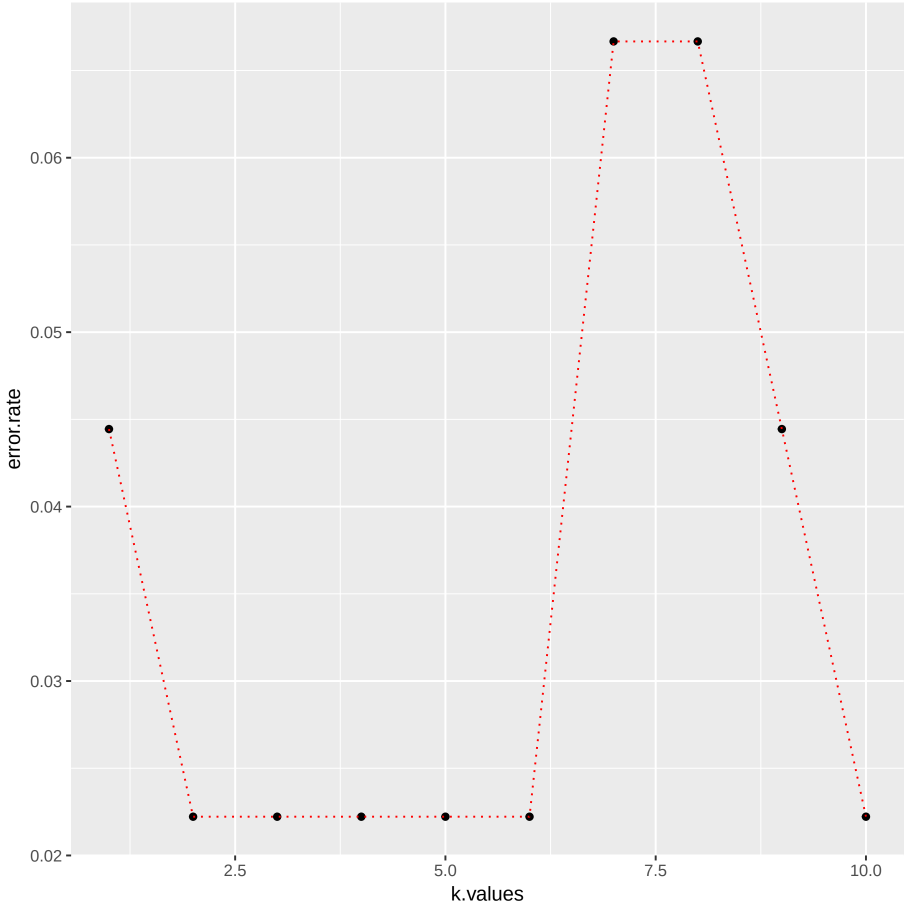
1 Writing code chunks in RStudio

```
# Geographically Weighted Regression

library(ggthemes)

## Warning: package 'ggthemes' was built under R version 3.4.3

library(dplyr)

## Warning: Installed Rcpp (0.12.14) different from Rcpp used to build
dplyr (0.12.11).
## Please reinstall dplyr to avoid random crashes or undefined behavior.
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ISLR)

## Warning: package 'ISLR' was built under R version 3.4.3

print(head(iris))

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa

# Scale the data
stand.features <- scale(iris[1:4])
print(var(stand.features[,1]))

## [1] 1

final.data <- cbind(stand.features, iris[5])

#Train test splits
set.seed(101)
library(caTools)
sample <- sample.split(final.data$Species, SplitRatio = 0.7)
```

```r
train <- subset(final.data, sample == T)
test <- subset(final.data, sample == F)

## KNN
library(class)
predicted.species <- knn(train[1:4], test[1:4], train$Species, k=1)
print(mean(test$Species != predicted.species))
```

```
## [1] 0.04444444
```

```r
# choose the K value

predicted.species <- NULL
error.rate <- NULL

for (i in 1:10) {
  set.seed(101)
  predicted.species <- knn(train[1:4],test[1:4], train$Species,k = i)
  error.rate[i] <- mean(test$Species != predicted.species)

}
 ## Plot this for the elbow method

library(ggplot2)
k.values <- 1:10
error.df <- data.frame(error.rate, k.values)
pl <- ggplot(error.df, aes(x=k.values, y=error.rate)) + geom_point()
pl <- pl + geom_line(lty = 'dotted', color = 'red')
print(pl)
```



```
library("pacman")

## Warning: package 'pacman' was built under R version 3.4.2

pacman::p_load(sp, rgdal, rgeos,tmap,raster)
install.packages("Rserve")

## Installing package into 'C:/Users/Arch/Pictures/Documents/R/win-library/3.4'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN without
setting a mirror

install.packages("plyr")

## Installing package into 'C:/Users/Arch/Pictures/Documents/R/win-library/3.4'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN without
setting a mirror
```

```
pacman::p_load("spgwr")
install.packages("ggthemes, ISLR")
```

```
## Installing package into 'C:/Users/Arch/Pictures/Documents/R/win-library/3.4'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN without
setting a mirror
```

```
install.packages("ISLR")
```

```
## Installing package into 'C:/Users/Arch/Pictures/Documents/R/win-library/3.4'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN without
setting a mirror
```

```
install.packages("ggthemes")
```

```
## Installing package into 'C:/Users/Arch/Pictures/Documents/R/win-library/3.4'
## (as 'lib' is unspecified)
## Error in contrib.url(repos, "source"): trying to use CRAN without
setting a mirror
```

```
URL <- "http://www.unified-democracy-scores.org/files/20140312/z/uds_summary.csv.gz"
temp <- tempfile()
download.file(URL, temp)
```

```
## Warning in download.file(URL, temp): InternetOpenUrl failed: 'The
server name or address could not be resolved'
## Error in download.file(URL, temp): cannot open URL 'http://www.unified-democracy-scores.
```

```
UDSData <- read.csv(gzfile(temp, "uds_summary.csv"))
```

```
## Warning in gzfile(temp, "uds_summary.csv"): cannot open compressed
file 'C:\Users\Arch\AppData\Local\Temp\Rtmpm2buDC\filecbd4ddc6a8',
probable reason 'No such file or directory'
## Error in gzfile(temp, "uds_summary.csv"): cannot open the connection
```

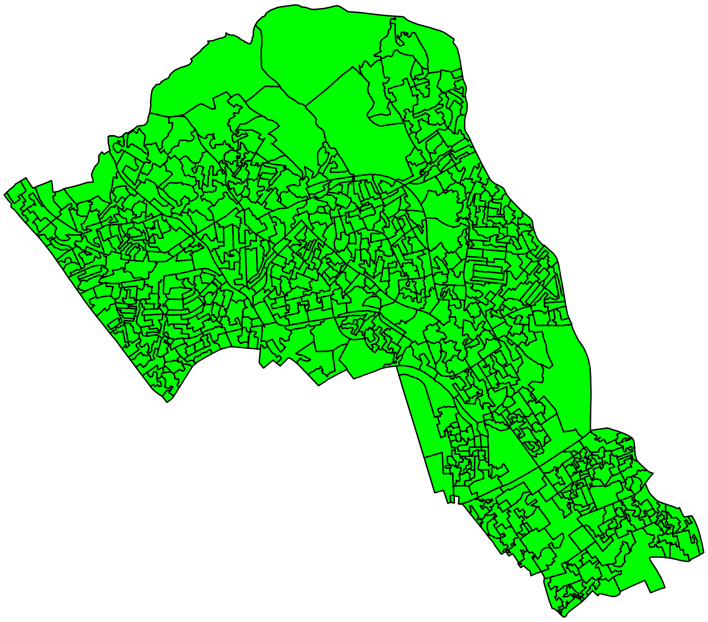
```
UDSData
```

```
## Error in eval(expr, envir, enclos): object 'UDSData' not found
```

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

```
##          OA White_British Low_Occupancy Unemployed Qualification
## 1 E00004120      42.35669     6.2937063   1.893939      73.62637
## 2 E00004121      47.20000     5.9322034   2.688172      69.90291
## 3 E00004122      40.67797     2.9126214   1.212121      67.58242
## 4 E00004123      49.66216     0.9259259   2.803738      60.77586
## 5 E00004124      51.13636     2.0000000   3.816794      65.98639
## 6 E00004125      41.41791     3.9325843   3.846154      74.20635
```

```
setwd("C:\\Users\\Arch\\Creative Cloud Files\\Desktop\\GeoR\\Ex5_Data")

getwd()

## [1] "C:/Users/Arch/Creative Cloud Files/Desktop/GeoR/Ex5_Data"

pacman::p_load(sp, rgdal,rgeos,tmap,raster)
Output.Areas <- readOGR(".", "Camden_oa11")

## OGR data source with driver: ESRI Shapefile
## Source: ".", layer: "Camden_oa11"
## with 749 features
## It has 1 fields

plot(Output.Areas, col = "green")
```



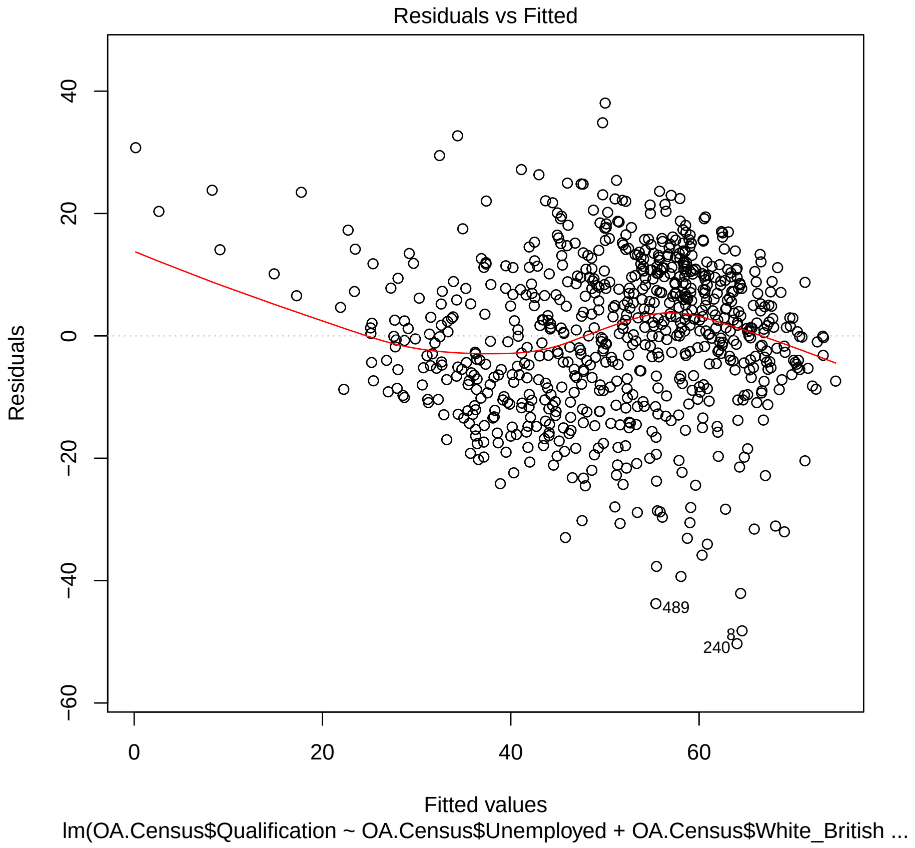
```
OA.Census <- merge(Output.Areas, Census.Data, by.x = "OA11CD", by.y = "OA")
head(OA.Census)
```

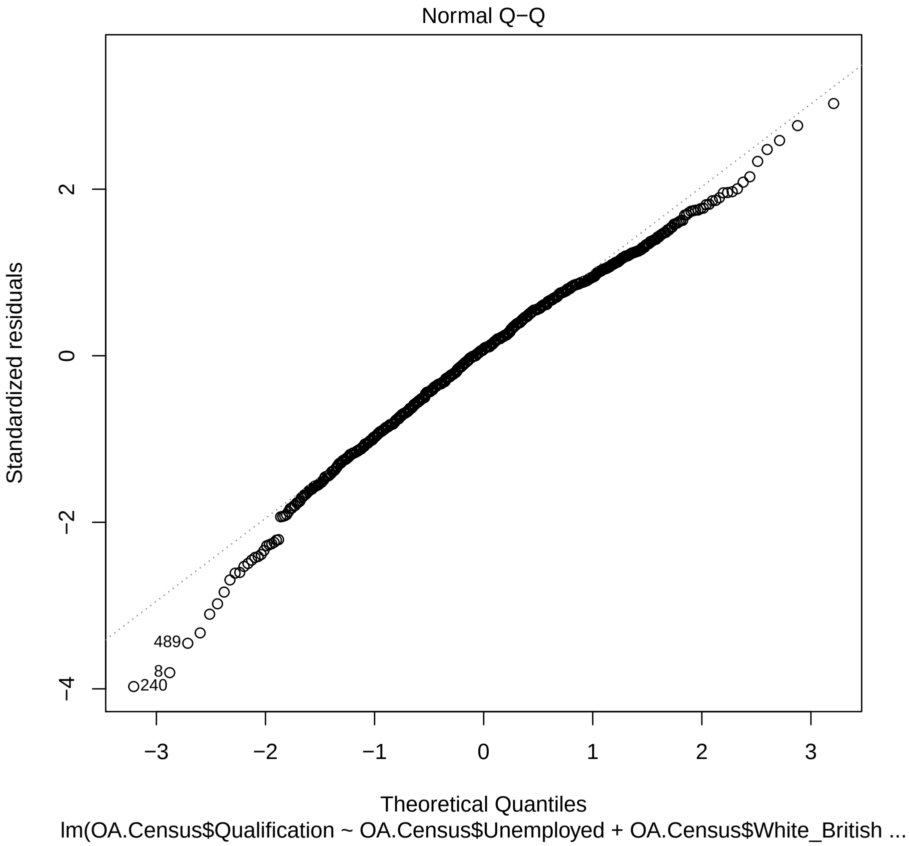
```
##        OA11CD White_British Low_Occupancy Unemployed Qualification
## 397 E00004527      48.29060     12.745098   7.511737      35.80786
## 395 E00004525      40.94488     16.806723   5.990783      42.41071
## 392 E00004522      44.16244      8.547009   2.116402      56.47668
## 160 E00004287      31.86813     12.612613   3.286385      67.85714
## 80  E00004206      56.45161     19.685039   7.983193      31.74603
## 74  E00004200      39.91935     11.764706   3.524229      57.20339
```

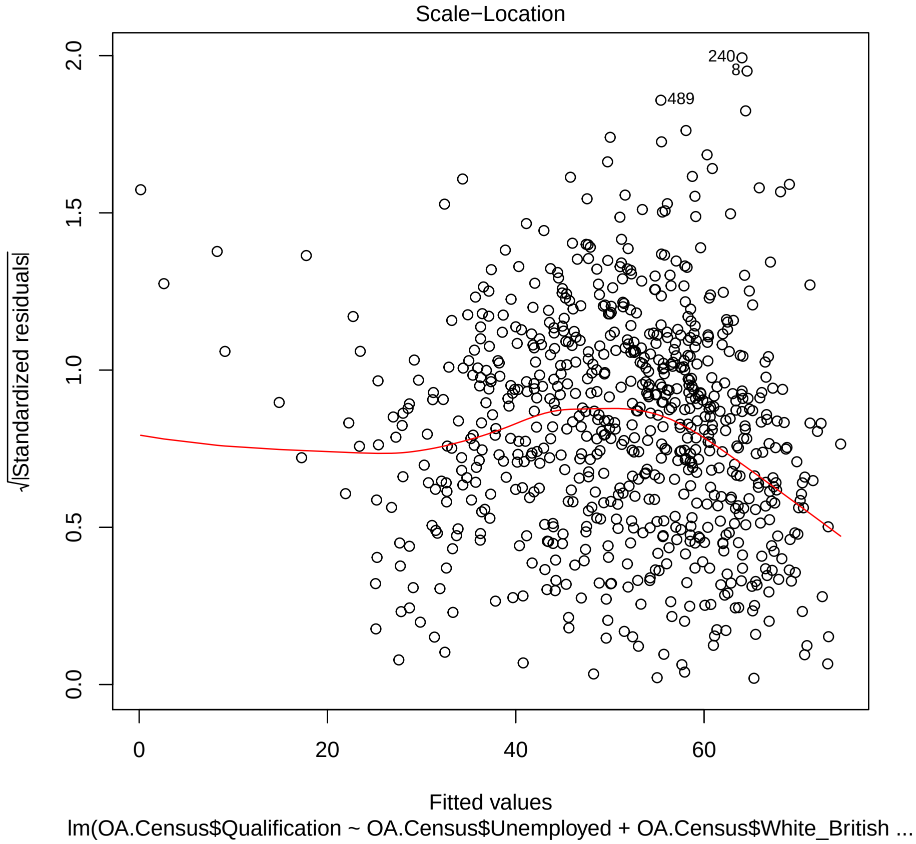
```
# Run a linear regression model on the OA.Census data
 model <- lm(OA.Census$Qualification ~ OA.Census$Unemployed + OA.Census$White_British)
 summary(model)
```

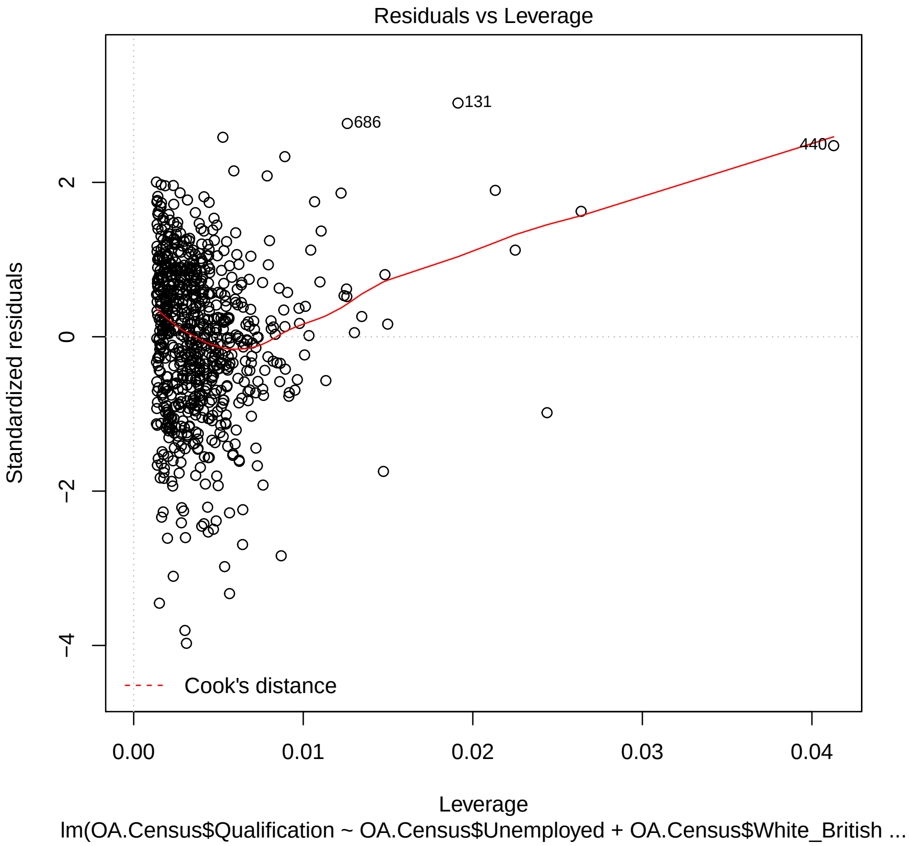
```
## 
## Call:
## lm(formula = OA.Census$Qualification ~ OA.Census$Unemployed + 
##     OA.Census$White_British)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -50.311  -8.014   1.006   8.958  38.046 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             47.86697    2.33574   20.49   <2e-16 ***
## OA.Census$Unemployed    -3.29459    0.19027  -17.32   <2e-16 ***
## OA.Census$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
```

```
 # Plot scatter plots
 plot(model)
```

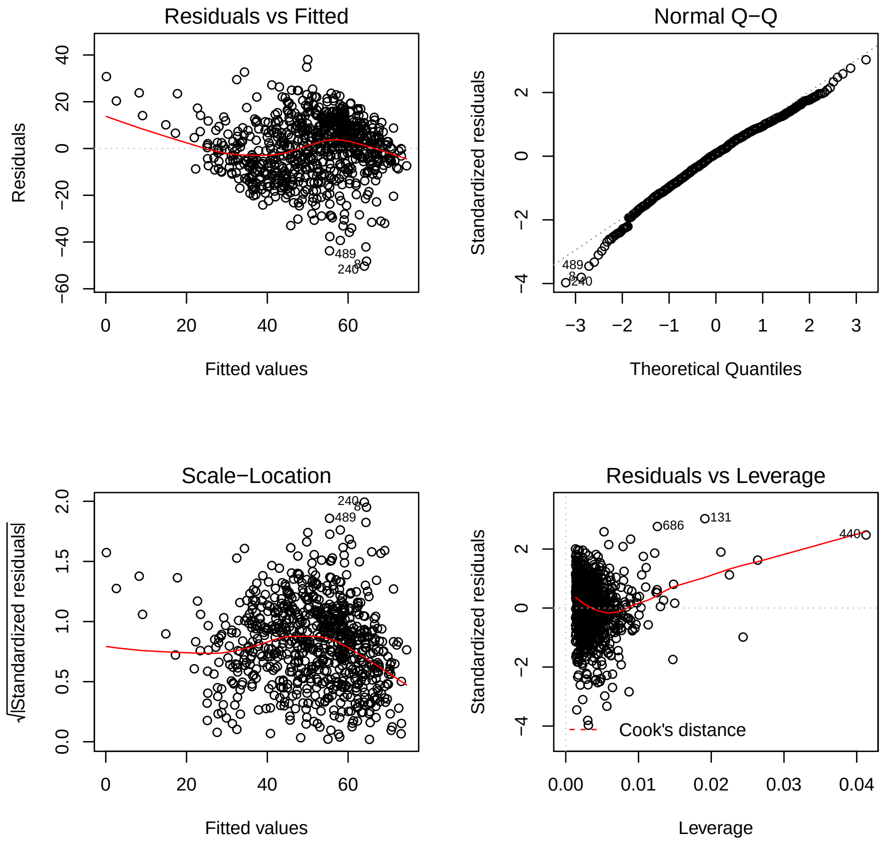

Residuals vs Fitted
40
20
0
−20
−40
−60
Residuals
489
8
240
0
20
40
60
Fitted values
lm(OA.Census$Qualification ~ OA.Census$Unemployed + OA.Census$White_British ...


Normal Q−Q
Standardized residuals
2
0
−2
−4
−3
−2
−1
0
1
2
3
489
8
240
Theoretical Quantiles
lm(OA.Census$Qualification ~ OA.Census$Unemployed + OA.Census$White_British ...


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
20
40
60
240
8
489
Fitted values
lm(OA.Census$Qualification ~ OA.Census$Unemployed + OA.Census$White_British ...



```r
# Plot them in a 2x2 frame
par(mfrow = c(2,2))
plot(model)
```



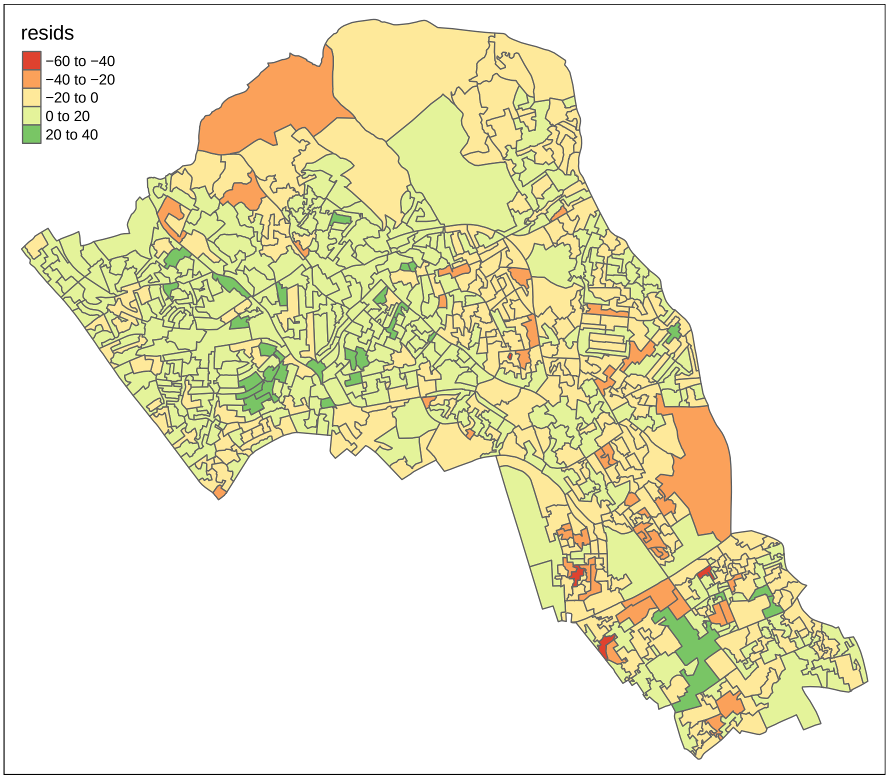
```
## Mapping the residuals
resids <- residuals(model)

map.resids <- cbind(OA.Census, resids)

# Rename the column header from resids file
head(map.resids)
```

```
##         OA11CD White_British Low_Occupancy Unemployed Qualification
## 397 E00004527      48.29060     12.745098   7.511737      35.80786
## 395 E00004525      40.94488     16.806723   5.990783      42.41071
## 392 E00004522      44.16244      8.547009   2.116402      56.47668
## 160 E00004287      31.86813     12.612613   3.286385      67.85714
## 80  E00004206      56.45161     19.685039   7.983193      31.74603
## 74  E00004200      39.91935     11.764706   3.524229      57.20339
##     structure.c..7.15455410912352...2.54412234089367...2.56480926103033..
## 397                                                              -7.154554
## 395                                                              -2.544122
## 392                                                              -2.564809
## 160                                                              17.722233
## 80                                                              -13.016647
## 74                                                                4.543675
```

```r
names(map.resids)[6] <- "resids"
head(map.resids)
```

```
##        OA11CD White_British Low_Occupancy Unemployed Qualification
## 397 E00004527      48.29060     12.745098   7.511737      35.80786
## 395 E00004525      40.94488     16.806723   5.990783      42.41071
## 392 E00004522      44.16244      8.547009   2.116402      56.47668
## 160 E00004287      31.86813     12.612613   3.286385      67.85714
## 80  E00004206      56.45161     19.685039   7.983193      31.74603
## 74  E00004200      39.91935     11.764706   3.524229      57.20339
##         resids
## 397  -7.154554
## 395  -2.544122
## 392  -2.564809
## 160  17.722233
## 80  -13.016647
## 74    4.543675
```

```r
# map the residuals using the quickmap function from tmap
qtm(map.resids, fill = "resids")
```



```
## Running a Geographically Weighted Regression
pacman::p_load(spgwr)

# Calculate kernel bandwidth
GWRbandwidth <- gwr.sel(OA.Census$Qualification ~ OA.Census$Unemployed + OA.Census$White_Br
```

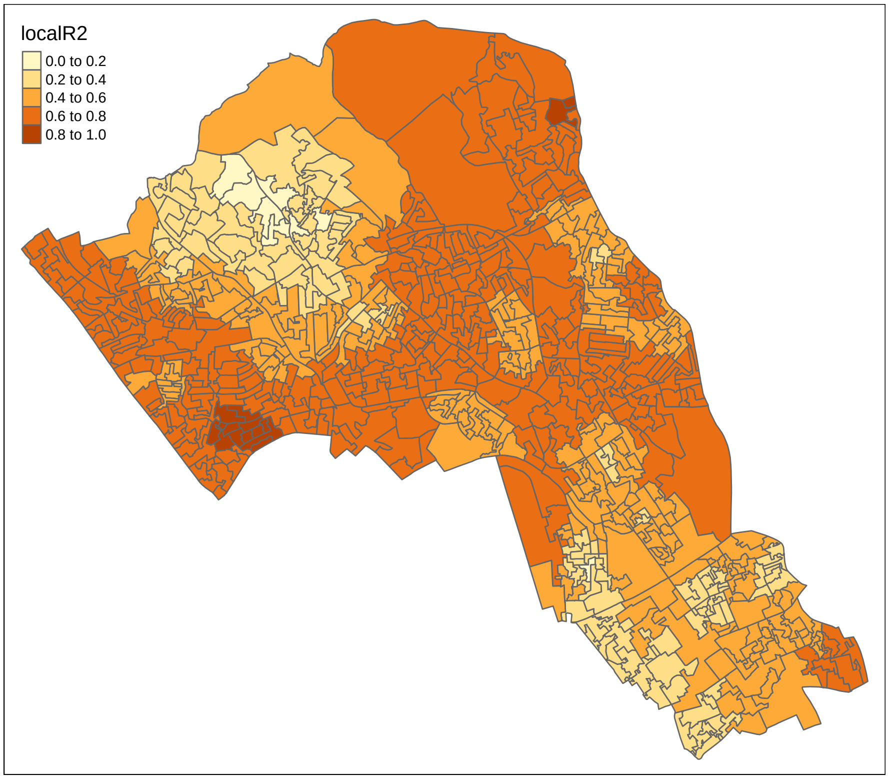
```
## Adaptive q: 0.381966 CV score: 101420.8
## Adaptive q: 0.618034 CV score: 109723.2
## Adaptive q: 0.236068 CV score: 96876.06
## Adaptive q: 0.145898 CV score: 94192.41
## Adaptive q: 0.09016994 CV score: 91099.75
## Adaptive q: 0.05572809 CV score: 88242.89
## Adaptive q: 0.03444185 CV score: 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
```

```
# Run the model and view the results
gwr.model <- gwr(OA.Census$Qualification ~ OA.Census$Unemployed + OA.Census$White_British,

# Print the model results
gwr.model
```

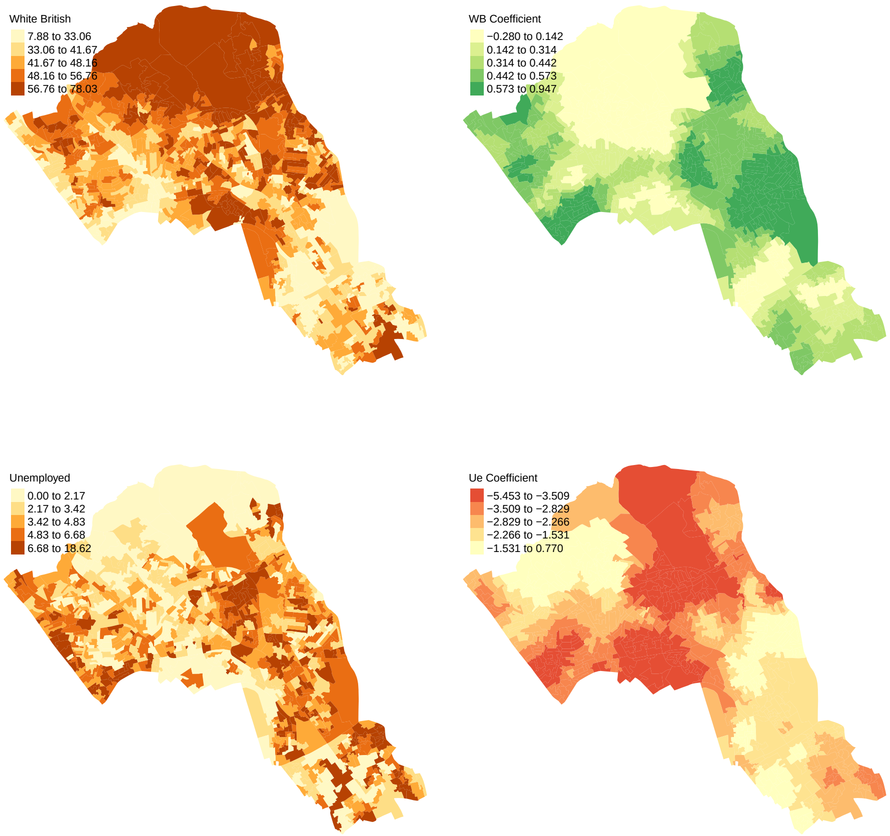
```
## Call:
## gwr(formula = OA.Census$Qualification ~ OA.Census$Unemployed +
##     OA.Census$White_British, data = OA.Census, adapt = GWRbandwidth,
##     hatmatrix = TRUE, se.fit = TRUE)
## Kernel function: gwr.Gauss
## Adaptive quantile: 0.01468774 (about 11 of 749 data points)
## Summary of GWR coefficient estimates at data points:
##                               Min.  1st Qu.   Median  3rd Qu.     Max.
## X.Intercept.              11.08183 34.43427 45.76862 59.75372 85.01866
## OA.Census.Unemployed      -5.45291 -3.28308 -2.55398 -1.79413  0.77019
## OA.Census.White_British -0.28046  0.19955  0.37788  0.53216  0.94678
##                           Global
## X.Intercept.              47.8670
## OA.Census.Unemployed      -3.2946
## OA.Census.White_British  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
```

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

```
names(results)

##  [1] "sum.w"                          "X.Intercept."
##  [3] "OA.Census.Unemployed"           "OA.Census.White_British"
##  [5] "X.Intercept._se"                "OA.Census.Unemployed_se"
##  [7] "OA.Census.White_British_se"     "gwr.e"
##  [9] "pred"                           "pred.se"
## [11] "localR2"                        "X.Intercept._se_EDF"
## [13] "OA.Census.Unemployed_se_EDF"    "OA.Census.White_British_se_EDF"
## [15] "pred.se.1"

gwr.map <- cbind(OA.Census, as.matrix(results))

# Mapping the model

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



```
## Using gridExtra

pacman::p_load(grid, gridExtra)
# Create tmap objects
map1 <- tm_shape(gwr.map) + tm_fill("White_British", n = 5, style = "quantile", title = "Wh
map2 <- tm_shape(gwr.map) + tm_fill("OA.Census.White_British", n = 5, style = "quantile", t
map3 <- tm_shape(gwr.map) + tm_fill("Unemployed", n = 5, style = "quantile") + tm_layout(fr
map4 <- tm_shape(gwr.map) + tm_fill("OA.Census.Unemployed", n = 5, style = "quantile", titl


# Create a clear grid
grid.newpage()

# Assign the cell size of the grid
pushViewport(viewport(layout = grid.layout(2, 2)))

# Print the map objects into predifed cells
print(map1, vp = viewport(layout.pos.col = 1, layout.pos.row = 1))
print(map2, vp = viewport(layout.pos.col = 2, layout.pos.row = 1))
print(map3, vp = viewport(layout.pos.col = 1, layout.pos.row = 2))
print(map4, vp = viewport(layout.pos.col = 2, layout.pos.row = 2))
```


White British
7.88 to 33.06
33.06 to 41.67
41.67 to 48.16
48.16 to 56.76
56.76 to 78.03
WB Coefficient
−0.280 to 0.142
0.142 to 0.314
0.314 to 0.442
0.442 to 0.573
0.573 to 0.947
Unemployed
0.00 to 2.17
2.17 to 3.42
3.42 to 4.83
4.83 to 6.68
6.68 to 18.62
Ue Coefficient
−5.453 to −3.509
−3.509 to −2.829
−2.829 to −2.266
−2.266 to −1.531
−1.531 to 0.770