Geographically Weighted

Regression

1. Under the Datasets for Module 5 there is a zipped file called “camdenoa11”. Camden is a borough (area) in the north-west of the city of London. The file “practical_data.csv” contains Census data for this borough. Unzip the contents of the zipped file and store them in your current working directory.

2. Install the new package spgwr, with any dependencies

3. Load the following libraries: sf, sp, tmap spdep and spgwr

4. Set your working directory and read in the census data

```
Census.Data <-read.csv("practical_data.csv")
```

This is a dataframe, so you can examine content using the str function

```
> str(Census.Data)
'data.frame':    749 obs. of  5 variables:
 $ OA           : chr  "E00004120" "E00004121" "E00004122" "E00004123" ...
 $ White_British: num  42.4 47.2 40.7 49.7 51.1 ...
 $ Low_Occupancy: num  6.294 5.932 2.913 0.926 2 ...
 $ Unemployed   : num  1.89 2.69 1.21 2.8 3.82 ...
 $ Qualification: num  73.6 69.9 67.6 60.8 66 ...
```

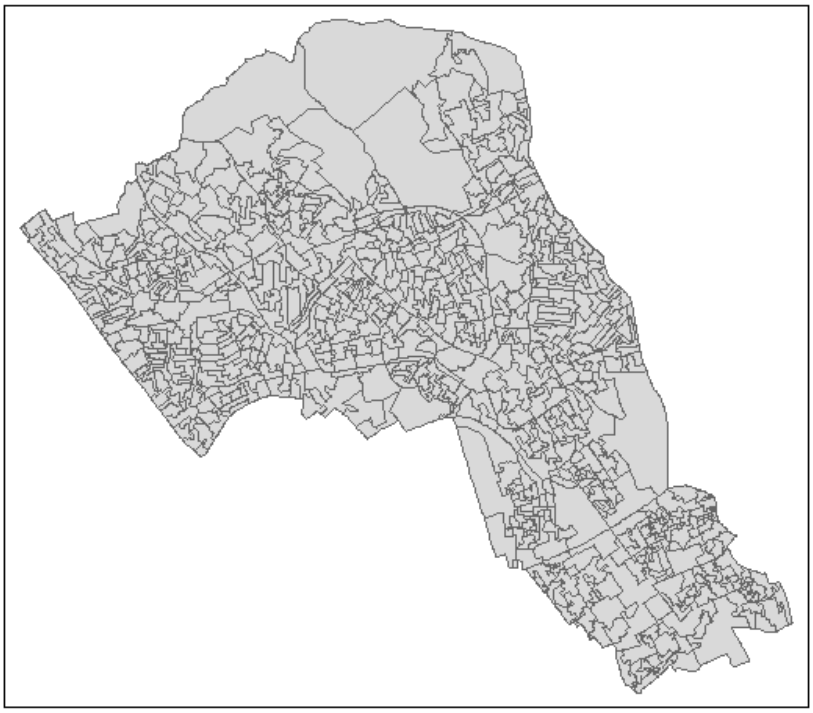
OA stands for Output Areas in British Census maps. Unemployed is percentage unemployed, Qualification is the percentage of people with specific work qualifications, and White_British is a weird ethnicity variable for the UK (there are also White Irish and White Other categories!). Try not to overthink low occupancy-- it reflects government concern around "underutilised" public housing supply (often elderly family in houses where kids have grown and moved out).

```
> Output.Areas <- st_read("Camden_oa11.shp")
```

And you can have a quick look at the OA boundaries within Camden.

```
> qtm(Output.Areas)
```

4. Read in the shape file for the Camden OAs.



5. Now let's merge the Census data with the Camden shape file

```
> OA.Census <- merge(Output.Areas, Census.Data, by.x="OA11CD", by.y="OA")
```

6. And let's run a non-spatial regression model where Qualification is the dependent variable and where Unemployed and White_British are independent variables.

```
> model <- lm(Qualification ~ Unemployed + White_British, data = Census.Data)
```

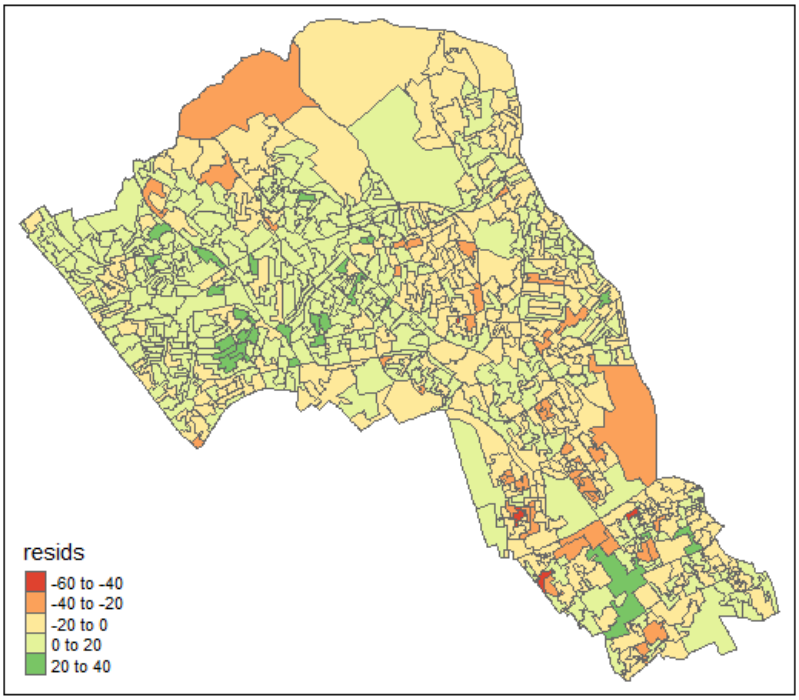
Have a look at the output

```
> summary(model)
```

```
Call:
lm(formula = Qualification ~ Unemployed + White_British, data = Census.Data)

Residuals:
    Min      1Q  Median      3Q     Max
-50.311  -8.014   1.006   8.958  38.046

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   47.86697    2.33574   20.49   <2e-16 ***
Unemployed    -3.29459    0.19027  -17.32   <2e-16 ***
White_British  0.41092    0.04032   10.19   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.69 on 746 degrees of freedom
Multiple R-squared:  0.4645,  Adjusted R-squared:  0.463
F-statistic: 323.5 on 2 and 746 DF,  p-value: < 2.2e-16
```

7. You know by now this model was estimated using OLS, and you should be able to interpret the output. Let's capture and plot the residuals from this regression.

```
> resids <- residuals(model)
> map.resids <- cbind(OA.Census, resids)
> qtm(map.resids, fill="resids")
```



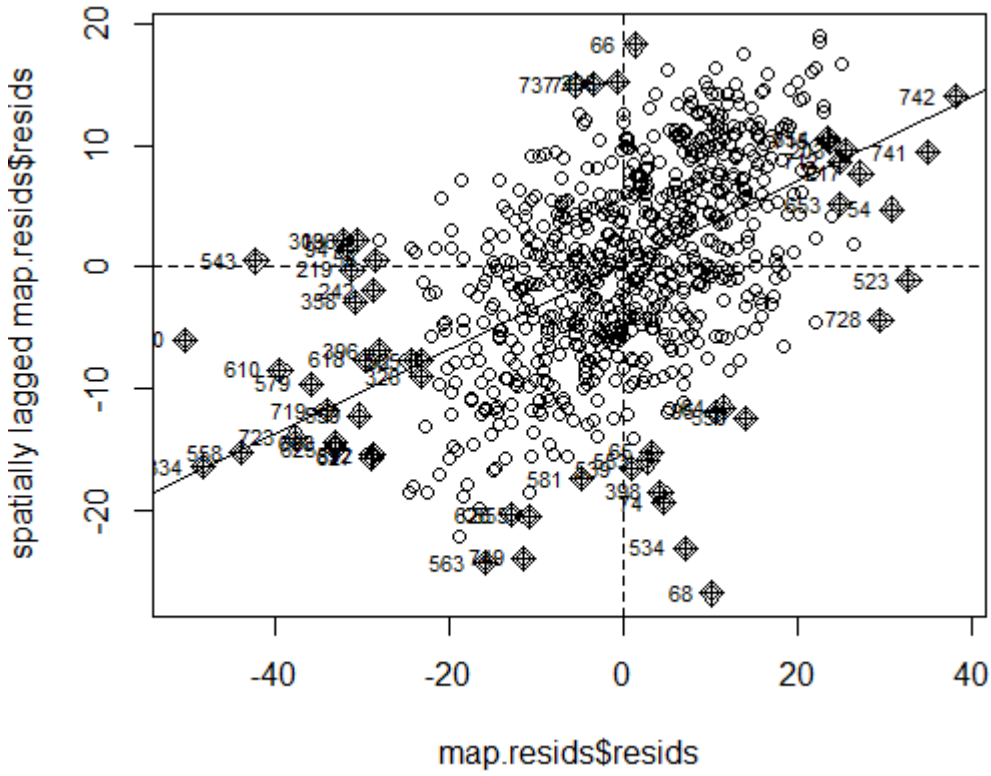
8. It's unclear if we have any spatial patterning within the residual map. Let's check by loading the spdep library, generating some weights and having a look.

```
> library(spdep)
> camden_q <- poly2nb(map.resids, queen=T)
> neighbors <-nb2listw(camden_q, zero.policy=T)
> moran.test(map.resids$resids, neighbors, zero.policy=T)
```

```
        Moran I test under randomisation

data:  map.resids$resids
weights: neighbors

Moran I statistic standard deviate = 15.773, p-value < 2.2e-16
alternative hypothesis: greater
sample estimates:
Moran I statistic       Expectation          Variance
     0.3463604896     -0.0013368984      0.0004859376
```

Well that is quite conclusive. Let's have a look at the Moran plot as well

```
> moran.plot(map.resids$resids, neighbors, zero.policy=T)
```



9. There is definitely some spatial autocorrelation in the residuals. At this point we could run a spatial lag or spatial error model. We could also try something a little different. Let's apply Geographically Weighted Regression (GWR) to have a look at how aspects of our regression model might be varying over space. From class, you know that GWR builds a series of local regressions, not from all the observations in a study region (the OAs in this case), but rather by taking local subsets of observations (points or polygons) and running the regression across these. To do this GWR uses a kernel function with a chosen bandwidth to vary the size of the local subsets used to fit the regression model over a sample of our study region's data points.

10. I am not going to worry too much about the type of kernel and the selection of the appropriate bandwidth. You could read more on GWR for this. I will use an adaptive kernel here that allows the kernel size to vary a little to make sure that estimation of the regression model over parts of our study region have roughly similar numbers of observations.

Now an immediate problem is that for GWR to run, it needs coordinate data from the shape file. Let's get this information by coercing our OA.Census object into a Spatial object

```
> OA.Census2<-as(OA.Census, "Spatial")
```

12. And now we can generate the GWR bandwidth

```
> GWRbandwidth <- gwr.sel(Qualification ~ Unemployed + White_British, data = OA.Census2, adapt = T)
```

```
Adaptive q: 0.381966 CV score: 101420.8
Adaptive q: 0.618034 CV score: 109723.2
Adaptive q: 0.236068 CV score: 96876.06
Adaptive q: 0.145898 CV score: 94192.41
Adaptive q: 0.09016994 CV score: 91099.75
Adaptive q: 0.05572809 CV score: 88242.89
Adaptive q: 0.03444185 CV score: 85633.41
Adaptive q: 0.02128624 CV score: 83790.04
Adaptive q: 0.01315562 CV score: 83096.03
Adaptive q: 0.008130619 CV score: 84177.45
Adaptive q: 0.01535288 CV score: 83014.34
Adaptive q: 0.01515437 CV score: 82957.49
Adaptive q: 0.01436908 CV score: 82857.74
Adaptive q: 0.01440977 CV score: 82852.4
Adaptive q: 0.01457859 CV score: 82833.25
Adaptive q: 0.01479852 CV score: 82855.45
Adaptive q: 0.01461928 CV score: 82829.32
Adaptive q: 0.01468774 CV score: 82823.82
Adaptive q: 0.01473006 CV score: 82835.89
Adaptive q: 0.01468774 CV score: 82823.82
```

13. Now armed with the bandwidth data you can run the GWR model.

```
> gwr.model = gwr(Qualification ~ Unemployed + White_British, data = OA.Census2, adapt=GWRbandwidth, hatmatrix=TRUE, se.fit=TRUE)
```

You don't ask for a summary of the GWR output, instead just specify the object

```
> gwr.model
Call:
gwr(formula = Qualification ~ Unemployed + White_British, data = OA.Census2,
    adapt = GWRbandwidth, hatmatrix = TRUE, se.fit = TRUE)
Kernel function: gwr.Gauss
Adaptive quantile: 0.01468774 (about 11 of 749 data points)
Summary of GWR coefficient estimates at data points:
                  Min.  1st Qu.   Median  3rd Qu.     Max.  Global
X.Intercept.  11.08183 34.43427 45.76862 59.75372 85.01866 47.8670
Unemployed    -5.45291 -3.28308 -2.55398 -1.79413  0.77019 -3.2946
White_British -0.28046  0.19955  0.37788  0.53216  0.94678  0.4109
Number of data points: 749
Effective number of parameters (residual: 2traceS - traceS'S): 132.6449
Effective degrees of freedom (residual: 2traceS - traceS'S): 616.3551
Sigma (residual: 2traceS - traceS'S): 9.903539
Effective number of parameters (model: traceS): 94.44661
Effective degrees of freedom (model: traceS): 654.5534
```

```
Sigma (model: traceS): 9.610221
Sigma (ML): 8.983902
AICc (GWR p. 61, eq 2.33; p. 96, eq. 4.21): 5633.438
AIC (GWR p. 96, eq. 4.22): 5508.777
Residual sum of squares: 60452.16
Quasi-global R2: 0.7303206
```

You know how to interpret most of this material.

14. Create a results dataframe

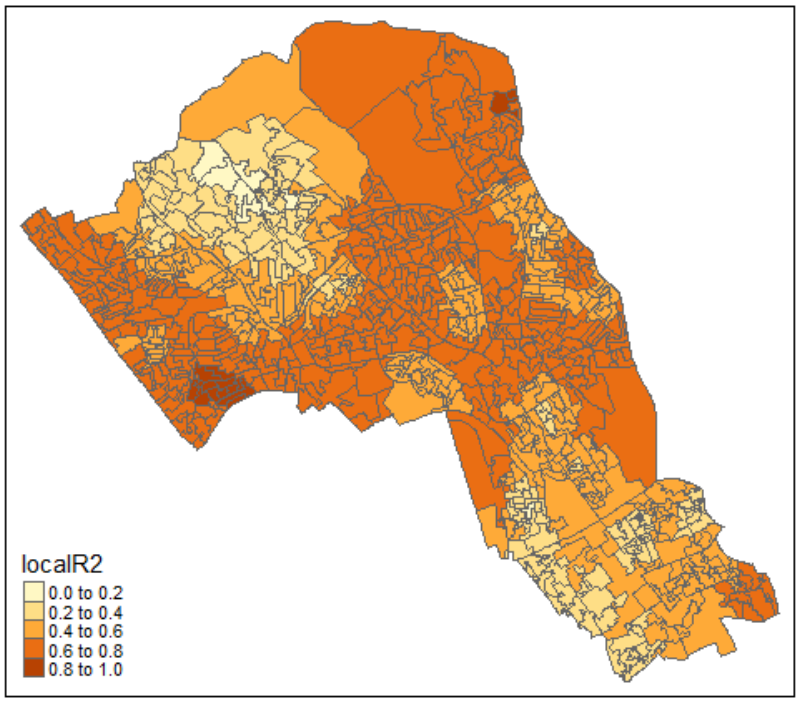
```
> results <-as.data.frame(gwr.model$SDF)
> names(results)
[1] "sum.w"                "X.Intercept."         "Unemployed"
 [4] "White_British"        "X.Intercept._se"      "Unemployed_se"
 [7] "White_British_se"     "gwr.e"                "pred"
[10] "pred.se"              "localR2"              "X.Intercept._se_EDF"
[13] "Unemployed_se_EDF"    "White_British_se_EDF" "pred.se.1"
```

15. Now you can plot selected outputs from the GWR model after binding the results and the map data.

```
> gwr.map <- cbind(OA.Census2, as.matrix(results))
```

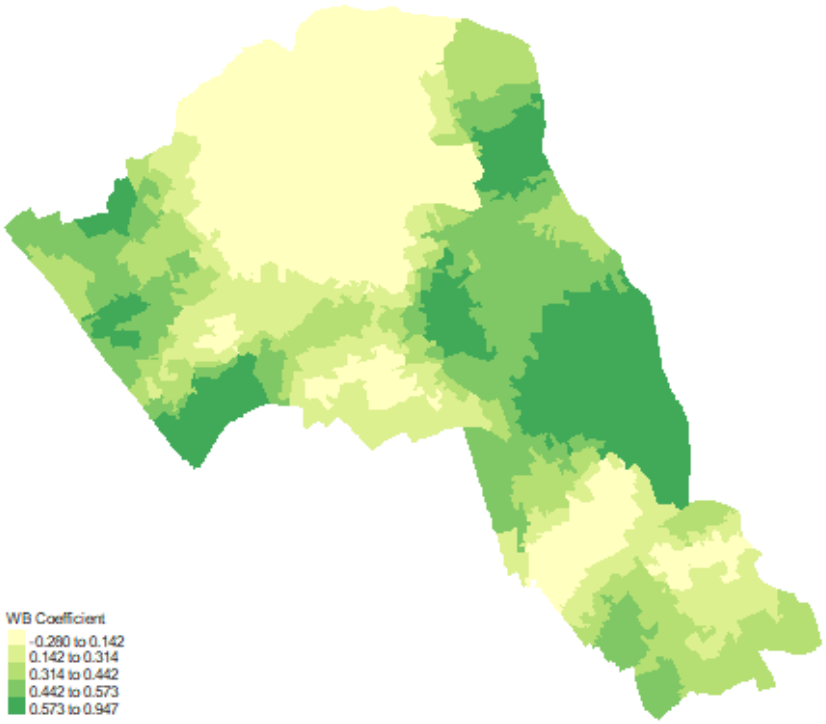
Then let's plot the local variation in R^2 values….this is showing how well our model fits the data as we look at different parts of the study region.

```
> qtm(gwr.map, fill = "localR2")
```



We might want to look at how regression coefficients in our model vary across the study region.

```
> map2 <- tm_shape(gwr.map) + tm_fill("White_British.1", n = 5, style = "quantile", title = "WB Coefficient") + tm_layout(frame = FALSE, legend.text.size = 0.5, legend.title.size = 0.6)
> map2
```

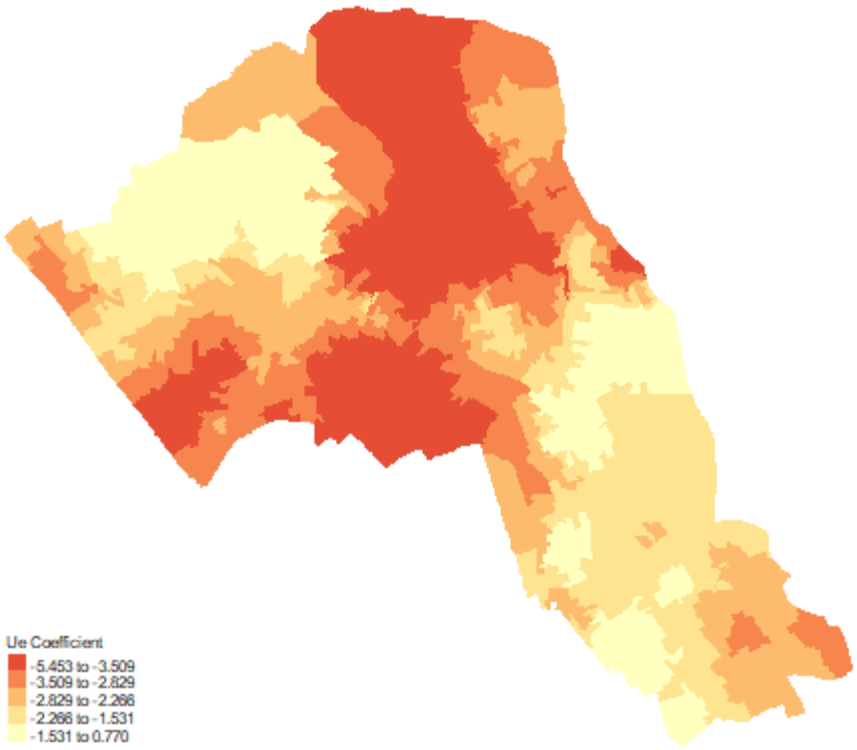


So you can see that the impact of ethnicity on qualification varies across the study region.

What about the effect of unemployment?

```
> map3 <- tm_shape(gwr.map) + tm_fill("Unemployed.1", n = 5, style = "quantile", title = "Ue Coefficient") + tm_layout(frame = FALSE, legend.text.size = 0.5, legend.title.size = 0.6)
> map3
```

And this map shows areas of the study region where unemployment has a higher/lower influence on the variance in qualification.



Here you see how unemployment impacts qualification in different ways across the different parts of the study region.