :-79.42   Mean   :43.69   Mean   : 7.845
##                     3rd Qu.:-79.39   3rd Qu.:43.71   3rd Qu.: 6.820
##                     Max.   :-79.03   Max.   :43.83   Max.   :30.286
##
##      O3AVG           NO2AVG         SO2AVG            COAVG          ...9
##  Min.   :25.50   Min.   : 8.643   Mode:logical   Min.   :1.643   Mode:logical
##  1st Qu.:27.05   1st Qu.: 9.161   NA's:33        1st Qu.:1.643   NA's:33
##  Median :28.07   Median :10.738                  Median :1.643
##  Mean   :27.61   Mean   :11.208                  Mean   :1.643
##  3rd Qu.:28.62   3rd Qu.:12.786                  3rd Qu.:1.643
##  Max.   :28.79   Max.   :14.714                  Max.   :1.643
##  NA's   :29      NA's   :29                      NA's   :31
##     ...10
##  Length:33
##  Class :character
##  Mode  :character
##
##
##
##
```

```r
AQ_data.sf <- AQ_data |>
  st_as_sf(coords = c("Lon", "Lat"),
           remove = "FALSE") # Keep X and Y columns
```

```r
# function to determine the mean and sd of the prediction variance obtained with different kriging para
# observations: sf object containing the points to be interpolated (in this case the air pollution read
# polynomial terms need to have already been calculated
# targets: points that are being interpolated to (in this case the centroids)
# vgm_model: model of theoretical variogram (string)
# trend: what to use for the trend surface (can be Linear, Quadratic, Cubic, or None)
krig_pred_var <- function(observations, targets, vgm_model, trend = "None"){

  if(trend == "Linear"){
    trend_model <- P ~ X + Y
  } else if (trend == "Quadratic") {
    trend_model <- P ~ X2 + X + XY + Y + Y2
  } else if (trend == "Cubic") {
    trend_model <- P ~ X3 + X2Y + X2 + X + XY + Y + Y2 + XY2 + Y3
  } else {
```

```
    trend_model <- P ~ 1
  }

  variogram_v <- variogram(trend_model,
                           data = observations)

  variogram_v.t <- fit.variogram(variogram_v, model = vgm(vgm_model))

  V.kriged <- krige(trend_model,
                    observations,
                    targets,
                    variogram_v.t)
  krig.list <- list(V.kriged, mean(V.kriged$var1.var), sd(V.kriged$var1.var))
  names(krig.list) <- c("Krig.output", "variance.mean", "variance.sd")
  return(krig.list)
}
```

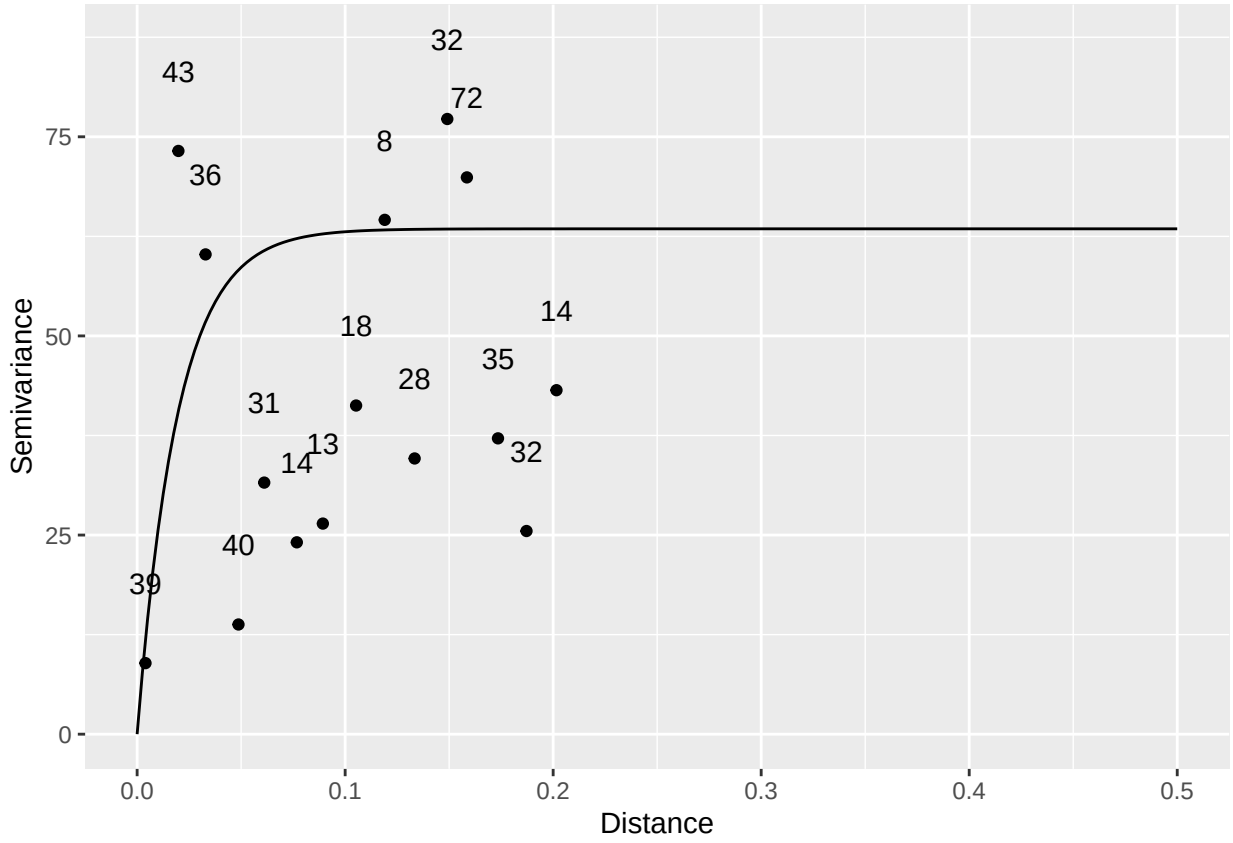
```
# Making air quality variogram
variogram_v <- variogram(P ~ 1,
                           data = AQ_data.sf)

# Create best-fitting theoretical curve
variogram_v.t <- fit.variogram(variogram_v, model = vgm("Exp", "Sph", "Gau"))
```

```
## Warning in fit.variogram(variogram_v, model = vgm("Exp", "Sph", "Gau")): No
## convergence after 200 iterations: try different initial values?
```

```
variogram_v.t
```

```
##   model    psill      range
## 1   Nug  0.00000 0.00000000
## 2   Exp 63.44689 0.01943117
```

```
gamma.t <- variogramLine(variogram_v.t, maxdist = 0.5)

# Plot semivariogram with fitted line
ggplot(data = variogram_v,
       aes(x = dist,
           y = gamma)) +
  geom_point() +
  # Add labels to indicate the number of pairs of observations used
  # in the calculation of each point in the variogram
  geom_text(aes(label = np),
            nudge_y = 10) +
  # Add theoretical semivariogram
  geom_line(data = gamma.t,
            aes(x = dist,
                y = gamma)) +
  # Add labels to axes
  xlab("Distance") +
  ylab("Semivariance")
```



Toronto Census Data

Go to https://mountainmath.github.io/cancensus/index.html and follow instructions for setting up CensusMapping account and generating API key to retrieve data.

More info: https://mountainmath.github.io/cancensus/articles/cancensus.html

Set your cache to the data folder:

```
cancensus::set_cancensus_cache_path(paste0(getwd(), "/data"),install = TRUE, overwrite = TRUE)
```

```
## Your original .Renviron will be backed up and stored in your R HOME directory if needed.

## Your cache path has been stored in your .Renviron and can be accessed by Sys.getenv("CM_CACHE_PATH")

## [1] "C:/Users/Farah Chin/Documents/McMaster/ENVSOCTY 4GA3/Air-Pollution-Correlates-4GA3/data"
```

```
#Getting median household income from cancensus:

# census tract level (Toronto CSD)
# torontoCTs.demographics.sf <- get_census(dataset='CA21', regions=list(CSD="3520005"),
#                           vectors=c("median_hh_income"="v_CA21_906",
#                                     "population_density" = "v_CA21_6",
#                                     "median_age" = "v_CA21_389",
#                                     "public_transit" = "v_CA21_7644",
#                                     "automobile" = "v_CA21_7635",
#                                     "walking" = "v_CA21_7647",
#                                     "cycling" = "v_CA21_7650"),
```

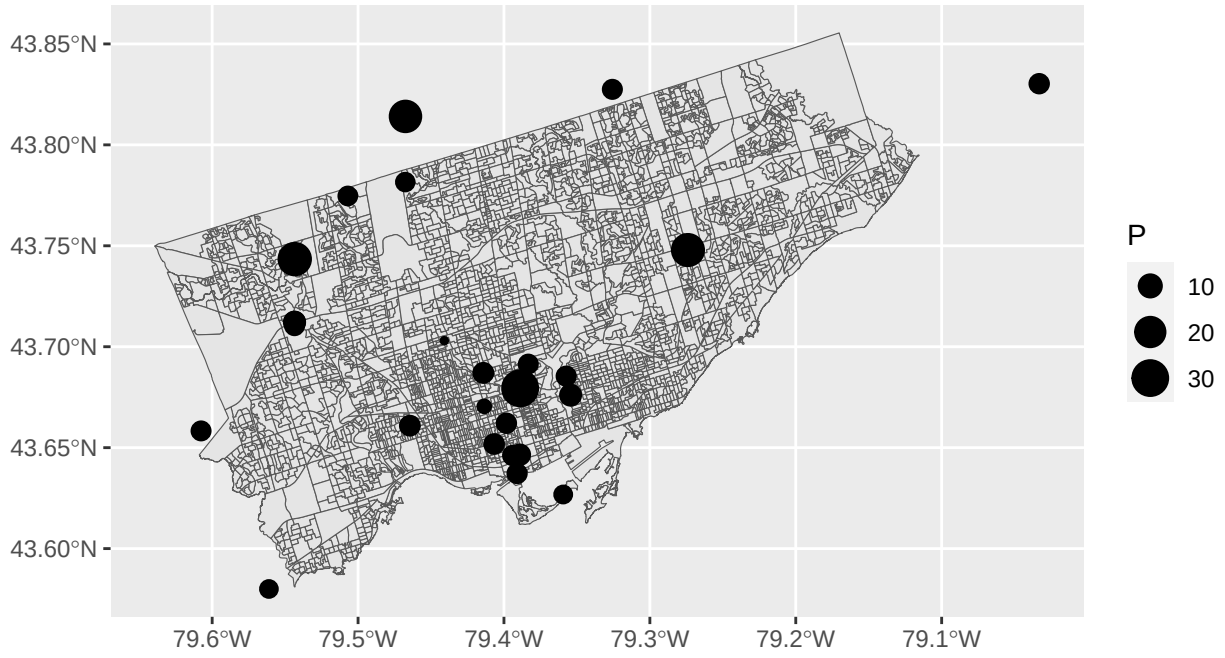
```r
#                           level='CT', quiet = TRUE,
#                           geo_format = 'sf', labels = 'short')

# dissemination area level (Toronto CSD)
torontoDAs.demographics.sf <- get_census(dataset='CA21', regions=list(CSD="3520005"),
                          vectors=c("median_hh_income"="v_CA21_906",
                                    "pop21" = "v_CA21_1",
                                    "population_density" = "v_CA21_6",
                                    "median_age" = "v_CA21_389",
                                    "public_transit" = "v_CA21_7644",
                                    "automobile" = "v_CA21_7635",
                                    "walking" = "v_CA21_7647",
                                    "cycling" = "v_CA21_7650",
                                    "minority_population" = "v_CA21_4875",
                                    "dwellings" = "v_CA21_434",
                                    "bachelors" = "v_CA21_5847"),
                          level='DA', quiet = TRUE,
                          geo_format = 'sf', labels = 'short')
```

```
## Simple feature collection with 3743 features and 25 fields
## Attribute-geometry relationships: constant (25)
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 609549.9 ymin: 4826362 xmax: 651615.7 ymax: 4857448
## Projected CRS: WGS 84 / UTM zone 17N
## First 10 features:
##    Shape Area Quality Flags   GeoUID Type Households CD_UID Dwellings
## 1      0.0504             0 35200002   DA        103   3520       106
## 2      0.0468             0 35200003   DA         88   3520        98
## 3      0.0488             0 35200004   DA        101   3520       106
## 4      0.0443             0 35200005   DA        110   3520       120
## 5      0.0572             0 35200006   DA        135   3520       146
## 6      0.0774             0 35200007   DA        179   3520       185
## 7      0.0638             0 35200009   DA         84   3520        86
## 8      0.0363             0 35200010   DA         75   3520        80
## 9      0.0474             0 35200012   DA        148   3520       153
## 10     0.0073             0 35200013   DA         18   3520        20
##    Population     name CSD_UID     CT_UID CMA_UID Region Name Area (sq km)
## 1         333 35200002 3520005 5350378.28   35535    35200002       0.0504
## 2         314 35200003 3520005 5350378.28   35535    35200003       0.0468
## 3         346 35200004 3520005 5350378.28   35535    35200004       0.0488
## 4         412 35200005 3520005 5350378.28   35535    35200005       0.0443
## 5         520 35200006 3520005 5350378.28   35535    35200006       0.0572
## 6         644 35200007 3520005 5350378.28   35535    35200007       0.0774
## 7         313 35200009 3520005 5350378.17   35535    35200009       0.0638
## 8         275 35200010 3520005 5350378.17   35535    35200010       0.0363
## 9         508 35200012 3520005 5350378.17   35535    35200012       0.0474
## 10         71 35200013 3520005 5350378.17   35535    35200013       0.0073
##    median_hh_income pop21 population_density median_age public_transit
## 1             94000   333             6607.1       43.6             20
## 2            118000   314             6709.4       47.6             10
## 3            123000   346             7090.2       45.6             25
## 4            105000   412             9300.2       36.8             50
## 5            113000   520             9090.9       40.4             20
```

```
## 6            117000   644             8320.4       41.6             50
## 7            104000   313             4906.0       43.6             40
## 8            109000   275             7575.8       39.2             10
## 9            116000   508            10717.3       42.4             70
## 10               NA    71             9726.0       39.2              0
##    automobile walking cycling minority_population dwellings bachelors
## 1          55       0       0                  280       105       120
## 2          85       0       0                  345        85        45
## 3          95       0       0                  295       105        75
## 4         145       0       0                  425       110        80
## 5         115       0       0                  410       135       115
## 6         155       0       0                  720       180       165
## 7         110       0       0                  320        85        80
## 8         120       0       0                  270        75        65
## 9         155       0       0                  480       150       140
## 10         15       0       0                   95        20        10
##                          geometry
## 1  MULTIPOLYGON (((644589.6 48...
## 2  MULTIPOLYGON (((644685.9 48...
## 3  MULTIPOLYGON (((644331.5 48...
## 4  MULTIPOLYGON (((644396.8 48...
## 5  MULTIPOLYGON (((644786.9 48...
## 6  MULTIPOLYGON (((644561.5 48...
## 7  MULTIPOLYGON (((644251.9 48...
## 8  MULTIPOLYGON (((644158.9 48...
## 9  MULTIPOLYGON (((643869.8 48...
## 10 MULTIPOLYGON (((643869.8 48...
```

```r
AQ_data.sf <- AQ_data.sf |>
  mutate(X = unlist(map(AQ_data.sf$geometry, 1)),
         Y = unlist(map(AQ_data.sf$geometry, 2))) |>
  # Add polynomials
  mutate(X3 = X^3, X2Y = X^2 * Y, X2 = X^2,
         XY = X * Y,
         Y2 = Y^2, XY2 = X * Y^2, Y3 = Y^3)
```

```r
ggplot(torontoDAs.demographics.sf) + geom_sf() + geom_sf(data = AQ_data.sf, aes(size = P))
```




```
# generating centroids
torontoDAs.centroids <- st_centroid(torontoDAs.demographics.sf)
#torontoCTs.centroids <- st_centroid(torontoCTs.demographics.sf)
```

```
# Extract coordinates as separate 'X' and 'Y' columns
torontoDAs.centroids <- torontoDAs.centroids |>
  mutate(X = unlist(map(torontoDAs.centroids$geometry, 1)),
         Y = unlist(map(torontoDAs.centroids$geometry, 2))) |>
  # Add polynomials
  mutate(X3 = X^3, X2Y = X^2 * Y, X2 = X^2,
         XY = X * Y,
         Y2 = Y^2, XY2 = X * Y^2, Y3 = Y^3)
```

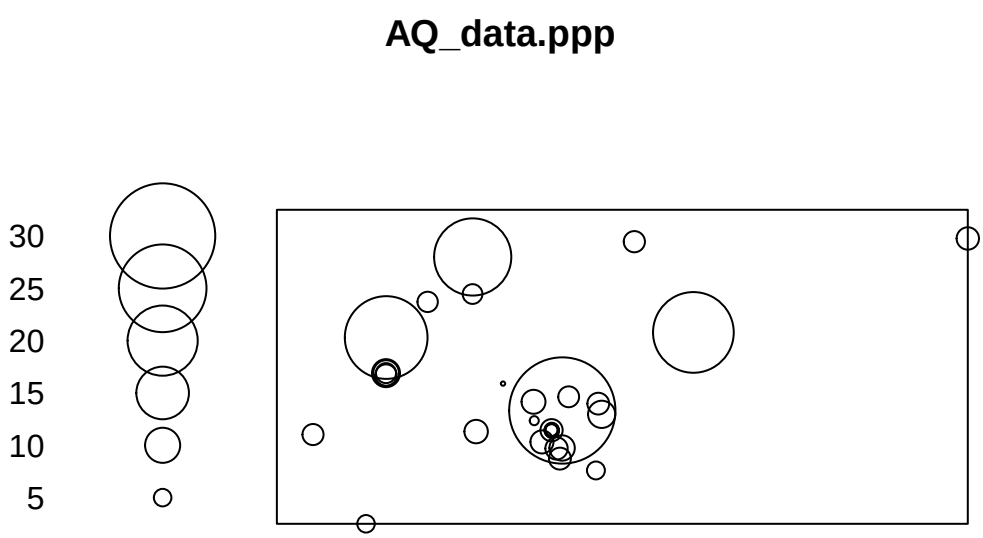
```
## IDW

to.bbox <- st_bbox(torontoDAs.demographics.sf)

# Get minimum and maximum values for aquifer data
min.X <- min(min(AQ_data$Lon), to.bbox$xmin)
min.Y <- min(min(AQ_data$Lat), to.bbox$ymin)
max.X <- max(max(AQ_data$Lon), to.bbox$xmax)
max.Y <- max(max(AQ_data$Lat), to.bbox$ymax)
x.range <- max.X - min.X
y.range <- max.Y - min.Y
```

```r
# Define region of interest
AQ.bbox <- st_polygon(list(rbind(c(min.X, min.Y),
                                 c(max.X, min.Y),
                                 c(max.X, max.Y),
                                 c(min.X, max.Y),
                                 c(min.X, min.Y))))

AQ.owin <- as.owin(AQ.bbox)

# We can create a `ppp` object with the coordinates of the points
AQ_data.ppp <- as.ppp(X = AQ_data[,2:4], W = AQ.owin)
```

```r
summary(AQ_data.ppp)
```

```
## Marked planar point pattern:  33 points
## Average intensity 197.6333 points per square unit
##
## Coordinates are given to 5 decimal places
##
## marks are numeric, of type 'double'
## Summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.236   4.994   6.102   7.845   6.820  30.286
##
## Window: polygonal boundary
## single connected closed polygon with 4 vertices
## enclosing rectangle: [-79.63931, -79.03316] x [43.58, 43.85547] units
##                      (0.6061 x 0.2755 units)
## Window area = 0.166976 square units
## Fraction of frame area: 1
```

```r
plot(AQ_data.ppp)
```



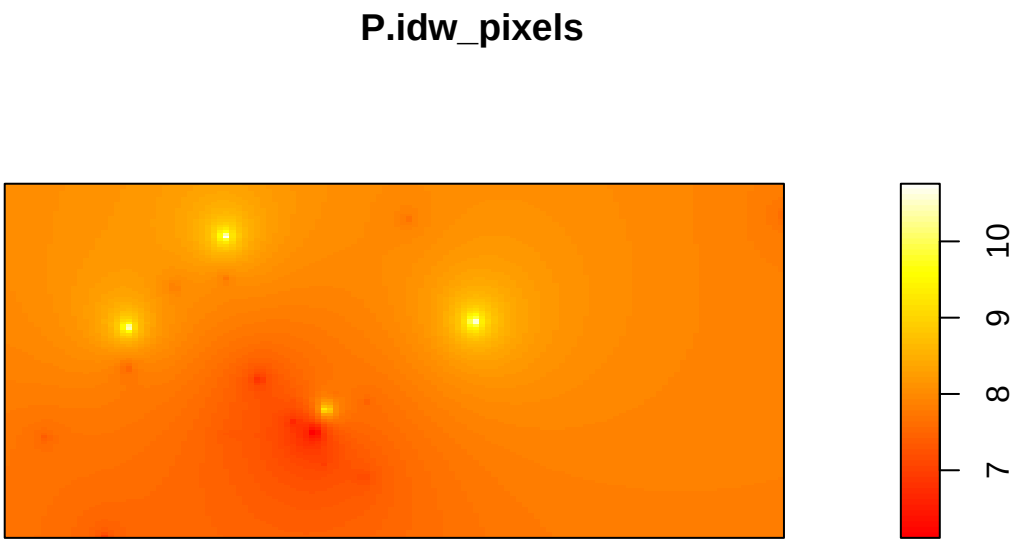
Here we use Leave One Out cross validation to determine the optimal power: https://www.statology.org/leave-one-out-cross-validation/

See also: https://rpubs.com/Dr_Gurpreet/interpolation_idw_R

```
# CROSS VALIDATION

powers <- seq(0.001, 10, 0.01)
mse_result <- NULL
for(power in powers){
  P_idw <- idw(AQ_data.ppp, power=power, at="points")
  mse_result <- c(mse_result,
                  Metrics::mse(AQ_data.ppp$marks,P_idw))
}
optimal_power <- powers[which.min(mse_result)]
optimal_power
```

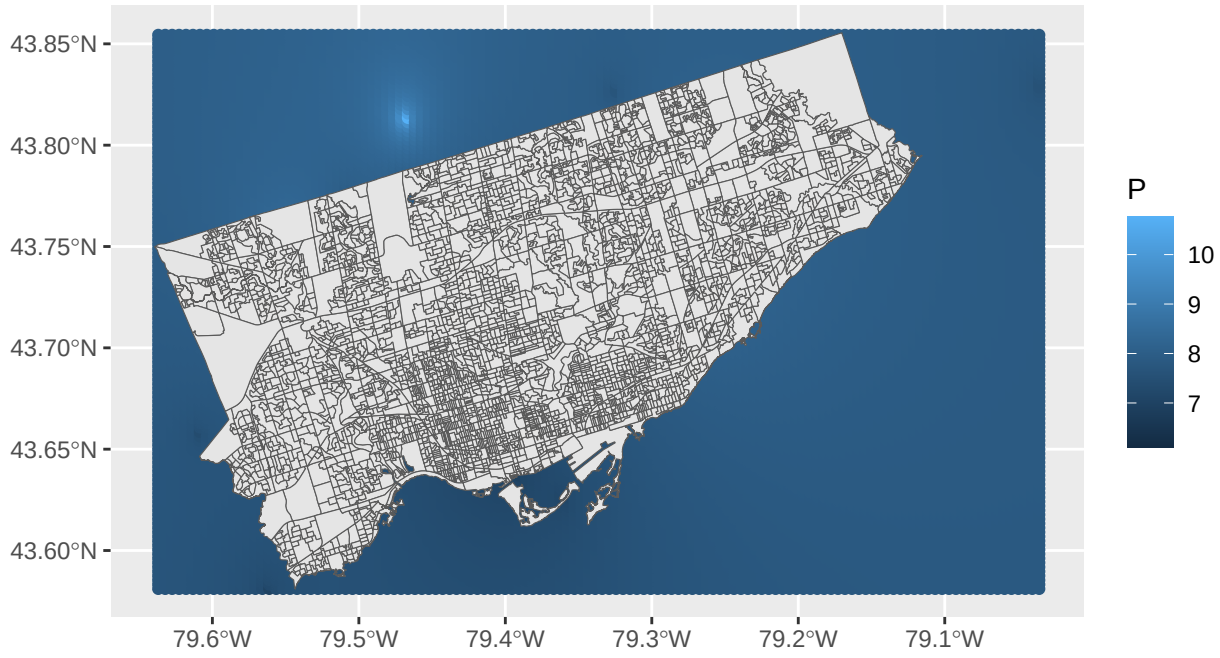
```
## [1] 0.471
```

```
P.idw_points <- idw(AQ_data.ppp, power = 0.471, at = "points")
P.idw_pixels <- idw(AQ_data.ppp, power = 0.471)
plot(P.idw_pixels, col=heat.colors(64))
```

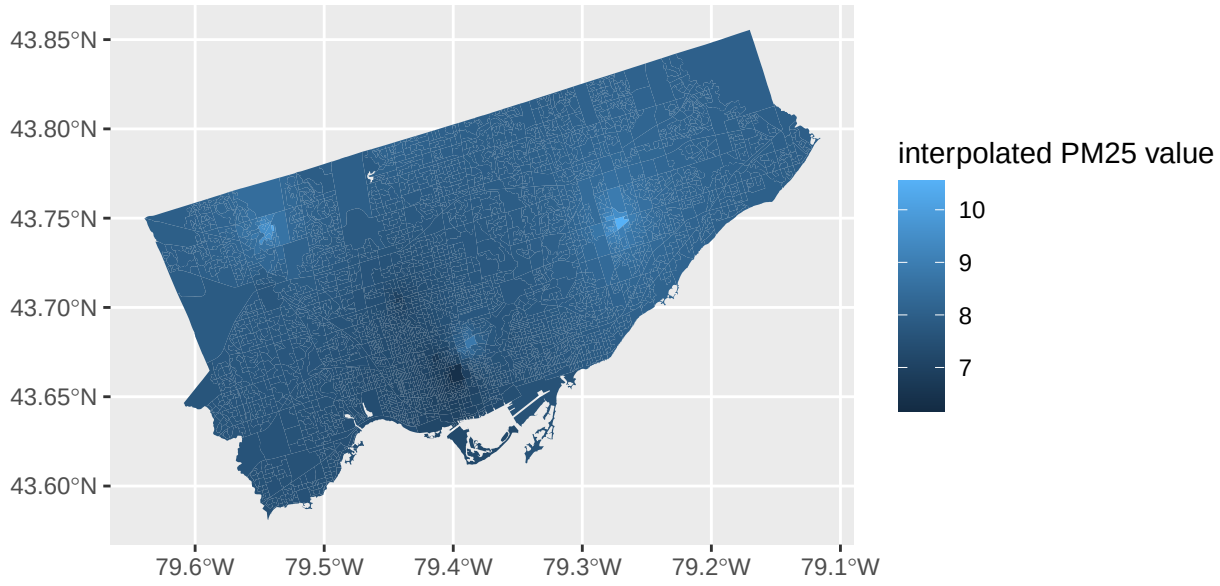



```
#Metrics::mse(AQ_data.ppp$marks, z_p.idw1)
```

```
# converting im object to a df
z.vec <- c(P.idw_pixels$v)
Y.vec <- rep(P.idw_pixels$yrow, 128)
X.vec <- rep(P.idw_pixels$xcol, each = 128)
P.idw.df <- data.frame("X" = X.vec, "Y" = Y.vec, "P" = z.vec)
P.idw.sf <- st_as_sf(P.idw.df, coords = c("X", "Y"))
st_crs(P.idw.sf) <- st_crs(torontoDAs.demographics.sf)


ggplot(P.idw.sf) + geom_sf(aes(colour = P)) + geom_sf(data = torontoDAs.demographics.sf)
```




```
# use spatial join to add the PM25 value of the nearest pixel to each centroid
toDAs.idw <- st_join(torontoDAs.centroids, P.idw.sf, join = st_nearest_feature)

#ggplot(toDAs.idw) + geom_sf(aes(colour = P))

toDAs.idw.poly <- mutate(torontoDAs.demographics.sf, P = toDAs.idw$P)
ggplot(toDAs.idw.poly) + geom_sf(aes(fill = P), colour = NA) + labs(fill = "interpolated PM25 value")
```




```
# toCTs.idw <- na.omit(st_join(torontoCTs.demographics.sf, P.idw.sf, join = st_intersects))
#
# ggplot(toCTs.idw) + geom_sf(aes(fill = P), colour = NA)
```

```
# k.vals <- seq(1, 32)
# mse_result <- c()
# for(k in k.vals){
#   se.vec <- c()
#   for (i in 1:nrow(AQ_data.sf)){
#     train.sf <- AQ_data.sf[-i,]
#     test.sf <- AQ_data.sf[i, -4]
#     kpoint.k <- kpointmean(source_xy = train.sf,
#                            target_xy = test.sf,
#                            z = P,
#                            k = k)
#
#     se.vec <- c(se.vec, (AQ_data.sf[i,]$P - kpoint.k$z)^2)
#
#   }
#   mse <- mean(se.vec)
#   mse_result <- c(mse_result, mse)
# }
# optimal_k <- k.vals[which.min(mse_result)]
# optimal_k
```

```
# k point means

# kpoint.23 <- kpointmean(source_xy = AQ_data.sf,
#                         target_xy = torontoDAs.centroids,
#                         z = P,
#                         k = 23) |>
#   rename(P = z)
#
# ct.kpoint <- kpointmean(source_xy = AQ_data.sf,
#                         target_xy = torontoCTs.centroids,
#                         z = P,
#                         k = 23) |>
#   rename(P = z)
#
# ggplot() +
#   geom_sf(data = kpoint.16,
#             aes(color = P)) +
#   scale_color_distiller(palette = "OrRd",
#                         direction = 1)
#
# ggplot() +
#   geom_sf(data = ct.kpoint,
#             aes(color = P)) +
#   scale_color_distiller(palette = "OrRd",
#                         direction = 1)
```

```
# Add interpolation to DA centroids
# toDA.centroids.kpoints <- mutate(torontoDAs.centroids,
#                                  PM25 = kpoint.23$P)
#
#
# toDA.kpoints <- mutate(torontoDAs.demographics.sf,
#                                  PM25 = kpoint.23$P)
#
# toCT.kpoints <- mutate(torontoCTs.demographics.sf,
#                                  PM25 = ct.kpoint$P)
#
# # Plot the centroids emissions
# ggplot() +
#   geom_sf(data = toDA.kpoints,
#             aes(fill = PM25), colour = NA) +
#   scale_color_brewer(palette = "OrRd",
#                         direction = 1)
#
# ggplot() +
#   geom_sf(data = toCT.kpoints,
#             aes(fill = PM25), colour = NA) +
#   scale_color_brewer(palette = "OrRd",
#                         direction = 1)
#
# ggplot() +
#   geom_sf(data = toDA.kpoints,
#             aes(color = PM25)) +
```

```
#   scale_color_distiller(palette = "OrRd",
#                         direction = 1)
#
# # Plot the populations
# ggplot() +
#   geom_sf(data = torontoDAs.demographics.sf,
#             aes(fill = Population)) +
#   scale_color_distiller(palette = "OrRd",
#                         direction = 1)
#
# # Plot the median HH income
# ggplot() +
#   geom_sf(data = torontoDAs.demographics.sf,
#             aes(fill = median_hh_income)) +
#   scale_color_distiller(palette = "OrRd",
#                         direction = 1)
#
# # Plot the centroids emissions
# ggplot() +
#   geom_sf(data = toDA.centroids.idw,
#             aes(color = automobile/Population)) +
#   scale_color_distiller(palette = "OrRd",
#                         direction = 1)
```

```
torontoDAs.nb <- poly2nb(pl = toDAs.idw.poly, queen = TRUE)
torontoDAs.w <- nb2listw(torontoDAs.nb, zero.policy = TRUE)

# torontoCTs.nb <- poly2nb(pl = toCTs.idw)
# torontoCTs.w <- nb2listw(torontoCTs.nb)
#
# torontoCTs.nb.k <- poly2nb(pl = toCT.kpoints)
# torontoCTs.k.w <- nb2listw(torontoCTs.nb.k, zero.policy = TRUE)
```

```
moran.test(toDAs.idw.poly$P, torontoDAs.w)
```

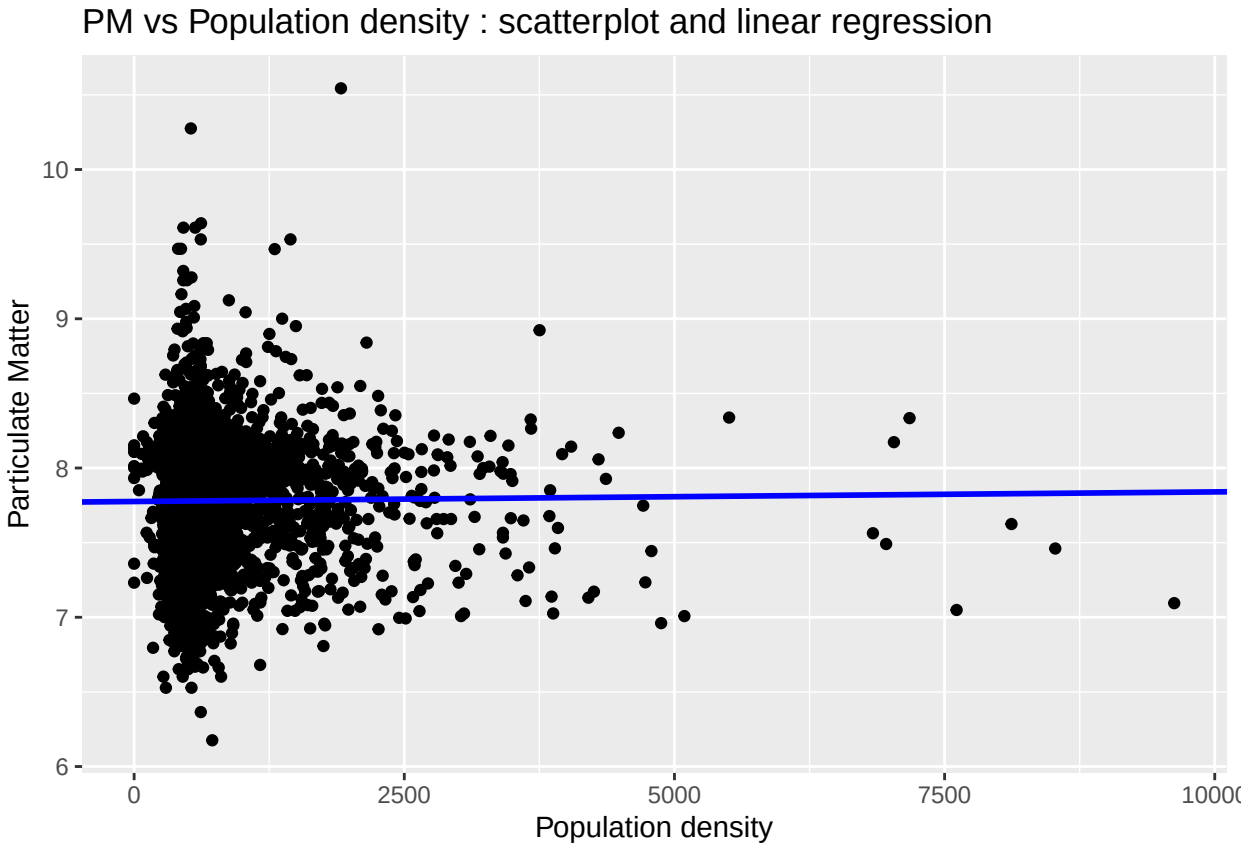
```
##
##  Moran I test under randomisation
##
## data:  toDAs.idw.poly$P
## weights: torontoDAs.w  n reduced by no-neighbour observations
##
##
## Moran I statistic standard deviate = 104.14, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      9.764162e-01     -2.673082e-04      8.796162e-05
```

```
# Linear regression function
# Ask for variable
linreg <- function(var, name){
  # Complete linear regression and output summary
  PM25_var_corr <- lm(formula = P ~ var, data = toDAs.idw.poly)
```

```
  #summary(PM25_var_corr)
  # Get variable name and title
  #title <- paste("PM25 &", name, "- Linear Regression")
  title <- paste("PM vs", name, ": scatterplot and linear regression")
  # Plot results
  ggplot(data = toDAs.idw.poly, aes(x = var, y = P)) +
    geom_point() +
    geom_abline(slope = PM25_var_corr$coefficients[2],
                intercept = PM25_var_corr$coefficients[1],
                colour = "Blue",
                linewidth = 1) +
    labs(title = title) +
    xlab(name) + ylab("Particulate Matter")

}
```

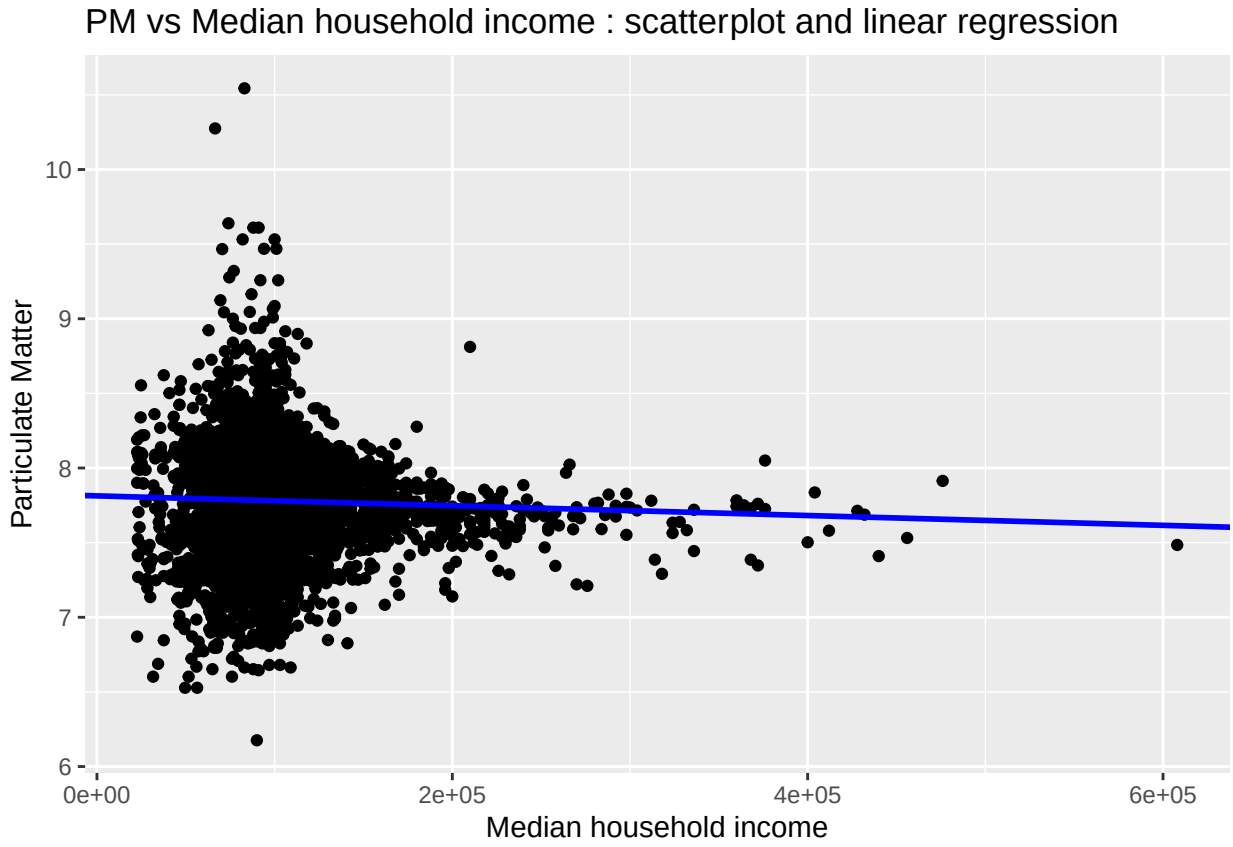
```
Pop_den.lm <- linreg(toDAs.idw.poly$Population, "Population density")
med_inc.lm <- linreg(toDAs.idw.poly$median_hh_income, "Median household income")
med_age.lm <- linreg(toDAs.idw.poly$median_age, "Median Age")
automobile.lm <- linreg(toDAs.idw.poly$automobile/toDAs.idw.poly$pop21, "Proportion commuting via autom
public_transit.lm <- linreg(toDAs.idw.poly$public_transit/toDAs.idw.poly$pop21, "Proportion commuting v
walking.lm <- linreg(toDAs.idw.poly$walking/toDAs.idw.poly$pop21, "Proportion commuting via walking")
cycling.lm <- linreg(toDAs.idw.poly$cycling/toDAs.idw.poly$pop21, "Proportion commuting via cycling")
minority.lm <- linreg(toDAs.idw.poly$minority_population/toDAs.idw.poly$pop21, "Proportion of visible
dwellings.lm <- linreg(toDAs.idw.poly$dwellings/toDAs.idw.poly$`Area (sq km)`, "Dwellings density")
```

```
par(mfrow = c(3, 3))
Pop_den.lm
```




```
med_inc.lm
```

```
## Warning: Removed 68 rows containing missing values (‘geom_point()‘).
```

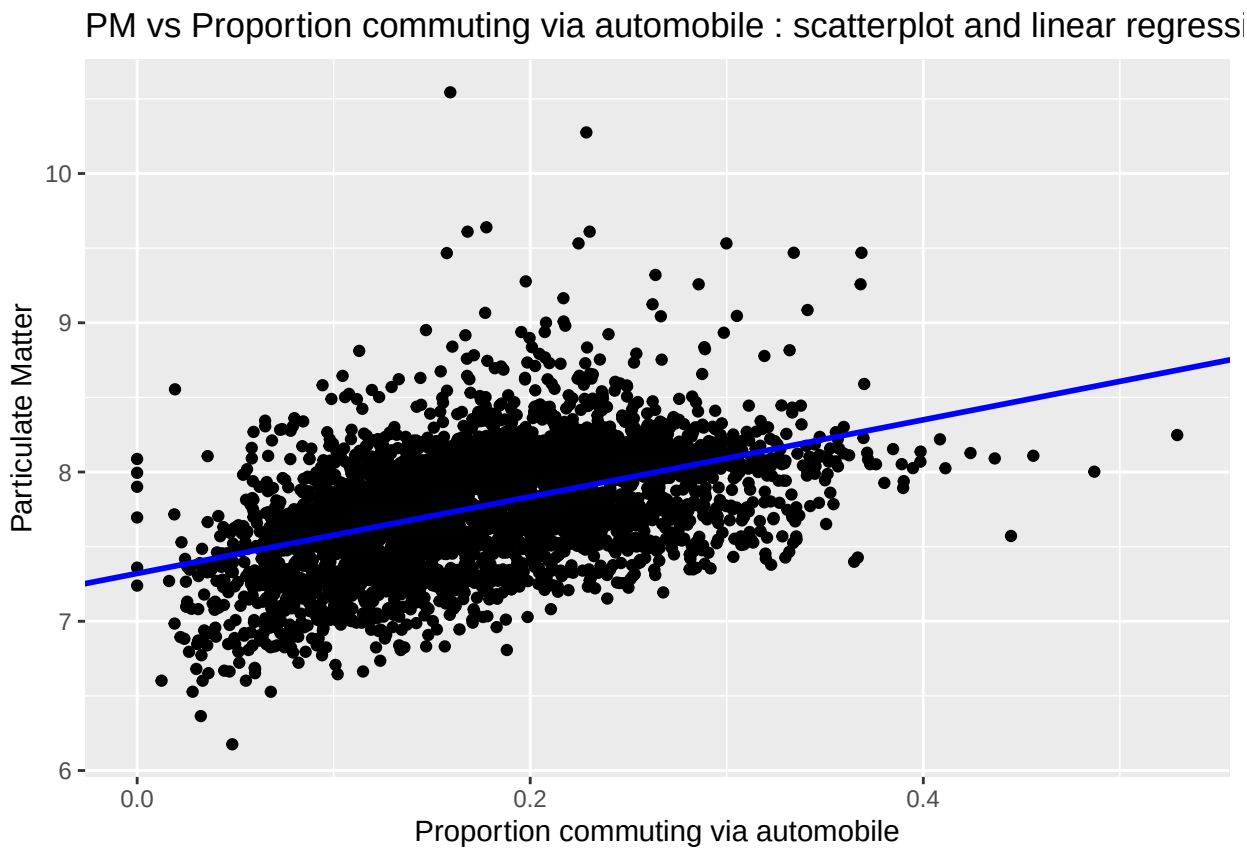



```
med_age.lm
```

```
## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```

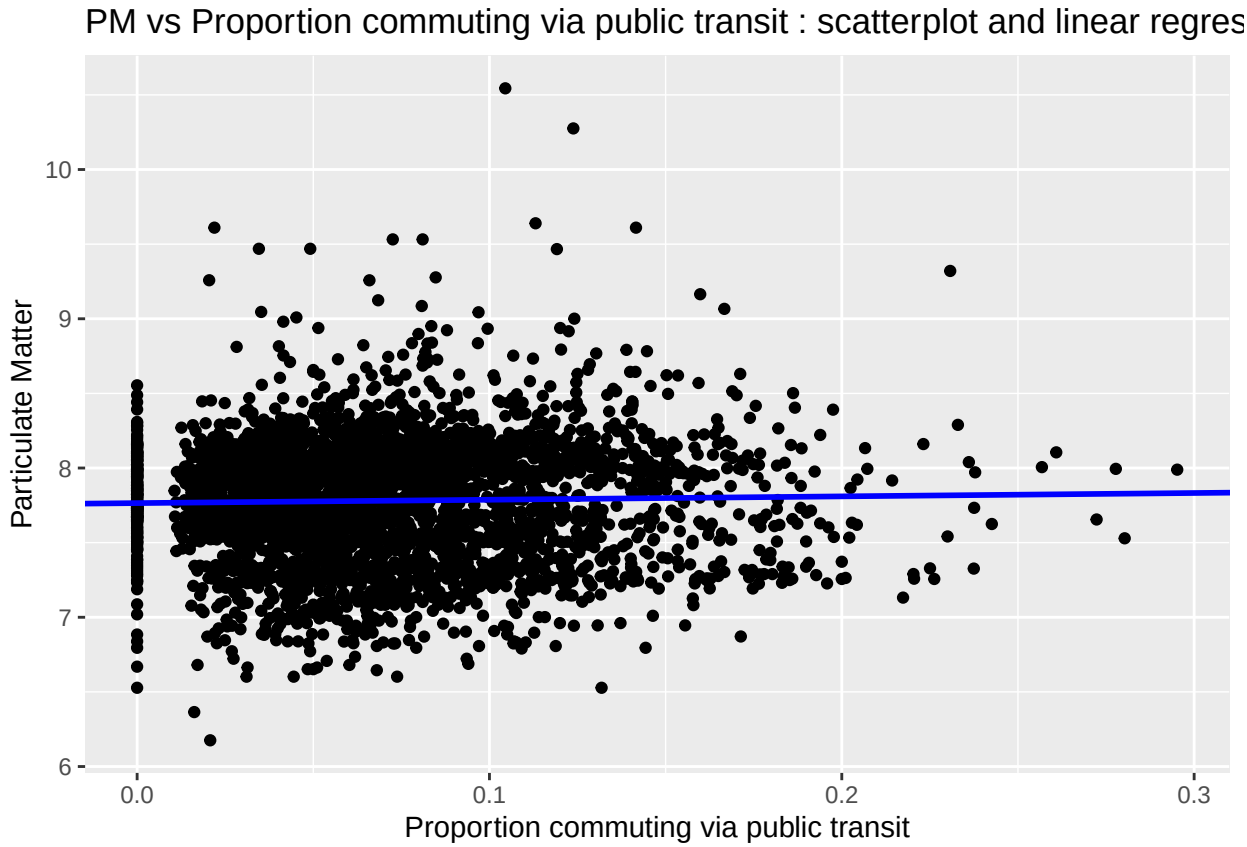

```
automobile.lm
```

```
## Warning: Removed 16 rows containing missing values (‘geom_point()‘).
```



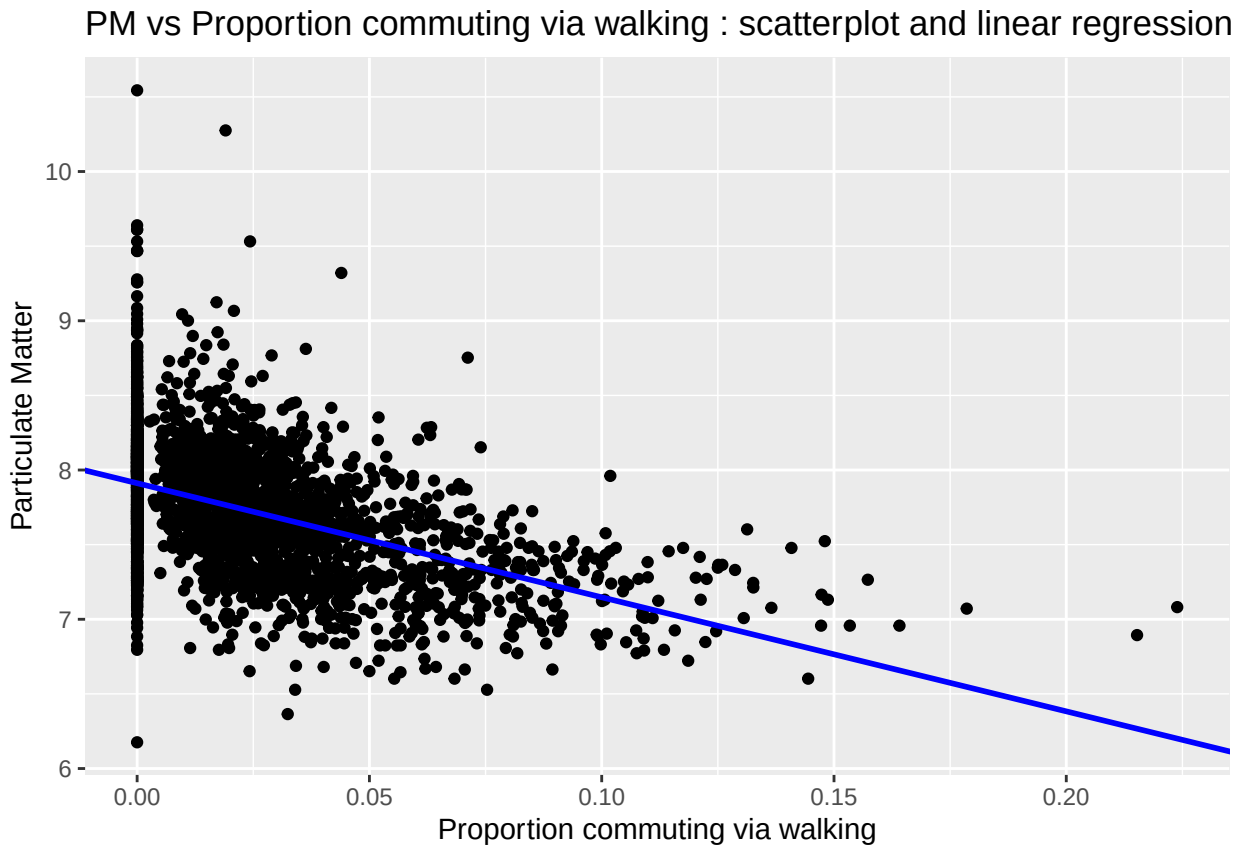

```
public_transit.lm
```

```
## Warning: Removed 16 rows containing missing values (‘geom_point()‘).
```



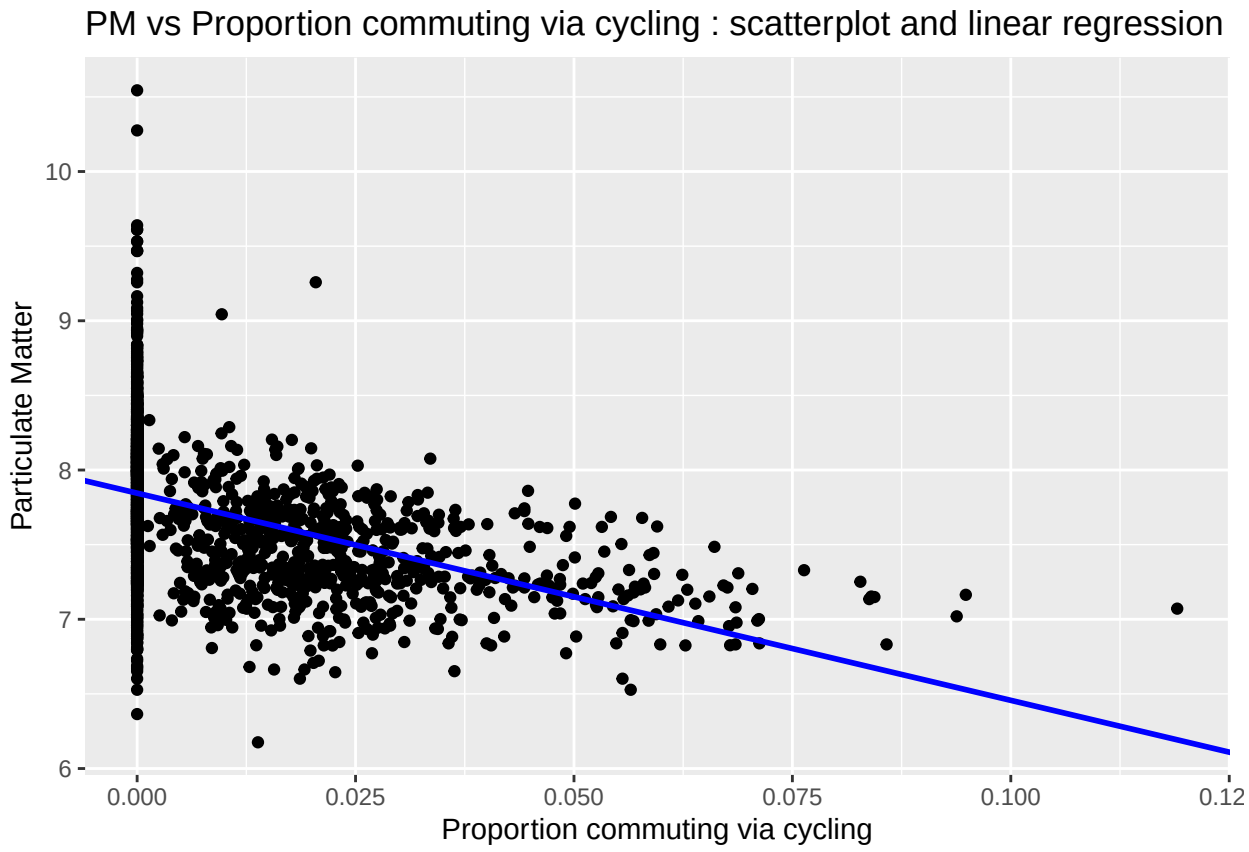

```
walking.lm
```

```
## Warning: Removed 16 rows containing missing values (‘geom_point()‘).
```



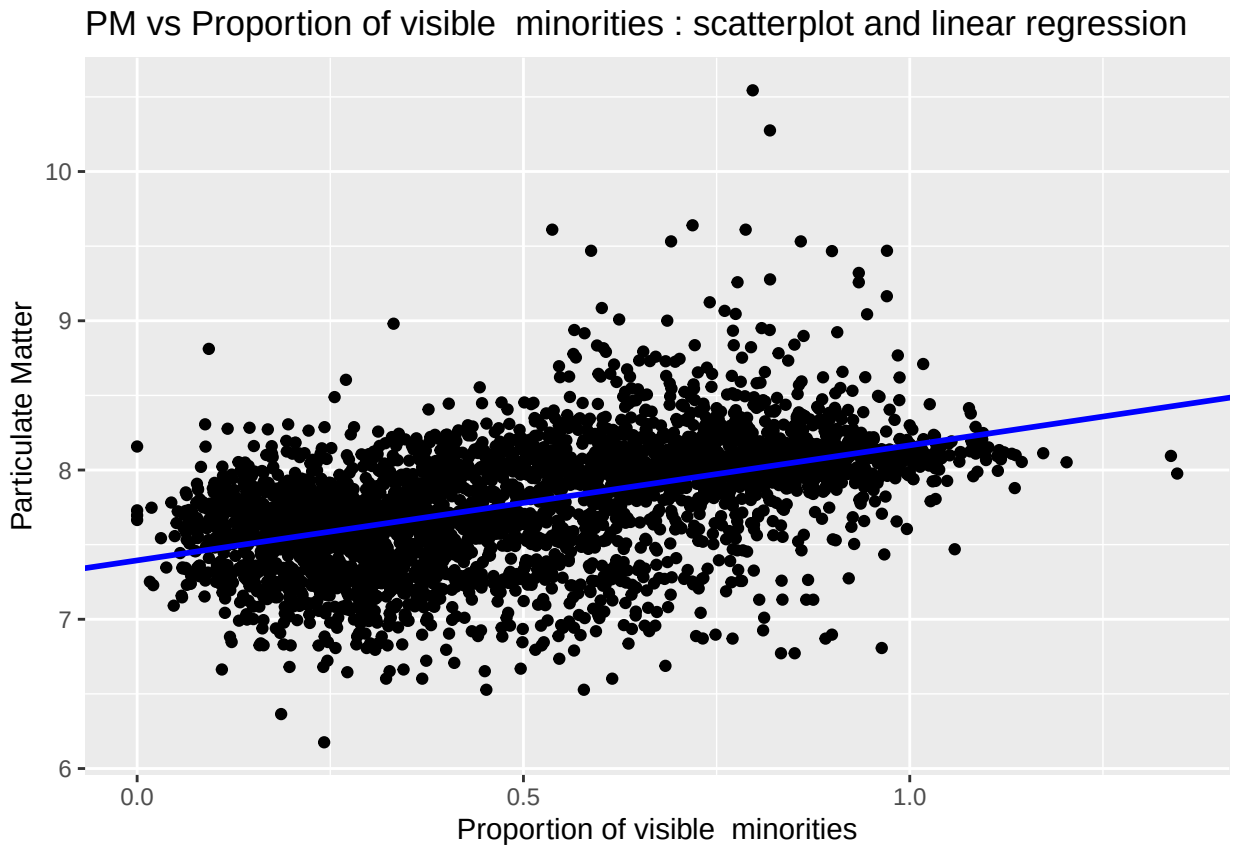

```
cycling.lm
```

```
## Warning: Removed 16 rows containing missing values (‘geom_point()‘).
```



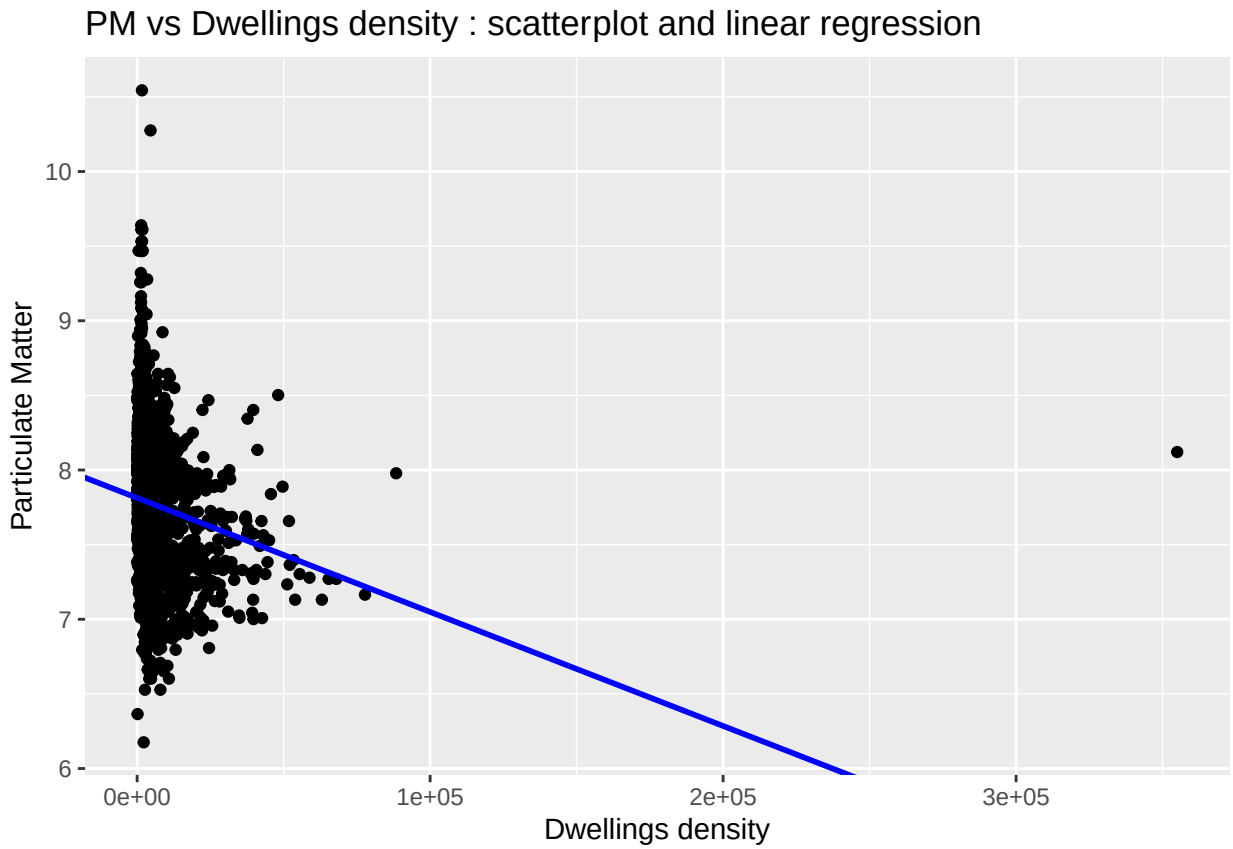

```
minority.lm
```

```
## Warning: Removed 16 rows containing missing values (‘geom_point()‘).
```




```
dwellings.lm
```

```
## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```



Words

Methods

Describing some methods.

Results and discussion

Here we will discuss the results of the analysis.

References

10 Hoek, Gerard, Rob Beelen, Kees De Hoogh, Danielle Vienneau, John Gulliver, Paul Fischer, and David Briggs. 2008. "A Review of Land-Use Regression Models to Assess Spatial Variation of Outdoor Air Pollution." *Atmospheric Environment* 42 (33): 7561–78.