Investigation to assess the impact of NO_2 air pollution on hospital admissions due to respiratory illness across Glasgow

Student 202078239

1 Introduction

In this study the impact of NO_2 pollution on respiratory illness related hospital admissions in 2012 will be investigated. The pollutant data has been provided by DEFRA and provides NO_2 values at 340 locations in and around Glasgow. Heavy industry and car emissions are leading sources of NO_2 and breathing it in can aggravate people's respiratory systems causing issues such as asthma, which sometimes require hospital admission. It is therefore important to monitor air quality in areas prone to high concentrations of these pollutants.

2 Exploratory Analysis

Figure 1 gives a visual image of the NO_2 observations that have been collected and shows that concentrations of NO_2 appear to be generally highest in the area slightly north east of the centre of Glasgow area and reducing gradually the further away you get from this high NO_2 concentration area. This area of high concentration is in an area of high population density and a busy part of the transport network.

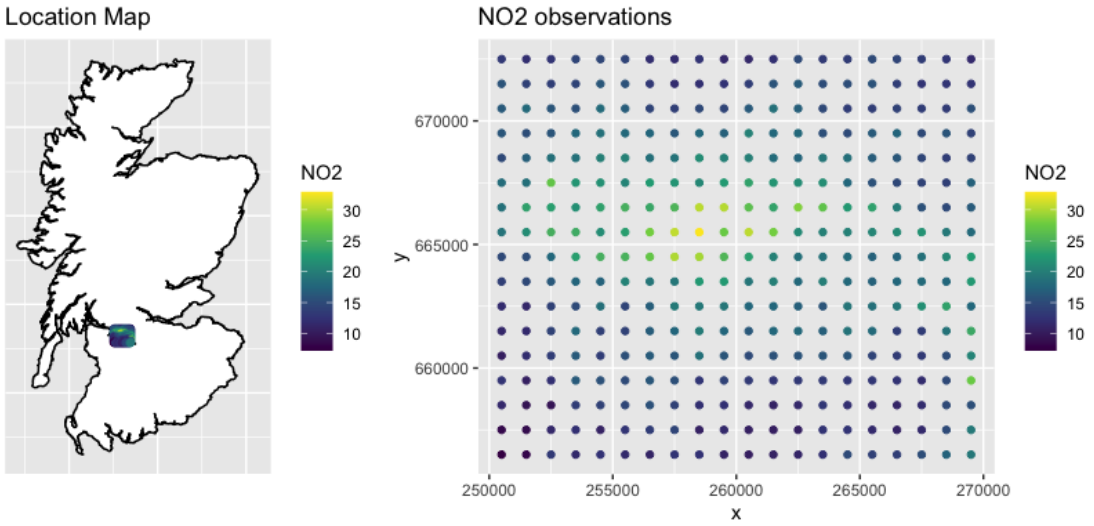

Figure 1: NO_2 Observations in Glasgow

Next, I explore making predictions of NO_2 using the observation data and Inverse Distance Weighted Interpolation (IDW). From Figure 2, it looks like p=5 is the best compromise between variability and overly smoothed data when using IDW. Reducing p showed an outline of prediction values around each observation point, whilst increasing p made the prediction grid very blocky. This gives us an idea of any spatial patterns that there may be.

Next I looked at the relationships between the covariates and NO_2. Figure 3 shows there is a trend in NO_2 values for northing and possibly slightly for easting. There seem to be particularly high concentrations centred around about (260000, 665000). I tested some linear regression models starting with x and y as covariates and then tried to account for this trend by adding polynomials to them both. Cubic polynomials seemed to work quite well for both, giving smaller p-values for these covariates but I don't want to complicate the model too much for interpretation purposes so I just include the quadratic polynomial as these are almost as good. Both variables were significant in the model so the model in equation 1 is used.

$$NO_2 = a_1x^2 + a_2x + b_1y^2 + b_2y + c \tag{1}$$

where a's and b's are coefficients and c is the intercept.

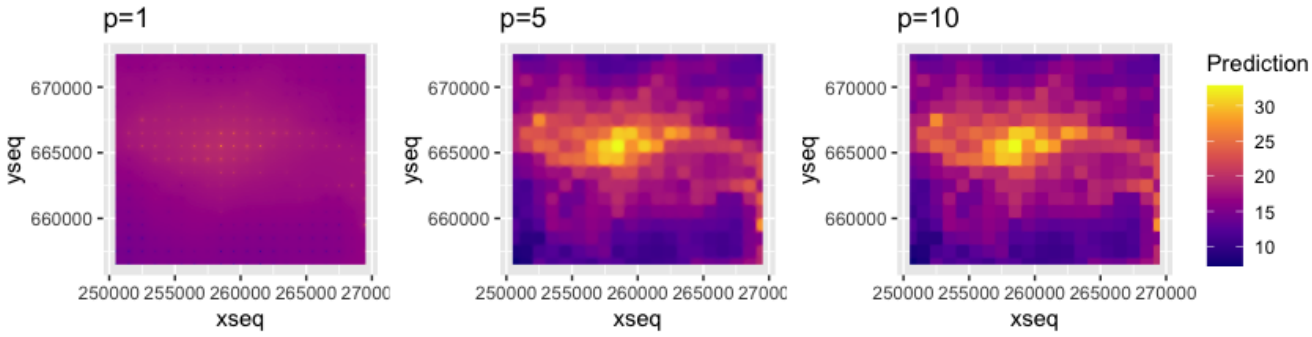

Figure 2: Inverse distance weighted interpolation

Examining the residuals of the NO_2 model in figure 4 shows that there is still some spatial dependence left. I therefore used the $\log(NO_2)$ in the model and figure 4 shows it reduces some residuals and increases others but the majority look much the same as in the NO_2 model, so for simplicity I decided to move forwards with the NO_2 model.

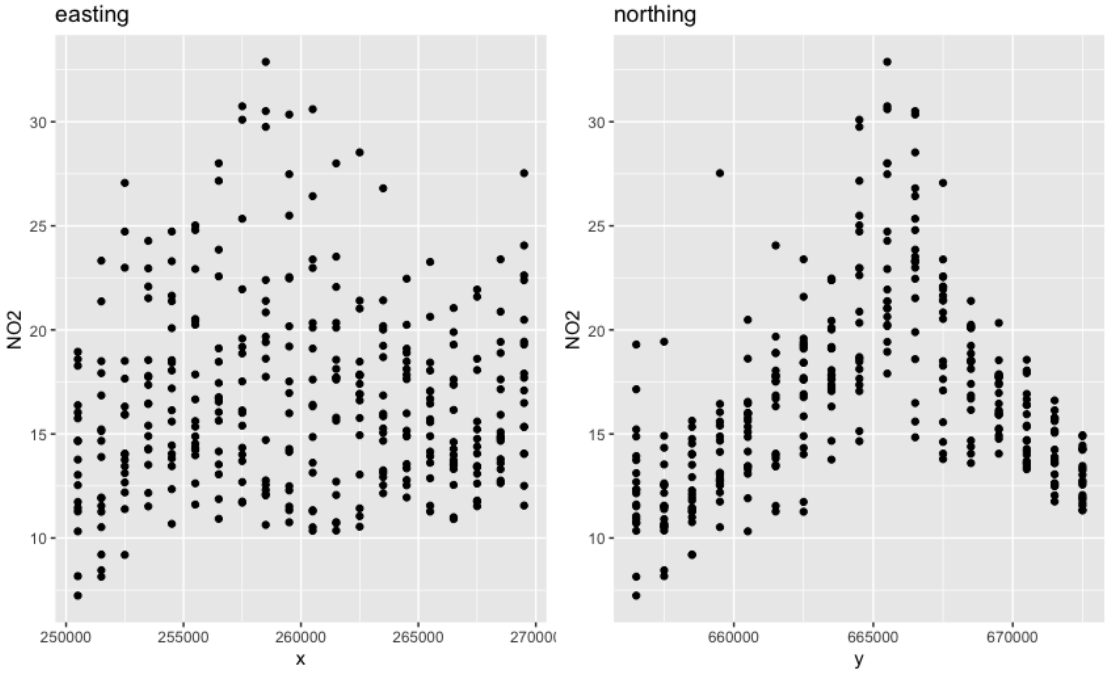

Figure 3: NO_2 against easting and northing plots

I then checked the residual versus fitted values plot and QQ-plot for this model and figure 5 shows that there is a clear non-linear trend and the QQ-plot is very skewed to the right, which is not good. I therefore investigated which was best between using $\log(NO_2)$ and NO_2 and cubic rather than quadratic polynomials in the model. The best model out of these options was modelling $\log(NO_2)$ against the cubic polynomials of x and y covariates. As shown in figure 5, the variance of the residuals looks more constant and centred around zero and the QQ-plot is linear apart from a slight right tail so I am now happy that the residuals look normally distributed and this model assumption is therefore valid. The final model is shown in equation 2.

$$Log(NO_2) = a_1x^3 + a_2x^2 + a_3x + b_1y^3 + b_2y^2 + b_3y + c. \tag{2}$$

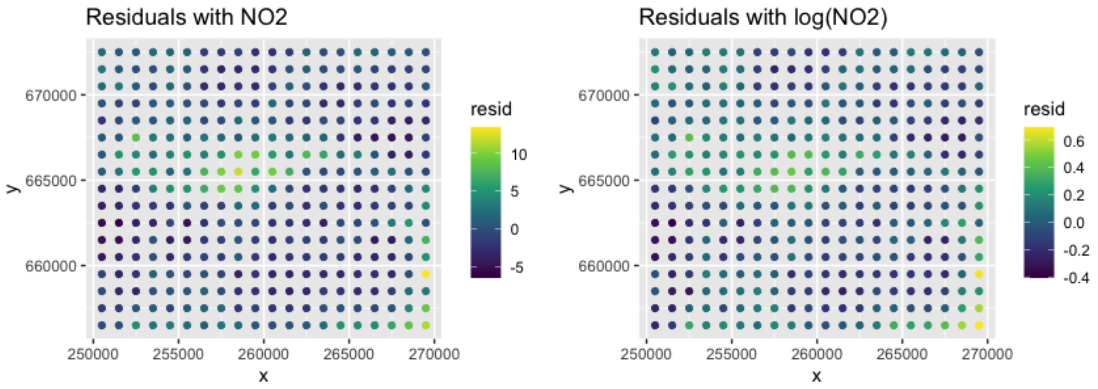

Figure 4: Residuals of the NO_2 and $\log(NO_2)$ models

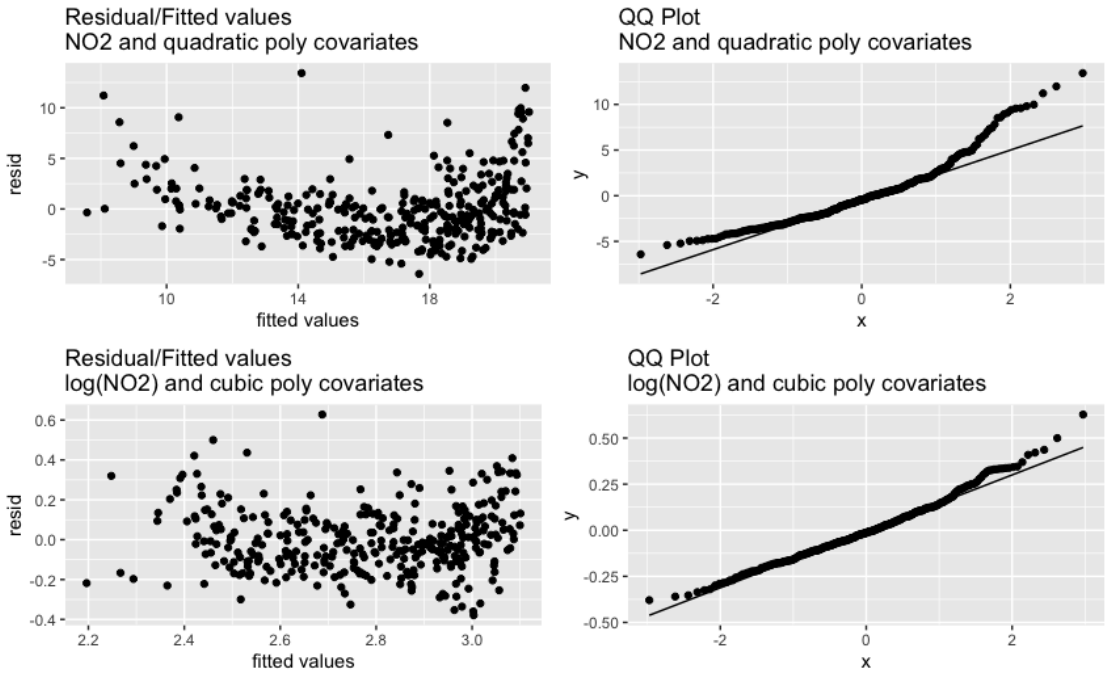

Figure 5: Residuals versus fitted values plots and QQ plots

3 Formal Spatial Analysis

Now I have an idea of what the predictions may look like and a suitable linear regression model that may be useful for including covariate information, I look at a binned sample variogram as this shows how the data changes over different separating distances and then constructing a variogram model allows me to estimate NO_2 values at any separating distance.

As shown in figure 6, taking the cut-off value to be a third of the maximum separating distance (8280) doesn't show us enough of the variogram to be confident where it is levelling off, therefore I looked at larger values to see the shape better and settled on a cut-off value of 9000 and used a bin-width of 300. This gives slightly more points than is perhaps necessary, however it gives a bit more data at the levelling off point, which would appear to be at about a range of 7500. Going forwards a value of 0.01 for the nugget and 0.08 for the partial sill seem appropriate to use to initialise a variogram model.

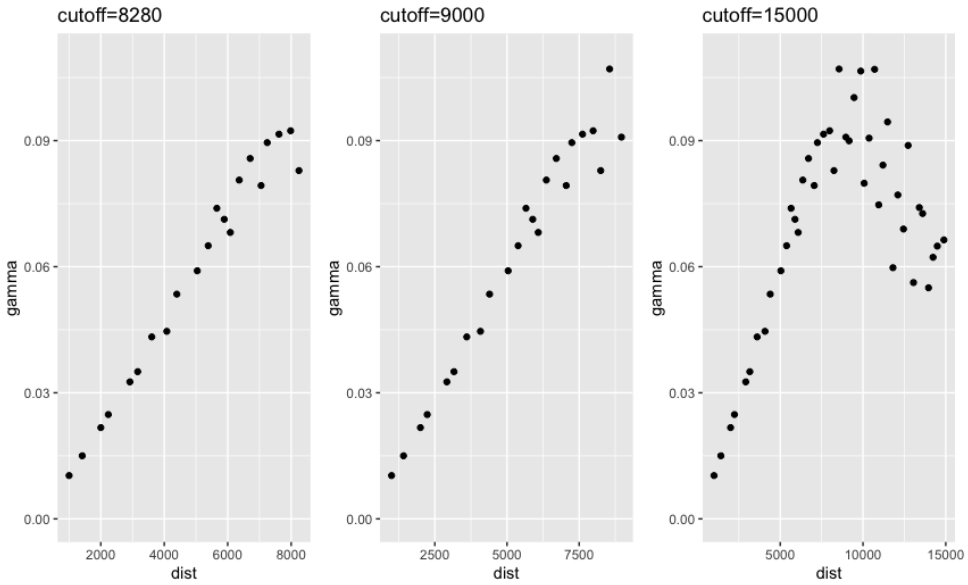

Figure 6: Variogram cut off values

When choosing which type of variogram model to use, exponential and spherical variogram models were not converging, however the gaussian one does converge. Figure 7 shows that the gaussian variogram model fits the sample variogram quite well. However, my linear regression model showed that the covariates added were significant, so I also produced a variogram model using the residuals from the regression model, also shown in figure 7. An exponential variogram model using the residuals is found to follow the sample variogram very well and since the partial sill is smaller, there is less dependence in the residuals compared to the raw data.

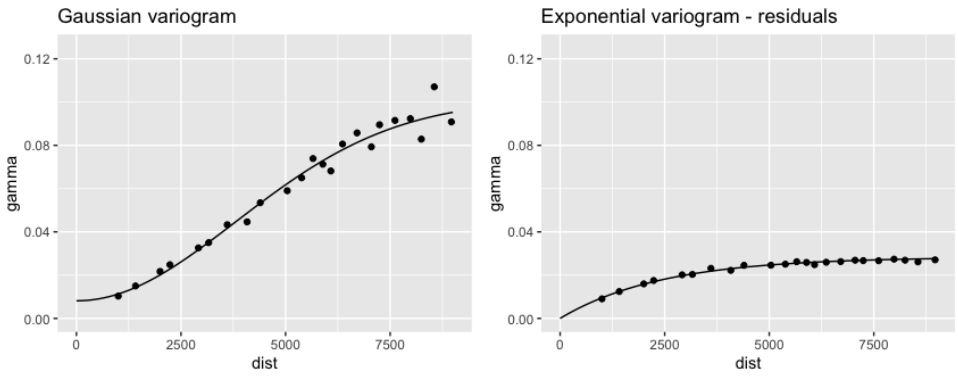

Figure 7: Variogram models

Next, I compare ordinary and universal kriging models using cross validation to form final prediction models of NO_2. The R^2 value for ordinary kriging is 0.835 and for universal kriging it is 0.843. These

values are both high, which is good and both very similar, therefore it seems sensible to use the simpler, ordinary kriging model for predictions. Figure 8 shows a plot of residuals and a QQ-plot to check assumptions of the cross validation residuals. The residual values are fairly evenly dispersed, showing no obvious spatial dependence and the QQ-plot is linear showing the residuals are normally distributed, so I am happy that no model assumptions have been violated. Figure 9 shows the NO_2 predictions for Glasgow area using this model.

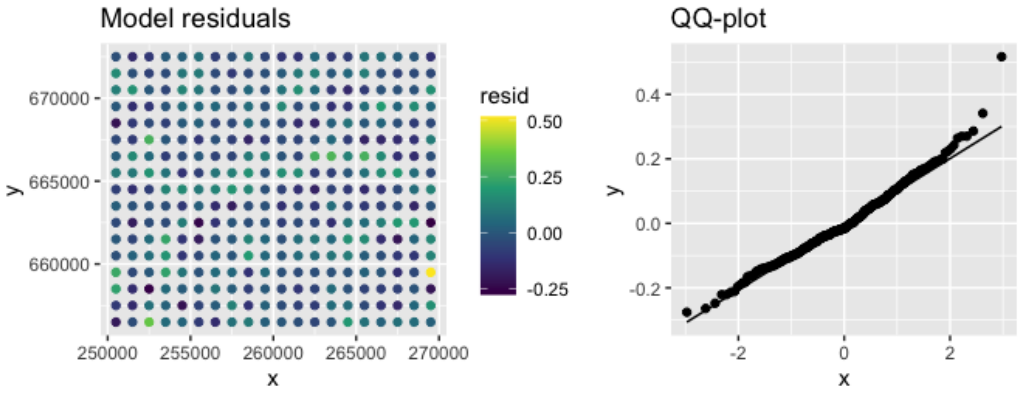

Figure 8: Assumption checking of the cross validation residuals

Now that I have a useful prediction model, I can get NO_2 predictions for a given set of locations where I also have other data that I want to use to build up a model for hospital admissions. In figure 10, I explore the spatial pattern in the respiratory admissions data using SIR values, so I can easily see if areas have more or less admissions than we would expect given their population sizes. The yellow regions show particularly high hospital admissions of at least twice the average, whilst the dark purple regions show areas of low admissions. It is interesting to note that there are areas of very low admissions that are neighbouring areas of high admissions and there's no obvious pattern, other than perhaps generally admissions are lower in the south of the city, but this is not always the case. I don't know Glasgow well but I believe that the purple area north of the river is a university area, where I assume high numbers of students live. There's a particularly high area in the north west and a few high areas in the east of the city. The Moran's I test statistic for the spatial dependence of hospital admissions is fairly large at 0.387 with a p-value of 5.00×10^{-13}. This p-value is very low suggesting there is significant spatial dependence that we need to try to account for.

Before building a model, I want to see if there are any interesting relationships between any of the covariates and respiratory admission values. Figure 11 shows that there is a clear positive, linear relationship between admissions and employment, with a correlation coefficient of 0.612 and also there is a negative relationship between admissions and house price with a correlation coefficient of -0.395. I think you could put a line through this data so I'm not going to transform this variable. The other correlations between admissions and the covariates are low, however, these covariates are considered to be confounding variables so will be included in the model anyway. For NO_2, it has a strong and weaker positive relationship with housing and house price. Also, it has a fairly strong, negative relationship with access and a weaker negative relationship with respiratory admissions, which is surprising, as we might expect a positive relationship, where hospital admissions would be higher where there are higher NO_2 values.

On checking for highly correlated covariates, house price and employment are strongly (negatively) correlated at -0.712, however this is not so high that we should exclude one from the model. There's no obvious pattern in these plots either so I'm not going to transform any of the variables at this point.

On running a Poisson regression model with all the covariates shown in figure 11, the Gelman Rubin diagnostic test shows issues with convergence, with a beta0 of 1.40 and a high beta4 value of 1.19. I immediately mean centre my covariates to see if this helps avoid correlation issues and my Gelman

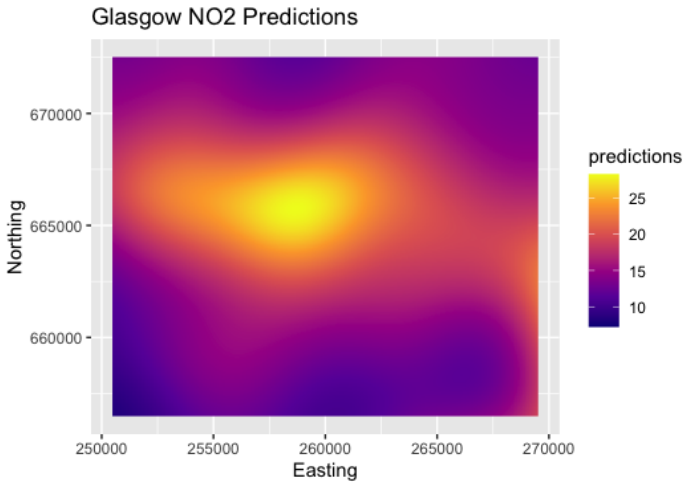

Figure 9: NO_2 predictions for Glasgow area

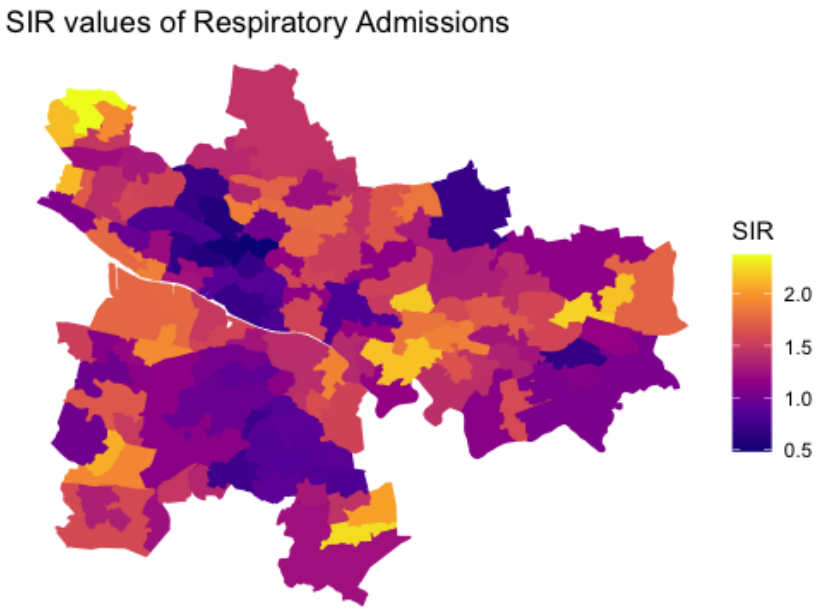

Figure 10: SIR values of respiratory hospital admissions

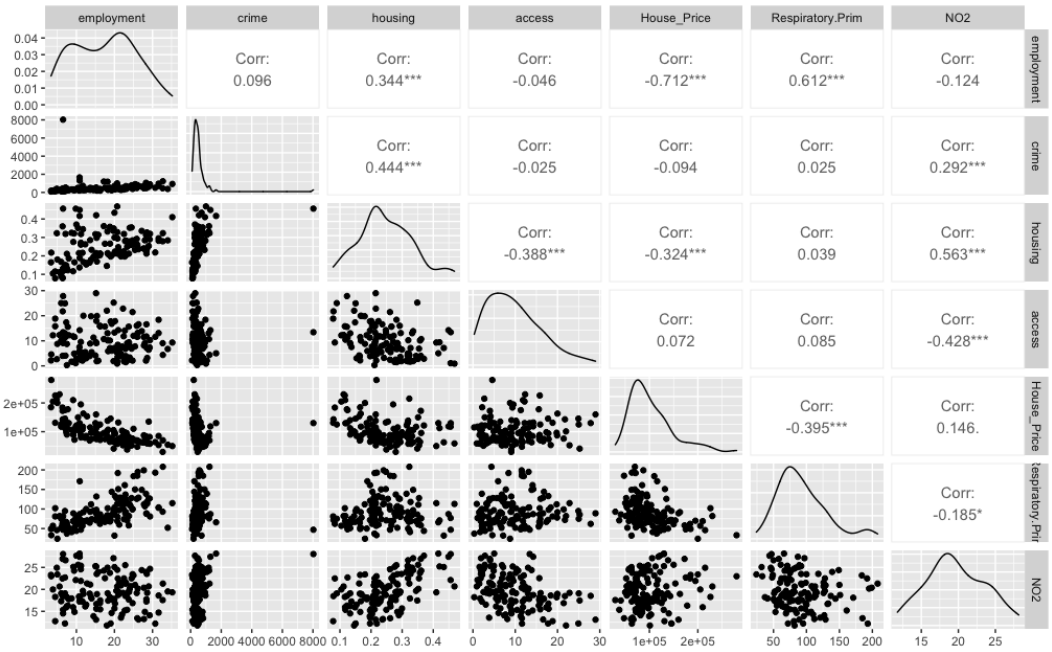

Figure 11: Correlations between the covariates

Rubin diagnostic test now shows 1.00 for all except for beta3 which is 1.01, so that suggests that it has converged. Looking at traceplots also shows convergence has happened, figure 12.

Checking the model assumption of mean=variance, my variance of the residuals is 3.40 which is quite small but still bigger than 1, suggesting I have slight overdispersion. In figure 13, it looks like the residuals are funnelling in, showing underdispersion, however, the values on the y-axis are not so very small, so since the variance of the residuals of 3.40 suggests very slightly over-dispersion and the plot looks slightly under-dispersed, but with fairly high residual values, then I think overall, the mean=variance assumption must be fairly close to being valid. I plotted graphs of residuals versus covariates to look for any relationship that had been missed from the model but there are no obvious trends to be seen, figure 14.

For the independence assumption, I do a Moran's I test. The Moran's I statistic is low at 0.099 with a p-value of 0.051. The p-value is just showing spatial independence, so continuing by forming a Poisson CAR model is a good idea, in case we can remove more spatial dependence. I'll include the same mean centred covariates in the Poisson CAR model as I included in the Poisson regression model. The Gelman Rubin diagnostic test shows 1.00 for most, a few are 1.01, so this test suggests convergence. The traceplots found in fig 15, look generally fine also, although a little bit of correlation in the chains can be seen, but there are no issues with convergence.

To check the Poisson CAR model assumptions, I again extract the Pearson residuals and look at the variance of the residuals and the residuals versus fitted plot for the mean=variance assumption and Moran's I test statistic for the independence assumption. The variance is 0.368, which is about 10 times smaller than for the Poisson regression model. It is closer to 1 than the Poisson regression model was but it suggests there is slight under-dispersion. Figure 16 of the residuals versus fitted plot also shows under-dispersion due to the funnelling in effect. So both tests for the mean=variance are showing signs of under-dispersion this time, which indicates that the model has possibly been overfit. An alternative model type may be preferable such as the Quasi-Poisson or the Negative Binomial. For the independence assumption check, Moran's I is -0.121 and it's p-value is 0.037, which shows that there is still spatial dependence that is being unaccounted for.

To compare the 2 models, I want to look at the DIC and pD values, along with the assumption information that we have. Table 1 shows that the DIC is lowest for the Poisson CAR model (1090) but the complexity (92.1) is also higher, Pearson's residual variance is closer to 1 for the Poisson CAR model too (0.368). The Moran's I statistic is slightly smaller for the Poisson regression model though (0.099) and with a p-value of 0.051, just shows spatial independence, whilst the Poisson CAR model still showed spatial dependence (-0.121 with p-value of 0.037). Whilst it is difficult to choose between

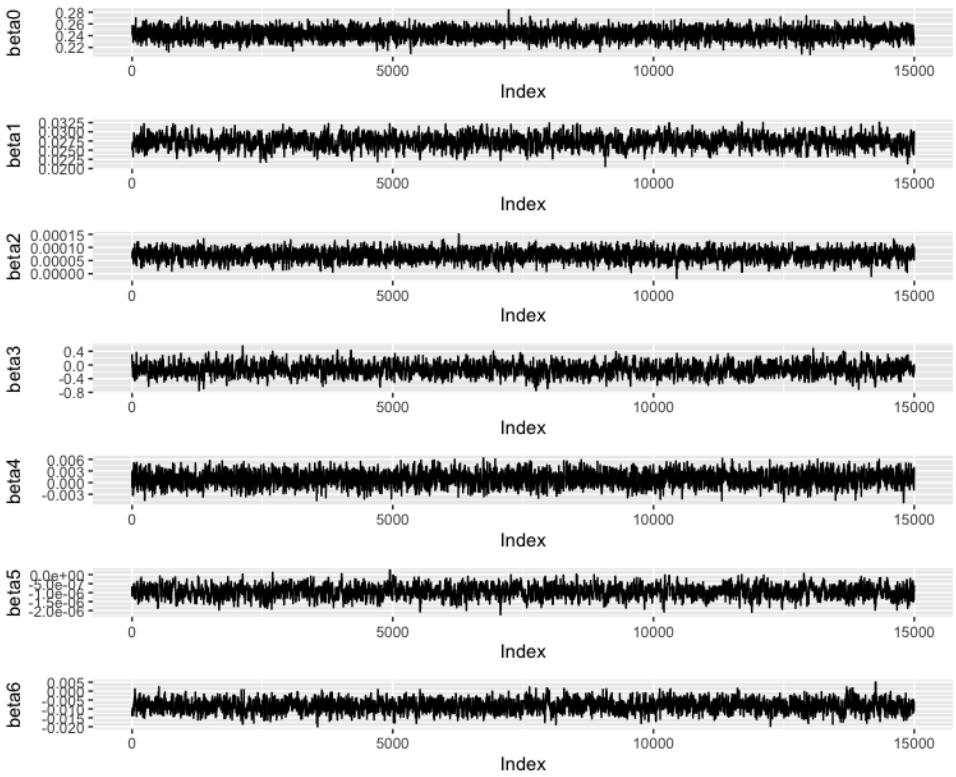

Figure 12: Traceplots of Poisson regression model

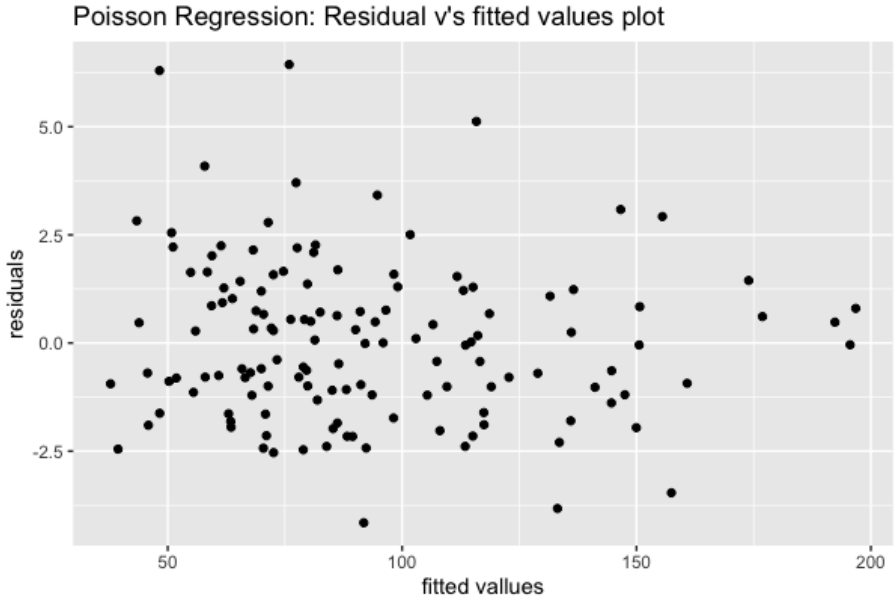

Figure 13: Poisson Regression

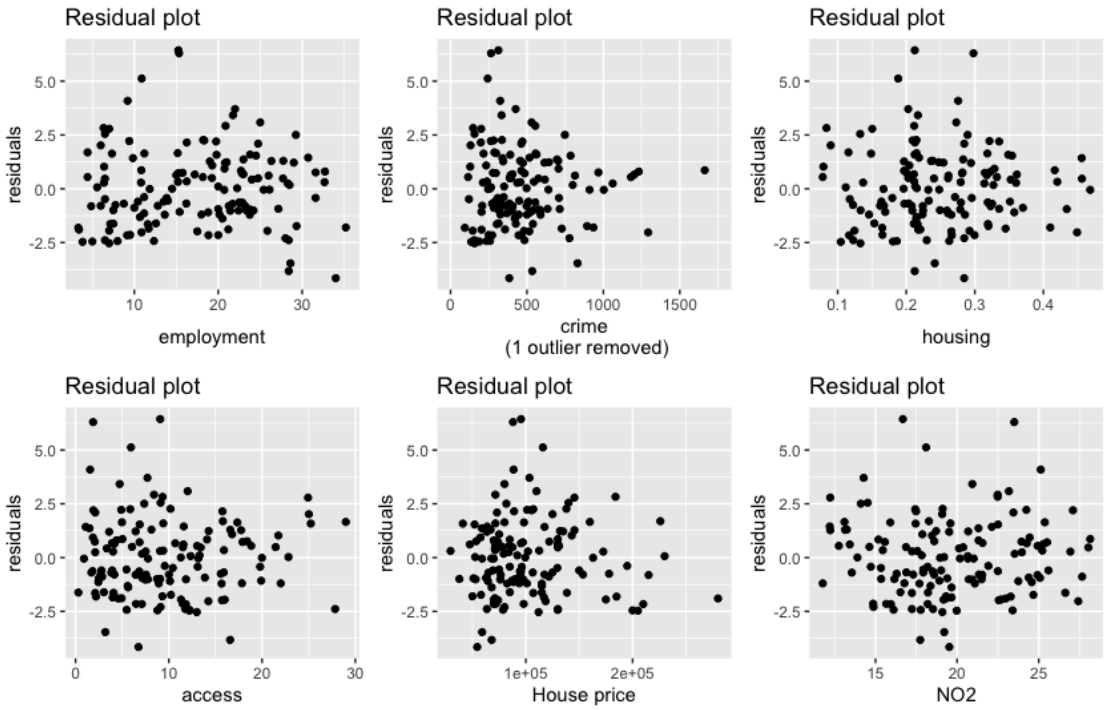

Figure 14: Plots of model residuals versus covariates to look for missed trends

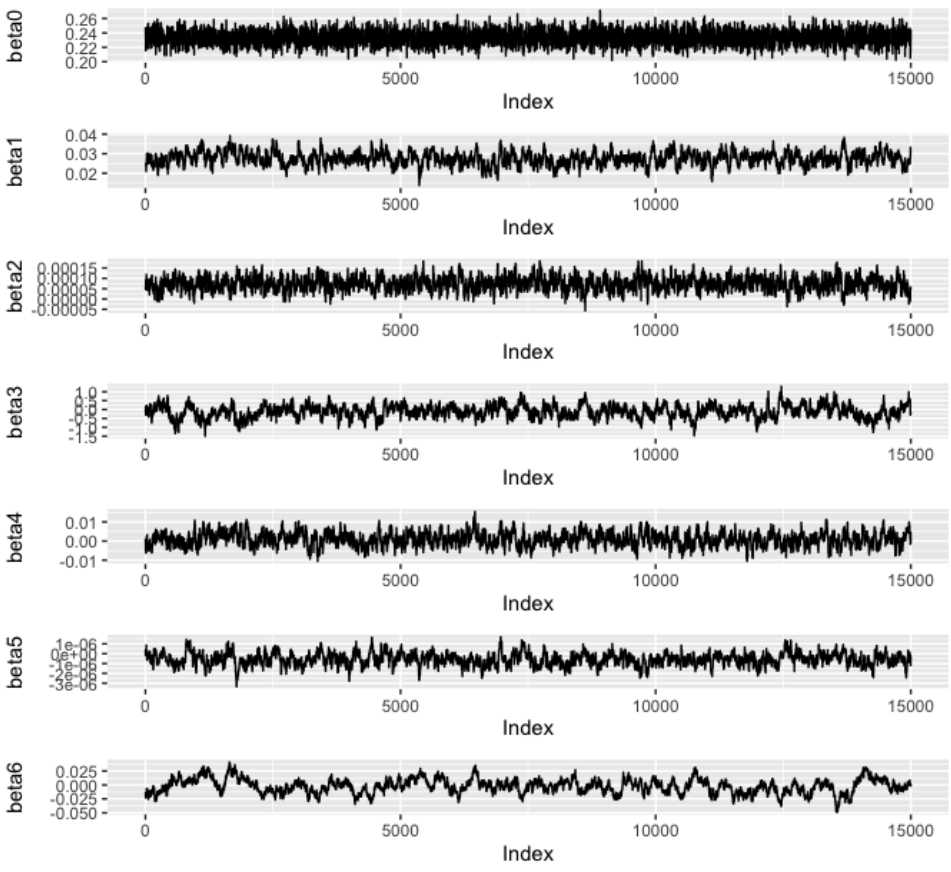

Figure 15: Traceplots for Poisson CAR model

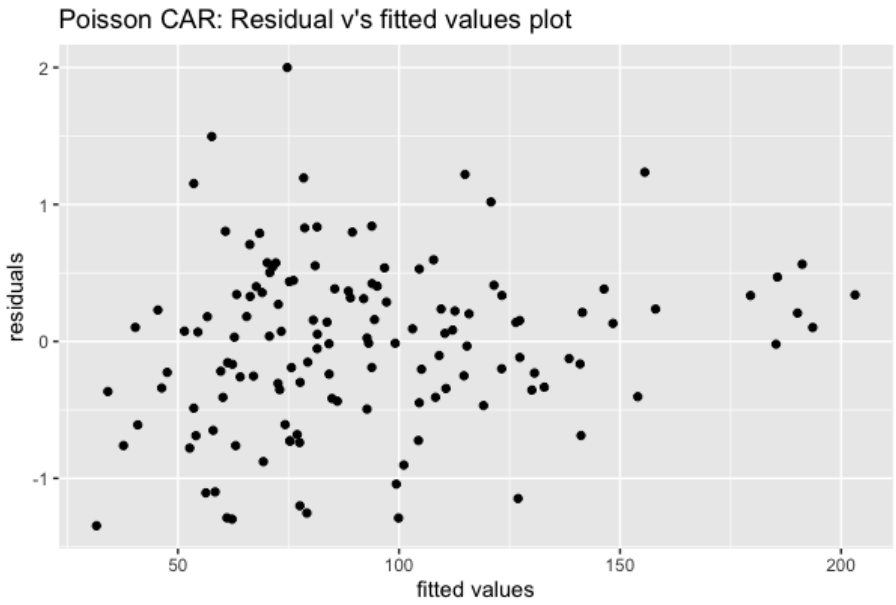

Figure 16: CAR model residuals versus fitted values

the 2 models, I believe that the Poisson regression model may be better since the Moran's I test for independence shows that the model is spatially independent, albeit only just and the mean=variance assumption was just about ok. So I think I would be more comfortable using this model.

Table 1: Table of values (to 3 s.f.

Model	Pearson's variance	Moran's I	P-value	pD	DIC
Poisson Regression	3.40	0.099	0.051	7.04	1310
Poisson CAR	0.368	-0.121	0.037	92.1	1090

Table 2: Values of model coefficients and their credibility intervals

Coefficient	Mean	95% lower CI	95% upper CI
beta0	1.275	1.253	1.297
beta1	1.028	1.024	1.031
beta2	1.000	1.000	1.000
beta3	0.884	0.635	1.241
beta4	1.001	0.998	1.004
beta5	1.000	1.000	1.000
beta6	0.992	0.986	0.998

Looking at the coefficient estimates for the Poisson regression model in table 2, beta6 is the coefficient for NO_2. This shows that for every unit increase in NO_2 in an area, there is a decrease in risk of hospital admissions of 0.8% but this could be anything between a 0.2% and 1.4% decrease in risk with 95% probability. This is very unexpected! However, it perhaps suggests that there are more important indicators of hospital admission due to respiratory illness. It would appear that employment deprivation, (beta1) is related to the biggest increase in risk of hospital admissions. For each unit increase in employment deprivation, there is an increase in risk of hospital admission of 2.8%, though this could be anywhere between 2.4% and 3.1% with 95% probability.

R code used for analysis

```
library(gstat)
library(sp)
library(dplyr)
library(ggplot2)
library(patchwork)
library(latticeExtra)
library(nimble)
library(rgdal)
library(sf)
library(spdep)
library(GGally)
library(coda)

#Read in data
load("scot_coords.RData")
polts <- read.csv("Pollutants.csv")
adm <- read.csv("Admissions.csv")

#Organising data
#remove PM10 data
polts <- select(polts,-PM10)
summary(polts)
nrow(polts)
#filter for just the data from 2012
polts <- filter(polts, year==2012)
nrow(polts)
summary(polts)

nrow(adm)
adm <- filter(adm, Year==2012)
nrow(adm)
summary(adm)

###
#Look at data on map
ggplot(polts)+
  #add border for scotland
geom_polygon(data=as.data.frame(scot_coords), aes(scot_coords[,1], scot_coords[,2]),
             fill="white", colour="black")+
  geom_point(aes(x,y, colour=NO2))+
  scale_colour_viridis_c()+
  coord_equal()+
  ggtitle("Location Map")+
  # There is an observation that is not from Scotland
  scale_y_continuous(limits=range(scot_coords[,2])) +
  theme(axis.text=element_blank(),
        axis.ticks=element_blank(),
        axis.title=element_blank()) -> p1

ggplot(polts)+
  geom_point(aes(x,y, colour=NO2))+
  scale_colour_viridis_c()+
  coord_equal() +
  ggtitle("NO2 observations")-> p2

p1+p2


#Create spatial object
polts.sp <- polts
coordinates(polts.sp) <- c("x", "y")

#Create grid of predictions locations
min(polts$x); max(polts$x)
min(polts$y); max(polts$y)
```

```
xseq <- seq(250500, 269500, length.out= 500)
yseq <- seq(656500, 672500, length.out= 500)
xygrid <- expand.grid(xseq=xseq, yseq=yseq)
grid.sp <- xygrid
coordinates(grid.sp) <- c("xseq", "yseq")

#IDW interpolation
idw1 <- idw(NO2~1, polts.sp, grid.sp, idp=1)
idw2 <- idw(NO2~1, polts.sp, grid.sp, idp=5)
idw3 <- idw(NO2~1, polts.sp, grid.sp, idp=10)
#idw4 <- idw(NO2~1, polts.sp, grid.sp, idp=6)
#idw5 <- idw(NO2~1, polts.sp, grid.sp, idp=5)
#idw8 <- idw(NO2~1, polts.sp, grid.sp, idp=8)

xygrid$pred1 <- idw1$var1.pred
xygrid$pred2 <- idw2$var1.pred
xygrid$pred3 <- idw3$var1.pred
#xygrid$pred4 <- idw4$var1.pred
#xygrid$pred5 <- idw4$var1.pred
#xygrid$pred8 <- idw4$var1.pred

ggplot(xygrid)+
  geom_raster(aes(xseq, yseq, fill=pred1))+
  scale_fill_viridis_c(option="C", limits=range(xygrid$pred1, xygrid$pred2,
                                                xygrid$pred3, xygrid$pred4), name="Prediction")+
  coord_equal()+
  ggtitle("p=1") -> ipred1

ggplot(xygrid)+
  geom_raster(aes(xseq, yseq, fill=pred2))+
  scale_fill_viridis_c(option="C", limits=range(xygrid$pred1, xygrid$pred2,
                                                xygrid$pred3, xygrid$pred4), name="Prediction")+
  coord_equal()+
  ggtitle("p=5") -> ipred2

ggplot(xygrid)+
  geom_raster(aes(xseq, yseq, fill=pred3))+
  scale_fill_viridis_c(option="C", limits=range(xygrid$pred1, xygrid$pred2,
                                                xygrid$pred3, xygrid$pred4), name="Prediction")+
  coord_equal()+
  ggtitle("p=10") -> ipred3

#ggplot(xygrid)+
 # geom_raster(aes(xseq, yseq, fill=pred4))+
  #scale_fill_viridis_c(option="C", limits=range(xygrid$pred1, xygrid$pred2,
   #                                             xygrid$pred3, xygrid$pred4), name="Prediction")+
  #coord_equal()+
  #ggtitle("p=6") -> ipred4

ipred1|ipred2|ipred3|plot_layout(guides="collect")

### Consider linear regression for prediction
ggplot(polts)+
  geom_point(aes(x, NO2))+
  ggtitle("easting")->p1a
ggplot(polts)+
  geom_point(aes(y, NO2))+
  ggtitle("northing")->p1b
p1a+p1b

#(Effectively) use backwards selection
mod1 <- lm(NO2 ~ poly(x, 2) + poly(y, 2), data=polts)
summary(mod1)

#Check model residuals
polts$resid <- resid(mod1)
```

```
ggplot(polts)+
  geom_point(aes(x,y, colour=resid))+
  scale_colour_viridis_c()+
  coord_equal()+
  ggtitle("Residuals with NO2")->P1
P1

ggplot(polts)+
  geom_point(aes(x, resid))+
ggtitle("Residual Plots")-> p2a
ggplot(polts)+
  geom_point(aes(y, resid)) -> p2b

p2a+p2b

#try log(NO2)
mod2 <- lm(log(NO2) ~ poly(x, 2) + poly(y, 2), data=polts)
summary(mod2)

polts$resid <- resid(mod2)
ggplot(polts)+
  geom_point(aes(x,y, colour=resid))+
  scale_colour_viridis_c()+
  coord_equal()+
  ggtitle("Residuals with log(NO2)")->P2

P1+P2

#Return to using NO2 model rather than log(NO2)
mod1 <- lm(NO2 ~ poly(x, 2) + poly(y, 2), data=polts)
polts$resid <- resid(mod1)

ggplot(polts)+
  geom_point(aes(x, resid))+
  ggtitle("Residual Plots")-> p1
ggplot(polts)+
  geom_point(aes(y, resid)) -> p2

#Check residuals v fitted and qq plot for further assumption check of NO2 model with quad poly terms
ggplot(polts)+
  geom_point(aes(mod1$fitted.values, resid))+
  xlab("fitted values")+
  ggtitle("Residual/Fitted values\nNO2 and quadratic poly covariates")-> pl1

ggplot(polts)+
  geom_qq(aes(sample=resid))+
  geom_qq_line(aes(sample=resid))+
  ggtitle("QQ Plot\nNO2 and quadratic poly covariates")-> pl2

pl1+pl2

mod3 <- lm(log(NO2) ~ poly(x, 3) + poly(y, 3), data=polts)
polts$resid <- resid(mod3)
polts.sp$resid <- resid(mod3)

ggplot(polts)+
  geom_point(aes(x, resid))+
  ggtitle("Residual Plots")-> p3
ggplot(polts)+
  geom_point(aes(y, resid)) -> p4

(p1+p2)/(p3+p4)

#Check residuals v fitted and qq plot for further assumption check log(NO2) and cubic poly terms
ggplot(polts)+
  geom_point(aes(mod3$fitted.values, resid))+
  xlab("fitted values")+
```

```
  ggtitle("Residual/Fitted values\nlog(NO2) and cubic poly covariates")-> pl3

ggplot(polts)+
  geom_qq(aes(sample=resid))+
  geom_qq_line(aes(sample=resid))+
  ggtitle("QQ Plot\nlog(NO2) and cubic poly covariates")-> pl4

(pl1+pl2)/(pl3+pl4)

### Sample variogram
#set cutoff value to 1/3 max separating value
xran <- range(polts$x)
yran <- range(polts$y)
dist <- sqrt(diff(xran)^2 + diff(yran)^2)
cutoff <- max(dist)/3
cutoff # ~ 8280

#choose bin width
vgm1 <- variogram(log(NO2)~1, polts.sp, cutoff=cutoff, width=300)
vgm2 <- variogram(log(NO2)~1, polts.sp, cutoff=9000, width=300)
vgm3 <- variogram(log(NO2)~1, polts.sp, cutoff=15000, width=300)

ggplot(vgm1)+
  geom_point(aes(dist, gamma))+
  scale_y_continuous(limits=c(0,0.11))+
  ggtitle("cutoff=8280") ->c1
ggplot(vgm2)+
  geom_point(aes(dist, gamma))+
  scale_y_continuous(limits=c(0,0.11))+
  ggtitle("cutoff=9000") ->c2
ggplot(vgm3)+
  geom_point(aes(dist, gamma))+
  scale_y_continuous(limits=c(0,0.11))+
  ggtitle("cutoff=15000") ->c3

c1+c2+c3

#Variogram model - which type - try exponential and spherical first - using vgm2
expon <- fit.variogram(vgm2, vgm(0.8, "Exp",7500, 0.01)) #no convergence
spher <- fit.variogram(vgm2, vgm(0.8, "Sph",7500, 0.01)) #no convergence
gaus <- fit.variogram(vgm2, vgm(0.8, "Gau",7500, 0.01)) #converges

#Check how well variogram model fits to sample variogram
ggplot(vgm2, aes(dist, gamma))+
  geom_point()+
  scale_y_continuous(limits=c(0, 0.125))+
  geom_line(data=variogramLine(gaus, maxdist=9000))+
  ggtitle("Gaussian variogram") -> g1
g1

vgm_resid <- variogram(resid~1, polts.sp, cutoff=9000, width=300)
ggplot(vgm_resid)+
  geom_point(aes(dist, gamma))+
  scale_y_continuous(limits=c(0, NA))+
  ggtitle("Sample variogram of regression model residuals")

max(vgm_resid$gamma) #help work out sill

exponr <- fit.variogram(vgm_resid, vgm(0.017, "Exp",5500, 0.009)) #converges
spherr <- fit.variogram(vgm_resid, vgm(0.017, "Sph",5500, 0.009)) #converges
gausr <- fit.variogram(vgm_resid, vgm(0.017, "Gau",5500, 0.009)) #no convergence

#Check how well variogram model fits to sample variogram
ggplot(vgm_resid, aes(dist, gamma))+
  geom_point()+
  scale_y_continuous(limits=c(0, 0.125))+
  geom_line(data=variogramLine(exponr, maxdist=9000))+
```

```
  ggtitle("Exponential variogram - residuals") -> g2
g1+g2

### Kriging ###
#set up no. of folds
nfold <- 10
#set up data partitions
part <- sample(1:nfold, nrow(polts), replace=T)
#set up somewhere to store the results for each model
ok <- numeric(nrow(polts))
uk <- numeric(nrow(polts))
for(i in 1:nfold){
  #extract the partitions
  modelling <- polts.sp[part!=i,]
  test <- polts.sp[part==i,]

#Ordinary Kriging
vgm_ok <- variogram(log(NO2)~1, modelling) #fit to modelling set
vfit_ok <- fit.variogram(vgm_ok, vgm(0.08, "Gau", 7500, 0.01))
ok[part==i] <- krige(log(NO2)~1, modelling, test, vfit_ok)$var1.pred

#Universal kriging
vgm_uk <- variogram(log(NO2) ~ poly(x,3) + poly(y, 3), modelling) #fit to modelling set
vfit_uk <- fit.variogram(vgm_uk, vgm(0.017, "Exp",5500, 0.009))
uk[part==i] <- krige(log(NO2) ~ poly(x,3) + poly(y, 3), modelling, test, vfit_uk)$var1.pred
}
#Find R^2 value to assess fit
1-sum((ok-log(polts$NO2))^2)/sum((log(polts$NO2) - mean(log(polts$NO2)))^2) #0.841058
1-sum((uk-log(polts$NO2))^2)/sum((log(polts$NO2) - mean(log(polts$NO2)))^2) #-0.7609823

#Use krige.cv function to check results
#Ordinary Kriging
vgm_okcv <- variogram(log(NO2)~1, polts.sp)
vfit_okcv <- fit.variogram(vgm_okcv, vgm(0.08, "Gau", 7500, 0.01))
okcv <- krige.cv(log(NO2)~1, polts.sp, vfit_okcv, nfold=10)

1-sum(okcv$residual^2)/sum((log(polts$NO2) - mean(log(polts$NO2)))^2) #0.8354685

#Universal Kriging
vgm_ukcv <- variogram(log(NO2)~1, polts.sp)
vfit_ukcv <- fit.variogram(vgm_ukcv, vgm(0.08, "Gau", 7500, 0.01))
ukcv <- krige.cv(log(NO2)~1, polts.sp, vfit_ukcv, nfold=10)

1-sum(ukcv$residual^2)/sum((log(polts$NO2) - mean(log(polts$NO2)))^2) #0.843031

#Plot residuals and QQ plot to check assumptions of CV residuals
ggplot(polts)+
  geom_point(aes(x, y, colour=okcv$residual))+
  labs(colour="resid")+
  scale_colour_viridis_c()+
  coord_equal()+
  ggtitle("Model residuals")-> cv1

ggplot()+
  geom_qq(aes(sample=okcv$residual))+
  geom_qq_line(aes(sample=okcv$residual))+
  ggtitle("QQ-plot")-> cv2

cv1+cv2

#Make Predictions
#Log NO2 predictions
ok_log_pred <- krige(log(NO2)~1, polts.sp, grid.sp, vfit_okcv)
#Converting to NO2 predictions
ok_pred$var1.pred <- exp(ok_log_pred$var1.pred)

#Plot NO2 predictions
```

```
ggplot(xygrid)+
  geom_raster(aes(xseq,yseq, fill=ok_pred$var1.pred))+
  scale_fill_viridis_c(option="C")+
  coord_equal() +
  labs(fill="predictions", x="Easting", y="Northing")+
  ggtitle("Glasgow NO2 Predictions")

########### Part 2 ##################
#Predictions are here: ok_pred$var1.pred

#Read in shapefile
shape <- readOGR("Shapefile.shp")

coord <- coordinates(shape)
coord <- as.data.frame(coord)
names(coord) <- c("x", "y")
coordinates(coord) <- c("x","y")

#Make predictions with new coordinates
#Log NO2 predictions
log_predictions <- krige(log(NO2)~1, polts.sp, coord, vfit_okcv)
#Converting to NO2 predictions
predictions <- exp(log_predictions$var1.pred)

#add to admissions data then merge both to shape data
adm$NO2 <- predictions
shape@data <- merge(shape@data, adm, by.x="InterZone", by.y="IZ", sort=FALSE)

#Add SIR column to data
shape$SIR <- shape$Respiratory.Prim/shape$Expected

#Plot SIR
shape.sf <- st_as_sf(shape)
ggplot(shape.sf)+
  geom_sf(aes(fill=SIR, colour=SIR))+
  scale_fill_viridis_c(option="C")+
  scale_colour_viridis_c(option="C")+
  theme_void()+
  ggtitle("SIR values of Respiratory Admissions")

#Moran's I test for spatial depepndence
moran.test(shape$SIR, nb2listw(poly2nb(shape)), alternative="two.sided")

#Explore relationships between admissions and covariates
ggpairs(shape@data[,c(12:17, 19)])

#Fit a poisson regression model
#model code
preg_code <- nimbleCode({
  for(i in 1:N){
    Y[i] ~ dpois(mu[i])
    log(mu[i]) <- log(E[i])+beta0+beta1*X1[i]+beta2*X2[i]+beta3*X3[i]+beta4*X4[i]+beta5*X5[i]+beta6*X6[i]
  }
  beta0 ~ dnorm(0,0.01)
  beta1 ~ dnorm(0,0.01)
  beta2 ~ dnorm(0,0.01)
  beta3 ~ dnorm(0,0.01)
  beta4 ~ dnorm(0,0.01)
  beta5 ~ dnorm(0,0.01)
  beta6 ~ dnorm(0,0.01)
})

Data <- list(Y =shape$Respiratory.Prim,
             E =shape$Expected,
             X1=shape$employment-mean(shape$employment),
             X2=shape$crime-mean(shape$crime),
             X3=shape$housing-mean(shape$housing),
```

```
            X4=shape$access-mean(shape$access),
            X5=shape$House_Price-mean(shape$House_Price),
            X6=shape$NO2-mean(shape$NO2))

Constants <- list(N=nrow(shape))

Inits <- list(list(beta0=0,beta1=0, beta2=0, beta3=0, beta4=0, beta5=0, beta6=0),
              list(beta0=0,beta1=0, beta2=0, beta3=0, beta4=0, beta5=0, beta6=0),
              list(beta0=0,beta1=0, beta2=0, beta3=0, beta4=0, beta5=0, beta6=0))

#Run the model
preg <- nimbleMCMC(data=Data,
                   constants=Constants,
                   code=preg_code,
                   monitors=c(paste0("beta",0:6), "mu"),
                   inits=Inits,
                   nchains=3,
                   niter=10000,
                   nburnin=5000,
                   summary=TRUE,
                   samplesAsCodaMCMC = TRUE)

#subset to remove mu's
for_diag <- lapply(preg$samples, function(x) x[,grepl("beta", names(x[1,]))])

#gelman diagnostic
gelman.diag(for_diag)

#check traceplots
samples <- as.data.frame(Reduce("rbind", preg$samples))

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta0))+
  labs(x="Index") -> p0

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta1))+
  labs(x="Index") -> p1

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta2))+
  labs(x="Index") -> p2

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta3))+
  labs(x="Index") -> p3

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta4))+
  labs(x="Index") -> p4

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta5))+
  labs(x="Index") -> p5

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta6))+
  labs(x="Index") -> p6

p0/p1/p2/p3/p4/p5/p6

#Check poisson model assumptions
mu_preg <- preg$summary$all.chains[grepl("mu", rownames(preg$summary$all.chains)),1]
res_preg <- (Data$Y - mu_preg)/sqrt(mu_preg)

#mean=variance
var(res_preg)
```

```
ggplot()+
  geom_point(aes(mu_preg, res_preg))+
  xlab("fitted vallues")+
  ylab("residuals")+
  ggtitle("Poisson Regression: Residual v's fitted values plot")

#independence assumption
moran.test(res_preg, nb2listw(poly2nb(shape)), alternative="two.sided")

#Plot residuals against covariates to see if any missed patterns
ggplot()+
  geom_point(aes(shape$employment, res_preg))+
  xlab("employment")+
  ylab("residuals")+
  ggtitle("Residual plot") -> a1

ggplot()+
  geom_point(aes(shape$crime, res_preg))+
  xlab("crime\n (1 outlier removed)")+
  ylab("residuals")+
  scale_x_continuous(limits = c(0, 1750))+
  ggtitle("Residual plot") -> a2

ggplot()+
  geom_point(aes(shape$housing, res_preg))+
  xlab("housing")+
  ylab("residuals")+
  ggtitle("Residual plot") -> a3

ggplot()+
  geom_point(aes(shape$access, res_preg))+
  xlab("access")+
  ylab("residuals")+
  ggtitle("Residual plot") -> a4

ggplot()+
  geom_point(aes(shape$House_Price, res_preg))+
  xlab("House price")+
  ylab("residuals")+
  ggtitle("Residual plot") -> a5

ggplot()+
  geom_point(aes(shape$NO2, res_preg))+
  xlab("NO2")+
  ylab("residuals")+
  ggtitle("Residual plot") -> a6

(a1+a2+a3)/(a4+a5+a6)

##Finding DIC and pD for Poisson regression
mu <- apply(samples[,grepl("mu", names(samples))], 2, median)
deviance <- sum(log(dpois(shape$Respiratory.Prim, lambda=mu)))
mu_samp <- samples[,grepl("mu", names(samples))]
ave_dev <- mean(
  sapply(1:nrow(samples),
         function(x){sum(log(dpois(shape$Respiratory.Prim,
                                   lambda=as.numeric(mu_samp[x,]))))}))

pD <- 2*(deviance-ave_dev)
DIC <- -2*deviance+2*pD
pD;DIC

####  Poisson CAR model #######
W <- nb2mat(poly2nb(shape), style="B")
Adj <- lapply(1:nrow(W), function(x) which(W[x,]==1))
Adj <- Reduce("c", Adj)
```

```
Num.Adj <- apply(W, 1, sum)
L <- length(Adj)

pCAR_code <- nimbleCode({
  for(i in 1:N){
    Y[i] ~ dpois(mu[i])
    log(mu[i])<-log(E[i])+beta0+beta1*X1[i]+beta2*X2[i]+beta3*X3[i]+beta4*X4[i]+beta5*X5[i]+beta6*X6[i]+phi[i]
  }
  phi[1:N] ~ dcar_normal(Adj[1:L], weights[1:L], Num.Adj[1:N], tau, zero_mean=1)
  tau ~ dgamma(0.01, 0.01)
  beta0 ~ dnorm(0,0.01)
  beta1 ~ dnorm(0,0.01)
  beta2 ~ dnorm(0,0.01)
  beta3 ~ dnorm(0,0.01)
  beta4 ~ dnorm(0,0.01)
  beta5 ~ dnorm(0,0.01)
  beta6 ~ dnorm(0,0.01)
})

#same Data as before can be used

Constants <- list(N=nrow(shape),
                  L=L,
                  Adj=Adj,
                  weights=rep(1, L),
                  Num.Adj=Num.Adj)

Inits <- list(list(beta0=0,beta1=0, beta2=0, beta3=0, beta4=0, beta5=0, beta6=0, tau=1, phi=rep(1, nrow(shape))),
              list(beta0=0,beta1=0, beta2=0, beta3=0, beta4=0, beta5=0, beta6=0, tau=1, phi=rep(1, nrow(shape))),
              list(beta0=0,beta1=0, beta2=0, beta3=0, beta4=0, beta5=0, beta6=0, tau=1, phi=rep(1, nrow(shape))))

#Run CAR model
pCAR <- nimbleMCMC(data=Data,
                   constants=Constants,
                   code=pCAR_code,
                   monitors=c(paste0("beta", 0:6), "phi", "mu"),
                   inits=Inits,
                   nchains=3,
                   niter=10000,
                   nburnin=5000,
                   summary=TRUE,
                   samplesAsCodaMCMC = TRUE)

#Check convergence
#subset to remove mu's
for_diag <- lapply(pCAR$samples, function(x) x[,grepl("mu", names(x[1,]))])

#gelman diagnostic
gelman.diag(for_diag)

#check traceplots
samples <- as.data.frame(Reduce("rbind", pCAR$samples))

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta0))+
  labs(x="Index") -> n0

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta1))+
  labs(x="Index") -> n1

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta2))+
  labs(x="Index") -> n2

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta3))+
```

```
  labs(x="Index") -> n3

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta4))+
  labs(x="Index") -> n4

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta5))+
  labs(x="Index") -> n5

ggplot(samples)+
  geom_line(aes(1:nrow(samples), beta6))+
  labs(x="Index") -> n6

n0/n1/n2/n3/n4/n5/n6

#Check poisson CAR model assumptions
mu_pCAR <- pCAR$summary$all.chains[grepl("mu", rownames(pCAR$summary$all.chains)),1]
res_pCAR <- (Data$Y - mu_pCAR)/sqrt(mu_pCAR)

#mean=variance
var(res_pCAR)

ggplot()+
  geom_point(aes(mu_pCAR, res_pCAR))+
  xlab("fitted values")+
  ylab("residuals")+
  ggtitle("Poisson CAR: Residual v's fitted values plot")

#independence assumption
moran.test(res_pCAR, nb2listw(poly2nb(shape)), alternative="two.sided")

##Finding DIC and pD for Poisson CAR
mu <- apply(samples[,grepl("mu", names(samples))], 2, median)
deviance <- sum(log(dpois(shape$Respiratory.Prim, lambda=mu)))
mu_samp <- samples[,grepl("mu", names(samples))]
ave_dev <- mean(
  sapply(1:nrow(samples),
         function(x){sum(log(dpois(shape$Respiratory.Prim,
                                   lambda=as.numeric(mu_samp[x,])))))}))

pD <- 2*(deviance-ave_dev)
DIC <- -2*deviance+2*pD
pD;DIC

### Going back to Poisson regression model ###
exp(preg$summary$all.chains[grepl("beta", rownames(preg$summary$all.chains)), c(1,4,5)])
```