wk8p

2023-12-11

In the worked example, we will explore the factors that might affect the average exam scores of 16 year-old across London. GSCEs are the exams taken at the end of secondary education and here have been aggregated for all pupils at their home addresses across the City for Ward geographies. This practical will walk you through the common steps that you should go through when building a regression model using spatial data to test a stated research hypothesis; from carrying out some descriptive visualisation and summary statistics, to interpreting the results and using the outputs of the model to inform your next steps. It will first cover linear regression which you may have covered in other modules. It will then move to spatial regression models. # Setting up your Data let's set up R and read in some data to enable us to carry out our analysis

```
#library a bunch of packages we may (or may not) use - install them first if not installed already.
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(tmap)
```

```
## Breaking News: tmap 3.x is retiring. Please test v4, e.g. with
## remotes::install_github('r-tmap/tmap')
```

```
library(geojsonio)
```

```
## Registered S3 method overwritten by 'geojsonsf':
##   method        from
##   print.geojson geojson
##
## Attaching package: 'geojsonio'
##
## The following object is masked from 'package:base':
##
##     pretty
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(rgdal)
```

```
## Loading required package: sp
## Please note that rgdal will be retired during October 2023,
## plan transition to sf/stars/terra functions using GDAL and PROJ
## at your earliest convenience.
## See https://r-spatial.org/r/2023/05/15/evolution4.html and https://github.com/r-spatial/evolution
## rgdal: version: 1.6-7, (SVN revision 1203)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.5.3, released 2022/10/21
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/rgdal
##  GDAL does not use iconv for recoding strings.
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 9.1.0, September 1st, 2022, [PJ_VERSION: 910]
## Path to PROJ shared files: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/rgdal
## PROJ CDN enabled: FALSE
## Linking to sp version:1.6-1
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading sp or rgdal.
```

```r
library(broom)
library(mapview)
library(crosstalk)
library(sf)
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```r
library(sp)
library(spdep)
```

```
## Loading required package: spData
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```r
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(fs)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
# download a zip file containing some boundaries we want to use

download.file("https://data.london.gov.uk/download/statistical-gis-boundary-files-london/9ba8c833-6370-
              destfile="statistical-gis-boundaries-london.zip")
```

```r
# Get the zip file and extract it
library(fs)
listfiles<-dir_info(here::here()) %>%
  dplyr::filter(str_detect(path, ".zip")) %>%
  dplyr::select(path)%>%
  pull()%>%
  #print out the .gz file
  print()%>%
  as.character()%>%
  utils::unzip(exdir=here::here())
```

```
## /Users/jijinting/Documents/GIS/gis_code/wk8p/statistical-gis-boundaries-london.zip
```

```r
# Look inside the zip and read in the .shp
# look what is inside the zip

Londonwards<-fs::dir_info(here::here("statistical-gis-boundaries-london",
                                     "ESRI"))%>%
  # $ means exact match $
  dplyr::filter(str_detect(path,
                           "London_Ward_CityMerged.shp$"))%>%
  dplyr::select(path)%>%
  dplyr::pull()%>%
  # read in the file in
  sf::st_read()
```

```
## Reading layer `London_Ward_CityMerged' from data source
##   `/Users/jijinting/Documents/GIS/gis_code/wk8p/statistical-gis-boundaries-london/ESRI/London_Ward_C
##   using driver `ESRI Shapefile'
## Simple feature collection with 625 features and 7 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 503568.2 ymin: 155850.8 xmax: 561957.5 ymax: 200933.9
## Projected CRS: OSGB36 / British National Grid
```

```r
#check the data
qtm(Londonwards)
```

Now we are going to read in some data from the London Data Store

```
#read in some attribute data
LondonWardProfiles <- read_csv("https://data.london.gov.uk/download/ward-profiles-and-atlas/772d2d64-e8
                               col_names = TRUE,
                               locale = locale(encoding = 'Latin1'))
```

```
## Rows: 660 Columns: 67
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (15): Ward name, Old code, New code, % children in reception year who ar...
## dbl (52): Population - 2015, Children aged 0-15 - 2015, Working-age (16-64) ...
##
## 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.
```

```
#check all of the columns have been read in correctly
Datatypelist <- LondonWardProfiles %>%
  summarise_all(class) %>%
  pivot_longer(everything(),
               names_to="All_variables",
               values_to="Variable_class")

Datatypelist
```

```
## # A tibble: 67 x 2
##    All_variables                               Variable_class
##    <chr>                                       <chr>
##   1 Ward name                          character
##   2 Old code                           character
##   3 New code                           character
##   4 Population - 2015                  numeric
##   5 Children aged 0-15 - 2015          numeric
##   6 Working-age (16-64) - 2015         numeric
##   7 Older people aged 65+ - 2015       numeric
##   8 % All Children aged 0-15 - 2015    numeric
##   9 % All Working-age (16-64) - 2015   numeric
##  10 % All Older people aged 65+ - 2015 numeric
## # i 57 more rows
```

Cleaning the data as you read it in

Examining the dataset as it is read in above, you can see that a number of fields in the dataset that should have been read in as numeric data, have actually been read in as character (text) data. If you examine your data file, you will see why. In a number of columns where data are missing, rather than a blank cell, the values 'n/a' have been entered in instead. Where these text values appear amongst numbers, the software will automatically assume the whole column is text. To deal with these errors, we can force read_csv to ignore these values by telling it what values to look out for that indicate missing data read_csv

```
#We can use readr to deal with the issues in this dataset - which are to do with text values being stor

#read in some data - couple of things here. Read in specifying a load of likely 'n/a' values, also spec

LondonWardProfiles <- read_csv("https://data.london.gov.uk/download/ward-profiles-and-atlas/772d2d64-e8
                               na = c("", "NA", "n/a"),
                               locale = locale(encoding = 'Latin1'),
                               col_names = TRUE)
```

```
## Rows: 660 Columns: 67
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Ward name, Old code, New code
## dbl (64): Population - 2015, Children aged 0-15 - 2015, Working-age (16-64) ...
##
## 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.
```

Or download it from the London data store and read it in…it's the ward Profiles excel download.

```
LondonWardProfiles <- read_csv("ward-profiles-excel-version.csv",
                               na = c("", "NA", "n/a"),
                               locale = locale(encoding = 'Latin1'),
                               col_names = TRUE)
```

```
## Rows: 660 Columns: 67
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Ward name, Old code, New code
## dbl (64): Population - 2015, Children aged 0-15 - 2015, Working-age (16-64) ...
##
## 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.
```

```r
#check all of the columns have been read in correctly
Datatypelist <- LondonWardProfiles %>%
  summarise_all(class) %>%
  pivot_longer(everything(),
               names_to="All_variables",
               values_to="Variable_class")

Datatypelist
```

```
## # A tibble: 67 x 2
##    All_variables                      Variable_class
##    <chr>                              <chr>
##  1 Ward name                          character
##  2 Old code                           character
##  3 New code                           character
##  4 Population - 2015                  numeric
##  5 Children aged 0-15 - 2015          numeric
##  6 Working-age (16-64) - 2015         numeric
##  7 Older people aged 65+ - 2015       numeric
##  8 % All Children aged 0-15 - 2015    numeric
##  9 % All Working-age (16-64) - 2015   numeric
## 10 % All Older people aged 65+ - 2015 numeric
## # i 57 more rows
```

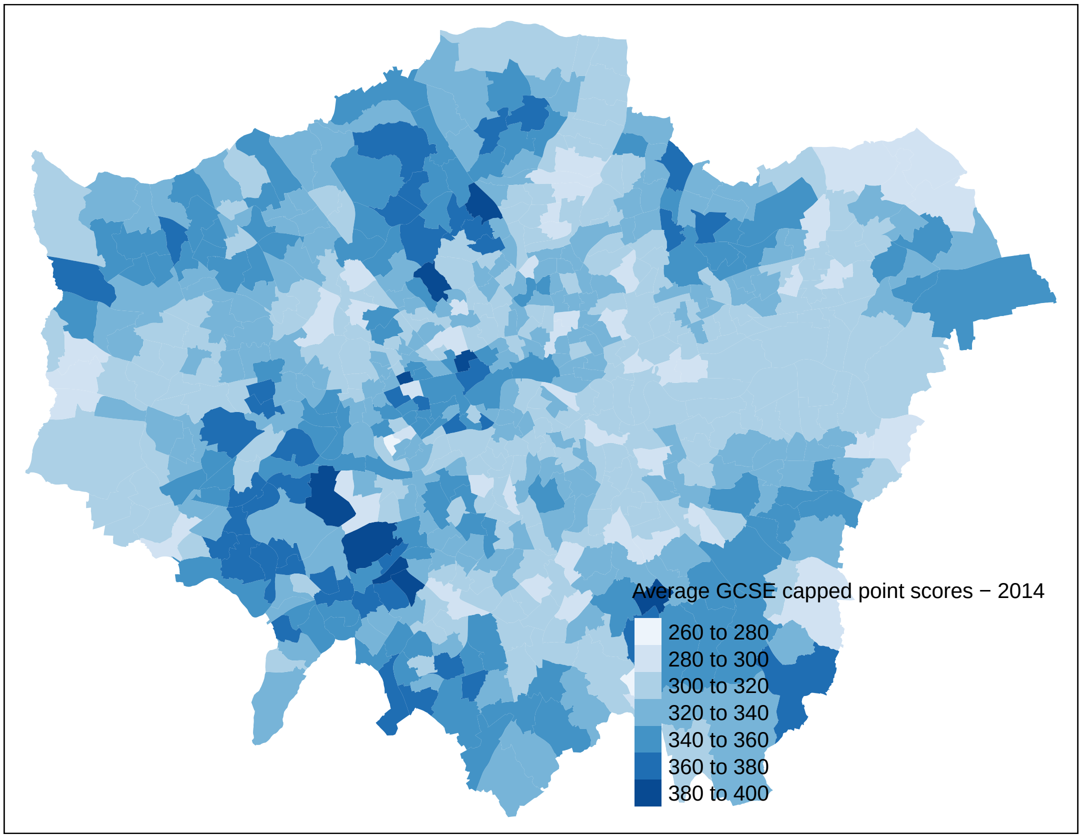
Now you have read in both your boundary data and your attribute data, you need to merge the two together using a common ID. In this case, we can use the ward codes to achieve the join

```r
# merge boundaries and data
LonWardProfiles <- Londonwards%>%
  left_join(.,
            LondonWardProfiles,
            by = c("GSS_CODE" = "New code"))
```

```r
#let's map our dependent variable to see if the join has worked:
tmap_mode("plot")
```

```
## tmap mode set to plotting
```

```r
qtm(LonWardProfiles,
    fill = "Average GCSE capped point scores - 2014",
    borders = NULL,
    fill.palette = "Blues")
```



Additional Data

In addition to our main datasets, it might also be useful to add some contextual data. While our exam results have been recorded at the home address of students, most students would have attended one of the schools in the City.

Let’s add some schools data as well. In the st_as_sf function x is longitude, y is latitude.

st_as_sf R sf sf Simple Features x y sf

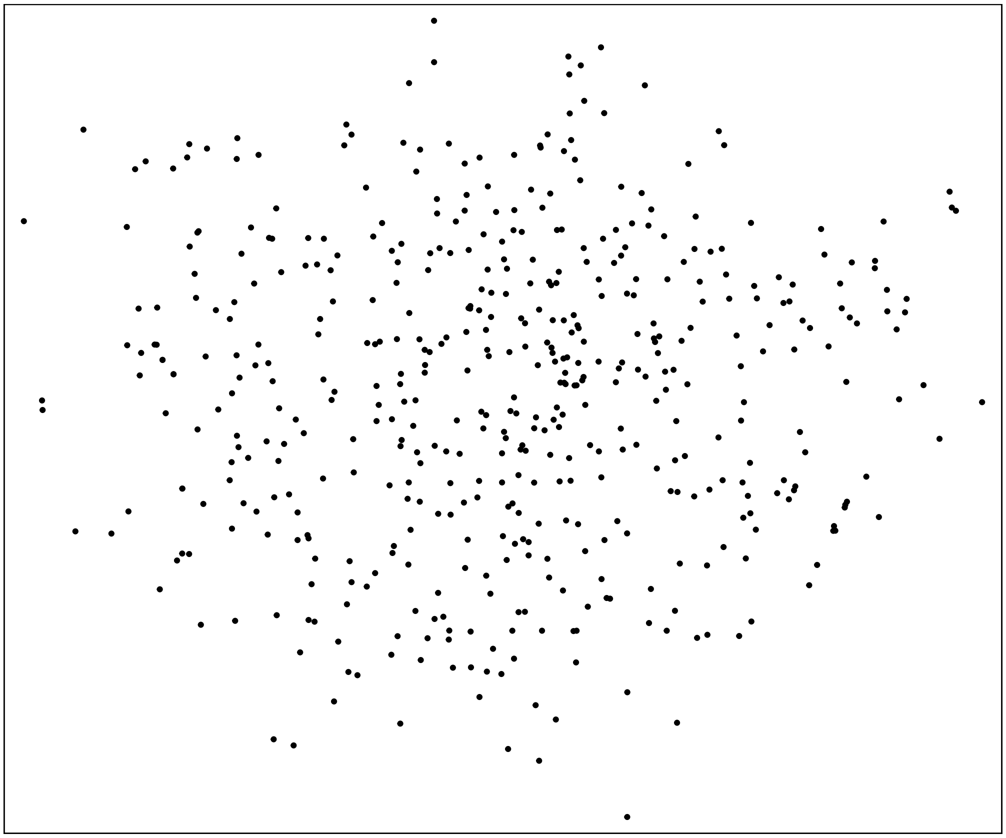
```
#might be a good idea to see where the secondary schools are in London too
london_schools <- read_csv("https://data.london.gov.uk/download/london-schools-atlas/57046151-39a0-45d9-
```

```
## Rows: 3889 Columns: 24
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (14): SCHOOL_NAM, TYPE, PHASE, ADDRESS, TOWN, POSTCODE, STATUS, GENDER, ...
## dbl  (7): URN, EASTING, NORTHING, map_icon_l, Primary, x, y
## num  (1): OBJECTID
## lgl  (2): NEW_URN, OLD_URN
##
## 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.
```

```
#from the coordinate values stored in the x and y columns, which look like they are latitude and longit
lon_schools_sf <- st_as_sf(london_schools,
                           coords = c("x","y"),
                           crs = 4326)

lond_sec_schools_sf <- lon_schools_sf %>%
  filter(PHASE=="Secondary")

tmap_mode("plot")
```

```
## tmap mode set to plotting
```

```
qtm(lond_sec_schools_sf)
```



Analysing GCSE exam performance - testing a research hypothesis GCSE -

To explore the factors that might influence GCSE exam performance in London, we are going to run a series of different regression models. A regression model is simply the expression of a linear relationship between our outcome variable (Average GCSE score in each Ward in London) and another variable or several variables that might explain this outcome. GCSE GCSE ## Research Question and Hypothesis Examining the spatial distribution of GSCE point scores in the map above, it is clear that there is variation across the city. My research question is:

What are the factors that might lead to variation in Average GCSE point scores across the city?

My research hypothesis that I am going to test is that there are other observable factors occurring in Wards in London that might affect the average GCSE scores of students living in those areas.

In inferential statistics, we cannot definitively prove a hypothesis is true, but we can seek to disprove that there is absolutely nothing of interest occurring or no association between variables. The null hypothesis that I am going to test empirically with some models is that there is no relationship between exam scores and other observed variables across London. GSCE

GCSE

GCSE

Regression Basics For those of you who know a bit about regression, you might want to skip down to the next section. However, if you are new to regression or would like a refresher, read on…

The linear relationship in a regression model is probably most easily explained using a scatter plot…

```
q <- qplot(x = `Unauthorised Absence in All Schools (%) - 2013`,
           y = `Average GCSE capped point scores - 2014`,
           data=LonWardProfiles)
```

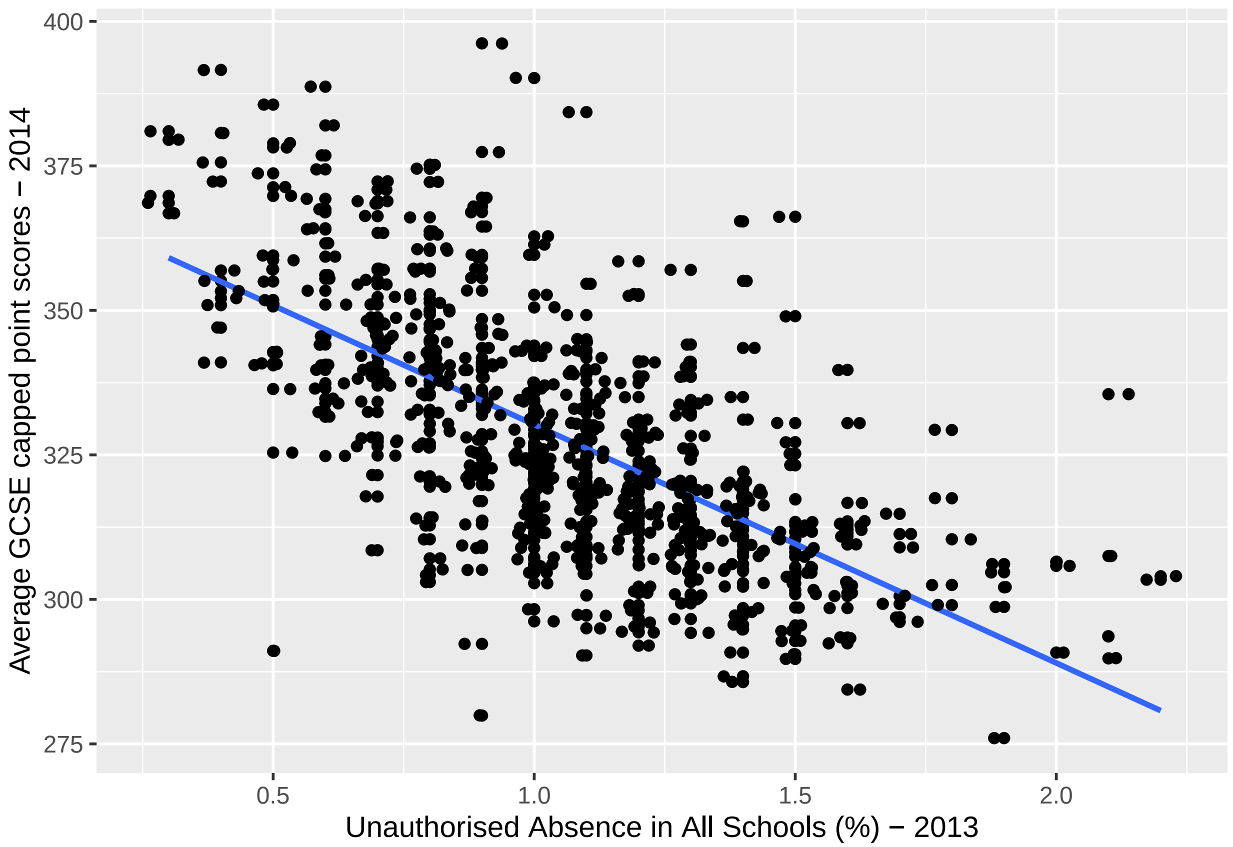
```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

“jitter” x x jittering

```
#plot with a regression line - note, I've added some jitter here as the x-scale is rounded
q + stat_smooth(method="lm", se=FALSE, size=1) +
  geom_jitter()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



GCSE

GCSE

X y GCSE

- y y X

- ” ”(OLS) Any value of y along the blue line can be modelled using the corresponding value of x and these parameter values. Examining the graph above we would expect the average GCSE point score for a student living in a Ward where 0.5% of school days per year were missed, to equal around 350, but we can confirm this by plugging the parameter values and the value of x into equation (1):

```
370 + (-40*0.5) + 0
```

```
## [1] 350
```

Running a Regression Model in R

In the graph above, I used a method called ‘lm’ in the stat_smooth() function in ggplot2 to draw the regression line. ‘lm’ stands for ‘linear model’ and is a standard function in R for running linear regression models. Use the help system to find out more about lm - ?lm

Below is the code that could be used to draw the blue line in our scatter plot. Note, the tilde ~ symbol means “is modelled by”.

First though, we’re going to clean up all our data names with Janitor then select what we want.

```r
#run the linear regression model and store its outputs in an object called model1
Regressiondata<- LonWardProfiles%>%
  clean_names()%>%
  dplyr::select(average_gcse_capped_point_scores_2014,
                unauthorised_absence_in_all_schools_percent_2013)

#now model
model1 <- Regressiondata %>%
  lm(average_gcse_capped_point_scores_2014 ~
               unauthorised_absence_in_all_schools_percent_2013,
     data=.)
```

Let's have a closer look at our model...

```r
#show the summary of those outputs
summary(model1)
```

```
##
## Call:
## lm(formula = average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_201
##     data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -59.753 -10.223  -1.063   8.547  61.842
##
## Coefficients:
##                                                  Estimate Std. Error t value
## (Intercept)                                       371.471      2.165   171.6
## unauthorised_absence_in_all_schools_percent_2013  -41.237      1.927   -21.4
##                                                  Pr(>|t|)
## (Intercept)                                        <2e-16 ***
## unauthorised_absence_in_all_schools_percent_2013   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.39 on 624 degrees of freedom
## Multiple R-squared:  0.4233, Adjusted R-squared:  0.4224
## F-statistic:   458 on 1 and 624 DF,  p-value: < 2.2e-16
```

Interpreting and using the model outputs

In running a regression model, we are effectively trying to test (disprove) our null hypothesis. If our null hypothesis was true, then we would expect our coefficients to = 0. = 0 In the output summary of the model above, there are a number of features you should pay attention to: Coefficient Estimates - these are the 0 (intercept) and 1 (slope) parameter estimates from Equation 1. You will notice that at 0=371.471 and 1=−41.237 they are pretty close to the estimates of 370 and -40 that we read from the graph earlier, but more precise. - 1 0 1 0=371.471 1=−41.237 370 -40 Coefficient Standard Errors - these represent the average amount the coefficient varies from the average value of the dependent variable (its standard deviation). So, for a 1% increase in unauthorised absence from school, while the model says we might expect GSCE scores to drop by -41.2 points, this might vary, on average, by about 1.9 points. As a rule of thumb, we are looking for a lower value in the standard error relative to the size of the coefficient. - 1% GSCE -41.2 1.9 Note that is the coefficient represents a one unit change, here it is %, as the variable is % unauthorized absence in school

So one unit is a 1% change… 1 …… Coefficient t-value - this is the value of the coefficient divided by the standard error and so can be thought of as a kind of standardised coefficient value. The larger (either positive or negative) the value the greater the relative effect that particular independent variable is having on the dependent variable (this is perhaps more useful when we have several independent variables in the model) . t -
Coefficient p-value - Pr(>|t|) - the p-value is a measure of significance. There is lots of debate about p-values which I won't go into here, but essentially it refers to the probability of getting a coefficient as large as the one observed in a set of random data. p-values can be thought of as percentages, so if we have a p-value of 0.5, then there is a 5% chance that our coefficient could have occurred in some random data, or put another way, a 95% chance that out coefficient could have only occurred in our data. As a rule of thumb, the smaller the p-value, the more significant that variable is in the story and the smaller the chance that the relationship being observed is just random. Generally, statisticians use 5% or 0.05 as the acceptable cut-off for statistical significance - anything greater than that we should be a little sceptical about. p - Pr(>|t|) -p p p p 0.5 5% 95% p 5% 0.05 —— In r the codes ***, , , . are used to indicate significance. We generally want at least a single * next to our coefficient for it to be worth considering. r*** . * R-Squared - This can be thought of as an indication of how good your model is - a measure of 'goodness-of-fit' (of which there are a number of others).r2 ( )is quite an intuitite measure of fit as it ranges between 0 and 1 and can be thought of as the % of variation in the dependent variable (in our case GCSE score) explained by variation in the independent variable(s). In our example, an r2 value of 0.42 indicates that around 42% of the variation in GCSE scores can be explained by variation in unathorised absence from school. In other words, this is quite a good model. The r2 value will increase as more independent explanatory variables are added into the model, so where this might be an issue, the adjusted r-squared value can be used to account for this affect R-Squared - - " " r2 0 1 GCSE r2 0.42 GCSE 42% r2 r ### broom
The output from the linear regression model is messy and like all things R mess can be tidied, in this case with a broom! Or the package broom which is also party of the package tidymodels. R tidymodels Here let's load broom and tidy our output…you will need to either install tidymodels or broom. The tidy() function will just make a tibble or the statistical findings from the model! broom …… tidymodels broom tidy() tibble

```r
library(broom)
tidy(model1)
```

```
## # A tibble: 2 x 5
##   term                                      estimate std.error statistic  p.value
##   <chr>                                        <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                                  371.       2.16     172.  0
## 2 unauthorised_absence_in_all_schools_per~     -41.2      1.93     -21.4 1.27e-76
```

We can also use glance() from broom to get a bit more summary information, such as r2 and the adjusted r-squared value. broom glance() r2 r Multiple R-squared Adjusted R-squared

Multiple R-squared ( ) 0.4233 42.33%

Adjusted R-squared ( ) 0.4224 R R
The closeness of these two values suggests that the number of independent variables added to your model has a positive contribution to the explanatory power of the model without introducing too many irrelevant variables. Usually, if the Adjusted R-squared is much lower than the Multiple R-squared, it might mean that there are too many insignificant independent variables in the model. R R

R R 0.4233

```r
glance(model1)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.423         0.422  16.4      458. 1.27e-76     1 -2638. 5282. 5296.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

But wait? Didn't we try to model our GCSE values based on our unauthorised absence variable? Can we see those predictions for each point, yes, yes we can…with the tidypredict_to_column() function from tidypredict, which adds the fit column in the following code. GCSE ……
tidypredict tidypredict_to_column()

```r
library(tidypredict)
Regressiondata %>%
  tidypredict_to_column(model1)
```

```
## Simple feature collection with 626 features and 3 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 503568.2 ymin: 155850.8 xmax: 561957.5 ymax: 200933.9
## Projected CRS: OSGB36 / British National Grid
## First 10 features:
##    average_gcse_capped_point_scores_2014
## 1                                  321.3
## 2                                  337.5
## 3                                  342.7
## 4                                  353.3
## 5                                  372.3
## 6                                  339.8
## 7                                  307.1
## 8                                  361.6
## 9                                  347.0
## 10                                 336.4
##    unauthorised_absence_in_all_schools_percent_2013
## 1                                               0.8
## 2                                               0.7
## 3                                               0.5
## 4                                               0.4
## 5                                               0.7
## 6                                               0.9
## 7                                               0.8
## 8                                               0.6
## 9                                               0.7
## 10                                              0.5
##                          geometry      fit
## 1  POLYGON ((516401.6 160201.8... 338.4815
## 2  POLYGON ((517829.6 165447.1... 342.6052
## 3  POLYGON ((518107.5 167303.4... 350.8525
## 4  POLYGON ((520480 166909.8, ... 354.9762
## 5  POLYGON ((522071 168144.9, ... 342.6052
## 6  POLYGON ((522007.6 169297.3... 334.3579
## 7  POLYGON ((517175.3 164077.3... 338.4815
## 8  POLYGON ((517469.3 166878.5... 346.7289
## 9  POLYGON ((522231.1 166015, ... 342.6052
## 10 POLYGON ((517460.6 167802.9... 350.8525
```

tidymodels

Before we move on it's worth pointing out that a new iteration of modelling is being developed through tidymodels…the benefit of this is that we can easily change the modelling method or as they term it…engine…(e.g. to RandomForest) tidymodels …… …… ……

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.1.1 --

## v dials        1.2.0     v rsample      1.2.0
## v infer        1.0.5     v tune         1.1.2
## v modeldata    1.2.0     v workflows    1.1.3
## v parsnip      1.1.1     v workflowsets 1.0.1
## v recipes      1.0.8     v yardstick    1.2.0

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x plotly::filter()  masks dplyr::filter(), stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x car::recode()     masks dplyr::recode()
## x car::some()       masks purrr::some()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Dig deeper into tidy modeling with R at https://www.tmwr.org
```

```
# set the model
lm_mod <- linear_reg()

# fit the model
lm_fit <-
  lm_mod %>%
  fit(average_gcse_capped_point_scores_2014 ~
               unauthorised_absence_in_all_schools_percent_2013,
     data=Regressiondata)

# we cover tidy and glance in a minute...
tidy(lm_fit)
```

```
## # A tibble: 2 x 5
##   term                                        estimate std.error statistic  p.value
##   <chr>                                          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                                    371.       2.16     172.  0
## 2 unauthorised_absence_in_all_schools_per~       -41.2      1.93     -21.4 1.27e-76
```

```
glance(lm_fit)
```

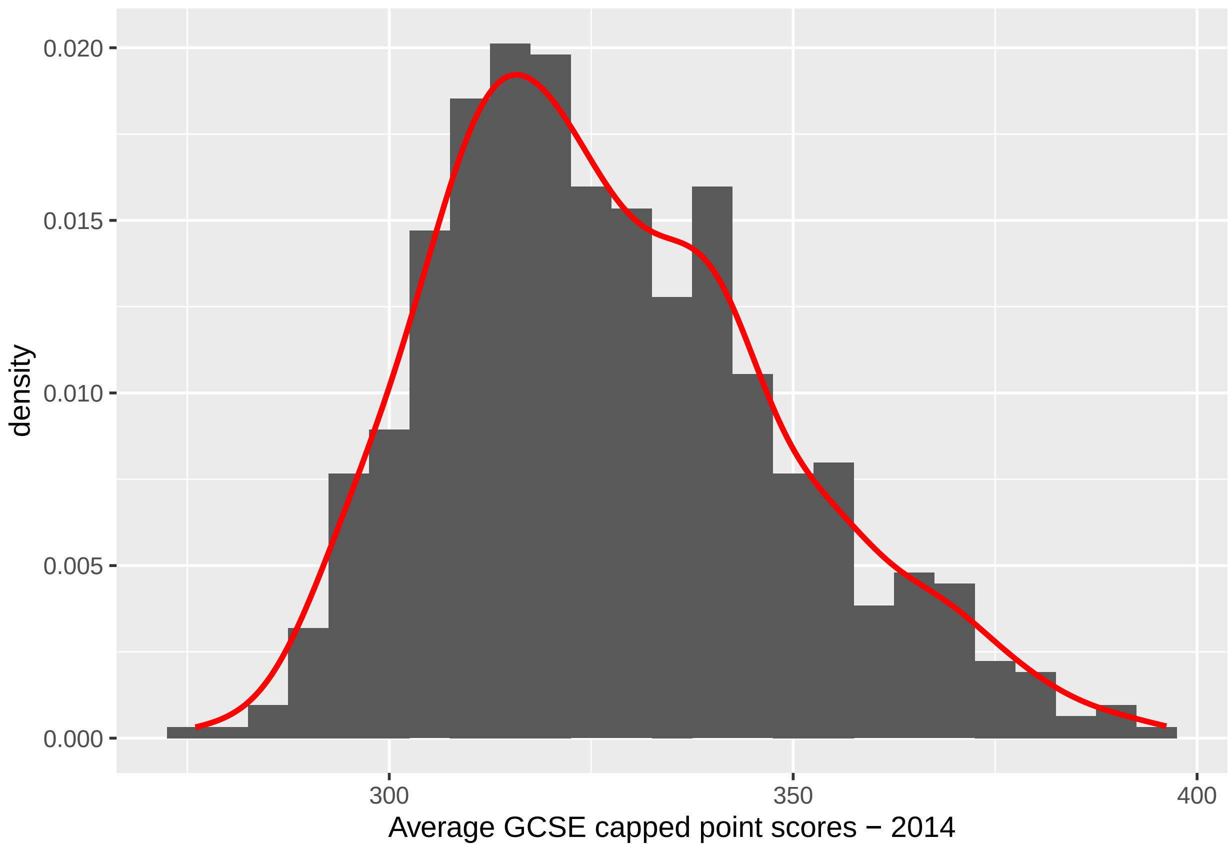
```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.423         0.422  16.4      458. 1.27e-76     1 -2638. 5282. 5296.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

However at the moment we can't do spatial modelling using tidymodels…but this is probably coming soon. tidymodels …… ## Bootstrap resampling If we only fit our model once, how can we be confident about that estimate? Bootstrap resampling is where we take the original dataset and select random data points from within it, but in order to keep it the same size as the original dataset some records are duplicated. This is known as bootstrap resampling by replacement. We used to briefly cover this within this practical but have recently removed it. If you wish to explore it then consult the bootstrap resampling section from previous years, but this is not a requirement and only for interest. ## Variables (What is confounding) ## Assumptions Underpinning Linear Regression ## Assumption 1 - There is a linear relationship between the dependent and independent variables 1 - The best way to test for this assumption is to plot a scatter plot similar to the one created earlier. It may not always be practical to create a series of scatter plots, so one quick way to check that a linear relationship is probable is to look at the frequency distributions of the variables. If they are normally distributed, then there is a good chance that if the two variables are in some way correlated, this will be a linear relationship.

For example, look at the frequency distributions of our two variables earlier: (Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0. Please use `after_stat(density)` instead.)

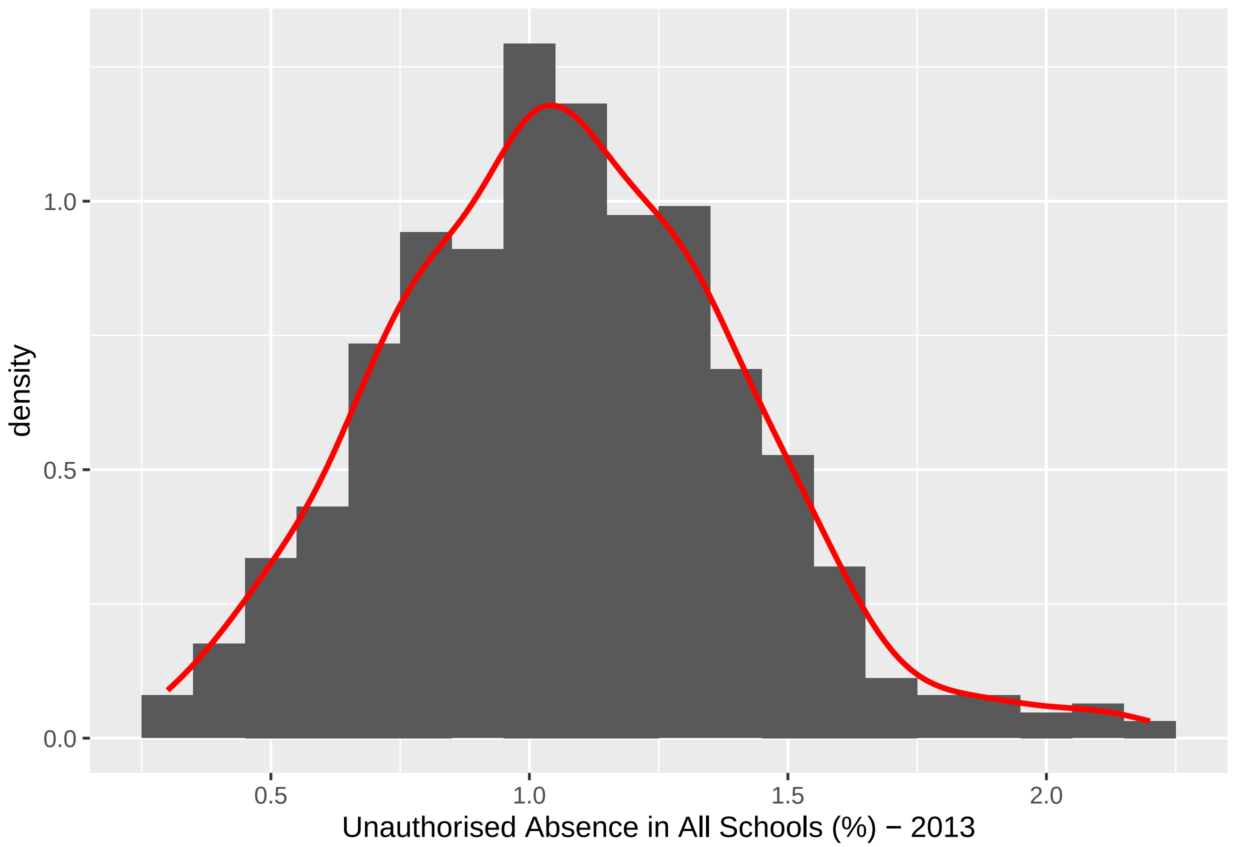
```r
#let's check the distribution of these variables first

ggplot(LonWardProfiles, aes(x=`Average GCSE capped point scores - 2014`)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 5) +
  geom_density(colour="red",
               size=1,
               adjust=1)
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Here, adding ..density.. means that the histogram is a density plot, this plots the chance that any value in the data is equal to that value. ..density..

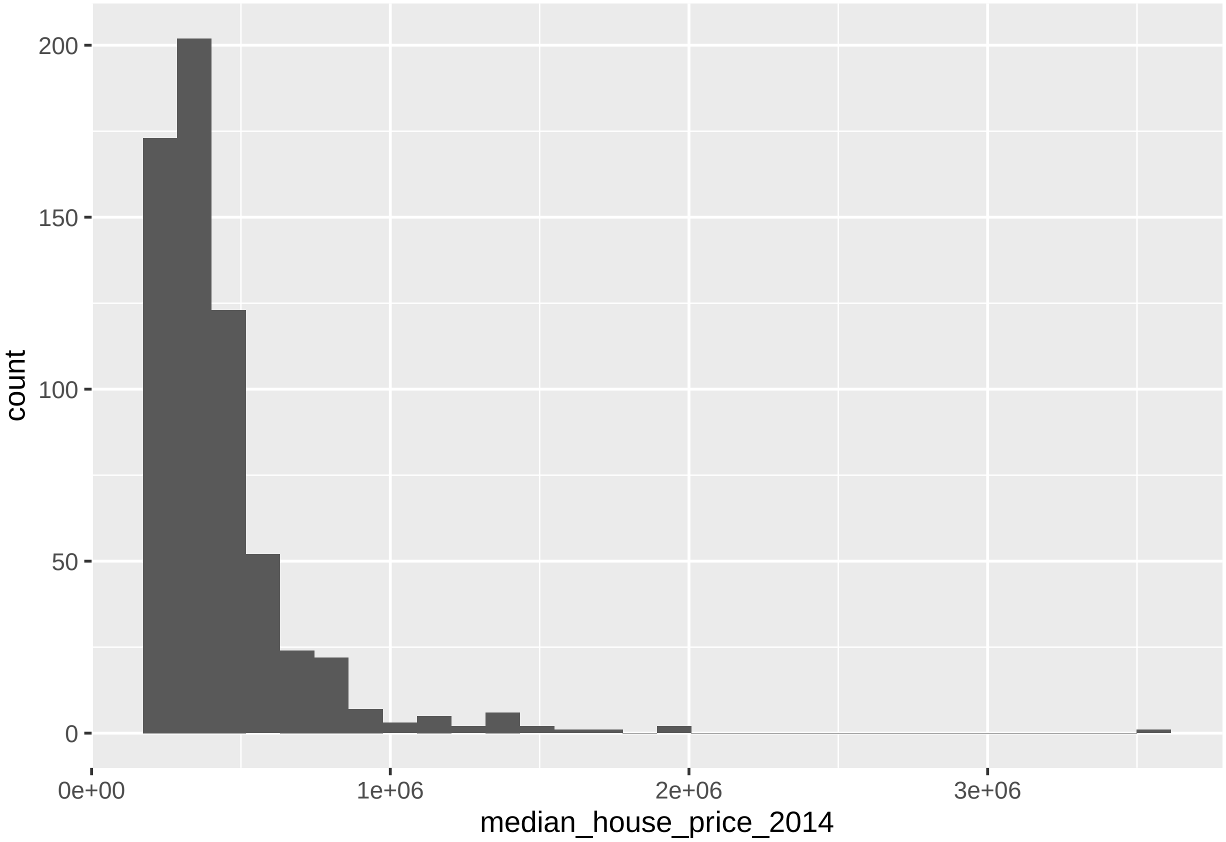
```
ggplot(LonWardProfiles, aes(x=`Unauthorised Absence in All Schools (%) - 2013`)) +
  geom_histogram(aes(y = ..density..),
                 binwidth = 0.1) +
  geom_density(colour="red",
               size=1,
               adjust=1)
```



We would describe both of these distribution as being relatively ‘normally’ or gaussian disributed, and thus more likely to have a linear correlation (if they are indeed associated). “ ”
Contrast this with the median house price variable: Median House Price (£) - 2014(don’t like it) -
So to fix this i just manually renamed the column and then used clean_names() for the rest of the columns.
Good code is code that works and doesn’t always need to be pretty / clean

```
library(ggplot2)

# from 21/10 there is an error on the website with
# median_house_price_2014 being called median_house_price<c2>2014
# this was corrected around 23/11 but can be corrected with rename..

LonWardProfiles <- LonWardProfiles %>%
  #try removing this line to see if it works...
  dplyr::rename(median_house_price_2014 =`Median House Price (£) - 2014`)%>%
  janitor::clean_names()

ggplot(LonWardProfiles, aes(x=median_house_price_2014)) +
  geom_histogram()
```

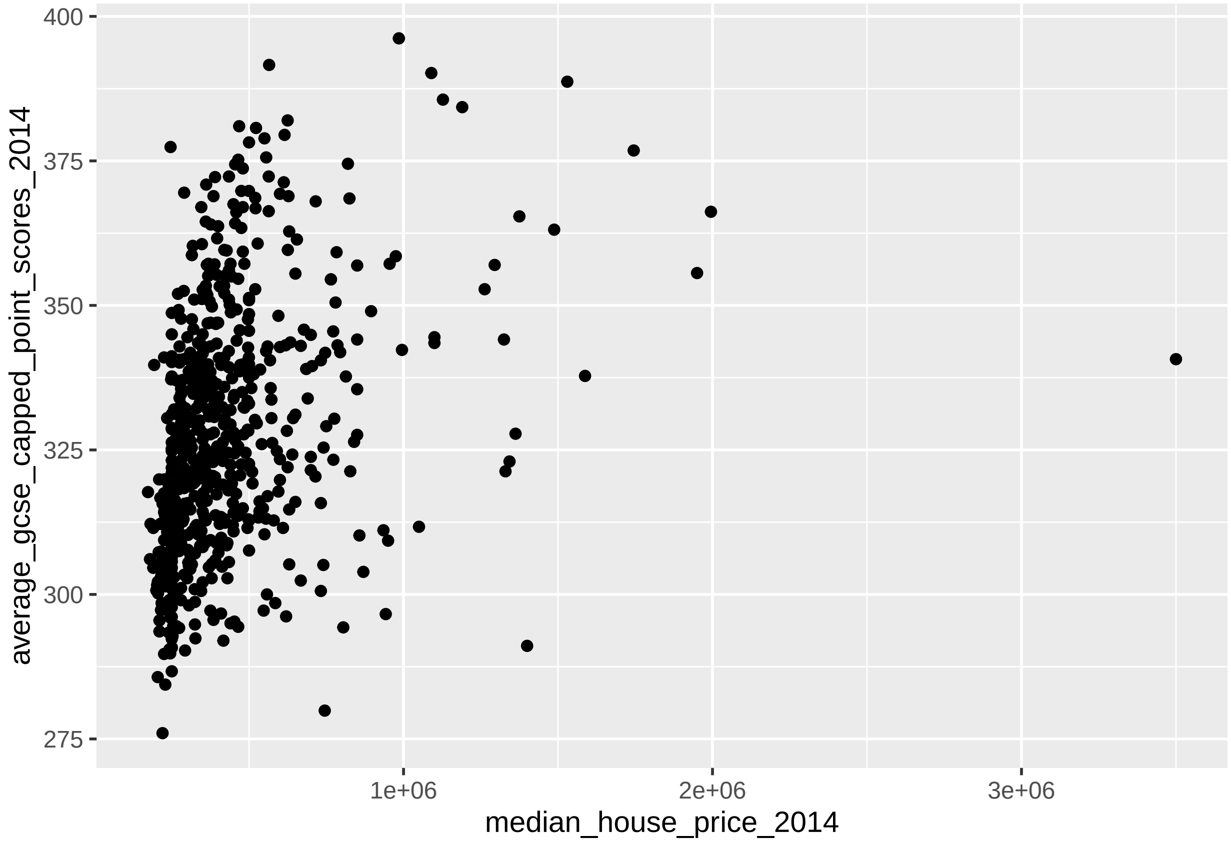
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We would describe this as a not normal and/or positively ‘skewed’ distribution, i.e. there are more observations towards the lower end of the average house prices observed in the city, however there is a long tail to the distribution, i.e. there are a small number of wards where the average house price is very large indeed. / “ ”

If we plot the raw house price variable against GCSE scores, we get the following scatter plot: GCSE

```
qplot(x = median_house_price_2014,
      y = average_gcse_capped_point_scores_2014,
      data=LonWardProfiles)
```



This indicates that we do not have a linear relationship, indeed it suggests that this might be a curvilinear relationship. ### Transforming variables One way that we might be able to achieve a linear relationship between our two variables is to transform the non-normally distributed variable so that it is more normally distributed.

There is some debate as to whether this is a wise thing to do as, amongst other things, the coefficients for transformed variables are much harder to interpret, however, we will have a go here to see if it makes a difference.

Tukey's ladder of transformations

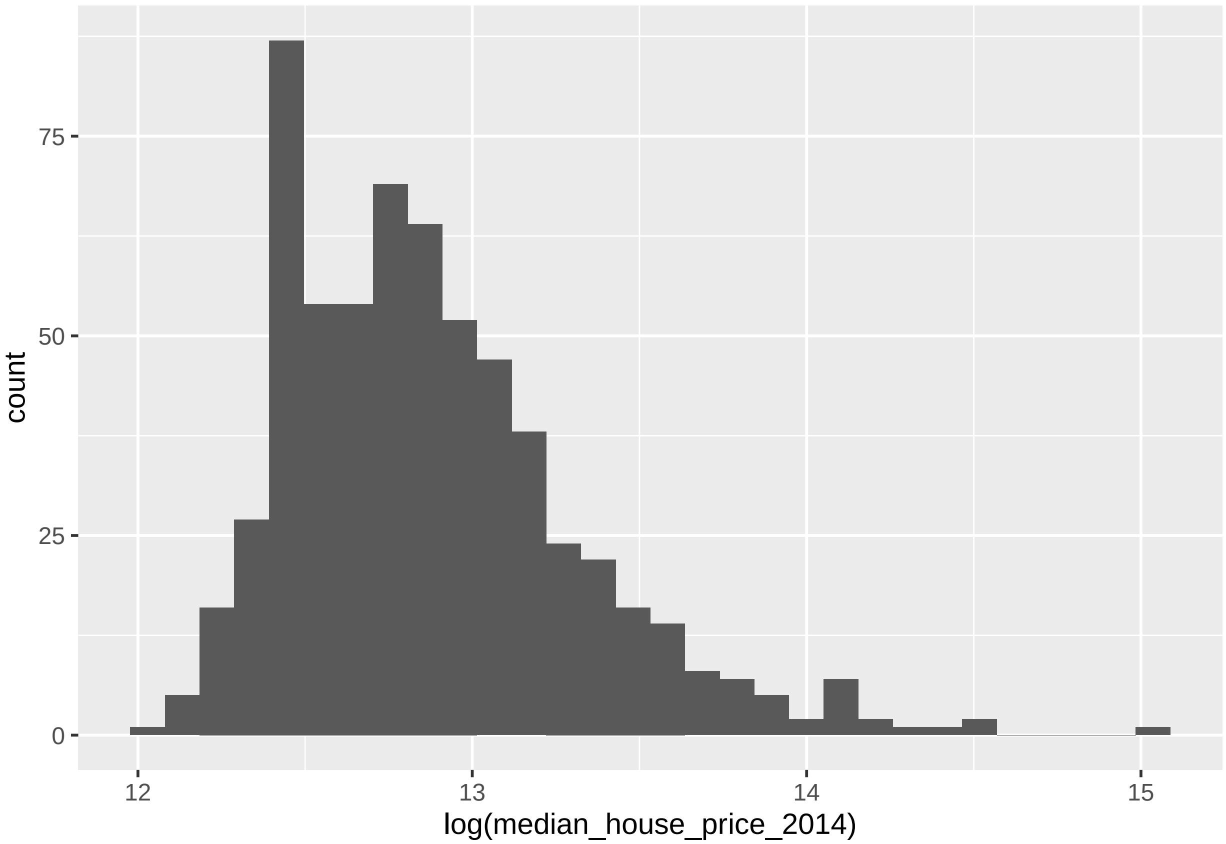
You might be asking how we could go about transforming our variables. In 1977, Tukey described a series of power transformations that could be applied to a variable to alter its frequency distribution.

In regression analysis, you analysts will frequently take the log of a variable to change its distribution, but this is a little crude and may not result in a completely normal distribution. For example, we can take the log of the house price variable:

1977 Tukey

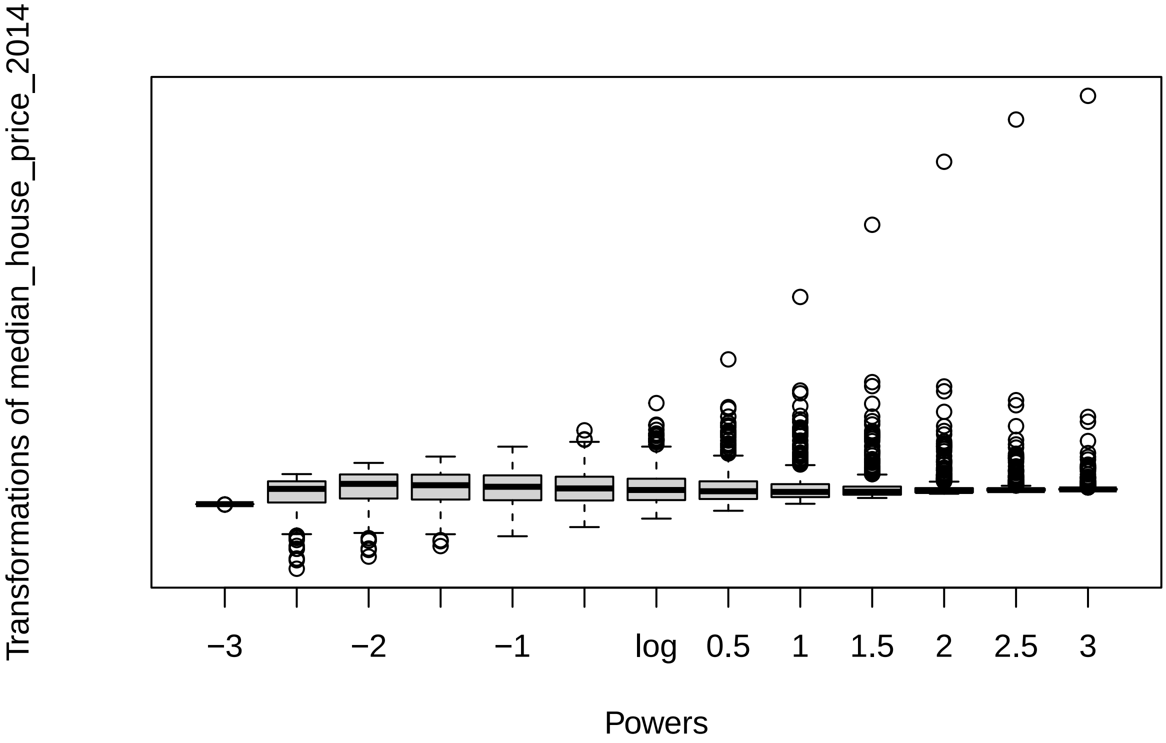
```
ggplot(LonWardProfiles, aes(x=log(median_house_price_2014))) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



This looks a little more like a normal distribution, but it is still a little skewed. Fortunately in R, we can use the symbox() function in the car package to try a range of transfomations along Tukey's ladder: R car symbox() Tukey

```
symbox(~median_house_price_2014,
       LonWardProfiles,
       na.rm=T,
       powers=seq(-3,3,by=.5))
```

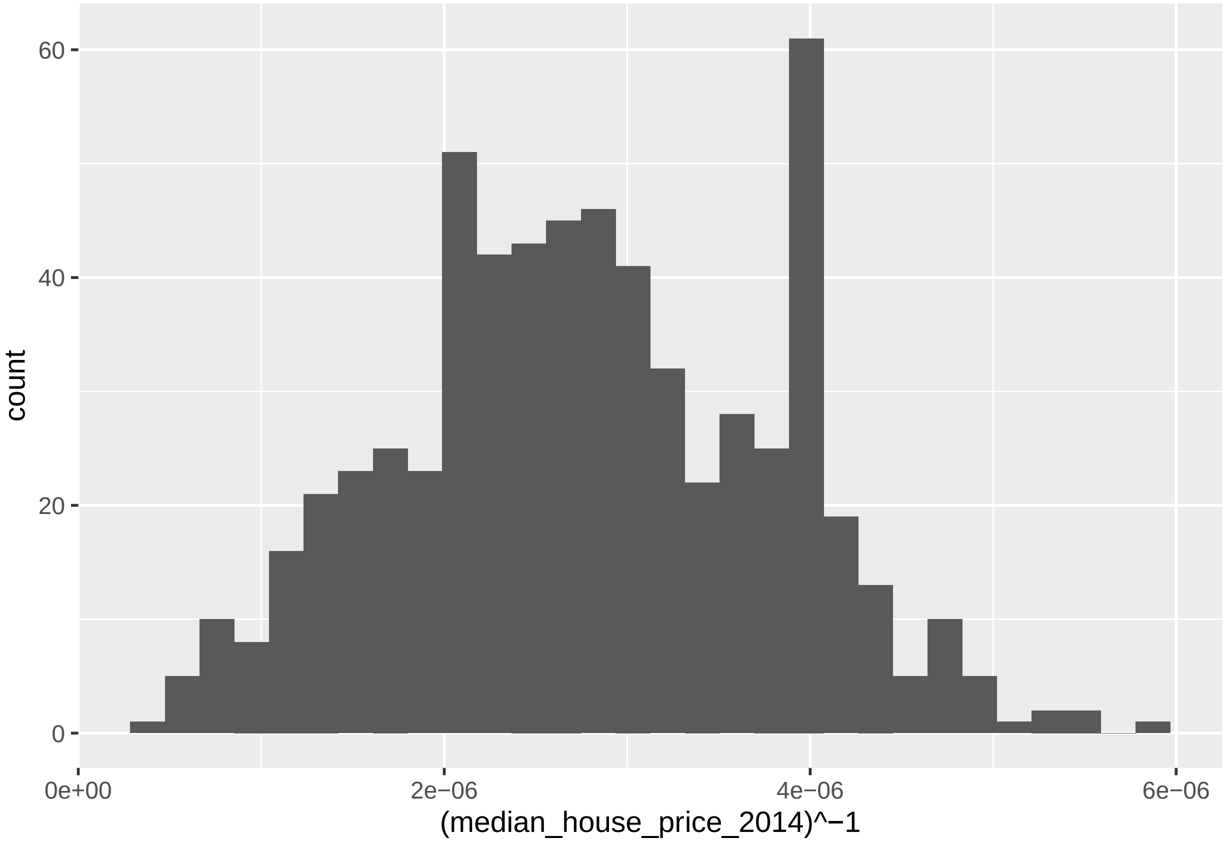


powers=seq(-3,3,by=.5))   seq(-3,3,by=.5)   -3   3   0.5   symbox

Observing the plot above, it appears that raising our house price variable to the power of -1 should lead to a more normal distribution:

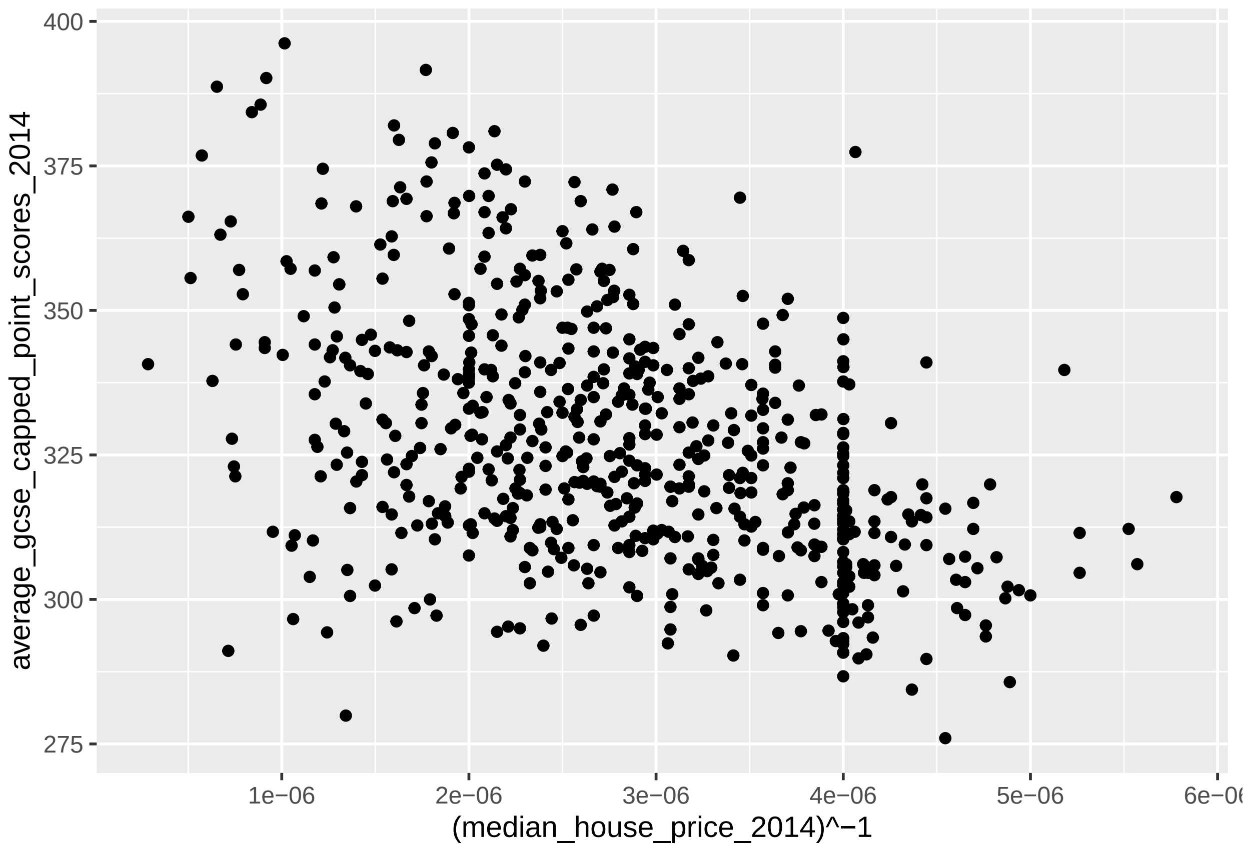
```
ggplot(LonWardProfiles, aes(x=(median_house_price_2014)^-1)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



qplot ggplot

```
qplot(x = (median_house_price_2014)^-1,
      y = average_gcse_capped_point_scores_2014,
      data=LonWardProfiles)
```



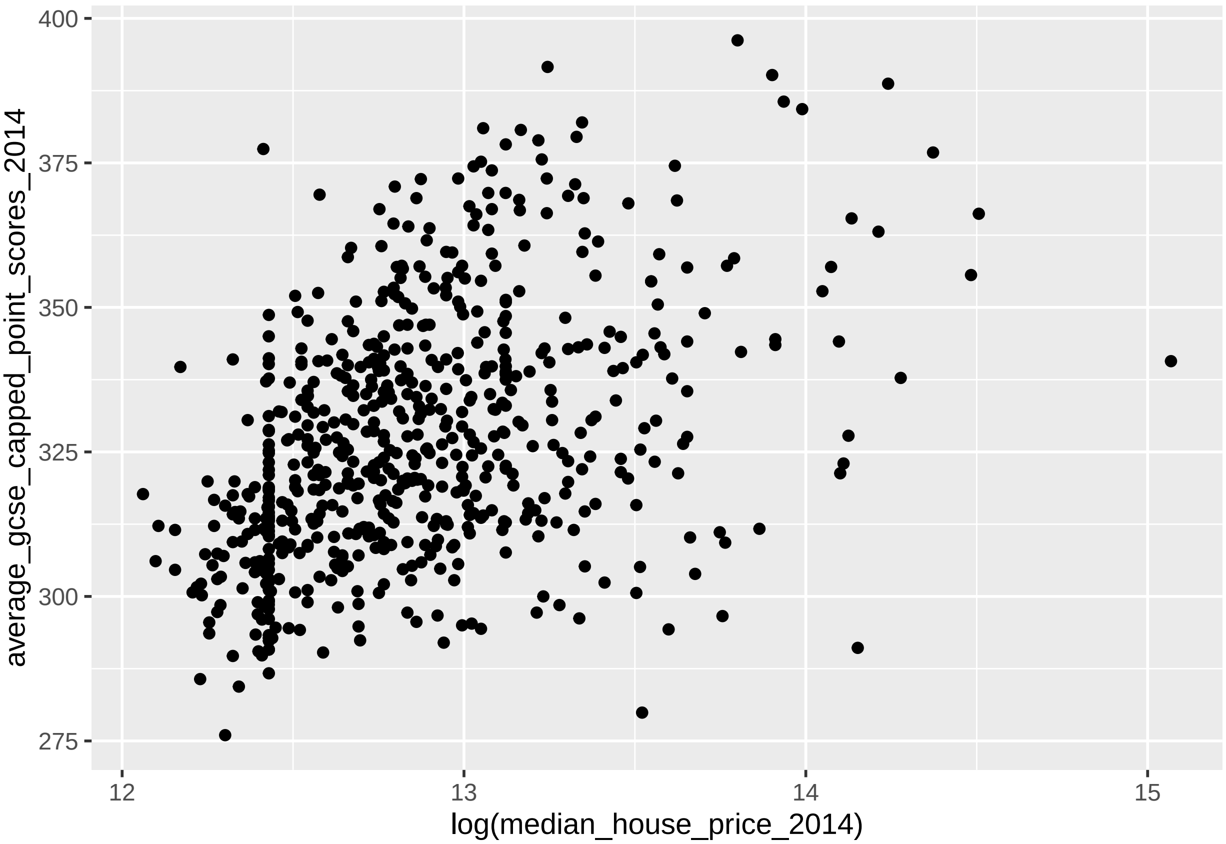
ggplot qplot ggplot2

ggplot ggplot2 ggplot2 [1]

qplot quick plot R plot() ggplot qplot [2]

ggplot2 geom_point() geom_bar() ggplot() geom_ qplot x y
x y Compare this with the logged transformation:

```
qplot(x = log(median_house_price_2014),
      y = average_gcse_capped_point_scores_2014,
      data=LonWardProfiles)
```



Depending on if the independent or dependent (GCSE point score) variables have been transformed depends on how we interpret them - see these rules for interpretation GCSE - ### Should I transform my variables? The decision is down to you as the modeller - it might be that a transformation doesn't succeed in normalising the distribution of your data or that the interpretation after the transformation is problematic, however it is important not to violate the assumptions underpinning the regression model or your conclusions may be on shaky ground. - Be careful The purpose of doing theses transformations is to make your data normally distributed, however you will be changing the relationship of your data - it won't be linear anymore! This could improve your model but is at the expense of interpretation. Aside from log transformation which has the rules in the link above.

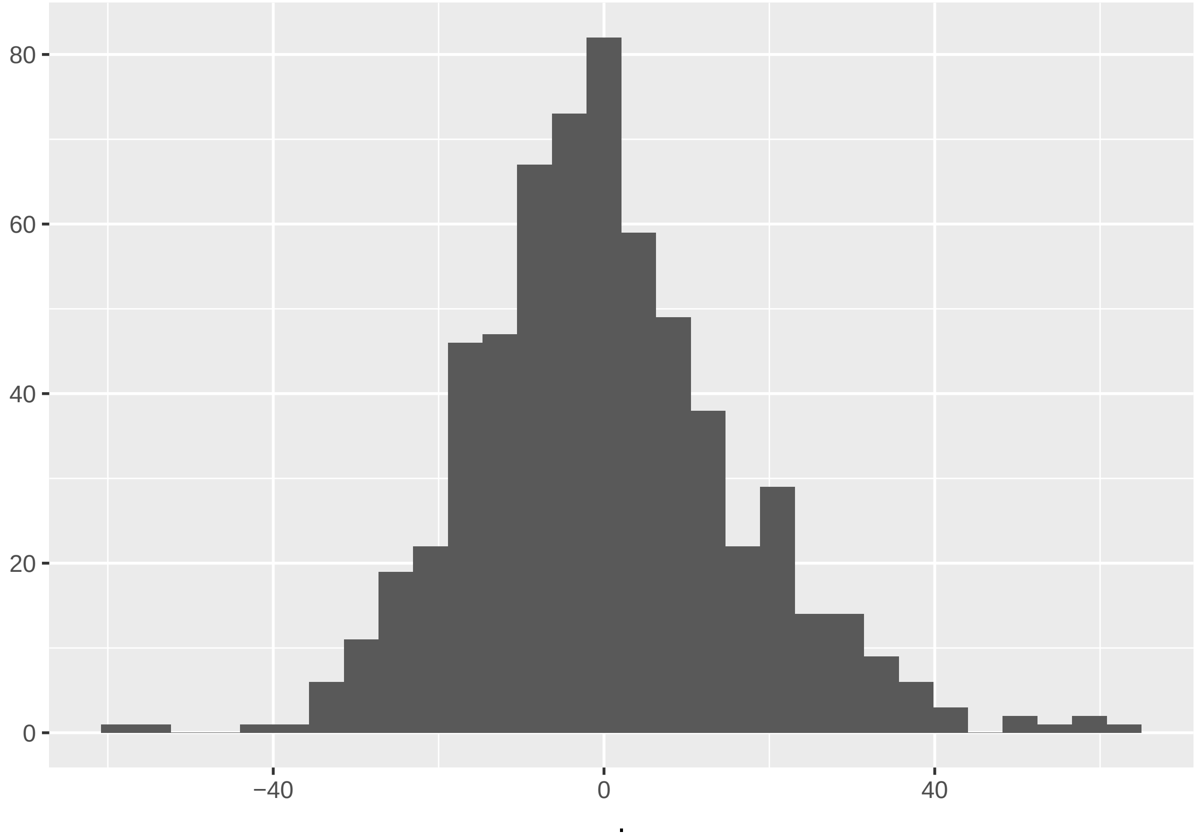
Typically if you do a power transformation you can keep the direction of the relationship (positive or negative) and the t-value will give an idea of the importance of the variable in the model - that's about it!

For more information here read Transforming Variables in Regression by Eric van Holm, 2021 - t - Eric van Holm 2021 ## Assumption 2 - The residuals in your model should be normally distributed This assumption is easy to check. When we ran our Model1 earlier, one of the outputs stored in our Model 1 object is the residual value for each case (Ward) in your dataset. We can access these values using augment() from broom which will add model output to the original GCSE data... Model1 1 (Ward) broom Augment() GCSE …… We can plot these as a histogram and see if there is a normal distribution:

```
#save the residuals into your dataframe
# augment   broom
#
model_data <- model1 %>%
  augment(., Regressiondata)
```

```
#plot residuals
model_data%>%
dplyr::select(.resid)%>%
  pull()%>%
  qplot()+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



pull() %>% dplyr::select(.resid) qplot() qplot() data pull() .resid Examining the histogram above, we can be happy that our residuals look to be relatively normally distributed. ## Assumption 3 - No Multicolinearity in the independent variables 3 - Now, the regression model we have be experimenting with so far is a simple bivariate (two variable) model. One of the nice things about regression modelling is while we can only easily visualise linear relationships in a two (or maximum 3) dimension scatter plot, mathematically, we can have as many dimensions / variables as we like. 3 / As such, we could extend model 1 into a multiple regression model by adding some more explanatory variables that we think could affect GSCE scores. Let's try the log or ^-1 transformed house price variable from earlier (the rationale being that higher house prices indicate more affluence and therefore, potentially, more engagement with education): GSCE 1 ^-1

```
Regressiondata2<- LonWardProfiles%>%
  clean_names()%>%
  dplyr::select(average_gcse_capped_point_scores_2014,
         unauthorised_absence_in_all_schools_percent_2013,
         median_house_price_2014)
```

```
model2 <- lm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_2013 +
               log(median_house_price_2014), data = Regressiondata2)

#show the summary of those outputs
tidy(model2)
```

```
## # A tibble: 3 x 5
##   term                                         estimate std.error statistic  p.value
##   <chr>                                           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                                     202.      20.1       10.0  4.79e-22
## 2 unauthorised_absence_in_all_schools_per~       -36.2       1.92     -18.9  3.71e-63
## 3 log(median_house_price_2014)                    12.8       1.50       8.50 1.37e-16
```

```
glance(model2)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.483         0.482  15.5      291. 4.78e-90     2 -2604. 5215. 5233.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
#and for future use, write the residuals out
model_data2 <- model2 %>%
  augment(., Regressiondata2)

# also add them to the shapelayer
LonWardProfiles <- LonWardProfiles %>%
  mutate(model2resids = residuals(model2))
```

Examining the output above, it is clear that including median house price into our model improves the fit from an r2 of around 42% to an r2 of 48%. Median house price is also a statistically significant variable. 42% r2 48% But do our two explanatory variables satisfy the no-multicoliniarity assumption? If not and the variables are highly correlated, then we are effectively double counting the influence of these variables and overstating their explanatory power.

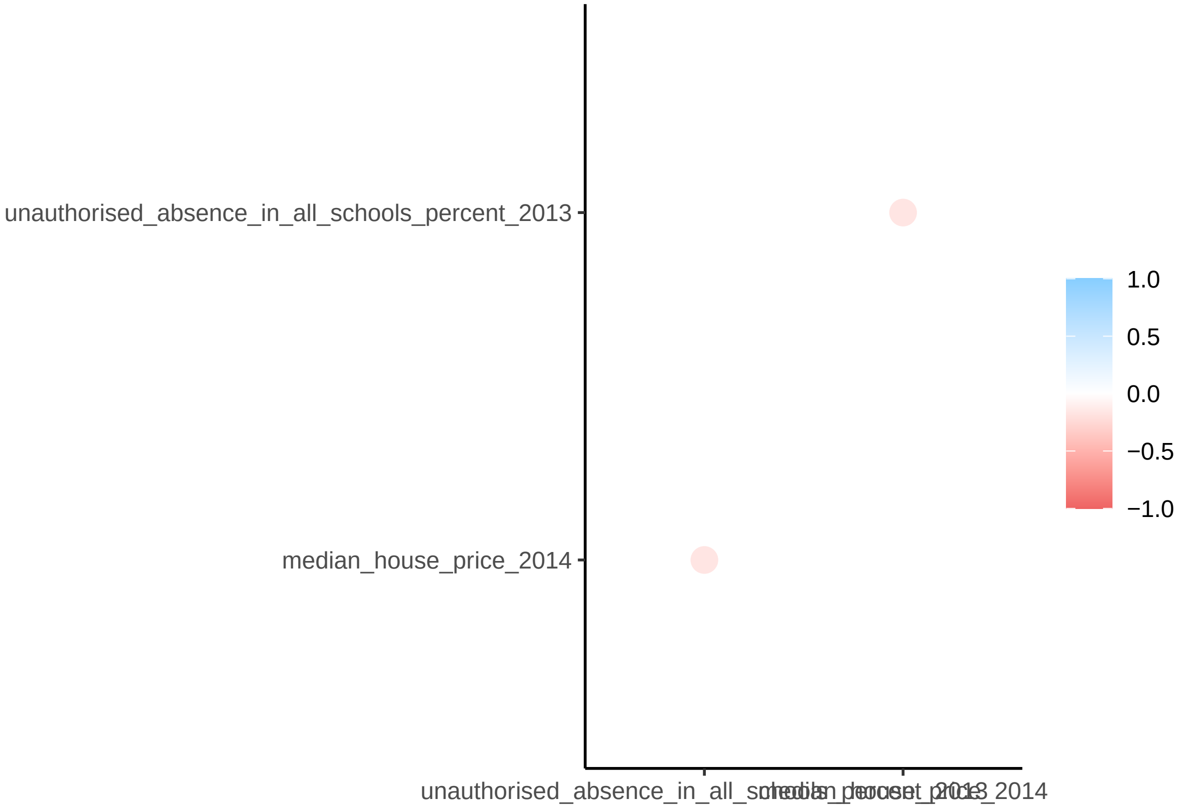
To check this, we can compute the product moment correlation coefficient between the variables, using the corrr() pacakge, that's part of tidymodels. In an ideal world, we would be looking for something less than a 0.8 correlation corrr() tidymodels 0.8

```
library(corrr)

Correlation <- LonWardProfiles %>%
  st_drop_geometry()%>%
  dplyr::select(average_gcse_capped_point_scores_2014,
         unauthorised_absence_in_all_schools_percent_2013,
         median_house_price_2014) %>%
  mutate(median_house_price_2014 =log(median_house_price_2014))%>%
    correlate() %>%
  # just focus on GCSE and house prices
  focus(-average_gcse_capped_point_scores_2014, mirror = TRUE)
```

```
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'
```

```
#visualise the correlation matrix
#   corrr   rplot()
```

```
#
rplot(Correlation)
```



Looking at either the correlation matrix or the correlation plot of that matrix, it's easy to see that there is a low correlation (around 30%) between our two independent variables. However, at this stage we might wish to introduce more variables into our model to improve our prediction of the dependent variable (GCSE scores), this is called multiple linear regression…multiple linear regression can be explained nicely with this example from allison_horst. 30% GCSE …… allison_horst ### Variance Inflation Factor (VIF) Another way that we can check for Multicolinearity is to examine the VIF for the model. If we have VIF values for any variable exceeding 10, then we may need to worry and perhaps remove that variable from the analysis:

```
vif(model2)
```

```
## unauthorised_absence_in_all_schools_percent_2013
##                                         1.105044
##                          log(median_house_price_2014)
##                                         1.105044
```

Both the correlation plots and examination of VIF indicate that our multiple regression model meets the assumptions around multicollinearity and so we can proceed further.

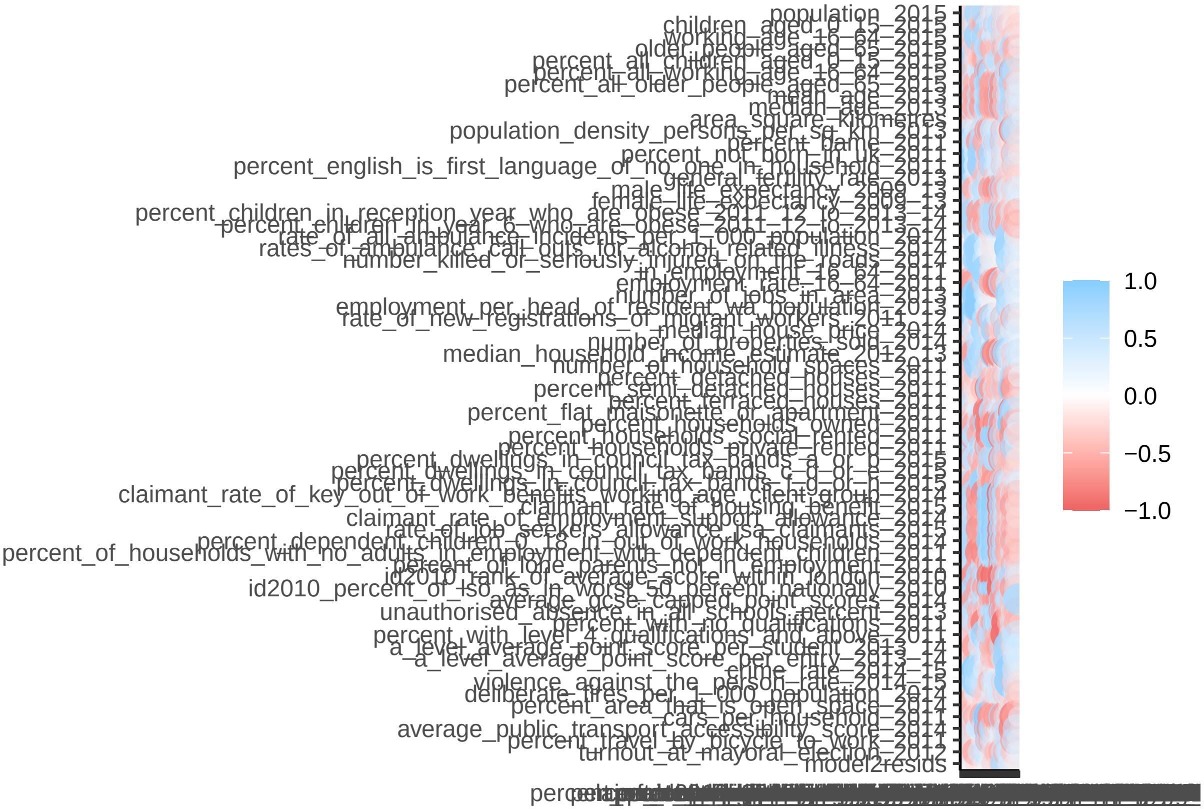
If we wanted to add more variables into our model, it would be useful to check for multicollinearity amongst every variable we want to include, we can do this by computing a correlation matrix for the whole dataset or checking the VIF after running the model: VIF

```
position <- c(10:74)

Correlation_all<- LonWardProfiles %>%
  st_drop_geometry()%>%
  dplyr::select(position)%>%
    correlate()
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(position)
##
##   # Now:
##   data %>% select(all_of(position))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'
```

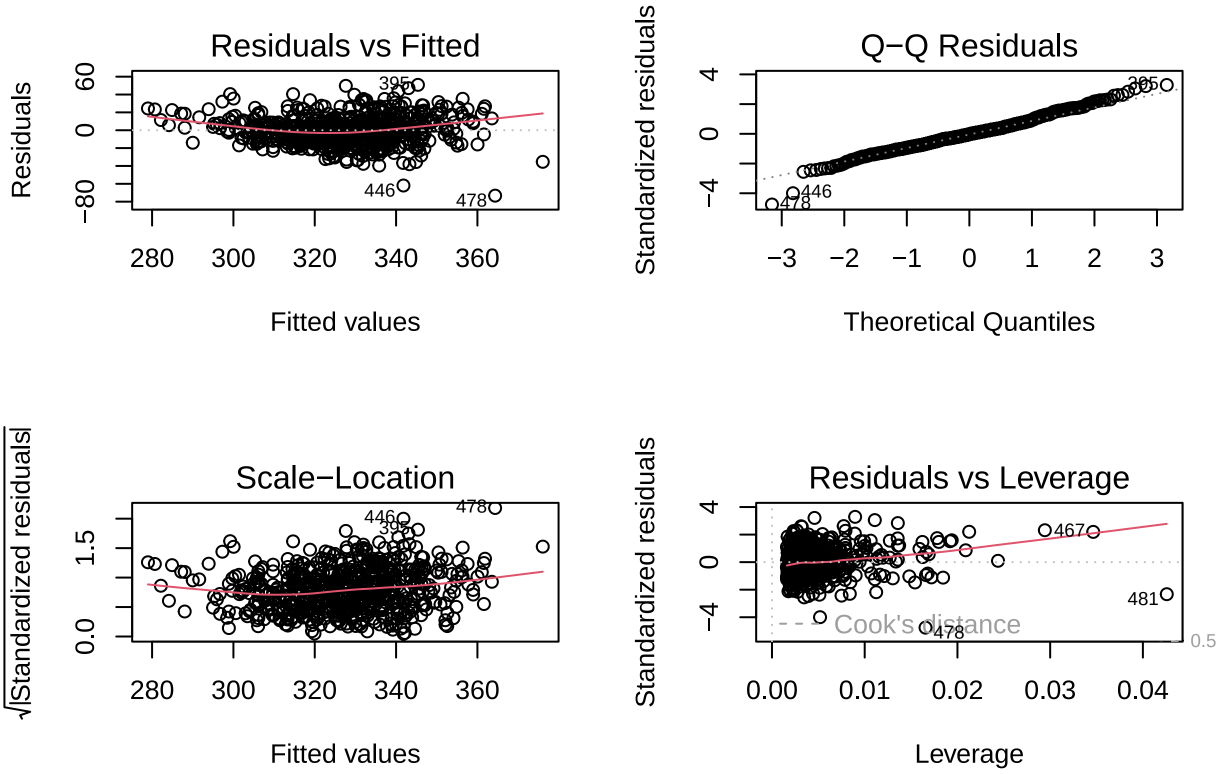
```
rplot(Correlation_all)
```



## Assumption 4 - Homoscedasticity 4 - Homoscedasticity means that the errors/residuals in the model exhibit constant / homogenous variance, if they don't, then we would say that there is hetroscedasticity present. Why does this matter? Andy Field does a much better job of explaining this in discovering statistics — but essentially, if your errors do not have constant variance, then your parameter estimates could be wrong, as could the estimates of their significance. / / / - Andy

Field – The best way to check for homo/hetroscedasticity is to plot the residuals in the model against the predicted values. We are looking for a cloud of points with no apparent patterning to them. /

```
#print some model diagnositcs.
par(mfrow=c(2,2))     #plot to 2 by 2 array
plot(model2)
```



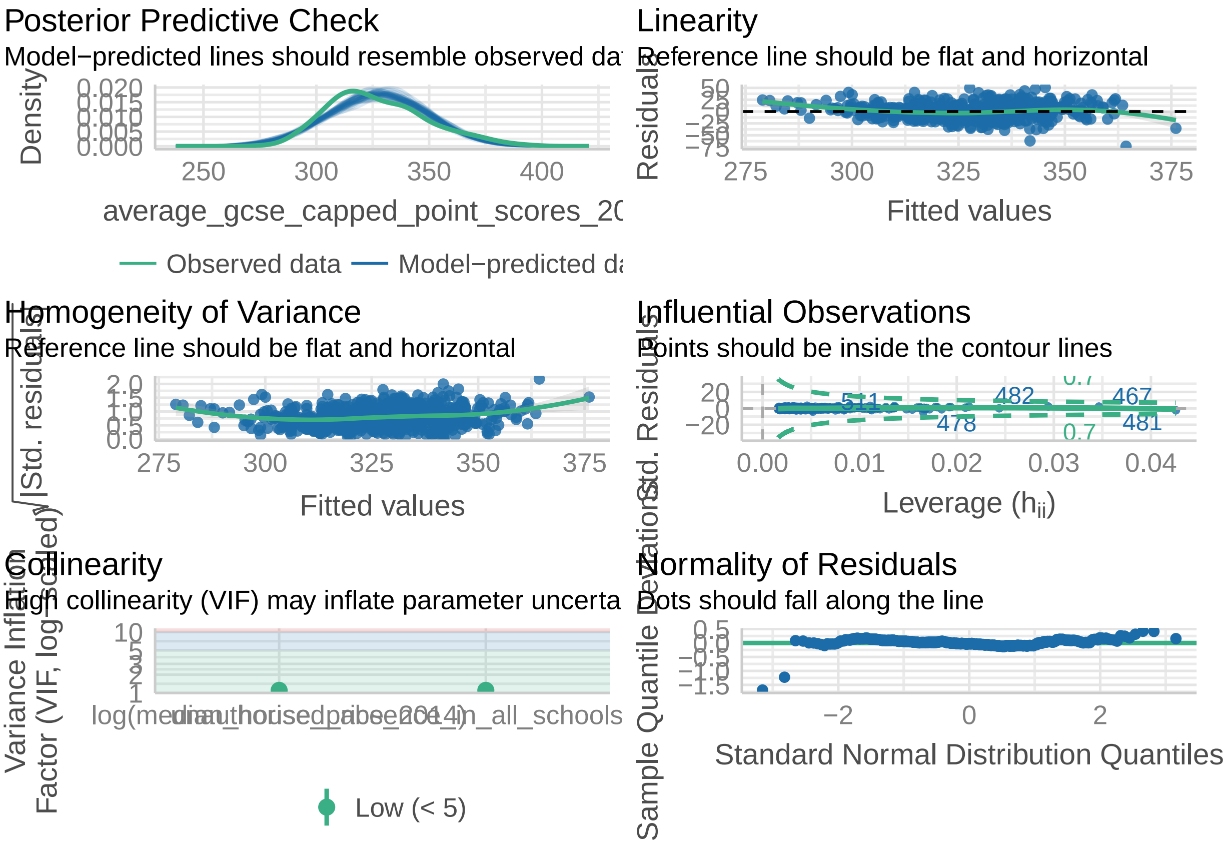
In the series of plots above, the first plot (residuals vs fitted), we would hope to find a random cloud of points with a straight horizontal red line. Looking at the plot, the curved red line would suggest some hetroscedasticity, but the cloud looks quite random. Similarly we are looking for a random cloud of points with no apparent patterning or shape in the third plot of standardised residuals vs fitted values. Here, the cloud of points also looks fairly random, with perhaps some shaping indicated by the red line. In the plots here we are looking for: Residuals vs Fitted: a flat and horizontal line. This is looking at the linear relationship assumption between our variables Normal Q-Q: all points falling on the line. This checks if the residuals (observed minus predicted) are normally distributed Q-Q Scale vs Location: flat and horizontal line, with randomly spaced points. This is the homoscedasticity (errors/residuals in the model exhibit constant / homogeneous variance). Are the residuals (also called errors) spread equally along all of the data. / / Residuals vs Leverage - Identifies outliers (or influential observations), the three largest outliers are identified with values in the plot. -

There is an easier way to produce this plot using check_model() from the performance package, that even includes what you are looking for…note that the Posterior predictive check is the comparison between the fitted model and the observed data. check_model() …… The default argument is check=all but we can specify what to check for…see the arguments section in the documentation…e.g. check = c("vif", "qq") check=all … … = c("vif", "qq")

```
library(performance)
```

```
##
## Attaching package: 'performance'

## The following objects are masked from 'package:yardstick':
##
##     mae, rmse
```

```
check_model(model2, check="all")
```



## Assumption 5 - Independence of Errors 5 - This assumption simply states that the residual values (errors) in your model must not be correlated in any way. If they are, then they exhibit autocorrelation which suggests that something might be going on in the background that we have not sufficiently accounted for in our model. ### Standard Autocorrelation If you are running a regression model on data that do not have explicit space or time dimensions, then the standard test for autocorrelation would be the Durbin-Watson test. Durbin-Watson

This tests whether residuals are correlated and produces a summary statistic that ranges between 0 and 4, with 2 signifiying no autocorrelation. A value greater than 2 suggesting negative autocorrelation and and value of less than 2 indicating postitve autocorrelation. 0 4 2 2 2

In his excellent text book, Andy Field suggests that you should be concerned with Durbin-Watson test statistics <1 or >3. So let's see: Andy Field Durbin-Watson <1 >3

```
#run durbin-watson test #
DW <- durbinWatsonTest(model2)
tidy(DW)
```

```
## # A tibble: 1 x 5
##   statistic p.value autocorrelation method             alternative
##       <dbl>   <dbl>           <dbl> <chr>              <chr>
## 1      1.61       0           0.193 Durbin-Watson Test two.sided
```

As you can see, the DW statistics for our model is 1.61, so some indication of autocorrelation, but perhaps nothing to worry about.

HOWEVER

We are using spatially referenced data and as such we should check for spatial-autocorrelation. The first test we should carry out is to map the residuals to see if there are any apparent obvious patterns:

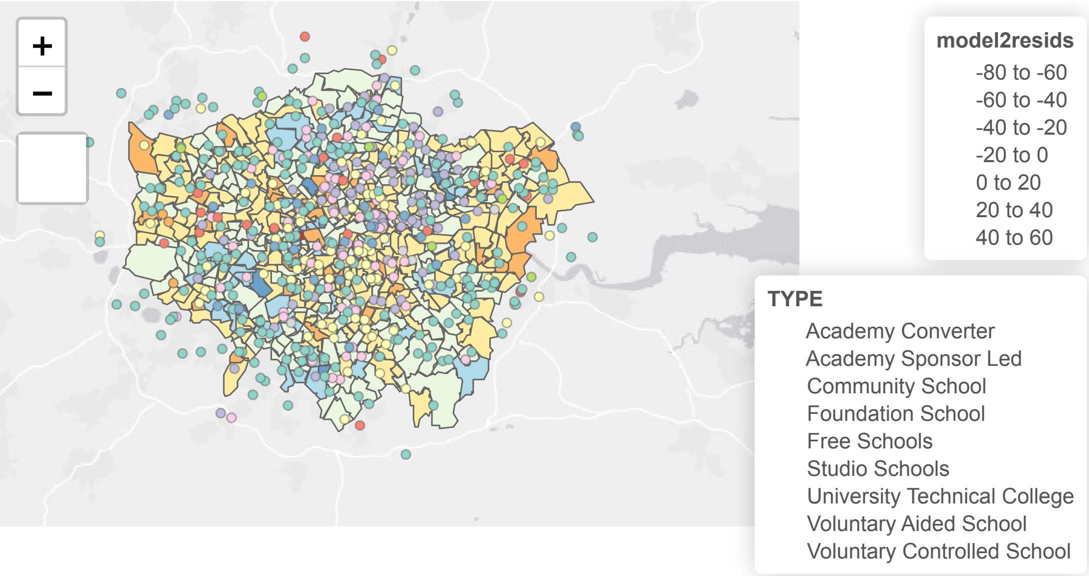
```
#now plot the residuals
tmap_mode("view")
```

```
## tmap mode set to interactive viewing
```

```
#qtm(LonWardProfiles, fill = "model1_resids")

tm_shape(LonWardProfiles) +
  tm_polygons("model2resids",
              palette = "RdYlBu") +
tm_shape(lond_sec_schools_sf) + tm_dots(col = "TYPE")
```

```
## Variable(s) "model2resids" contains positive and negative values, so midpoint is set to 0. Set midpoi
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please
```



Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ

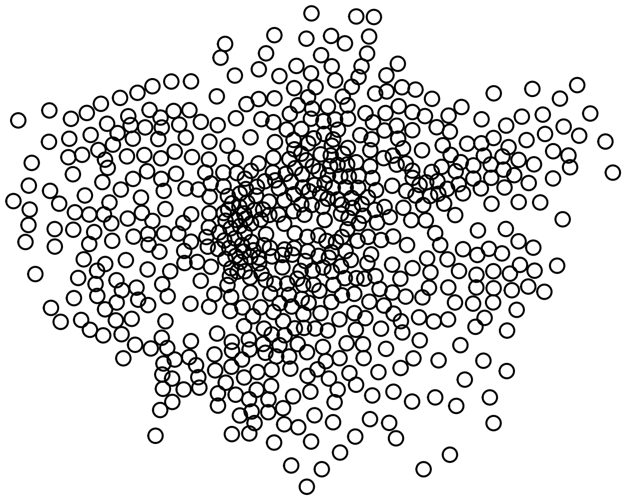
Looking at the map above, there look to be some blue areas next to other blue areas and some red/orange areas next to other red/orange areas. This suggests that there could well be some spatial autocorrelation biasing our model, but can we test for spatial autocorrelation more systematically?

/ / Yes - and some of you will remember this from the practical two weeks ago. We can calculate a number of different statistics to check for spatial autocorrelation - the most common of these being Moran's I. - - Moran's I

```
#calculate the centroids of all Wards in London
coordsW <- LonWardProfiles%>%
  st_centroid()%>%
  st_geometry()
```

```
## Warning: st_centroid assumes attributes are constant over geometries
```

```
plot(coordsW)
```

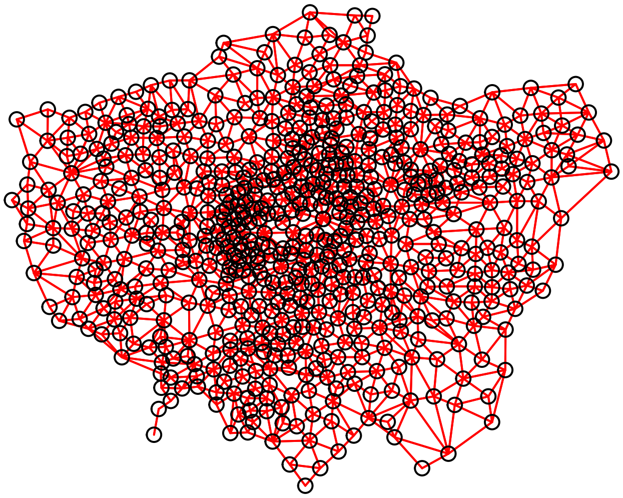


```
#Now we need to generate a spatial weights matrix
#(remember from the lecture a couple of weeks ago).
#We'll start with a simple binary matrix of queen's case neighbours

LWard_nb <- LonWardProfiles %>%
  poly2nb(., queen=T)

#or nearest neighbours
knn_wards <-coordsW %>%
  knearneigh(., k=4)
```

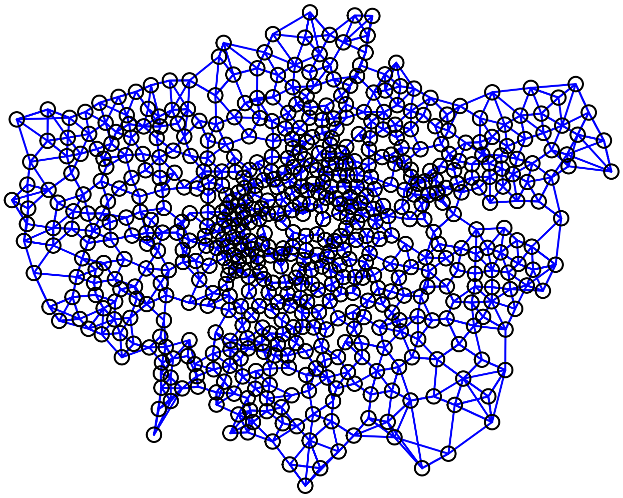
```
## Warning in knearneigh(., k = 4): knearneigh: identical points found
```

```
## Warning in knearneigh(., k = 4): knearneigh: kd_tree not available for
## identical points
```

```
LWard_knn <- knn_wards %>%
  knn2nb()

#plot them
plot(LWard_nb, st_geometry(coordsW), col="red")
```



```
plot(LWard_knn, st_geometry(coordsW), col="blue")
```



```
#create a spatial weights matrix object from these weights

Lward.queens_weight <- LWard_nb %>%
  nb2listw(., style="W")

Lward.knn_4_weight <- LWard_knn %>%
  nb2listw(., style="W")
```

The style argument means the style of the output — B is binary encoding listing them as neighbours or not, W row standardised that we saw last week.

Now run a moran's I test on the residuals, first using queens neighbours

```
Queen <- LonWardProfiles %>%
  st_drop_geometry()%>%
  dplyr::select(model2resids)%>%
  pull()%>%
  moran.test(., Lward.queens_weight)%>%
  tidy()
```

Then nearest k-nearest neighbours

```
Nearest_neighbour <- LonWardProfiles %>%
  st_drop_geometry()%>%
  dplyr::select(model2resids)%>%
  pull()%>%
  moran.test(., Lward.knn_4_weight)%>%
  tidy()
```

```
Queen
```

```
## # A tibble: 1 x 7
##   estimate1 estimate2 estimate3 statistic  p.value method           alternative
##       <dbl>     <dbl>     <dbl>     <dbl>    <dbl> <chr>            <chr>
## 1     0.282   -0.0016  0.000556      12.0 1.54e-33 Moran I test und~ greater
```

```
Nearest_neighbour
```

```
## # A tibble: 1 x 7
##   estimate1 estimate2 estimate3 statistic  p.value method           alternative
##       <dbl>     <dbl>     <dbl>     <dbl>    <dbl> <chr>            <chr>
## 1     0.292   -0.0016  0.000718      10.9 3.78e-28 Moran I test und~ greater
```

Observing the Moran's I statistic for both Queen's case neighbours and k-nearest neighbours of 4, we can see that the Moran's I statistic is somewhere between 0.27 and 0.29. Remembering that Moran's I ranges from between -1 and +1 (0 indicating no spatial autocorrelation) we can conclude that there is some weak to moderate spatial autocorrelation in our residuals. Queen 4 k Moran's I Moran's I 0.27 0.29 Moran's I -1 +1 0 This means that despite passing most of the assumptions of linear regression, we could have a situation here where the presence of some spatial autocorrelation could be leading to biased estimates of our parameters and significance values. #### waywiser This process of detecting spatial autocorrelation is becoming much easier. Whilst this is a beyond the scope of the module, the new package waywiser let's you conduct this analysis (build a weight matrix and then run spatial autocorrelation in model residuals) in just a few lines of code…This is beyond the scope here. waywiser …… # Spatial Regression Models ## Dealing with Spatially Autocorrelated Residuals - Spatial Lag and Spatial Error models - ### The Spatial Lag (lagged dependent variable) model In the example models we ran above we were testing the null-hypothesis that there is no relationship between the average GCSE scores recorded for secondary school pupils in different Wards in London and other explanatory variables. Running regression models that tested the effects of absence from school and average house price, early indications were that we could reject this null-hypothsis as the regression models ran indicated that close to 50% of the variation in GCSE scores could be explained by variations in unauthorised absence from school and average house prices. GCSE GCSE 50% However, running a Moran's I test on the residuals from the model suggested that there might be some spatial autocorreation occurring suggesting that places where the model over-predicted GCSE scores (those shown in blue in the map above with negative residuals) and under-predicted (those shown in red/orange) occasionally were near to each other. Moran's I GCSE / Overlaying the locations of secondary schools in London onto the map reveals why this could be the case. Many of the schools in London lie on or near the bounaries of the wards that pupils will live in. Therefore, it is likely that pupils attending a school could come from a number of neighbouring wards.

As such the average GCSE score in one ward could be related to that in another as the pupils living in these wards may be attending the same school. This could be the source of the autocorrelation. GCSE GCSE practical 8.6.1.1 (w_i.y_i) Wards GCSE GCSE GCSE GCSE

Let's run the original model again to remind ourselves of the paramters:

```
#Original Model
model2 <- lm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_2013 +
               log(median_house_price_2014), data = LonWardProfiles)

tidy(model2)
```

```
## # A tibble: 3 x 5
##   term                                      estimate std.error statistic  p.value
##   <chr>                                        <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                                  202.      20.1      10.0  4.79e-22
## 2 unauthorised_absence_in_all_schools_per~     -36.2      1.92    -18.9  3.71e-63
## 3 log(median_house_price_2014)                  12.8      1.50      8.50 1.37e-16
```

```
library(spatialreg)
```

Queen’s case lag

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
##
## Attaching package: 'spatialreg'
```

```
## The following objects are masked from 'package:spdep':
##
##     get.ClusterOption, get.coresOption, get.mcOption,
##     get.VerboseOption, get.ZeroPolicyOption, set.ClusterOption,
##     set.coresOption, set.mcOption, set.VerboseOption,
##     set.ZeroPolicyOption
```

```
slag_dv_model2_queen <- lagsarlm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_sc
               log(median_house_price_2014),
               data = LonWardProfiles,
               nb2listw(LWard_nb, style="C"),
               method = "eigen")

#what do the outputs show?
tidy(slag_dv_model2_queen)
```

```
## # A tibble: 4 x 5
##   term                                      estimate std.error statistic  p.value
##   <chr>                                        <dbl>     <dbl>     <dbl>    <dbl>
## 1 rho                                        5.16e-3   0.00759     0.679 4.97e- 1
## 2 (Intercept)                                2.02e+2  20.1        10.1   0
## 3 unauthorised_absence_in_all_schools_per~ -3.62e+1   1.91      -18.9    0
## 4 log(median_house_price_2014)               1.26e+1   1.53        8.21  2.22e-16
```

```
#glance() gives model stats (     ) but this need something produced from a linear model #here we have
glance(slag_dv_model2_queen)
```

```
## # A tibble: 1 x 6
##   r.squared   AIC   BIC deviance logLik  nobs
##       <dbl> <dbl> <dbl>    <dbl>  <dbl> <int>
## 1     0.484 5217. 5239.  150150. -2604.   626
```

```
t<-summary(slag_dv_model2_queen)

sum(t$residuals)
```

```
## [1] -8.570922e-13
```

Running the spatially-lagged model with a Queen's case spatial weights matrix reveals that in this example, there is an insignificant and small effect associated with the spatially lagged dependent variable. However, a different conception of neighbours we might get a different outcome

Here:

Rho is our spatial lag (0.0051568) that measures the variable in the surrounding spatial areas as defined by the spatial weight matrix. We use this as an extra explanatory variable to account for clustering (identified by Moran's I). If significant it means that the GCSE scores in a unit vary based on the GCSE scores in the neighboring units. If it is positive it means as the GCSE scores increase in the surrounding units so does our central value

Log likelihood shows how well the data fits the model (like the AIC, which we cover later), the higher the value the better the models fits the data.

Likelihood ratio (LR) test shows if the addition of the lag is an improvement (from linear regression) and if that's significant. This code would give the same output…

```
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
lrtest(slag_dv_model2_queen, model2)
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): original model was of
## class "Sarlm", updated model is of class "lm"

## Likelihood ratio test
##
## Model 1: average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_2013 +
##     log(median_house_price_2014)
## Model 2: average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_2013 +
##     log(median_house_price_2014)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -2603.5
## 2   4 -2603.7 -1 0.4618     0.4968
```

Lagrange Multiplier (LM) is a test for the absence of spatial autocorrelation in the lag model residuals. If significant then you can reject the Null (no spatial autocorrelation) and accept the alternative (is spatial autocorrelcation)

Wald test (often not used in interpretation of lag models), it tests if the new parameters (the lag) should be included it in the model…if significant then the new variable improves the model fit and needs to be included. This is similar to the LR test and i've not seen a situation where one is significant and the other not. Probably why it's not used!

In this case we have spatial autocorrelation in the residuals of the model, but the model is not an improvement on OLS — this can also be confirmed with the AIC score (we cover that later) but the lower it is the better. Here it is 5217, in OLS (model 2) it was 5215. The Log likelihood is the other way around but very close, model 2 (OLS) it was -2604, here it is -2603. OLS

OLS AIC AIC

AIC AIC 5217 OLS AIC 5215 OLS Log likelihood 2604 -2603 OLS AIC #### Lag impacts Warning according to Solymosi and Medina (2022) you must not not compare the coefficients of this to regular OLS…Why ? Well in OLS recall we can use the coefficients to say…a 1 unit change in the independent variable means a drop or rise in the dependent (for a 1% increase in unauthorised absence from school GSCE scores to drop by -41.2 points). BUT here the model is not consistent as the observations will change based on the weight matrix neighbours selected which might vary (almost certainly in a distance based matrix). This means we have a direct effect (standard OLS) and then an indirect effect in the model (impact of the spatial lag). OLS …… 1 1% GSCE -41.2 OLS We can compute these direct and indirect effects using code from Solymosi and Medina (2022) and the spatialreg package. Here the impacts() function calculates the impact of the spatial lag. We can fit this to our entire spatial weights…. Solymosi Medina (2022) Spatialreg impact() ……

```
# weight list is just the code from the lagsarlm
weight_list<-nb2listw(LWard_knn, style="C")

imp <- impacts(slag_dv_model2_queen, listw=weight_list)

imp
```

```
## Impact measures (lag, exact):
##                                                 Direct   Indirect     Total
## unauthorised_absence_in_all_schools_percent_2013 -36.1758 -0.1873222 -36.36312
## log(median_house_price_2014)                     12.5879  0.0651815  12.65308
```

Now it is appropriate to compare these coefficients to the OLS outputs…however if you have a very large matrix this might not work, instead a sparse matrix that uses approximation methods (see Solymosi and Medina (2022) and within that resource, Lesage and Pace 2009). This is beyond the scope of the content here, but essentially this makes the method faster on larger data…but only row standardised is permitted here… OLS …… Solymosi Medina (2022) Lesage Pace2009 …… ……

```
slag_dv_model2_queen_row <- lagsarlm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all
               log(median_house_price_2014),
               data = LonWardProfiles,
               nb2listw(LWard_nb, style="W"),
               method = "eigen")


W <- as(weight_list, "CsparseMatrix")

trMatc <- trW(W, type="mult")
trMC <- trW(W, type="MC")

imp2 <- impacts(slag_dv_model2_queen_row, tr=trMatc, R=200)

imp3 <- impacts(slag_dv_model2_queen_row, tr=trMC, R=200)

imp2
```

```
## Impact measures (lag, trace):
##                                                    Direct   Indirect
## unauthorised_absence_in_all_schools_percent_2013 -29.581803 -18.450158
## log(median_house_price_2014)                       9.908963   6.180216
##                                                    Total
## unauthorised_absence_in_all_schools_percent_2013 -48.03196
## log(median_house_price_2014)                      16.08918
```

```
imp3
```

```
## Impact measures (lag, trace):
##                                                      Direct   Indirect
## unauthorised_absence_in_all_schools_percent_2013 -29.584952 -18.447010
## log(median_house_price_2014)                       9.910017   6.179161
##                                                      Total
## unauthorised_absence_in_all_schools_percent_2013 -48.03196
## log(median_house_price_2014)                      16.08918
```

We can also get the p-values (where an R is set, this is the number of simulations to use)…from the sparse computation p R

```
sum <- summary(imp2,  zstats=TRUE, short=TRUE)

sum
```

```
## Impact measures (lag, trace):
##                                                      Direct   Indirect
## unauthorised_absence_in_all_schools_percent_2013 -29.581803 -18.450158
## log(median_house_price_2014)                       9.908963   6.180216
##                                                      Total
## unauthorised_absence_in_all_schools_percent_2013 -48.03196
## log(median_house_price_2014)                      16.08918
## ========================================================
## Simulation results ( variance matrix):
## ========================================================
## Simulated standard errors
##                                                    Direct Indirect    Total
## unauthorised_absence_in_all_schools_percent_2013 1.846482 2.797153 3.350249
## log(median_house_price_2014)                     1.509159 1.215325 2.414184
##
## Simulated z-values:
##                                                      Direct  Indirect
## unauthorised_absence_in_all_schools_percent_2013 -15.931194 -6.715060
## log(median_house_price_2014)                       6.636831  5.245413
##                                                       Total
## unauthorised_absence_in_all_schools_percent_2013 -14.386902
## log(median_house_price_2014)                       6.789423
##
## Simulated p-values:
##                                                  Direct     Indirect
## unauthorised_absence_in_all_schools_percent_2013 < 2.22e-16 1.8799e-11
## log(median_house_price_2014)                     3.205e-11  1.5593e-07
##                                                  Total
## unauthorised_absence_in_all_schools_percent_2013 < 2.22e-16
## log(median_house_price_2014)                     1.1258e-11
```

The results on the entire datasets will differ as that used C which is a globally standardised weight matrix. In the sparse example, there are different two examples using slightly different arguments that control the sparse matrix, this is beyond the scope here (so don't worry about it) but for reference…. mult which is (default) for powering a sparse matrix (with moderate or larger N, the matrix becomes dense, and may lead to swapping) MC for Monte Carlo simulation of the traces (the first two simulated traces are replaced by their analytical equivalents) The purpose of providing this extra step is in case you have a large

data set in the exam and wish to do compute the direct and indirect. For more details and another example see Fitting and interpreting a spatially lagged model by Solymosi and Medina (2022).
C …… mult N MC
Solymosi Medina (2022) #### KNN case lag Now let's run a model with nearest neigh ours as opposed to queens neighbours

```
#run a spatially-lagged regression model
slag_dv_model2_knn4 <- lagsarlm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_scho
               log(median_house_price_2014),
               data = LonWardProfiles,
               nb2listw(LWard_knn,
                        style="C"),
               method = "eigen")

#what do the outputs show?
tidy(slag_dv_model2_knn4)
```

```
## # A tibble: 4 x 5
##   term                                      estimate std.error statistic  p.value
##   <chr>                                        <dbl>     <dbl>     <dbl>    <dbl>
## 1 rho                                          0.374    0.0409      9.14 0
## 2 (Intercept)                                116.      20.1         5.76 8.39e- 9
## 3 unauthorised_absence_in_all_schools_per~   -28.5      1.97      -14.5   0
## 4 log(median_house_price_2014)                 9.29     1.48        6.28 3.36e-10
```

Using the 4 nearest neighbours instead of just considering all adjacent zones in the spatial weights matrix, the size and significance of the spatially lagged term changes quite dramatically. In the 4 nearest neighbour model it is both quite large, positive and statistically significant (<0.05), conversely the effects of unauthorised absence and (log (median house price)) are reduced. Remember…this is how we interpret the coefficients…
4 4 <0.05 …… … Before a 1% increase (or 1 unit, it is % as the variable is %) in unauthorized absence meant GCSE scores dropped by -36.36 points, now they just drop by -28.5 points. 1% 1 % % GCSE
-36.36 -28.5

Here as we have logged the median house price we must follow the rules… Divide the coefficient by 100 (it was 12.65 it is now 9.29 = 0.1265 and 0.0929) For every 1% increase in the independent variable (median house price) the dependent (GCSE scores) increases by around 0.09 points (previously 0.12)
100 12.65 9.29 = 0.1265 0.0929 1% GCSE 0.09 0.12

What this means is that in our study area, the average GCSE score recorded in Wards across the city varies partially with the average GCSE score found in neighbouring Wards. Given the distribution of schools in the capital in relation to where pupils live, this makes sense as schools might draw pupils from a few close neighbouring wards rather than all neighbour bordering a particular Ward. GCSE GCSE
Wards GCSE GCSE GCS
Effectively, by ignoring the effects of spatial autocorrelation in the original OLS model, the impacts of unauthorised absence and affluence (as represented by average house price) were slightly overplayed or exaggerated (meaning the OLS coefficients were too high). OLS OLS We can also now check that the residuals from the spatially lagged model are now no-longer exhibiting spatial autocorrelation:

```
#write out the residuals

LonWardProfiles <- LonWardProfiles %>%
  mutate(slag_dv_model2_knn_resids = residuals(slag_dv_model2_knn4))

KNN4Moran <- LonWardProfiles %>%
  st_drop_geometry()%>%
  dplyr::select(slag_dv_model2_knn_resids)%>%
  pull()%>%
  moran.test(., Lward.knn_4_weight)%>%
  tidy()

KNN4Moran
```

```
## # A tibble: 1 x 7
##   estimate1 estimate2 estimate3 statistic p.value method             alternative
##       <dbl>     <dbl>     <dbl>     <dbl>   <dbl> <chr>              <chr>
## 1    0.0468   -0.0016  0.000717      1.81  0.0353 Moran I test unde~ greater
```

8.6.1.2 The Spatial Error Model

Another way of coneptualising spatial dependence in regression models is not through values of the dependent variable in some areas affecting those in neighbouring areas (as they do in the spatial lag model), but in treating the spatial autocorrelation in the residuals as something that we need to deal with, perhaps reflecting some spatial autocorrelation amongst unobserved independent variables or some other mis-specification of the model. Ward and Gleditsch (2008) characterise this model as seeing spatial autocorrelation as a nuisance rather than being particularly informative, however it can still be handled within the model, albeit slightly differently. Ward Gleditsch 2008 We can run a spatial error model on the same data below:

```
sem_model1 <- errorsarlm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_per
               log(median_house_price_2014),
               data = LonWardProfiles,
               nb2listw(LWard_knn, style="C"),
               method = "eigen")

tidy(sem_model1)
```

```
## # A tibble: 4 x 5
##   term                                     estimate std.error statistic  p.value
##   <chr>                                       <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                               154.      28.2         5.44 5.28e- 8
## 2 unauthorised_absence_in_all_schools_per~  -32.3      2.22      -14.5  0
## 3 log(median_house_price_2014)               16.2      2.12        7.62 2.55e-14
## 4 lambda                                      0.475    0.0451     10.5  0
```

Comparing the results of the spatial error model with the spatially lagged model and the original OLS model, the suggestion here is that the spatially correlated errors in residuals lead to an over-estimate of the importance of unauthorised absence in the OLS model and an under-estimate of the importance of affluence, represented by median house prices. Conversely, the spatial error model estimates higher parameter values for both variables when compared to the spatially lagged model. OLS OLS Note, here we can compare to OLS as there is no spatial lag. OLS Both the parameter in the spatial error model and the parameter in the spatially lagged model are larger than their standard errors, so we can conclude that spatial dependence should be borne in mind when interpreteing the results of this regression model.

[2] p 0.05

## 8.6.2 Key advice The lag model accounts for situations where the value of the dependent variable in one area might be associated with or influenced by the values of that variable in neighbouring zones (however we choose to define neighbouring in our spatial weights matrix). With our example, average GCSE scores in one neighbourhood might be related to average GCSE scores in another as the students in both

neighbourhoods could attend the same school. You may be able to think of other examples where similar associations may occur. You might run a lag model if you identify spatial autocorrelation in the dependent variable (closer spatial units have similar values) with Moran's I.

The error model deals with spatial autocorrelation (closer spatial units have similar values) of the residuals (vertical distance between your point and line of model – errors – over-predictions or under-predictions) again, potentially revealed though a Moran's I analysis. The error model is not assuming that neighbouring independent variables are influencing the dependent variable but rather the assumption is of an issue with the specification of the model or the data used (e.g. clustered errors are due to some un-observed clustered variables not included in the model). For example, GCSE scores may be similar in bordering neighbourhoods but not because students attend the same school but because students in these neighbouring places come from similar socio-economic or cultural backgrounds and this was not included as an independent variable in the original model. There is no spatial process (no cross Borough interaction) just a cluster of an un-accounted for but influential variable.

Usually you might run a lag model when you have an idea of what is causing the spatial autocorrelation in the dependent variable and an error model when you aren't sure what might be missing. However, you can also use a more scientific method - the Lagrange Multiplier test.

But! recall from a few weeks ago when I made a big deal of type of standardisation for the spatial weight matrix? This test expects row standardisation.

The Lagrange multiple tests are within the function lm.LMtests from the package spdep:

LMerr is the spatial error model test LMlag is the lagged test With each also having a robust form, being robust to insensitivities to changes (e.g. outliers, non-normality).
GCSE GCSE I

– – I
GCSE

–

spdep lm.LMtests

LMerr LMlag LMerr LMlag

```r
library(spdep)

Lward.queens_weight_ROW <- LWard_nb %>%
  nb2listw(., style="W")

lm.LMtests(model2, Lward.queens_weight_ROW, test = c("LMerr","LMlag","RLMerr","RLMlag","SARMA"))
```

```
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = average_gcse_capped_point_scores_2014 ~
## unauthorised_absence_in_all_schools_percent_2013 +
## log(median_house_price_2014), data = LonWardProfiles)
## weights: Lward.queens_weight_ROW
##
## LMerr = 141.05, df = 1, p-value < 2.2e-16
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
```

```
## data:
## model: lm(formula = average_gcse_capped_point_scores_2014 ~
## unauthorised_absence_in_all_schools_percent_2013 +
## log(median_house_price_2014), data = LonWardProfiles)
## weights: Lward.queens_weight_ROW
##
## LMlag = 97.669, df = 1, p-value < 2.2e-16
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = average_gcse_capped_point_scores_2014 ~
## unauthorised_absence_in_all_schools_percent_2013 +
## log(median_house_price_2014), data = LonWardProfiles)
## weights: Lward.queens_weight_ROW
##
## RLMerr = 43.746, df = 1, p-value = 3.738e-11
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = average_gcse_capped_point_scores_2014 ~
## unauthorised_absence_in_all_schools_percent_2013 +
## log(median_house_price_2014), data = LonWardProfiles)
## weights: Lward.queens_weight_ROW
##
## RLMlag = 0.36458, df = 1, p-value = 0.546
##
##
##  Lagrange multiplier diagnostics for spatial dependence
##
## data:
## model: lm(formula = average_gcse_capped_point_scores_2014 ~
## unauthorised_absence_in_all_schools_percent_2013 +
## log(median_house_price_2014), data = LonWardProfiles)
## weights: Lward.queens_weight_ROW
##
## SARMA = 141.42, df = 2, p-value < 2.2e-16
```

Here, we look to see if either the standard tests LMerr or LMlag are significant (p <0.05), if one is then that is our answer. If both are move to the robust tests and apply the same rule.

If everything is significant then Prof Anselin (2008) proposed: One robust test will often be much more significant than the other. In which select the most significant model. In the case both a highly significant select the largest test statistic value - but here, there may be violations of the regression assumptions

Here, based on this test and guidance which is the right model to select?

For more information on the Lagrange Multipler see: Lagrange multiple tests from crime mapping in R Luc Anselin's 2003 tutorial

LMerr LMlag p <0.05

2008 “ ” –

R Luc Anselin 2003

Lagrange LM

LMerr LMlag

LMerr 141.05 df 1 p 2.2e-16 LMlag 97.669 1 p 2.2e-16 RLMerr RLMlag

LM

RLMerr 43.746 1 p 3.738e-11 RLMlag 0.36458 1 p 0.546 RLMerr RLMlag SEM

SARMA

SARMA 141.42 2 p 2.2e-16 RLM Anselin LM SEM
Degrees of Freedom df

n-1 n n n-1

t t

n n n-1

t p
“ ”

Geographically weighted regression (GWR) will be explored next, but simply assumes that spatial autocorrelation is not a problem, but a global regression model of all our data doesn’t have the same regression slope - e.g. in certain areas (Boroughs, Wards) the relationship is different, termed non-stationary. GWR runs a local regression model for adjoining spatial units and shows how the coefficients can vary over the study area. GWR – GWR
GWR “ ”

GWR GWR

GWR 1 2000 500 GWR
## 8.6.3 Which model to use Usually you will run OLS regression first then look for spatial autocorrelation of the residuals (Moran’s I).

Once at this stage you need to make a decision about the model: Is it a global model (error / lag) or a local model (GWR)? Can a single model (error/lag) be fitted to the study area? Is the spatial autocorrelation a problem (error) or showing local trends (GWR)?

Of course you could do a OLS, spatial lag and GWR as long as they all contribute something to your analysis.
OLS (Moran’s I)

/ GWR / GWR OLS

OLS GWR ## 8.6.4 More data We will now read in some extra data which we will use shortly

```
extradata <- read_csv("https://www.dropbox.com/s/qay9q1jwpffxcqj/LondonAdditionalDataFixed.csv?raw=1")
```

```
## Rows: 625 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): WardName, WardCode, Wardcode, Candidate, InnerOuter
## dbl (6): PctSharedOwnership2011, PctRentFree2011, x, y, AvgGCSE2011, UnauthA...
##
## 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.
```

```
#add the extra data too
LonWardProfiles <- LonWardProfiles%>%
  left_join(.,
            extradata,
            by = c("gss_code" = "Wardcode"))%>%
  clean_names()

#print some of the column names
LonWardProfiles%>%
  names()%>%
  tail(., n=10)
```

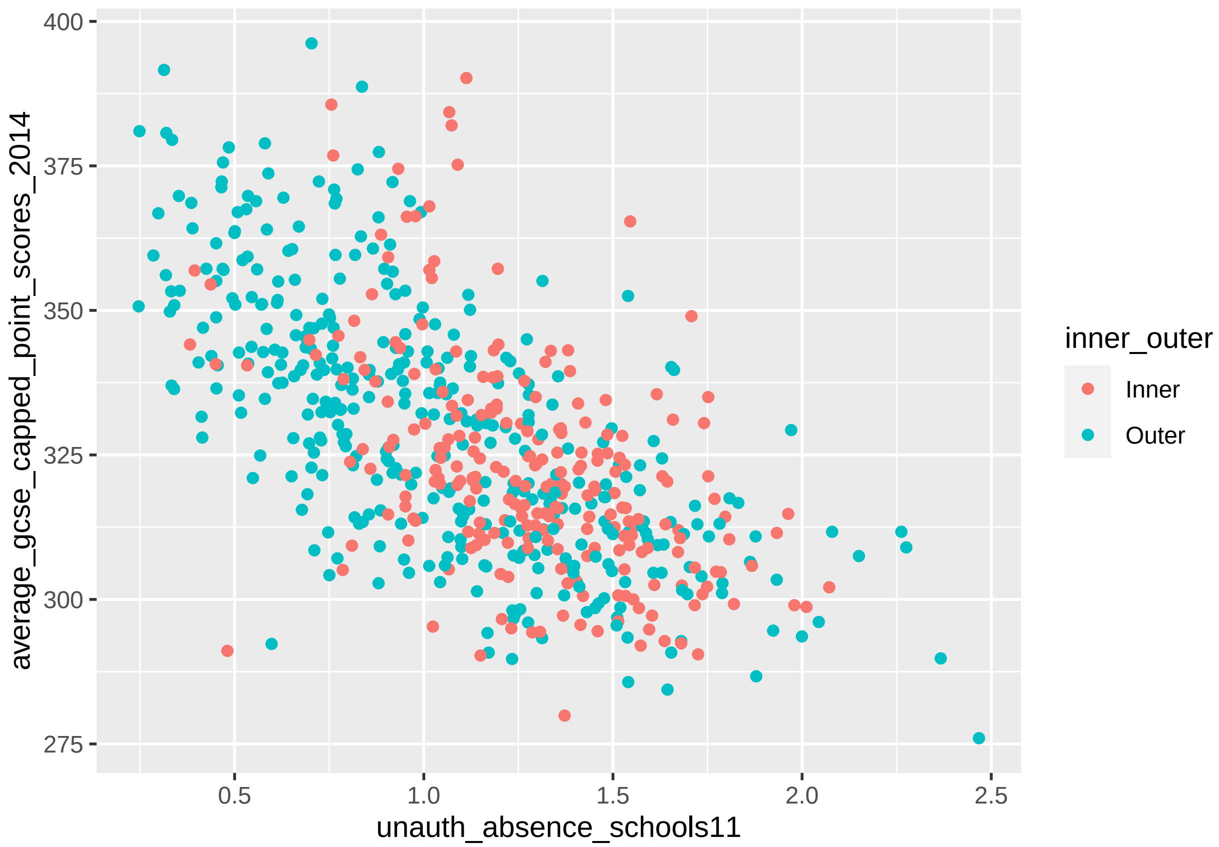
```
##  [1] "ward_code"                 "pct_shared_ownership2011"
##  [3] "pct_rent_free2011"         "candidate"
##  [5] "inner_outer"               "x"
##  [7] "y"                         "avg_gcse2011"
##  [9] "unauth_absence_schools11" "geometry"
```

8.6.5 Extending your regression model - Dummy Variables

What if instead of fitting one line to our cloud of points, we could fit several depending on whether the Wards we were analysing fell into some or other group. What if the relationship between attending school and achieving good exam results varied between inner and outer London, for example. Could we test for that? Well yes we can - quite easily in fact.

—— If we colour the points representing Wards for Inner and Outer London differently, we can start to see that there might be something interesting going on. Using 2011 data (as there are not the rounding errors that there are in the more recent data), there seems to be a stronger relationship between absence and GCSE scores in Outer London than Inner London. We can test for this in a standard linear regression model. 2011 GCSE

```
p <- ggplot(LonWardProfiles,
            aes(x=unauth_absence_schools11,
                # "unauth"   "unauthorized"
                y=average_gcse_capped_point_scores_2014))
p + geom_point(aes(colour = inner_outer))
```



Dummy variables are always categorical data (inner or outer London, or red / blue etc.). When we incorporate them into a regression model, they serve the purpose of splitting our analysis into groups. In the graph above, it would mean, effectively, having a separate regression line for the red points and a separate line for the blue points. / Let's try it!

```
#first, let's make sure R is reading our InnerOuter variable as a factor
#see what it is at the moment...
isitfactor <- LonWardProfiles %>%
  dplyr::select(inner_outer)%>%
  summarise_all(class)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## i The deprecated feature was likely used in the dplyr package.
##   Please report the issue at <https://github.com/tidyverse/dplyr/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
isitfactor
```

```
## Simple feature collection with 2 features and 1 field
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 503568.2 ymin: 155850.8 xmax: 561957.5 ymax: 200933.9
## Projected CRS: OSGB36 / British National Grid
##   inner_outer                       geometry
## 1   character POLYGON ((517066.3 167341.1...
## 2   character POLYGON ((517066.3 167341.1...
```

```
# change to factor

LonWardProfiles<- LonWardProfiles %>%
  mutate(inner_outer=as.factor(inner_outer))

#now run the model
model3 <- lm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_2013 +
               log(median_house_price_2014) +
               inner_outer,
             data = LonWardProfiles)

tidy(model3)
```

```
## # A tibble: 4 x 5
##   term                                         estimate std.error statistic  p.value
##   <chr>                                           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                                      97.6     24.1       4.06 5.62e- 5
## 2 unauthorised_absence_in_all_schools_per~        -30.1      2.03    -14.8  7.02e-43
## 3 log(median_house_price_2014)                     19.8      1.74     11.4  2.09e-27
## 4 inner_outerOuter                                 10.9      1.51      7.24 1.30e-12
```

So how can we interpret this?

Well, the dummy variable is statistically significant and the coefficient tells us the difference between the two groups (Inner and Outer London) we are comparing. In this case, it is telling us that living in a Ward in outer London will improve your average GCSE score by 10.93 points, on average, compared to if you lived in Inner London. The R-squared has increased slightly, but not by much. Ward GCSE 10.93 R You will notice that despite there being two values in our dummy variable (Inner and Outer), we only get one coefficient. This is because with dummy variables, one value is always considered to be the control (comparison/reference) group. In this case we are comparing Outer London to Inner London. If our dummy variable had more than 2 levels we would have more coefficients, but always one as the reference. / The order in which the dummy comparisons are made is determined by what is known as a 'contrast matrix'. This determines the treatment group (1) and the control (reference) group (0). We can view the contrast matrix using the contrasts() function: " " (1) (0) contrasts()

```
contrasts(LonWardProfiles$inner_outer)
```

```
##       Outer
## Inner     0
## Outer     1
```

If we want to change the reference group, there are various ways of doing this. We can use the contrasts() function, or we can use the relevel() function. Let's try it: contrasts() relevel()

```
LonWardProfiles <- LonWardProfiles %>%
  mutate(inner_outer = relevel(inner_outer,
                               ref="Outer"))

model3 <- lm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_2013 +
               log(median_house_price_2014) +
               inner_outer,
             data = LonWardProfiles)

tidy(model3)
```

```
## # A tibble: 4 x 5
##   term                                        estimate std.error statistic  p.value
##   <chr>                                          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                                    109.      23.2       4.68 3.53e- 6
## 2 unauthorised_absence_in_all_schools_per~       -30.1      2.03    -14.8  7.02e-43
## 3 log(median_house_price_2014)                    19.8      1.74     11.4  2.09e-27
## 4 inner_outerInner                               -10.9      1.51     -7.24 1.30e-12
```

You will notice that the only thing that has changed in the model is that the coefficient for the inner_outer variable now relates to Inner London and is now negative (meaning that living in Inner London is likely to reduce your average GCSE score by 10.93 points compared to Outer London). The rest of the model is exactly the same. inner_outer GCSE 10.93 ## 8.6.6 TASK: Investigating Further - Adding More Explanatory Variables into a multiple regression model Now it's your turn. You have been shown how you could begin to model average GCSE scores across London, but the models we have built so far have been fairly simple in terms of explanatory variables.
GCSE You should try and build the optimum model of GCSE performance from your data in your LondonWards dataset. Experiment with adding different variables - when building a regression model in this way, you are trying to hit a sweet spot between increasing your R-squared value as much as possible, but with as few explanatory variables as possible. GCSE –
R ### 8.6.6.1 A few things to watch out for… …… You should never just throw variables at a model without a good theoretical reason for why they might have an influence. Choose your variables carefully! Be prepared to take variables out of your model either if a new variable confounds (becomes more important than) earlier variables or turns out not to be significant. “ ”

“ ”

For example, let's try adding the rate of drugs related crime (logged as it is a positively skewed variable, where as the log is normal) and the number of cars per household… are these variables significant? What happens to the spatial errors in your models? ……

[1]

[2]

[3]

Box-Cox Box-Cox lambda [4]

[1]

[2]

Q-Q # 8.7 Task 3 - Spatial Non-stationarity and Geographically Weighted Regression Models (GWR) (GWR) “Spatial Non-stationarity” “Geographically Weighted Regression Models (GWR)” GWR

Here's my final model from the last section:

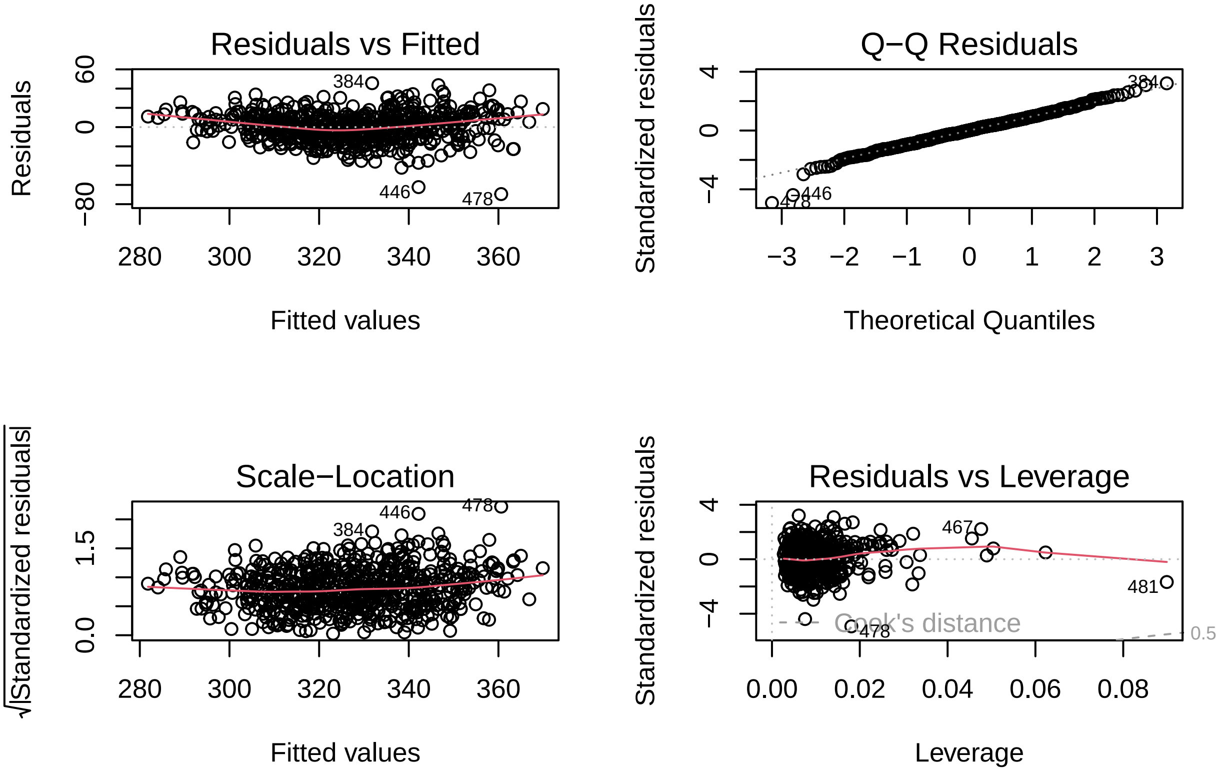
```
#select some variables from the data file
myvars <- LonWardProfiles %>%
  dplyr::select(average_gcse_capped_point_scores_2014,
         unauthorised_absence_in_all_schools_percent_2013,
         median_house_price_2014,
         rate_of_job_seekers_allowance_jsa_claimants_2015,
         percent_with_level_4_qualifications_and_above_2011,
         inner_outer)

#check their correlations are OK
Correlation_myvars <- myvars %>%
  st_drop_geometry()%>%
  dplyr::select(-inner_outer)%>%
  correlate()
```

```
## Correlation computed with
## * Method: 'pearson'
## * Missing treated using: 'pairwise.complete.obs'
```

```
#run a final OLS model
model_final <- lm(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_20
                    log(median_house_price_2014) +
                    inner_outer +
                    rate_of_job_seekers_allowance_jsa_claimants_2015 +
                    percent_with_level_4_qualifications_and_above_2011,
                  data = myvars)

tidy(model_final)
```

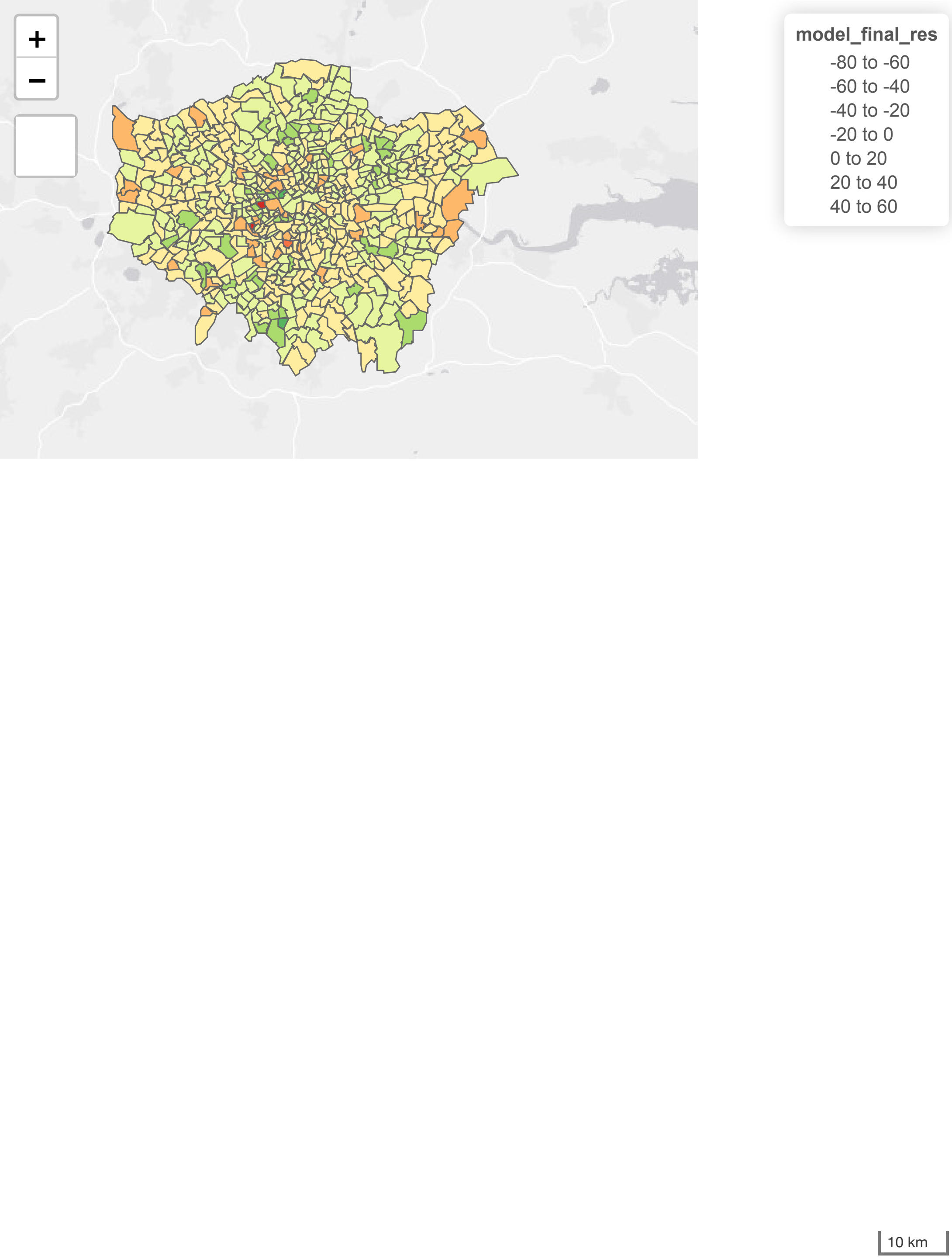
```
## # A tibble: 6 x 5
##   term                                     estimate std.error statistic  p.value
##   <chr>                                       <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                               240.      27.9         8.59 7.02e-17
## 2 unauthorised_absence_in_all_schools_per~  -23.6      2.16      -10.9   1.53e-25
## 3 log(median_house_price_2014)                8.41     2.26        3.72 2.18e- 4
## 4 inner_outerInner                          -10.4      1.65       -6.30 5.71e-10
## 5 rate_of_job_seekers_allowance_jsa_claim~   -2.81     0.635      -4.43 1.12e- 5
## 6 percent_with_level_4_qualifications_and~    0.413    0.0784      5.27 1.91e- 7
```

```
LonWardProfiles <- LonWardProfiles %>%
  mutate(model_final_res = residuals(model_final))

par(mfrow=c(2,2))
plot(model_final)
```



```r
qtm(LonWardProfiles, fill = "model_final_res")
```

```
## Variable(s) "model_final_res" contains positive and negative values, so midpoint is set to 0. Set mi
```



Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ

```r
final_model_Moran <- LonWardProfiles %>%
  st_drop_geometry()%>%
  dplyr::select(model_final_res)%>%
  pull()%>%
  moran.test(., Lward.knn_4_weight)%>%
  tidy()

final_model_Moran
```

```
## # A tibble: 1 x 7
##   estimate1 estimate2 estimate3 statistic  p.value method           alternative
##       <dbl>     <dbl>     <dbl>     <dbl>    <dbl> <chr>            <chr>
## 1     0.224   -0.0016  0.000718      8.42 1.91e-17 Moran I test und~ greater
```

Now, we probably could stop at running a spatial error model at this point, but it could be that rather than spatial autocorrelation causing problems with our model, it might be that a “global” regression model does not capture the full story. In some parts of our study area, the relationships between the dependent and independent variable may not exhibit the same slope coefficient. While, for example, increases in unauthorised absence usually are negatively correlated with GCSE score (students missing school results in lower exam scores), in some parts of the city, they could be positively correlated (in affluent parts of the city, rich parents may enrol their children for just part of the year and then live elsewhere in the world for another part of the year, leading to inflated unauthorised absence figures. Ski holidays are cheaper during the school term, but the pupils will still have all of the other advantages of living in a well off household that will benefit their exam scores. “ ”

GCSE

If this occurs, then we have ‘non-stationarity’ - this is when the global model does not represent the relationships between variables that might vary locally. “ ”—— This part of the practical will only skirt the edges of GWR, for much more detail you should visit the GWR website which is produced and maintained by Prof Chris Brunsdon and Dr Martin Charlton who originally developed the technique - http://gwr.nuim.ie/ GWR GWR Chris Brunsdon Martin Charlton - http://gwr.nuim.ie/ There are various packages which will carry out GWR in R, for this pracical we we use spgwr (mainly because it was the first one I came across), although you could also use GWmodel or gwrr. R GWR spgwr GWmodel gwrr I should also acknowledge the guide on GWR produced by the University of Bristol, which was a great help when producing this exercise. GWR

```
library(spgwr)
```

```
## NOTE: This package does not constitute approval of GWR
## as a method of spatial analysis; see example(gwr)
```

```
# coordsW
coordsW2 <- st_coordinates(coordsW)

LonWardProfiles2 <- cbind(LonWardProfiles,coordsW2)
#    x y

GWRbandwidth <- gwr.sel(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_perc
                    log(median_house_price_2014) +
                    inner_outer +
                    rate_of_job_seekers_allowance_jsa_claimants_2015 +
                    percent_with_level_4_qualifications_and_above_2011,
                  data = LonWardProfiles2,
                        coords=cbind(LonWardProfiles2$X, LonWardProfiles2$Y),
                  adapt=T)
```

```
## Adaptive q: 0.381966 CV score: 124832.2
## Adaptive q: 0.618034 CV score: 126396.1
## Adaptive q: 0.236068 CV score: 122752.4
## Adaptive q: 0.145898 CV score: 119960.5
## Adaptive q: 0.09016994 CV score: 116484.6
## Adaptive q: 0.05572809 CV score: 112628.7
## Adaptive q: 0.03444185 CV score: 109427.7
## Adaptive q: 0.02128624 CV score: 107562.9
## Adaptive q: 0.01315562 CV score: 108373.2
## Adaptive q: 0.02161461 CV score: 107576.6
## Adaptive q: 0.0202037 CV score: 107505.1
## Adaptive q: 0.01751157 CV score: 107333
## Adaptive q: 0.01584775 CV score: 107175.5
## Adaptive q: 0.01481944 CV score: 107564.8
## Adaptive q: 0.01648327 CV score: 107187.9
## Adaptive q: 0.01603246 CV score: 107143.9
## Adaptive q: 0.01614248 CV score: 107153.1
## Adaptive q: 0.01607315 CV score: 107147.2
## Adaptive q: 0.01596191 CV score: 107143
## Adaptive q: 0.01592122 CV score: 107154.4
## Adaptive q: 0.01596191 CV score: 107143
```

```
GWRbandwidth
```

```
## [1] 0.01596191
```

Setting adapt=T here means to automatically find the proportion of observations for the weighting using k nearest neighbours (an adaptive bandwidth), False would mean a global bandwidth and that would be in meters (as our data is projected).

Occasionally data can come with longitude and latitude as columns (e.g. WGS84) and we can use this straight in the function to save making centroids, calculating the coordinates and then joining - the argument for this is longlat=TRUE and then the columns selected in the coords argument e.g. coords=cbind(long, lat). The distance result will then be in KM.

The optimal bandwidth is about 0.015 meaning 1.5% of all the total spatial units should be used for the local regression based on k-nearest neighbours. Which is about 9 of the 626 wards.

This approach uses cross validation to search for the optimal bandwidth, it compares different bandwidths to minimise model residuals — this is why we specify the regression model within gwr.sel(). It does this with a Gaussian weighting scheme (which is the default) - meaning that near points have greater influence in the regression and the influence decreases with distance - there are variations of this, but Gaussian is a fine to use in most applications. To change this we would set the argument gweight = gwr.Gauss in the gwr.sel() function — gwr.bisquare() is the other option. We don't go into cross validation in this module.

However we could set either the number of neighbours considered or the distance within which to considered points ourselves, manually, in the gwr() function below.

To set the number of other neighbours considered simply change the adapt argument to the value you want - it must be the number of neighbours divided by the total (e.g. to consider 20 neighbours it would be 20/626 and you'd use the value of 0.0319)

The set the bandwidth, remove the adapt argument and replace it with bandwidth and set it, in this case, in meters.

To conclude, we can:

set the bandwidth in gwr.sel() automatically using: the number of neighbors a distance threshold Or, we can set it manually in gwr() using: a number of neighbors a distance threshold adapt=T k False

WGS84 – longlat=TRUE coords coords=cbind(long,lat)

0.015 1.5% k 626 9

– gwr.sel() gwr.sel() gweight = gwr.Gauss gwr.bisquare()

gwr()

adapt 20 20/626 0.0319

adapt bandwidth

gwr.sel() gwr()

BUT a problem with setting a fixed bandwidth is we assume that all variables have the same relationship across the same space (using the same number of neighbours or distance)…(such as rate_of_job_seekers_allowance_jsa_claimants_2015 and percent_with_level_4_qualifications_and_above_2011). We can let these bandwidths vary as some relationships will operate on different spatial scales…this is called Multiscale GWR and Lex Comber recently said that all GWR should be Multisacle (oops!). We have already covered a lot here so i won't go into it. If you are interested Lex has a good tutorial on Multiscale GWR …… 2015 2011 …… GWR Lex Comber GWR Multisacle Lex GWR Geographically Weighted Regression, GWR 2015 JSA 2011 4

GWR Multiscale GWR Lex Comber GWR "oops!"

GWR Lex Comber GWR

```
#run the gwr model
gwr.model = gwr(average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_201
                    log(median_house_price_2014) +
                    inner_outer +
                    rate_of_job_seekers_allowance_jsa_claimants_2015 +
                    percent_with_level_4_qualifications_and_above_2011,
                  data = LonWardProfiles2,
                coords=cbind(LonWardProfiles2$X, LonWardProfiles2$Y),
                adapt=GWRbandwidth,
                #matrix output
                hatmatrix=TRUE,
                #standard error
                se.fit=TRUE)
```

```
## Warning in sqrt(betase): NaNs produced

## Warning in sqrt(betase): NaNs produced
```

```
#print the results of the model
gwr.model
```

```
## Call:
## gwr(formula = average_gcse_capped_point_scores_2014 ~ unauthorised_absence_in_all_schools_percent_20
##     log(median_house_price_2014) + inner_outer + rate_of_job_seekers_allowance_jsa_claimants_2015 +
##     percent_with_level_4_qualifications_and_above_2011, data = LonWardProfiles2,
##     coords = cbind(LonWardProfiles2$X, LonWardProfiles2$Y), adapt = GWRbandwidth,
##     hatmatrix = TRUE, se.fit = TRUE)
## Kernel function: gwr.Gauss
## Adaptive quantile: 0.01596191 (about 9 of 626 data points)
## Summary of GWR coefficient estimates at data points:
```

```
## Warning in print.gwr(x): NAs in coefficients dropped
##                                                           Min.    1st Qu.
## X.Intercept.                                       -345.01649  -14.73075
## unauthorised_absence_in_all_schools_percent_2013    -47.06120  -31.08397
## log.median_house_price_2014.                         -0.55994   11.18979
## inner_outerInner                                    -24.36827  -10.44459
## rate_of_job_seekers_allowance_jsa_claimants_2015      1.43895   10.72734
## percent_with_level_4_qualifications_and_above_2011   -0.06701    0.49946
##                                                         Median    3rd Qu.
## X.Intercept.                                         81.81663  179.15965
## unauthorised_absence_in_all_schools_percent_2013    -14.04901   -5.00033
## log.median_house_price_2014.                         18.00032   22.78750
## inner_outerInner                                     -6.58838   -3.33210
## rate_of_job_seekers_allowance_jsa_claimants_2015     16.11748   26.08932
## percent_with_level_4_qualifications_and_above_2011    0.72555    1.07515
##                                                           Max.   Global
## X.Intercept.                                        318.94967 239.9383
## unauthorised_absence_in_all_schools_percent_2013      6.79870 -23.6167
## log.median_house_price_2014.                         44.78874   8.4136
## inner_outerInner                                      3.98708 -10.3690
## rate_of_job_seekers_allowance_jsa_claimants_2015     52.82565  -2.8135
## percent_with_level_4_qualifications_and_above_2011    3.04231   0.4127
## Number of data points: 626
## Effective number of parameters (residual: 2traceS - traceS'S): 160.9269
## Effective degrees of freedom (residual: 2traceS - traceS'S): 465.0731
## Sigma (residual: 2traceS - traceS'S): 12.35905
## Effective number of parameters (model: traceS): 116.0071
## Effective degrees of freedom (model: traceS): 509.9929
## Sigma (model: traceS): 11.80222
## Sigma (ML): 10.65267
## AICc (GWR p. 61, eq 2.33; p. 96, eq. 4.21): 5026.882
## AIC (GWR p. 96, eq. 4.22): 4854.513
## Residual sum of squares: 71038.1
## Quasi-global R2: 0.7557128
```

The output from the GWR model reveals how the coefficients vary across the 626 Wards in our London Study region. You will see how the global coefficients are exactly the same as the coefficients in the earlier lm model. In this particular model (yours will look a little different if you have used different explanatory variables), if we take unauthorised absence from school, we can see that the coefficients range from a minimum value of -47.06 (1 unit change in unauthorised absence resulting in a drop in average GCSE score of -47.06) to +6.8 (1 unit change in unauthorised absence resulting in an increase in average GCSE score of +6.8). For half of the wards in the dataset, as unauthorised absence rises by 1 point, GCSE scores will decrease between -30.80 and -14.34 points (the inter-quartile range between the 1st Qu and the 3rd Qu).

You will notice that the R-Squared value (Quasi global R-squared) has improved - this is not uncommon for GWR models, but it doesn't necessarily mean they are definitely better than global models. The small number of cases under the kernel means that GW models have been criticised for lacking statistical robustness.

The best way to compare models is with the AIC (Akaike Information Criterion) or for smaller sample sizes the sample-size adjusted AICc, especially when you number of points is less than 40! Which it will be in GWR. The models must also be using the same data and be over the same study area!

AIC is calculated using the:

number of independent variables maximum likelihood estimate of the model (how well the model reproduces the data). The lower the value the better the better the model fit is, see scribbrif you want to know more here..although this is enough to get you through most situations.

Coefficient ranges can also be seen for the other variables and they suggest some interesting spatial patterning. To explore this we can plot the GWR coefficients for different variables. Firstly we can attach the coefficients to our original dataframe - this can be achieved simply as the coefficients for each ward appear in the same order in our spatial points dataframe as they do in the original dataframe. GWR 626 lm -47.06 1 GCSE -47.06 +6.8 1 GCSE +6.8 1 GCSE -30.80 -14.34 1 Qu 3 Qu

R R – GWR

AIC AICc 40 GWR

AIC

scribbr

GWR –

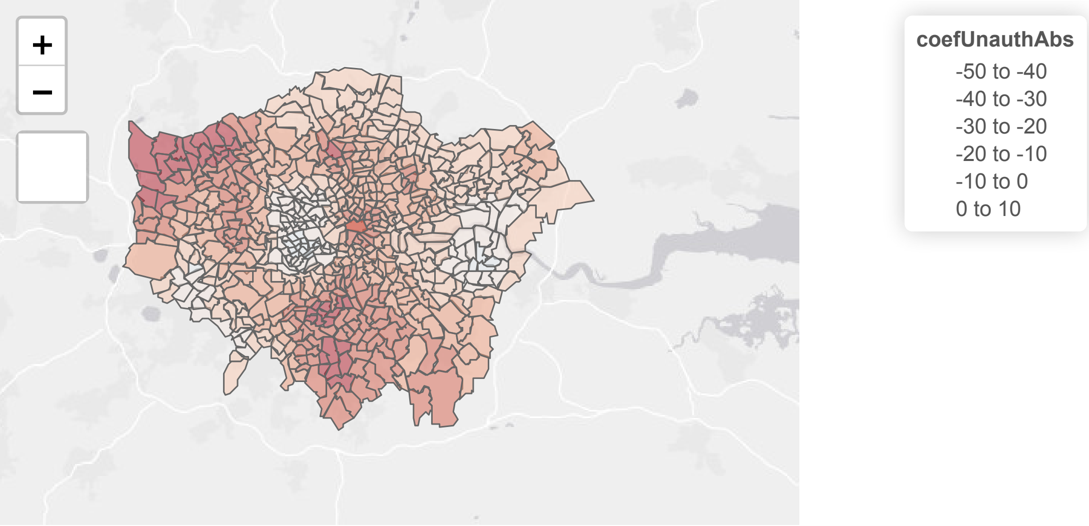
```r
results <- as.data.frame(gwr.model$SDF)
names(results)
```

```
##  [1] "sum.w"
##  [2] "X.Intercept."
##  [3] "unauthorised_absence_in_all_schools_percent_2013"
##  [4] "log.median_house_price_2014."
##  [5] "inner_outerInner"
##  [6] "rate_of_job_seekers_allowance_jsa_claimants_2015"
##  [7] "percent_with_level_4_qualifications_and_above_2011"
##  [8] "X.Intercept._se"
##  [9] "unauthorised_absence_in_all_schools_percent_2013_se"
## [10] "log.median_house_price_2014._se"
## [11] "inner_outerInner_se"
## [12] "rate_of_job_seekers_allowance_jsa_claimants_2015_se"
## [13] "percent_with_level_4_qualifications_and_above_2011_se"
## [14] "gwr.e"
## [15] "pred"
## [16] "pred.se"
## [17] "localR2"
## [18] "rate_of_job_seekers_allowance_jsa_claimants_2015_EDF"
## [19] "X.Intercept._se_EDF"
## [20] "unauthorised_absence_in_all_schools_percent_2013_se_EDF"
## [21] "log.median_house_price_2014._se_EDF"
## [22] "inner_outerInner_se_EDF"
## [23] "rate_of_job_seekers_allowance_jsa_claimants_2015_se_EDF"
## [24] "percent_with_level_4_qualifications_and_above_2011_se_EDF"
## [25] "pred.se.1"
## [26] "coord.x"
## [27] "coord.y"
```

```r
#attach coefficients to original SF


LonWardProfiles2 <- LonWardProfiles %>%
  mutate(coefUnauthAbs = results$unauthorised_absence_in_all_schools_percent_2013,
         coefHousePrice = results$log.median_house_price_2014.,
         coefJSA = rate_of_job_seekers_allowance_jsa_claimants_2015,
         coefLev4Qual = percent_with_level_4_qualifications_and_above_2011)
```

```
tm_shape(LonWardProfiles2) +
  tm_polygons(col = "coefUnauthAbs",
              palette = "RdBu",
              alpha = 0.5)
```

```
## Variable(s) "coefUnauthAbs" contains positive and negative values, so midpoint is set to 0. Set midp
```



Leaflet | Tiles © Esri — Esri, DeLorme, NAVTEQ

Now how would you plot the House price coeffeicent, Job seekers allowance and level 4 qualification coefficient? Taking the first plot, which is for the unauthorised absence coefficients, we can see that for the majority of boroughs in London, there is the negative relationship we would expect - i.e. as unauthorised absence goes up, exam scores go down. For three boroughs (Westminster, Kensington & Chelsea and Hammersmith and Fulham, as well as an area near Bexleyheath in South East London - some of the richest in London), however, the relationship is positive - as unauthorised school absence increases, so does average GCSE score. This is a very interesting pattern and counterintuitive pattern, but may partly be explained the multiple homes owned by many living in these boroughs (students living in different parts of the country and indeed the world for significant periods of the year), foreign holidays and the over representation of private schooling of those living in these areas. If this is not the case and unauthorised absence from school is reflecting the unauthorised absence of poorer students attending local, inner city schools, then high GCSE grades may also reflect the achievements of those who are sent away to expensive fee-paying schools elsewhere in the country and who return to their parental homes later in the year. Either way, these factors could explain these results. Of course, these results may not be statistically significant across the whole of London. Roughly speaking, if a coefficient estimate is more than 2 standard errors away from zero, then it is “statistically significant”. —— - - GCSE GCSE 2 “ ” Remember from earlier the standard error is “the average amount the coefficient varies from the average value of the dependent variable (its standard deviation). So, for a 1% increase in unauthorised absence from school, while the model says we might expect GSCE scores to drop by -41.2 points, this might vary, on average, by about 1.9 points. As a rule of thumb, we are looking for a lower value in the standard error relative to the size of the coefficient.” “ 1% GSCE -41.2 1.9 ” To calculate standard errors, for each variable you can use a formula similar to this:

```
#run the significance test
sigTest = abs(gwr.model$SDF$"log(median_house_price_2014)")-2 * gwr.model$SDF$"log(median_house_price_20


#store significance results
LonWardProfiles2 <- LonWardProfiles2 %>%
  mutate(GWRUnauthSig = sigTest)
```

If this is greater than zero (i.e. the estimate is more than two standard errors away from zero), it is very unlikely that the true value is zero, i.e. it is statistically significant (at nearly the 95% confidence level) 95% “ ” ” ”

This is a combination of two ideas: 95% of data in a normal distribution is within two standard deviations of the mean Statistical significance in a regression is normally measured at the 95% level. If the p-value is less than 5% — 0.05 — then there’s a 95% probability that a coefficient as large as you are observing didn’t occur by chance 95% 95% p 5% 0.05 95%

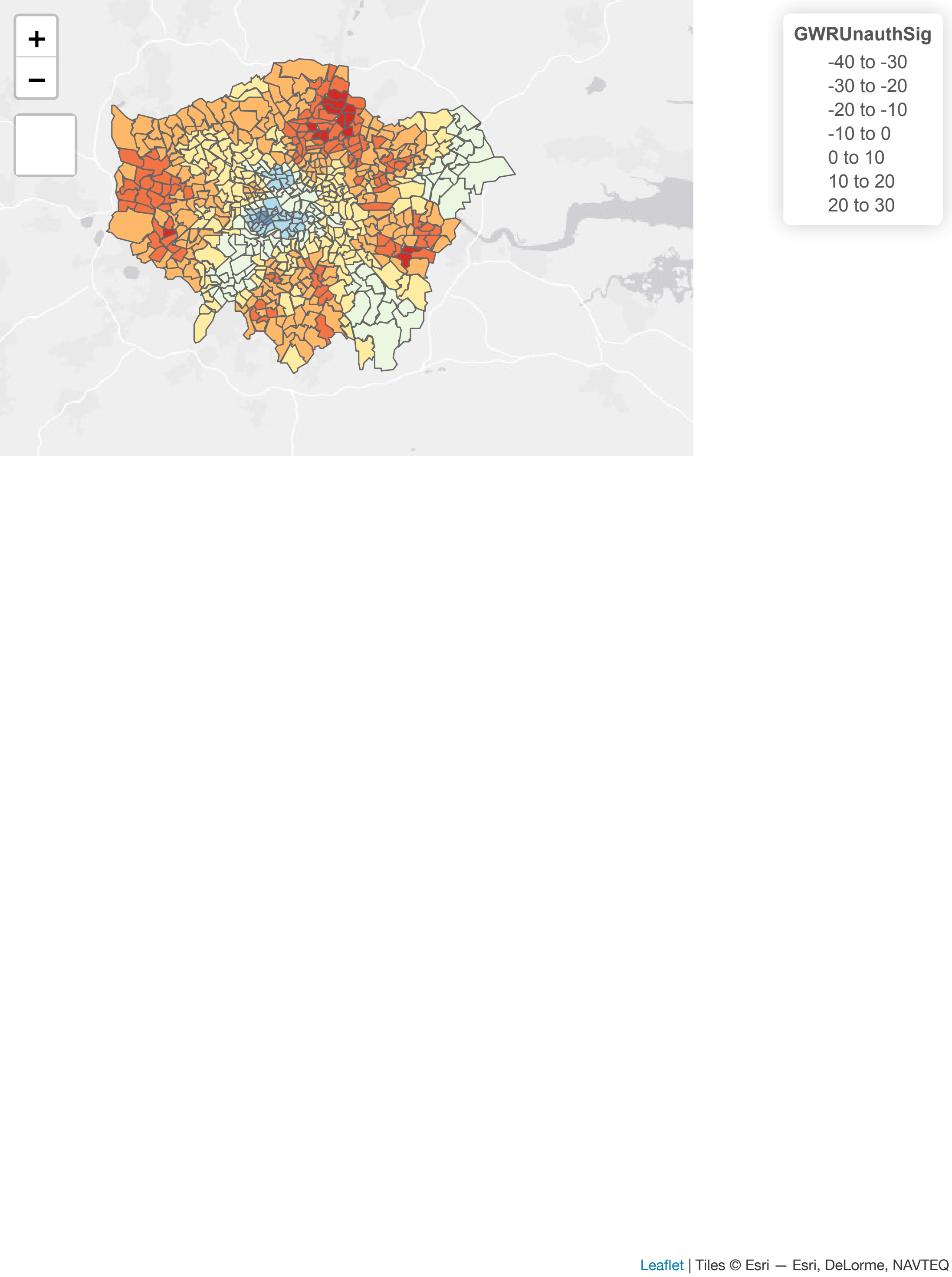
Combining these two means if… the coefficient is large in relation to its standard error and the p-value tells you if that largeness statistically acceptable - at the 95% level (less than 5% — 0.05) p 95% 5% - 0.05

You can be confident that in your sample, nearly all of the time, that is a real and reliable coefficient value.

You should now calculate these for each variable in your GWR model and See if you can plot them on a map, for example: GWR

```
tm_shape(LonWardProfiles2) +
  tm_polygons(col = "GWRUnauthSig",
              palette = "RdYlBu")
```

```
## Variable(s) "GWRUnauthSig" contains positive and negative values, so midpoint is set to 0. Set midpo
```



From the results of your GWR exercise, what are you able to conclude about the geographical variation in your explanatory variables when predicting your dependent variable? GWR
1. GWR 2. spatial non-

stationarity 3. GWR 4.
5. GWR GWR